Development and Applications of AI for Offshore Wind Power Forecasting

Shahram Hanifi

Submitted in fulfilment of the requirements for the

Degree of Doctor of Philosophy

School of Engineering

College of Science and Engineering University of Glasgow

July 2024

Abstract

Wind energy plays a vital role in securing a sustainable and low-carbon future, strengthening energy independence, enhancing economic growth, and preserving the environment. In addition to reducing climate change impacts, wind power is able to facilitate the development of a more resilient and sustainable energy system. There is one obstacle, though, that prevents its penetration into the power grid: its high variability in terms of wind speed fluctuations. Wind power forecasting plays a vital role in addressing the inherent uncertainty of wind power generation. Accurate power forecasting, while making maintenance more efficient, leads to profit maximisation of power traders, whether for a wind turbine or a wind farm.

Several studies have been conducted in the past to investigate factors affecting the performance of power forecasting methods, and several models have also been developed. It is, however, necessary to develop a method that not only provides high prediction accuracy, but also provides good efficiency as well.

This thesis explores different forecasting approaches for wind energy and uses machine learning to develop an accurate, efficient, and robust prediction model. First, background and literature review is presented which covers analysis methods, forecasting time scales, error measurement, and accuracy improvement. Following this, in order to provide high-quality and noise-free data for wind power forecasting, several pre-processing techniques were investigated. Next, the research focused on fine-tuning the hyperparameters of machine learning models to increase forecasting accuracy and efficiency. Scikit-opt, Hyperopt, and Optuna, three hyperparameter optimisation techniques, are used to tune CNNs and LSTMs, two commonly used deep learning models.

After analyzing the results of the previous sections, a new wind power forecasting method is proposed using Wavelet Packet Decomposition (WPD) models, optimised LSTM models and CNN models. After pre-processing the raw data and removing the outliers, WPD is employed to decompose wind power time series into multiple subseries with different frequencies. Comparing the prediction results of all involved models demonstrates that the developed model improves the prediction accuracy by at least 77.4% compared to methods that do not use WPD. In addition, the proposed combination of optimised CNN and LSTM improves the forecasting accuracy by 26.25% compared to methods that use only one deep learning model to forecast all sub-series. In light of the success of the one step ahead forecast, different strategies of multi-step ahead forecasting were explored for the first time in the field of wind power forecasting. The results show that in two-step ahead wind power forecasting, all strategies produce similar results, in both wind turbines. In all forecast horizons of more than two steps ahead, the MIMO approach is best when the dataset does not contain any outliers. In contrast, the direct approach is best when the dataset does contain outliers. It was also concluded

that when datasets contain outliers, wind power forecasting using recursive strategies results in the highest errors for forecasts over two steps.

List of Publications

Journals:

1) **Hanifi, S.**, Liu, X., Lin, Z. and Lotfian, S. 2020 A Critical Review of Wind Power Forecasting Methods-Past, Present and Future *Energies* 13 1–24

2) **Hanifi, S.**, Lotfian, S., Zare-behtash, H. and Cammarano, A. 2022 Offshore Wind Power Forecasting—A New Hyperparameter Optimisation Algorithm for Deep Learning Models Energies 15 6919

3) **Hanifi, S.**, Lotfian, S., Zare-behtash, H. and Cammarano, A. 2023 Offshore Wind Power Forecasting based on WPD and optimised deep learning methods, Journal of Renewable Energy, 218, 119241

4) **Hanifi, S.**, Zare-behtash, H. and Cammarano, A. 2023, Advanced Hyperparameter Optimisation of Deep Learning Methods for Wind Power Forecasting, Journal of Renewable Energy, **Accepted to be published.**

5) **Hanifi, S.**, Zare-behtash, H. and Cammarano, A. 2023, Comparative analysis of strateies for multi-step ahead wind power forecasting, Journal of Energy, In Progress

Conferences:

1) **Hanifi, S.**, Lotfian, S., Zare-behtash, H. and Cammarano, A. Offshore Wind Power Forecasting based on WPD and optimised deep learning methods. the International Conference on Time Series and Forecasting (ITISE-2023), Gran Canaria (Spain). 12th-14th July, 2023.

Contents

List of Tables

List of Figures

List of Acronyms

ACC	Anomaly Correlation Coefficient
ACF	Autocorrelation Function
ADAM	Adaptive Moment Estimation
ADF	Augmented Dickey Fuller
AI	Artificial Intelligence
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
AR	Autoregressive model
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ARX	Autoregressive model with exogenous variable
AWNN	Adaptive Wavelet Neural Network
BP	Back Propagation
BPNN	Back Propagation Neural Network
CEC	Constant Error Carousel
CNN	Convolutional Neural Network
DB	Daubechies
DBN	Deep Belief Network
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DCNN	Deep Convolution Neural Network
DGF	Double Gaussian Function
DGF	Double Gaussian Function
DWT	Discrete Wavelet Transform
EE	Elliptic Envelope
EEMD	Ensemble Empirical Mode Decomposition
EI	Expected Improvement
ELM	Extreme Learning Machine
ENN	Elman Neural Network
EVS	Explained Variance Score
EWT	Empirical Wavelet Transform
FFNN	Feed Forward Neural Network
GA	Genetic Algorithm
GFS	Global Forecasting System
GHG	greenhouse gas
GMM	Gaussian Mixture Model
GP	Gaussian Process
GRU	Gated Recurrent Unit
ICEEMDAN	Improved Complementary Ensemble Empirical Mode Decomposition with Adaptive Noise
IF	Isolation Forest
IMF	Intrinsic Mode Function
LCB	Lower Confidence Bound
LDT	Levenmouth Demonstration Turbine
LM	Learning Machine
LR	Linear Regression
LSSVM	Least Square Support Vector Machine
LSTM	Long Short Term Memory

MA	Moving Average
MBE	Mean Bias Error
MDN	Mixture Density Neural Network
ME	Mean Error
MFNN	Multilayer Feed-Forward Neural Network
MIMO	Multi Input Multi Output
ML	Machine Learning
MLP	Multilayer Perceptron
MLPNN	Multilayer Perceptron Neural Network
MODA	Multi Objective Dragonfly Algorithm
MOMFO	Multi Objective Moth-Flame Optimisation
MSLE	Mean Squared Logarithmic Error
NAAE	Normal Absolute Average Error
NMAE	Normal Mean Absolute Error
NMBE	Normal Mean Bias Error
NMSE	Normal Mean Square Error
NN	Neural Network
NREL	National Renewable Energy Laboratory
NRMSE	Normal Root Mean Square Error
NSC	Nash-Sutcliffe Coefficient
NWP	Numerical Weather Prediction
OCSVM	One Class Support Vector Machine
ORE	Offshore Renewable Energy
PACF	Partial Autocorrelation Function
PCC	Pearson Correlation Coefficient
PI	Probability of Improvement
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
RELM	Regularized Extreme Learning Machine
ReLU	Rectified Linear Unit
RF	Random Forest
RMAE	Root Mean Absolute Error
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RVM	Relative Vector Machine
SCADA	Supervisory Control and Data Acquisition
SDE	Standard Deviation Error
SMBO	Sequential Model Based Optimisation
SNMAE	Sigma Normalized Mean Absolute Error
SVR	Support Vector Regression
TCN	Temporal Convolution Network
TPE	Tree Parzen Estimator
TSO	transmission system operator
UCB	Upper Confidence Bound
VMD	Variational Mode Decomposition
WNN	Wavelet Neural Network
WPD	Wavelet Packet Decomposition
WPT	Wavelet Packet Transform

List of Nomenclatures

b	Activation function of CNN model	-
b_f	Biases in LSTM model	-
b_i	Biases in LSTM model	-
b_o	Biases in LSTM model	-
C_p	coefficient of performance in percentage	-
d	Degree of difference	-
$\widehat{E_\iota}$	Mean normalised error	-
EI	Acquisition function	-
f	Activation function of CNN model	-
f_t	Forget gate activation vectors of LSTM model	-
$f(x)$	Gaussian process	-
$g(x)$	Function for calculation of conditional probability	-
i	Hour	hr
I	Probability density	-
I	Degree of improvement	-
i_t	Input gate activation vectors of LSTM model	-
k	Covariance function in Gaussian process	-
l	Time horizon	-
$\ell(x)$	Function for calculation of conditional probability	-
M	Total number of predictions	-
n	Number of data points	-
o_t	Output gate control signal of LSTM model	-
P	Wind power	MW
p	Order of the autoregressive model	-
PI	Acquisition function	-
$P_N(i,l)$	Target measured power	MW
$(P_{measured})_k$	Measured power of the kth data point	MW
$\left(P_{predicted}\right)_k$	Predicted wind power of the kth data point	MW
$(P(x\|y))$	Conditional probability	-
$P(y\|x)$	Probability of score given the hyperparameter	-
q	Order of the moving average model	-
R	blade length	-
R^2	R-square	-
SD_{bias}	Amplitude error	-
$\tilde{S}_t$	State value of LSTM model	-
$T_N(i,l)$	Forecasted power	MW
u	Wind speed	m/s
U_i	Assigned weights of LSTM model	-
U_o	Assigned weights of LSTM model	-
U_f	Assigned weights of LSTM model	-
W_i	Assigned weights of LSTM model	-
W_o	Assigned weights of LSTM model	-
W^k	Kernel weight of CNN model	-
x	Activation function of CNN model	-
x	Hyperparameter of ML models	-
x_i	*i*th sample in Gaussian process	-
x_j	*j*th sample in Gaussian process	-

x_t	Neuron input of LSTM model	-
X_t	Forecasted wind power at time t	m/s
χ	Search space of hyperparameters	-
y	Score of the f(x)	MW
α_t	White noise	-
θ_j	Moving average parameter	-
θ_t	Coefficient of ARIMA model	-
$\mu(x)$	Mean of normal distribution	-
ρ	Air density	kg/m3
σ_l	Activation function of LSTM model	-
σ_{lh}	Activation function	-
$\sigma^2(x)$	Standard deviation of normal distribution	-
ϕ	Cumulative distribution function	-
φ	Probability density function of the standard normal distribution	-
φ_i	Autoregressive parameter	-
ϕ_t	Coefficient of ARIMA model	-
$\otimes$	Convolution function of CNN model	-

Acknowledgement

I would like to take this opportunity to express my sincere gratitude to all those who have supported me during my research study at the University of Glasgow.

My deepest gratitude goes first and foremost to both my supervisors, Dr. Hossein Zare-Behtash and Dr. Andrea Cammarano, for their constant encouragement and instructive advices. Dr. Zare-Behtash and Dr. Cammarano given me a lot of helpful suggestions not only on research but also on the plan for my career. Without their support, the research and thesis would have not reached the present form.

Secondly, I would like to thank Dr. Saeid Lotfian, my mentor, who has always been available with helpful advice, support and encouragement.

Then I would like to dedicate my PhD thesis to my beloved parents, who, though recently departed, remain the enduring architects of my academic pursuits. Their untimely absence casts a poignant shadow over this accomplishment, but it is precisely in their memory that I find the strength to pen these words. This work stands not only as a testament to my academic endeavours but as a living memorial to the indelible influence of their love and wisdom. Though they cannot witness the culmination of this journey, I dedicate these pages to their memory, acknowledging that every milestone achieved is a reflection of the values they so profoundly instilled.

Lastly, I would like to express my gratitude to my wife, Parisa. I am overwhelmed with gratitude for having you not only as my life partner but as my closest confidante and unwavering support. Your friendship has been my greatest treasure, and your support has been my anchor through life's highs and lows. Thank you for being the one I can always count on.

Chapter 1 - Introduction

One of the biggest problems humanity is facing is climate change, which according to latest reports, there has been an increase in its intensity and rate [1]. The most important factor in the occurrence of climate change is the use of fossil fuels in various sectors, including transportation, construction, and most importantly, in the production of electricity for various applications.

Several agreements have been made between different nations in the past years to limit global warming, including the Paris Agreement 2015 or the Glasgow Agreement 2021, where one of the issues and challenges is always the amount of pollutants produced in different countries and methods to reduce them.

The forecasts show that even considering the improvement in energy efficiency in the various sectors, (e.g., transportation and heating) demanded by the regulations, by 2050, due to the growing demand for products and services, electricity generation needs to be increased by a factor of 2.5 [1].

Currently, the majority of the electricity generated in the world comes from power plants that burn fossil fuels. As an example, the UK, gets 42% of its electricity from gas-fired power plants. However, using this energy source leads to many environmental issues such as global warming and increased greenhouse gas (GHG) emissions [1].

To prevent these adverse effects, and take steps towards decarbonisation, the share of electricity produced from fossil fuels needs to be gradually reduced and replaced with renewable energies in a way that by the end of 2050 three-quarters of the total electricity is supplied from clean sources of energy. Figure 1.1 shows the trend of electricity production of different power plants until 2050 in the UK [2]. As demonstrated in Figure 1.1, the main renewable energy source considered to replace fossil fuels is wind energy, either onshore or offshore.

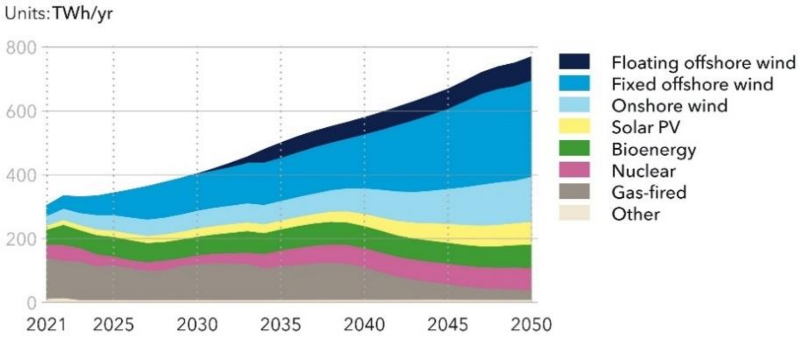


Figure 1.1. UK grid-connected power generation by different power stations [2]

1.1 Wind power and obstacles against its high-scale integration

Wind energy is the fastest-growing renewable energy with the ability to guarantee national security against global threats caused by the reduction of fossil fuel reserves and their price increase [1]. Building wind power plants does have some negative environmental effects on communities and wildlife including noise, visual effects, etc [1]. However, the main obstacle against the integration of wind power into the power grid comes from its inherent variability and unpredictability [3]. Because wind power is dependent on weather conditions, it can be difficult to predict how much power will be generated at any given time since wind speed and direction change.

In the power grid, there must always be a balance between the electricity consumption and production. Otherwise, the supply of electricity will be impossible or of low quality. Variability in wind power can therefore cause challenges for grid stability. The provision of a high share of grid power by non-dispatchable wind energy makes this balance more difficult. Unless an accurate forecast of generated power is provided to the electricity distributor, certain measures cannot be effectively implemented. It is therefore recognised that wind power forecasting is essential for large-scale wind power integration [4].

1.2 Problem Statement

The inability to provide accurate wind power prediction has various consequences. Technically, forecast divergence impedes the ability of transmission system operators to plan the fulfilment of demand based on the available power capacity. Furthermore, from an economic point of view, the uncertainty of the total wind power forecast leads to uncertainties in day-ahead and balancing market costs, which will bring financial losses to electricity providers [5].

The transmission system operator (TSO) is responsible for balancing the electricity production and consumption based on the price-quantity bids of electricity producers. In practice, if the delivery power of each supplier is not equal to its committed level, changes will be made in supplying electricity from TSO’s own facilities or purchases from other producers through bilateral contracts or power pools. The cost of these changes is usually covered by penalties for electricity producers who miss their obligations. Sometimes these penalties are so high that a large part of the production income is used to cover the costs.

By increasing the accuracy of forecasting, these penalties can be avoided, and production costs can be reduced. For example, in the All-Island of Ireland electrical system, each percentage improvement in wind power forecast accuracy saves 0.27% of the total generation costs which equals €4.1 million [6].

Wind power forecasting also provides the possibility of saving costs during the maintenance of wind turbines. Maintenance of wind turbines can sometimes take hours to several weeks. Postponing the maintenance to the times when the lowest amount of energy is expected to be produced, causes less loss in profits [7].

1.3 Motivation

This project is motivated by the recent advancements in artificial intelligence (AI) and its applications in various aspects of working with big data and creating predictive models. Due to the importance of wind power forecasting, it is necessary to use these advances in the analysis and pre-processing of wind power datasets of wind turbines, as well as the development of accurate and efficient forecasting models. The added benefits in this project is having access to the SCADA data of an offshore wind turbine in Scotland which makes testing of the developed models possible.

1.4 Objectives

As mentioned earlier, this thesis focuses on the accurate and efficient prediction of generated power of wind turbines using machine learning algorithms. The efficient wind power forecasting method is an approach that, while providing accurate and reliable forecasts of future electricity production, minimizes computing resources and the time required for forecasting. The developed wind power prediction models can be used for regulation actions, real-time grid operations, market clearing and wind turbine control systems.

For this purpose, various pre-processing methods including data resampling, outlier detection and treatment, wavelet transform and deep learning methods such as CNN, LSTM, and advanced hyperparameter optimisation algorithms are used to develop a novel predictive model with a high accuracy, efficiency and robustness of prediction for the available SCADA wind power data.

A descriptive literature review is presented for this thesis to examine different methods of wind power forecasting. This is followed by the contributions presented in Chapters 3, 4, 5 and 6.

1. The focus of chapter 3 is on examining the data received from the offshore wind turbine from the perspective of outlier data. In this section, after a detailed discussion about the reasons for having outliers, various methods for data pre-processing and outlier detection including isolation forest (IF), elliptic envelope (EE), and the one-class support vector machine (OCSVM), are used to detect and treat them. Different outlier detection methods are examined to determine the best method that leads to the highest prediction accuracy. A comparison of the results will help researchers to choose the best outlier detection method for future studies.
2. In chapter 4, the hyperparameter optimisation of machine learning models is investigated. In this chapter, after reviewing common hyperparameter tuning methods such as grid search and random search, three hyperparameter optimisation techniques including Scikit-opt, Optuna, and Hyperopt are

utilized to tune the hyperparameters of Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) models employed for the short-term wind power forecasting of an offshore wind turbine in Scotland. In this section the impact of the hyperparameter optimisation methods on the accuracy and efficiency of the CNN and LSTM models are assessed by comparing the root mean square error (RMSE) of the predictions and the required time to tune the models.

3. In Chapter 5, a novel hybrid forecasting method is developed based on the wavelet packet decomposition [WPD], Long Short-Term Memory Network [LSTM], and Convolutional Neural Network [CNN] to improve the accuracy of wind power forecasting. WPD is employed to decompose pre-processed wind power data into sublayers with different frequencies. Sequential Model-Based Optimisation (SMBO) with the Tree Parzen Estimator (TPE) is then used to tune the hyper-parameters of LSTM and CNN, efficiently. The optimised LSTM is employed to predict the low-frequency sub-layer that has both long-term and short-term dependencies, and CNN is used to forecast the high-frequency sub-layers with short-term dependencies.
4. In Chapter 6, to increase the forecast horizon of the proposed forecasting model, for the first time in the field of wind turbine power prediction, a comprehensive comparison of three main strategies for multi-step ahead prediction including the recursive, direct, and multi-input multi-output (MIMO) strategies are investigated. In this section, two real-world wind power datasets are used to clarify the best strategy. In addition, the impact of outlier presence in the accuracy of multi-step ahead predictions at different forecasting horizons is assessed.

1.5 Research Contributions

Based on the aforementioned objectives, to develop the accurate, efficient and robust wind power forecasting model, this thesis uses pre-processing methods to clean the dataset, deep learning models as the core of prediction and advanced hyperparameter optimisation algorithms to increase the accuracy and efficiency. As such, after assessment of different pre-processing methods, ML algorithms and hyperparameter tuning tools, a novel method is proposed. The contributions of this thesis can be summarised as follows:

1. Providing an extensive literature review of wind power forecasting methods and presenting a flowchart as a guideline for wind power forecasting process screening, allowing the wind turbine/farm operators to identify the most appropriate forecasting methods based on time horizons, input features, computational time and error measurements, etc. The literature review includes the latest research conducted in the last two decades in the field of wind power forecasting, which examines various strategies to improve the forecasting performance.
2. To ensure the best performance of WPD for decomposition of power time series to different subseries with different frequencies, sixteen mother wavelets from four widely used wavelet families in the

literature (Daubechies, Haar, Sym, and Coif) were selected, and their performance in prediction improvement of forecasting models are assessed.

3. As opposed to grid search or random search which are time-consuming and unreliable, three advanced hyperparameter optimisation techniques, Scikit-opt, Hyperopt, and Optuna, are used to tune CNN and LSTM prediction models, for wind power forecasting. To the best of the author's knowledge this is the first time that theses hyperparameter optimisation techniques are used in this field. The results showed that the Optuna optimisation technique using Tree-structured Parzen estimator (TPE) search algorithm and Expected Improvement (EI) acquisition function, has the highest efficiency for both CNN and LSTM models. Also, regarding the improvement of the prediction accuracy, it has been demonstrated that while for the CNN model, all the optimisation methods perform almost the same, the LSTM model optimised by the Hyperopt algorithm based on the annealing search method achieves the highest accuracy.
4. A novel wind power forecasting method is proposed based on the combination of WPD, optimised LSTM and CNN models. In the developed WPD-LSTM-CNN model, first, the obvious outliers that diminish the prediction accuracy are removed and the resolution of data averaged over 10 minutes in order to mitigate the influence of turbulence. Next, Wavelet Packet Decomposition (WPD) is employed to decompose the pre-processed wind power time series into multiple sub-series with different frequencies. This increases the stationarity of the data, thereby enhancing the efficiency of the prediction models. Three tuned independent CNNs are employed for the prediction of the high-frequency sub-series, and one optimised LSTM model is adopted to complete the forecasting of the low-frequency sub-layer. For the optimisation of these deep learning models, the SMBO method as a formalisation of Bayesian optimisation, provided in the Optuna optimisation package, is used to reduce the dependence on computational resources.
5. The sensitivity of different structures of CNN and LSTM models to seed changes was investigated and the most resistant structure against randomness was selected for both models. The proposed models not only do not require initial settings of random seeds, but also provide the highest level of accuracy, efficiency, and robustness for the utilised offshore wind power dataset. This level of performance in short-term wind power forecasting can be utilized for regulation actions, real-time grid operations, market clearing, and wind turbine control systems, ensuring efficient balancing of supply and demand in the electricity market.
6. Based on the results of comparison between three main strategies including the recursive, direct, and multi-input multi-output (MIMO) strategies for multi-step ahead forecasting of two wind turbine datasets, it is concluded that in two-step ahead (20 min) wind power forecasting, all different strategies come up with almost identical results. The multi-step ahead wind power prediction with the MIMO

strategy performs best when the dataset contains no outlier data, and vice versa, when it contains outlier data, the direct strategy performs best in forecast horizons of more than two steps ahead. In the case of datasets containing outliers, for forecast horizons above two steps, wind power forecasting using the recursive strategy results in the highest error. The errors in previous wind energy forecasts, used to predict next steps accumulate, leading to a decrease in accuracy.

Chapter 2 - Background and Literature Review

As mentioned earlier, the power generated by wind turbines varies rapidly due to the fluctuation of wind speed and wind direction. It is also dependent on terrain, humidity, date and time of the day [8]. This continuous change makes wind power management challenging for distribution networks, where a balance is highly desired between the power supply and demand [9]. Therefore, one of the major reasons for wind power forecasting is to decrease the risk of uncertainties in wind, allowing higher penetration of wind power into the grid. It is also vital for better dispatch, maintenance planning, determination of required operating equipment, etc.

In recent studies that have been carried out for wind power forecasting, there is no obvious evidence or guide to compare different predictive methods [4, 9, 10]. Therefore, it is necessary to systemically examine the critical methods of wind power forecasting. It is important to further compare the details of each method to clarify the prediction accuracy, input features, dataset specifications, size of the database and their sampling rate, evaluation criteria, and loss function.

2.1 Classification of wind power forecasting methods

Theoretically, wind power forecasting can be classified based on either time horizons or applied methodology [4]. Based on different time scales, the prediction can be divided into a very short-term scenario, for which the range of predictions is usually below 30 minutes, up to a month (long-term predictions), which has seen a progressive development over the past decade. On the other hand, the forecasting methods have benefited from the evolution of high-performance computing tools, with increasingly newer computational methods established.

2.1.1 Prediction horizons

Depending on different functional requirements, predictive horizons can be divided into four major time scales summarised in Table 2.1. The shorter the forecasting period, the more accurate the results, but less time to make important decisions in wind power production. Long-term forecasts can provide information on future wind energy, however they are typically less accurate [9, 10, 11, 12].

Table 2.1. Prediction horizons in wind power forecasting.

Time horizon	Range	Applications
very short-term	few minutes to 30 minutes	regulation actions, real-time grid operations, market clearing, turbine control
short-term	30 minutes to 6 hours	load dispatch planning, load intelligent decisions
medium-term	6 hours to 1 day	operational security in the electricity market, energy trading, on-line and off-line generating decisions
long-term	1 day to a month	reserve requirements, maintenance schedules, optimum operating cost, operation management

2.1.2 Prediction methodologies

According to applied methodologies, wind power forecasting models can be further divided into persistence methods, physical methods, and statistical methods. Their differences are located in the required input data, the accuracy at different time scales, and the complexity of the process.

- **Persistence methods**

In this method, which is normally used as a reference, the predicted future wind power is equal to the measured power in the present. This approach was commonly used to be compared with novel short-term forecasting methods to identify their improvements [10, 11, 12, 13]. The accuracy of this method can quickly deteriorate with the increment of prediction timescale [10]. Apart from being simple and economical, the main advantage of this method is that neither a parameter evaluation nor external variables are required [14].

- **Physical methods**

Physical methods use detailed physical characterisations to model wind turbines/farms. This modelling effort is often carried out by downscaling the Numerical Weather Prediction (NWP) data, which requires a description of the area, such as roughness and obstacles, as well as weather forecasting data of temperature, pressure, etc. These variables are used in complex mathematical models that are time-consuming when it comes to predicting wind speed. Then, the predicted wind speed is related to the wind turbine power curve (normally provided by the turbine manufacturer) to forecast the wind power. This method doesn’t need to be trained with historical data, but it depends on physical data [12]. In recent decades, many physical methods have been proposed. For example, Focken et al. [15] created a physical wind power forecasting approach for time scales up to 48 hours ahead. The method was founded on a physical approach that received input data from a weather prediction model. The boundary layer, the layer of air directly influenced by friction from the ground, was first shaped taking into account roughness, terrain and wake effect. Furthermore, the day-to-day change of the thermal stratification of the atmosphere was taken into account to estimate the wind speed at hub height [15]. De Felice et al. [16] used a physical model to predict the electricity demand in Italy by considering 14 months of hourly temperature as inputs. Comparing their proposed method with the Persistence approach using the mean absolute error (MAE) showed that NWP models can improve the forecasting performance, especially for the hottest regions. Even though this method is perhaps the best choice for medium to long-term wind power prediction, it is computationally complex and therefore needs considerable computing resources [17].

- **Statistical methods**

These methods are generally based on developing the non-linear and the linear relationships between NWP data (such as wind speed, wind direction, and temperature) and the generated power. To define this statistical relationship, historical data must be used as the training data. The model is then tuned by comparing the model prediction and the on-line measured power. After that, the model is ready to predict using NWP forecast of the next few hours and the on-line measurements. This method is easy to model and inexpensive [18]. It is for short term periods, hence as the estimation time increases, its prediction accuracy decreases [13]. More specifically, statistical methods can be divided into two main sub-classifications: Time series based, and Neural Network (NN) established.

Time series models

These models, proposed by Box-Jenkins [12], apply historical data to generate a mathematical approach for developing the model, estimating parameters, and checking the simulation characteristics. The general form of the model can be described as:

$$X_t = \sum_{i=1}^{p} \varphi_i \, X_{t-i} + \alpha_t - \sum_{j=1}^{q} \theta_j \, \alpha_{t-j} \tag{1}$$

while φ_i represents the autoregressive parameter, θ_j is the moving average parameter, α_t is the white noise, p is the order of the autoregressive, q is the order of the moving average model, and X_t is the forecasted wind power at time t.

The entire equation (1) represents an Autoregressive Moving Average (ARMA) model, but if p is assumed to be zero, it will represent a moving average model (MA). Moreover, when q is assumed to be zero, it represents an autoregressive model (AR) [12]. Statistical methods based on this approach are easy to formulate and very applicable in short-term wind power forecasting [19]. They also require low computation times, but they may not provide adequate prediction capabilities, especially when the time series are nonstationary [20]. Table 2.2 shows two-time series models with the specifications of their selected input features, including data size and sampling rate.

Table 2.2. Time series wind power prediction models.

References	Method	Input Features	Datasets	Data Size	Sampling rate
M. Duran et al. 2007 [21]	ARX	wind speed	Spanish wind farms	12 months	6 hr
Gallego et al., 2011 [22]	AR model	wind speed, wind direction	offshore 160 MW wind farm of Horns Rev in Denmark	12 months	10 min

Firat et al. [23] proposed a statistical model based on independent component analysis and the AR model for wind speed forecasting. Using six years of hourly wind speed data of a wind farm in the Netherland, the authors claimed that the proposed model provided higher accuracy compared to direct forecasting methods for 2 to 14 hours ahead.

De Felice et al. [16] used NWP data and ARIMA models to forecast electricity demand in Italy. The temperature of the 14 months in the years between 2003 and 2009 was used as the main input value. Duran et al. [21] designed an AR model with exogenous variable (ARX) model. Using the wind speed as an exogenous variable, they compared the mean error of their model with persistence and traditional AR models and showed significant improvements in accuracy.

Artificial Neural Networks (ANNs)

ANNs, as one of the most commonly used methods for wind power prediction, can identify the non-linear relationships between input features and output data [24]. One of the reasons for the tendency to use neural networks is to avoid the complexity of the mechanical structure in wind turbines [25]. Typically, an ANN model consists of an input layer, one or more hidden layers, and an output layer, where the historical data/features are fed for training and testing [11]. It also consists of processing units called neurons, which are connected with certain weighted connections. The ANN adjusts the weight of these interconnections through the training process. If the desired output is known at the beginning of the process, it will be named supervised; contrarily, it will be called un-supervised [26]. Figure 2.1 shows a typical structure of an ANN model.

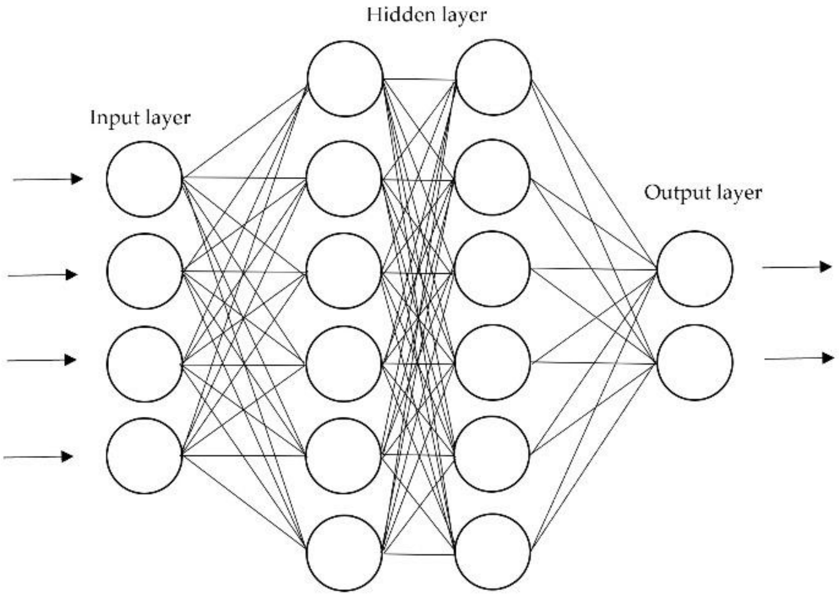


Figure 2.1. Typical structure of an ANN model.

The performance of ANNs is dependent on many factors, including data pre-processing, data structure, learning method, connections between input & output data, etc [27]. There are more than 50 forms of ANNs, including multilayer perceptron (MLP) [28], wavelet neural network (WNN) [29], back-propagation neural network (BPNN), radial basis function neural network (RBFNN) [30], Elman neural network (ENN) [14], long-short-term memory (LSTM) [17], and convolutional neural network (CNN) [31]. Designing an ANN model involves two steps: first, the selection of the proper structure of the network and specifying the direction of the passed information. There are two major topologies, including feed-forward for passing data in one direction from the input to the output layers and recurrent for mutual directions [12].

Table 2.3 shows a summary of the ANN models reviewed in this study. The selected input features were introduced along with the specifications of the wind turbine or wind farm, data size, and sampling rate.

Table 2.3. ANN wind power prediction models.

References	Algorithm	Input Features	Datasets	Data Size	Sampling rate
Pelletier et al., 2016 [32]	MLP	wind speed, air density, turbulence intensity, wind shear, and wind direction	140 wind turbines in Nordic	12 months	10 min
Sideratos et al., 2007 [33]	RBFNN	past power measurements, NWPs	wind farm in Denmark including 35 600-KW turbines	26 months	1 min
Bilal et al., 2018 [28]	MLP	wind speed	four sites on the northwest of Senegal.	6 ~ 9 months	1 & 10 min
Jyothi & Rao, 2016 [34]	Adaptive wavelet NN (AWNN)	wind speed, air density, ambient temperature, and wind direction	two wind turbines in North India	15 days	10 min
Zhao et al., 2016 [20]	Bidirectional ELM	wind power	onshore wind farm in the USA	12 months	60 min
De Giorgi et al., 2011 [14]	ENN	wind power, wind speed, pressure temperature and relative humidity	wind farm in southern Italy	12 months	60 min
Xu & Mao, 2016 [35]	ENN	wind speed, wind direction, temperature, humidity, pressure	a single 15kW wind turbine in a west wind farm in China	6 days	15 min
Catalao et al., 2009 [36]	MFNN	historical data	wind farm in Portugal	4 days	
Singh et al. 2007 [8]	MLP	wind speed, wind direction and air density	Fort Davis Wind Farm the in Texas, USA	2 months	10 min
Chang, 2013 [37]	BPNN	wind power	a wind turbine in Taiwan	6 days	10 min
Carolin & Fernandez, 2008 [38]	Feed-forward NN (MLP)	wind speed, relative humidity, and generation hour	137 wind turbines from seven wind farms located in Muppandal, (India)	36 months	-

Jyothi and Rao [34] used an AWNN for short wind power prediction. The minimum NRMSE that they achieved was 0.02. Bilal et al. [28] designed an MLP network to forecast the wind power of four different wind farms in Senegal. The main input of their model was wind speed, but they also assessed different combinations of input variables like wind direction, temperature, humidity and solar radiation. The results showed that, except wind speed, air temperature has the highest impact on improving the accuracy of the model. Regarding the best structure for MLP, the authors considered the Levenberg-Marquardt backpropagation algorithm as training the algorithm and log-sigmoid transfer as the activation function. It was concluded that the MLP with three hidden layers (5,7 and 8 neurons in each hidden layer) has the lowest NMSE.

Xu and Mao [35] used ENN to study a single 15kW wind turbine on a west wind farm of China. Using input variables of wind speed, wind direction, humidity and temperature, the authors presented acceptable accuracy, particularly after the application of the particle swarm optimisation algorithm.

In another investigation, Chang [37] developed a model based on BPNN for 10 min ahead wind power prediction. The historical wind power data of a wind turbine in Taiwan were used to verify the efficiency of this method. The results showed that the proposed neural network could predict wind power easily with an average absolute error of 0.278%.

- **Hybrid approach**

Combinations of different forecasting methods, such as ANNs and fuzzy logic models, are called hybrid approaches [31]. The main aim of hybrid approach is to retain the merits of each technique and improve the overall accuracy. A combination may not always lead to a better result, compared to the constituent models. However, it has been demonstrated that there are fewer risks in most of the situations [12]. Many hybrid methods have been proposed based on the combination of different models. Table 2.4 shows the reviewed hybrid methods carried out in this study, including input features for training models, specifications of the used dataset, database size, and sampling rate. What is clear is that although combining different methods improves the overall performance, on the other hand, it complicates the model and increases the required computation time. Hence, it is vital to obtain a balance between accuracy and efficiency. A number of these methods are introduced and evaluated in more details in the following section.

Table 2.4. Hybrid wind power prediction (part 1).

References	Method	Input Features	Datasets	Data Size	Sampling rate
Du et al. 2019 [39]	ICEEMDAN, MOMFO, Wavelet NN	wind power	?	?	?
Shahid et al. 2021 [40]	GA, LSTM	wind power	?	?	?
Wang et al. 2018 [41]	MODA, ELNN	Wind speed	?	?	?
Kisvari et al. 2021 [42]	IF, GRU, LSTM	wind power	?	?	?
Hong et al. 2019 [31]	CNN, RBFNN, DGF	wind power	historical power data of a wind farm in Taiwan	12 months	60 min
Lin et al., 2020 [43]	Isolation Forest (IF), deep learning NN	wind speed, nacelle orientation, yaw error, blade pitch angle, and ambient temperature	SCADA data of a wind turbine in Scotland	12 months	1 s
Zhang et al., 2019 [17]	LSTM, Gaussian Mixture Model (GMM)	wind speed	a wind farm of 123 units in north China	3 months	15 min
Marcos et al. 2017 [44]	Kalman filter, statistical regression or power curve	NWP data	Palmas and RN05 wind farms in Brazil	7 and 12 months	10 min
Wang et al., 2018 [41]	ELM optimised by MODA	wind speed	two observation sites in Penglai, China	37 days	10 min
Shetty et al., 2016 [45]	RBFNN, PSO in optimising and ELM in training	wind speed, wind direction, blade pitch angle, density, rotor speed	SCADA of a 1.5 MW horizontal wind turbine	6 months	10 min
De Giorgi et al., 2011 [14]	Elman and MLP network	wind power, wind speed, pressure, temperature, humidity	wind farm in southern Italy	12 months	60 min
De Giorgi et al., 2011 [14]	Wavelet decomposition and Elman network	wind power, wind speed, pressure, temperature and relative humidity	wind farm in southern Italy	12 months	60 min

Table 2.5. Hybrid wind power prediction (part2).

References	Method	Input Features	Datasets	Data Size	Sampling rate
Liu et al., 2017 [46]	BPNN, RBFNN and LSSVM	wind speed, wind direction, and temperature at the wind turbine hub height	16 MW wind farm located in Sichuan, China	2 months	15 min
Zhao et al., 2012 [47]	Kalman filter and MFNN	wind speed, direction, temperature, pressure, humidity power data from SCADA	an outermost domain which covers the eastern half of China	12 months	6 hr
Lin and Liu 2020 [43]	IF feed-forward NN	wind speed, blade pitch angle, temperature, yaw error and nacelle orientation	7 MW wind turbine in Scotland owned by the ORE Catapult	12 months	1 s
Peng et al., 2013 [48]	Physical model and ANN	wind speed, wind direction, temperature	50 MW wind farm with 40 wind turbines in China	3 months	10 min

Hong and Riofloriodo [31] proposed a hybrid 24 hour-ahead wind power prediction model based on CNN. Different operations in CNN, such as convolution, pooling, and kernel, were used to pull out the input features. The defined features were then fed to an RBFNN, implementing the Double Gaussian Function (DGF) as an activation function. The authors also used Adaptive Moment Estimation (ADAM) to further improve CNN and RBFNN. Using one-year historical power data from a wind farm in Taiwan, the proposed approach provided the best performance compared with other method. The authors also concluded that the application of DGF in the RBFNN generated better results than conventional RBFNN with a Gaussian function.

Lin et al. [43] implemented the isolation forest (IF) technique along with deep learning neural network to detect outliers for more accurate wind power forecasting. The isolation forest method works by constructing random decision trees that isolate observations, effectively identifying outliers as those requiring fewer splits to isolate. Wind speed, wind direction, air temperature etc. were extracted from a SCADA dataset of an offshore wind turbine to be used as inputs while employing wind power as the output in the predictive model. Comparison results showed that IF is a more effective method of providing accurate forecasting, especially when the investigated data does not follow a normal distribution. In another paper [25], the authors critically evaluated eleven features from a 7MW wind turbine in Scotland, including four wind speeds at different heights, average blade pitch angle, three measured pitch angles for three blades, ambient temperature, yaw error, and nacelle orientation. The results revealed that the blade pitch angle had the greatest effect on the performance of the prediction model, even more than wind speed and wind shear. On the contrary, wind direction and air density contributed the least, which allowed their elimination for reducing computational time. Zhang et al. [17] used the LSTM network to predict wind power production of a wind farm in China. Three-month wind speed data from NWP were used as inputs, and the produced wind power was treated as output. The authors compared their model with RBF, wavelet, DBN, BP and ELMAN. The results showed that the proposed model improved the accuracy of forecasting, although its operating time was longer than the others. The performance of the model was found to strongly depend on the wind speed. In high speeds, the wavelet provided better performance, while the other methods provided better prediction capability in lower speed areas. In high wind speeds, the wavelet method likely provided better performance due to its ability to effectively capture and analyse the transient, localized variations in the wind speed data. The uncertainty of the forecasted power was assessed by three different methods, including mixture density neural network (MDN), relative vector machine (RVM) and Gaussian Mixture Model (GMM). The results showed that the GMM gave the best performance.

Marcos et al. [44] used a combination of a physical and a statistical model. The input data were atmospheric global-scale forecasts, which were provided by the Global Forecasting System (GFS) [44]. A Brazilian NWP model (BRAMS) was also used to refine the atmospheric global-scale forecasts by using physical considerations regarding the terrain, such as vegetation cover, soil texture etc. Afterwards, a systematic error correction filter with the capability of learning the dynamic behaviour of wind data was used to reduce the biases of the forecasted wind. Following the elimination of the biases, two main methods were used for wind power forecasting, the manufacture's power curve and regression equations, which were derived from wind measurements and generated power data from SCADA systems. For generating polynomial regressions, the observed one-year data of wind and power were considered using four equations: linear, quadratic, quadratic considering previous power outputs, and cubic. Comparing these four equations with statistical indices such as RMSE showed that the cubic regression provides the best results. As other factors can also influence the power output, such as air density, wake effect, orography etc., a Kalman filter was used to eliminate systematic errors from the conversion model. Finally, it was concluded that using a Kalman filter decreased the RMSE value and increased the values of ACC and NSC, all representing better forecasting.

Zhao et al. [20] proposed a bidirectional model for 1 ~ 6 hour-ahead wind power forecasting. In their model, the forecasted power from the forward model was used as the input for the optimisation algorithm of the backward model. By comparing the difference between the forwarding and backward results, the authors were able to provide the final forecasting. Eight months' worth of hourly measured wind power of an American wind farm were used for training, while another four months were used for further evaluation. Comparing the evaluation criteria of this model with the forward, backward, and persistence methods showed that it outperformed the others.

Liu at al. [46] combined three different prediction models including BPNN, RBFNN, and the Least Square Support Vector Machine (LSSVM) by an adaptive neuro-fuzzy inference system (ANFIS) for 48 hour-ahead wind power forecasting. As the first step, a Pearson Correlation Coefficient (PCC) based method was used to eliminate outliers. Sixty-day datasets of a wind farm in China, containing wind speed, wind direction, temperature and generated power, were used as inputs to train the three methods. The evaluation of the proposed hybrid model showed that it outperformed the three individual forecasting models and can predict with remarkable accuracy improvement. Zhao et al. [47] used a Kalman filter to decrease the systematic errors of wind speed generated from a weather research and forecasting model. The model was used with wind direction, temperature, and humidity as input variables for a multilayer feed-forward neural network to forecast a day-ahead wind power. The results showed that filtering the raw speed and application of MFNN can decrease the NRMSE from 17.81% to 16.47%.

2.2 Factors to compare different methods

Methods of estimating wind power can be compared through various parameters, such as accuracy, input features, computational time, etc., presented in Tables 2.2 for time series models, Table 2.3 for ANN models, and Table 2.4 for hybrid methods. Local features, such as temperature, and humidity are highly dependent on the selected regions, directly affecting the output power. The different forecasting methods have been evaluated based on various criteria, these criteria are examined in Table 2.5 (time series models), Table 2.6 (ANN models), and Table 2.7 (hybrid models), where the metric obtained from each reviewed article are summarised.

Table 2.6. Performance evaluation in time series wind power prediction models

References	Algorithm	Evaluation Criteria	Evaluation Value	Evaluation Unit	Results
Duran et al., 2007 [21]	ARX	ME	34.6-63.2	-	Accuracy improvement was found in comparison with the persistence method and conventional AR models.
Gallego et al., 2011 [22]	AR	NRMSE	3.93	-	local measurement of both wind speed and direction improves the forecasting performance

Table 2.7. Performance evaluation in ANN wind power prediction models (part 1)

References	Algorithm	Evaluation Criteria	Evaluation Value	Evaluation Unit	Results
Pelletier et al., 2016 [32]	MLP	MAE	15.3-15.9	kW	multi-stage ANN with 6 inputs performed better than parametric, non-parametric and discrete models
Sideratos et al., 2007 [33]	RBFNN	NMAE NRMSE	5-14 20	% %	effectively predicted for 1-48 h ahead performed better than the persistence method
Bilal et al., 2018 [28]	MLP	NMSE NMAE SNMAE R (fitting rate)	3.51 14.85 25.7 0.98	% % - %	wind speed + temperature as input is better than only wind speed. Considering all variables improve performance
Jyothi & Rao, 2016 [34]	AWNN	NRMSE	0.1647	-	The minimum NRMSE showed that WNN performs well.
Zhao et al., 2016 [20]	Bidirectional ELM	NMBE NMAE NRMSE	-0.53 16.61 21.27	% % %	lower values of NMAE and NRMSE showed that bidirectional performed better than forward, backward and persistence method.
De Giorgi et al., 2011 [14]	ENN	NAAE	15 12.5	% %	among the different NWPs data, pressure, and temperature had the highest positive impact. The hybrid method performed better than other methods especially in long term forecast (24 h)
Xu & Mao, 2016 [35]	ENN	MSE MAE	16.55 10.52	% %	application of particle swarm optimisation algorithm improves the performance/accuracy

Table 2.8. Performance evaluation in ANN wind power prediction models (part 2)

References	Algorithm	Evaluation Criteria	Evaluation Value	Evaluation Unit	Results
Catalao et al., 2009 [36]	MFNN (trained by LM algorithm)	MAPE	7.26	%	performed better than Persistence method in less than 5 seconds of computing time.
Singh et al., 2007 [8]	MLP	percentage difference	0.303-1.082	%	better results than traditional methods
Chang, 2013 [37]	BPNN	AAE	0.278	%	the proposed model can predict win power easily and correctly
Carolin & Fernandez, 2008 [38]	feed forward NN (MLP)	RMSE	8.06	%	helpful model for energy planners

Table 2.9. Performance evaluation in hybrid wind power prediction models (part 1)

References	Method	Evaluation Criteria	Evaluation Value	Evaluation Unit	Results
Hong et al., 2019 [31]	CNN, RBFNN, DGF	R2 RMSE NMSE MAPE	0.92 76.97 2.75 5.048	- % % %	best performance rather than CNN-RBFNN and CNN-MFNN (lower RMSE, NMSE, MAPE and higher R2)
Lin et al., 2020 [43]	IF, deep learning NN	MSE	0.003	-	using IF for outlier detection instead of Elliptic envelope increased the performance
Zhang et al., 2019 [17]	LSTM, GMM	RMSE	6.37	%	the accuracy of LSTM was higher than RBF, wavelet, DBN, BP and ELMAN. GMM was better in analysing the uncertainty of the prediction than MDN and RVM.
Marcos et al., 2017 [44]	Kalman filter, statistical regression or power curve	MBE RMSE	4.32 101.11	kW	using Kalman filter decreased the RMSE and increased the ACC and NSC, all represent better forecasting

Table 2.10. Performance evaluation in hybrid wind power prediction models (part 2)

References	Method	Evaluation Criteria	Evaluation Value	Evaluation Unit	Results
Wang et al., 2018 [41]	ELM optimised by MODA	AE MAE RMSE NMSE MAPE	19.05 77.67 107.96 0.0001 0.9824	MW MW MW - %	-
Shetty et al., 2016 [45]	RBFNN, PSO in optimising and ELM in training	MSE	0.0003	-	ELM as a learning algorithm makes the learning process quicker.
De Giorgi et al., 2011 [14]	Elman and MLP network based on historical data and NWP	NAAE	10.98	%	among the different NWPs data, pressure and temperature had the highest positive impact. The hybrid method performed better than other methods especially in long term forecast (24 h)
De Giorgi et al., 2011 [14]	Wavelet decomposition and Elman network	NAAE	15.5	%	
Liu et al., 2017 [46]	BPNN, RBFNN and LSSVM	MAPE NMAE NRMSE	6.7-27.4 1.01-6.35 2.37-9.45	% % %	the combined model performed better than all three individual models
Zhao et al., 2012 [47]	Kalman filter, MFNN	NRMSE	16.47	%	NRMSE value improved from 17.81% to 16.47%, by using a Kalman filter.
Lin and Liu, 2020 [43]	IF, feed-forward neural network	RMSE MAE R-square MSLE EVS	517.33 374.41 0.91 0.29 0.91	- - - - -	blade pitch angle had the greatest effect on the performance of the prediction model even more than wind speed and wind shear.
Peng et al., 2013 [48]	Physical + ANN	MAE RMSE	760 2.01	kW %	combining physical and statistical prediction techniques rather than the application of ANN improved accuracy.

2.2.1 Accuracy

The accuracy of wind power forecasting is the most important factor for comparing different predictive methods, which can be determined by evaluating certain metrics. Typically, levels are provided for these evaluation factors in different systems, based on which it can be ensured that the model has enough accuracy. For example, in some references, it has been mentioned that the rate of RMSE should be within 10% of the installed capacity for most of the models. In China, State Grid Corporation defined a value of 20% for the maximum acceptable RMSE for short term wind power forecasting and 15% for the forecasted value of 4 hour-ahead [17]. Methods with higher RMSE do not provide the required performance. In Ireland, the system managers (EirGrid and SONI) require a target accuracy of 6% ~ 8% [18].

2.2.2 Efficiency (Computational time)

Forecasting computational time (time required for training/learning) is considered as a significant factor for selection of proper prediction models, especially for short term forecasting. It is also useful to understand whether it can be applied in real-time. For example, the proposed approach by Marcos et al. [44], which needed about 60 ~ 70 min computational time for each 72 hour NWP model simulation, was demonstrated that it could be used in real-time for power system operation. The computational time depends on the used methods, required accuracy, volume and sampling rate of input data, computing power etc. It also relies on the training algorithm. For instance, as Zhao et al. [20] discussed, Extreme Learning Machine (ELM) with a feed-forward neural network performed faster than networks based on the backpropagation algorithm.

Singh et al. [8] showed that the training and testing of two-month input data with 10 min sampling rate for the proposed MLP network could be finished in 30 min on a Pentium 150 MHz computer. The authors also claimed that using a separate neural network for each turbine rather than the wind farm guarantees fast training because the size and complexity of the network will be minimised. Another benefit of this scheme is that the separate models will not be affected by the off-line turbines.

Lin and Liu [25], in an effort to reduce the computational time, removed minor influencer features in the proposed model, including air temperature, nacelle orientation, and yaw error. Even though this reduction resulted in a small savings in processing time (0.77 min) for a single wind turbine, the saved simulation time can be considered worthwhile when taking into account a typical wind farm comprising of more than 100 turbines.

2.3 Performance evaluation in wind power forecasting

To assess the performance of the wind power forecasting methods, there are several statistical metrics which can show deviations of the forecasted from the measured wind power. The statistical description of how the evaluation criteria were chosen in the prior research is outlined in Figure 2.2, which is based on the information from Table 2.5, 2.6 and 2.7. As reflected in Figure 2.2, the majority of studies consider a number of evaluation criteria such as RMSE, NRMSE, MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage

Error). In the following sections, the types of techniques used for the accuracy assessment of wind power forecasting methods are discussed in detail.

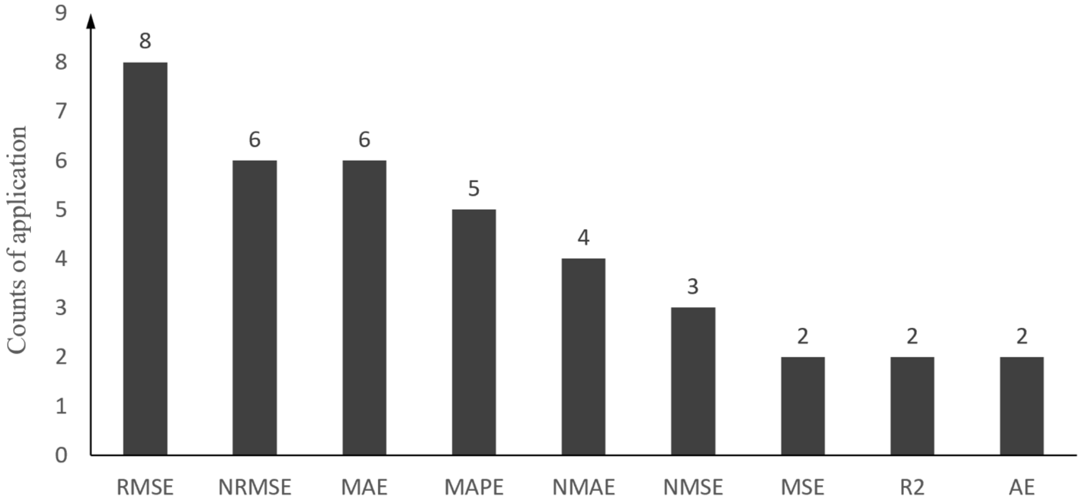


Figure 2.2. Statistical description of chosen evaluation criteria in the prior research

2.3.1 Error measurements

The most common error measurements that evaluate performance by specifying the degree of similarity between forecasted and measured data are [49]:

- **Normalised error**

$$E_i(l) = P_N(i,l) - T_N(i,l) \tag{2}$$

where i = hour of the predicted data, l = time horizon, $T_N(i,l)$ is the forecasted power, $P_N(i,l)$ is the target measured power and M is the total number of predicted data described in Eq (3) and Eq (4).

$$T_N(i,l) \ = \frac{T(i,l)}{\text{Max}_{i=1}^{M}(P(i,l))} \tag{3}$$

$$P_N(i,l) = \frac{P(i,l)}{\text{Max}_{i=1}^{M}(P(i,l))} \tag{4}$$

- **Normalised mean bias error (NMBE)**

The normalised mean bias error shows the difference between the average forecasted and observed wind power. The value shows if the method over-estimates (NMBE > 0) or under-estimates (NMBE < 0). This statistical metric displays systematic errors instead of the forecasting method’s capability [20]. This statistical index doesn’t offer enough information about the accuracy of the forecasting method when it is used by only itself. The NBME can be calculated from Eq (5).

$$\text{NMBE}\ (l) = \left(\frac{1}{M}\sum_{i=1}^{M} E_i\ (l)\right) \times 100 \tag{5}$$

- **Normalised mean absolute error (NMAE)**

One of the most common wind power prediction performance indexes is normalised mean absolute error [17]. It provides more precise random and systematic error analysing. The NMAE can be calculated from Eq (6).

$$\text{NMAE}(l) = \left(\frac{1}{M}\sum_{i=1}^{M} |E_i\ (l)|\right) \times 100 \tag{6}$$

- **Normalised root mean square error (NRMSE)**

This is another widely used accuracy checking parameter. Similar to NMAE, NRMSE shows both random and systematic errors. Higher values of NRMSE indicate deviations, while successful forecasts show lower values of NRMSE. It should also be mentioned that a large gap between NMAE and NRMSE for the results of a method indicates that the predicted values are extremely spread from the measures data [20]. The NRMSE can be calculated from Eq (7).

$$NRMSE(l) = \sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M} (E_i\ (l))^2} \times 100 \tag{7}$$

- **Mean squared logarithmic error (MSLE)**

The MSLE is a risk metric according to the expected value of the squared logarithmic error and can be expressed as:

$$MSLE = \frac{1}{n}\sum_{k=0}^{n-1} (\log_e(1 + (P_{measured})_k) - \log_e\left(1 + \left(P_{predicted}\right)_k\right))^2 \tag{8}$$

in this equation, n is the number of data points, $(P_{measured})_k$is the measured value of the kth data point from the SCADA database and $\left(P_{predicted}\right)_k$ is the predicted wind power of the kth data point from deep learning modelling.

- **R-square (R^2)**

This coefficient of determination shows the variance of the prediction from the measured data. The maximum possible value of the R^2 is 1.0, while negative values indicate worse prediction. It can be defined as:

$$R^2 = 1 - \frac{\sum_{k=1}^{n}\left[\left(P_{predicted}\right)_k - (P_{measured})_k\right]^2}{\sum_{k=1}^{n}\left[(P_{measured})_k - \frac{1}{n}\sum_{k=1}^{n}(P_{measured})_k\right]^2} \tag{9}$$

- **Explained variance score (EVS)**

Explained variation estimates the proportion to which a forecasting model scores for the dispersion of a specified dataset. For the best prediction, the value of EVS is 1.0, while lower scores represent worse prediction [25]. The EVS is defined by Eq (10):

$$EVS = 1 - \frac{Var\{P_{measured} - P_{predicted}\}}{Var\{P_{measured}\}} \tag{10}$$

- **Median absolute error (MAE)**

MAE is a risk metric according to the expected value of the absolute error. It is a non-negative floating-point, and its best value is 0. MAE is defined as:

$$MAE = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \left|(P_{measured})_i - (P_{predicted})_i\right| \tag{11}$$

Where $P_{measured}$ is the measured value from the SCADA database, and $P_{predicted}$is the predicted wind power from deep learning modelling.

2.3.2 The amplitude and phase error

The amplitude error shows if the predicted power is overestimated or underestimated, but the phase error is a result of a timing shift between forecasted and real data (Figure 2.3). With these two types of errors, the standard deviation error (SDE) can be defined [20]:

$$SDE = \sqrt{\frac{1}{M-1}\sum\nolimits_{i=1}^{M} \left(E_i(l) - \widehat{E}_\iota(l)\right)^2} \tag{12}$$

$$SDE^2 = SD_{bias}^2 + DISP^2 \tag{13}$$

here the $\widehat{E}_\iota$ is the mean normalised error, SD_{bias} is amplitude error, and DISP is the phase error.

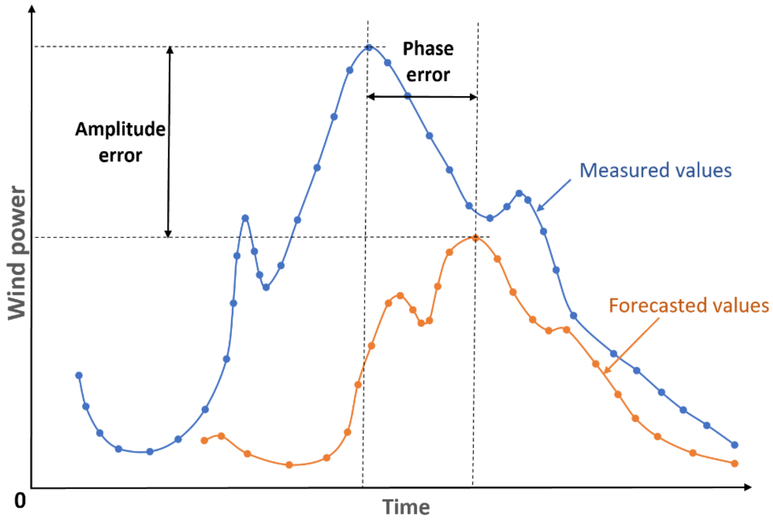


Figure 2.3. Amplitude and phase errors of wind power forecasting

2.3.3 Statistical error distribution

Error distribution can be investigated by two statistical metrics: the skewness (SKEW) and the Kurtosis (KURT). The SKEW is an estimate of the symmetry of the distribution. If it is near zero, the distribution is symmetric, but negative or positive values shows the distribution is inclined to the left or the right respectively. On the other hand, the KURT explains the degree of the distribution peak and shows the distribution of the data rather than a normal distribution [49]. These parameters are expressed in Eq (14) and Eq (15):

$$SKEW = \frac{M}{(M-1)(M-2)} \sum_{i=1}^{M} \left(\frac{E_i - \widehat{E_\imath}}{SD} \right)^3 \tag{14}$$

$$KURT = \left\{ \frac{M(M-1)}{(M-1)(M-2)(M-3)} \sum_{i=1}^{M} \left(\frac{E_i - \widehat{E_\imath}}{SD}\right)^4 \right\} \times \frac{3(M-1)^2}{(M-2)(M-3)} \tag{15}$$

2.4 Enhancement of predictive accuracy

Almost all current modelling efforts to predict wind power generation are aimed at forecasting errors. These efforts have led to various enhancements, which are summarised below.

2.4.1 Kalman filtering

Since the accuracy of NWP data has a significant effect on the accuracy of wind power prediction, one method to improve its performance is to reduce the uncertainty of NWPs. For this purpose, the Kalman filtering algorithm is often used to eliminate systematic errors. The Kalman filter, as a group of mathematical equations, presents the optimal estimation by merging last weighted observations to mitigate related biases. This method can easily adapt to any change in observations and does not need a long series of basic information.

In a research by Louka et al. [50], Kalman filtering was applied to improve the input data for the model that predicted wind power. The results showed that Kalman filtered wind information improved the model for long-term forecasting. These results also showed that, instead of spending money for high-resolution applications (< 6 km), a combined more moderate NWP resolution and a flexible statistical technique of Kalman filter can be used to provide more accurate results. Marcos et al. [44] used a Kalman filter twice in their forecasting model for a wind farm in Brazil. The first implementation of the Kalman filter was for wind speed forecasting error, while, in the second application, the goal was eliminating the systematic errors of the wind speed to wind power conversion model. The latter, in particular, was due to the impact of other variables on the generated power, such as roughness, air density, and wake effect. The results showed a clear reduction in RMAE after the application of the Kalman filter.

2.4.2 Outlier detection

Outliers of SCADA data, which can lead to the inaccuracy of wind power prediction, are usually caused by non-calibration of sensors or degradation over time [51]. As a technique of improving the model accuracy, detection and elimination of these outliers have been investigated in previous studies. Yang et al. [52] used an algorithm for pre-processing SCADA data for CM quality enhancement after examining the influencing factors of a wind turbine, including structural integrity and turbulence. Manobel et al. [53] applied a Gaussian Process (GP) for detecting and removing outliers from SCADA data, where RMSE was improved by 25% in comparison with standard forecasting methods. Besides, Lin et al. [30, 36] used IF to deal with outliers to increase accuracy. The results showed that pre-processing the SCADA data leads to more accurate forecasting.

2.4.3 Optimal combinations

By combining different NWP data and prediction methods, individual benefits can be merged. This combination also reduces the negative effect of errors on each technique in certain situations. In the explanation section of the hybrid methods in section 2.1.2, references were made to several compounds of different approaches and their effect on increasing efficiency. In this section, the optimal combination of NWP data will be discussed.

Vaccaro et al. [54] designed an adaptive framework for wind power forecasting based on a combination of different data sources. The novel part of their investigation was a flexible supervised learning system called the Lazy Learning algorithm, which combined meteorological data from different sources. This algorithm was able to be continuously updated. Using 12 months of wind speed observations of a generator site in Italy, the authors assessed the forecasting data by MAE and Mean Square Error (MSE). The results showed that the proposed model outperformed local atmospheric models in wind power prediction. Peng et al. [55] showed that combining physical and statistical forecasting techniques can improve the accuracy after evaluating a proposed ANN model. The authors achieved an 80% reduction in RMSE. In another study, Lange and Focken [56] presented in details the benefits of the combination of NWP models in a German weather service.

2.4.4 Input parameters

To establish the most efficient wind power prediction model, the next critical factor is the selection of the best input features from the system [14]. This selection is extremely important in increasing the accuracy of the predicting models. As shown in Table 2.2 ~ 2.4, wind speed is the most used input variable for wind power forecasting. This is because the wind power is proportional to the cube of wind speed according to the Eq. 16 [57], where ρ is air density, R is radius of the wind turbine blades, C_p is power coefficient and u is wind speed.

$$\mathrm{P} = \frac{1}{2}\rho\pi R^2 C_p u^3 \tag{16}$$

Zhang et al. [17] considered wind power data of three wind farms in North China in their investigations. The results showed that wind speed, among all the NWP data, is the most important influencing parameter in terms of accuracy. The authors displayed this fact by comparing the forecasting results of two high accuracy wind speed data of wind farms #10 & #16 with wind farm #58. The authors also noted that the performance of the forecasting method in very sensitive to the change in the location of the wind turbines, because of the change in the wake effect, topography, and shadow effect.

Wind direction is another factor that influences power generation. Considering the current design of a wind turbine, turbines are allowed to face into the wind during the time of operation [25]. Singh et al. [8] showed that wind speed and wind direction were the two most influential factors on wind power prediction through their MLP prediction model.

Lin and Liu [25] identified that wind speed, wind direction, temperature, and humidity had been the most used input features through their reviewed literature. They proposed a novel hybrid model, using wind speed in different heights, blade pitch angle, temperature, yaw error, and nacelle orientation as input features. Blade pitch angle was used because it plays a vital role in the adjustment of the blades to obtain safe power generation. After discussing the effect of air density on wind power (according to Eq. 16), the authors demonstrated that air density itself depends on air temperature, air pressure and relative humidity. Figure 2.4 gives a view of how different input features were used in the reviewed literature. As can be seen, apart from wind speed, other variables like temperature, wind direction, relative humidity and air pressure are also often used.

Choosing the correct type of input features for a wind energy prediction model is critical to its performance. This has led to various research into input selection, data processing, as well as combining different input information to increase the accuracy of the models. As discussed, wind speed is the most significant parameter for wind power prediction. However, additional parameters have also been used to consider the benefits of atmospheric data etc. Giorgi et al. [14] investigated the impact of numerical weather parameters on the performance of a wind farm power prediction, such as daily wind speed, pressure, relative humidity and ambient temperature. The authors designed eight different forecasting models with a variety of combinations of different ANNs and numerical weather parameters. The assessment of those models by normalised mean absolute percentage error (NMAPE) revealed that, apart from the clear importance of the predicted wind velocity, the pressure and temperature bring the highest influence on the prediction model among other NWPs parameters. Furthermore, Lange et al. [58] included the predicted wind speed at 100 m height in their investigations. The results showed that the prediction errors (RMSE) decreased by more than 20%.

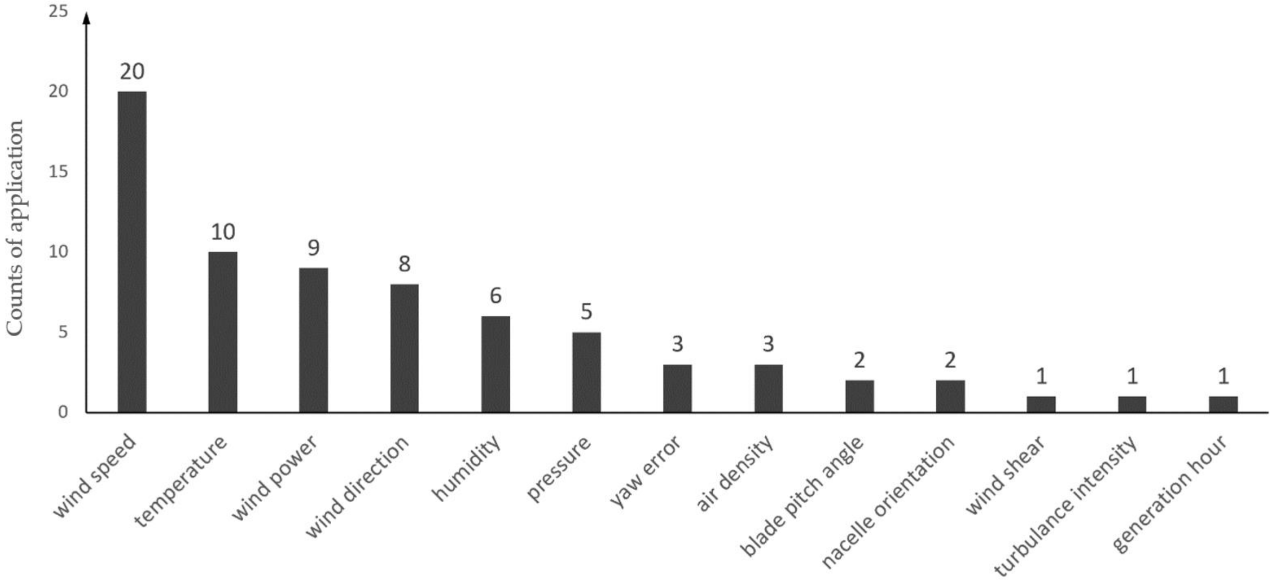


Figure 2.4. Statistics of used input features in the reviewed literature

In another investigation, Velazquez et al. [59] assessed the influence of three input variables for ANN models, including wind speed, wind power density, and power output. It was concluded that considering the wind direction as an input will lead to a decrease in forecasting error (MARE).

The results of the research from Bilal et al. [28] on input and output data of four different sites in Senegal showed that higher rates of the standard deviation of wind velocity could lead to lower average fitting rate for prediction. The authors also proved that considering other climatic variables such as temperature, humidity, and solar radiation can lead to an improvement of 0.3% in accuracy. They also showed that using wind direction improved the fitting rate of their method's prediction by 0.25%.

Apart from the effect of the type of input data on accuracy and performance of wind power prediction models, the data period, as well as the sampling rate, is also important. The short period of data cannot provide sufficient information for training refined prediction models. On the other hand, when the period is long, it will not be representative for the current state of the wind farm. Recent investigations have shown that the older part of the long data will lead to distortions in the prediction [21]. Duran et al. [21] used different training periods from 3 months to 2.5 years to assess the training period on the prediction accuracy. The results showed that, although data with a period longer than one year have similar accuracies, the results of the case of 2.5 years is worse than two years. The reason for the distortion in the prediction caused by the influence of the older part of the data. Among the cases with lower than one year periods, the shortest period had the least accuracy.

2.4.5 Hyperparameter optimisation

One of the most important ways to increase the accuracy of machine learning methods is the appropriate selection of their hyperparameters [3]. Hyperparameters are parameters that govern the behaviour of the learning algorithm and influence its performance on new data. Unlike model parameters, hyperparameters are not directly learned from the training data but can be inferred through methods such as maximum likelihood estimation or learned using techniques like Reversible Jump MCMC. Hyperparameters determine how the algorithm behaves and how it generalises to new data and how well it is trained.

Hyperparameters can be tuned manually, however this takes a lot of time. As a result, grid and random search methods are widely used to set up a network of hyper-parameters and then run the train, predict, and evaluate cycle automatically [60]. Nevertheless, without considering the past evaluated hyper-parameters, these tuning methods are relatively inefficient as they spend a significant amount of time evaluating improper hyper-parameters, i.e., the inaccurate selection of activation functions of deep learning models. Bayesian model-based methods in contrast, through evaluation of hyper-parameters that appear more promising, can find better hyper-parameters in less time [61]. Bayesian model-based methods have been used in the literature for hyperparameter optimisation of ML models. For example, for detecting network intrusion, Masum et al. [62] used Bayesian optimisation to find the best hyperparameters for deep neural networks. In the field of wind power forecasting, Zha et al. [63] utilised the Tree-structured Parzen Estimator (TPE) algorithm to obtain the best hyperparameters of the temporal convolution network (TCN), but they did not compare the optimisation performance of the applied algorithm with other optimisation methods. In another study, Hanifi et al.[3] used the TPE search method to optimise the LSTM model for wind power prediction. Although in the comparison of the accuracy and efficiency of the proposed method with the conventional grid search method, the authors proved the better optimisation performance of the TPE, they did not investigate other smart and advanced hyperparameter tuning methods.

2.4.6 Data decomposition

Signal processing methods such as data decomposition, data denoising, or data feature selection can improve the accuracy of power forecasting methods. All decomposition-based forecasting models published in the literature use the same framework. In this framework, the original non-stationary time series is decomposed into stationary sub-series. Afterwards, independent forecasting models are used to predict each sub-series. Finally, all predictions are added together to form the final forecast. Independent forecasting of each sub-series can efficiently enhance the prediction accuracy [64]. Decomposition is also effective when frequency is considered. Due to the subseries' concentrated frequency bands, predictors only need to focus on a single band,

and forecasting becomes easier. Several researchers have used data decomposition to improve wind speed/power forecasting models in the past few years, some of which are highlighted below.

Su et al. [65] decomposed the wind speed data into four low-frequency and four high-frequency components by WPD. Then the four high-frequency components were decomposed into 60 intrinsic mode functions (IMFs) through ensemble empirical mode decomposition (EEMD). These components were then fed to individual LSTM models with yaw error and rotor speed data. The power prediction results of the proposed approach showed an improvement in accuracy. However, the effect of the direct application of the wind power dataset for prediction was not investigated. Zu et al. [66] used WPD to decompose wind power time series into three levels. The gained sub-series were fed to a gated recurrent unit (GRU), and the predictions were reconstructed to obtain the results. Experimental results showed that the proposed WPD-GRU-SELU model had a higher prediction accuracy than other Recurrent Neural Network (RNN) models. In another research, Mujeeb et al. [67] combined the Wavelet Packet Transform (WPT) method and Deep convolutional neural network (DCNN) to predict the day-ahead hourly wind power of ISO New England's wind farm. However, the authors did not attempt to forecast the sub-series with different independent methods. In addition to WPD, other wavelet transform methods have recently been used in the field of wind power prediction. For example, Azimi et al. [68], with a combination of the K-means clustering method with discrete wavelet transform (DWT) and multilayer perceptron neural network (MLPNN), improved the wind power forecasting accuracy of the National Renewable Energy Laboratory (NREL). Shi et al. [69] employed variational mode decomposition (VMD) and LSTM to provide hourly predictions of day-ahead wind power of a Chinese wind farm. In another study, Liu et al. [70] combined empirical mode decomposition (EMD), LSTM and Elman neural network (ENN) to develop a hybrid model and obtained satisfactory results for multi-step ahead wind speed predictions. To obtain better forecasting results, some researchers use error correction mechanisms through application of the double decomposition methods. For example, Ma et al. [71] used this decomposition approach with the LSTM model and obtained a better prediction performance of the proposed model compared to models without double decomposition.

2.4.7 Statistical downscaling

Statistical downscaling to increase the quality of NWP data is used to improve wind power forecasting. NWPs are usually provided for a wider area than the wind farm location while by statistical downscaling higher-resolution computations can be employed to estimate wind speeds at the wind turbine location. Power predictions with this downscaled NWP have higher accuracy. Al-Yahyai et al. [72] showed the impact of this factor by discussing the reduction of the prediction error as a result of higher resolution. The authors proved that the increment of the model's resolution to 7 km, significantly enhances the accuracy of wind speed estimations.

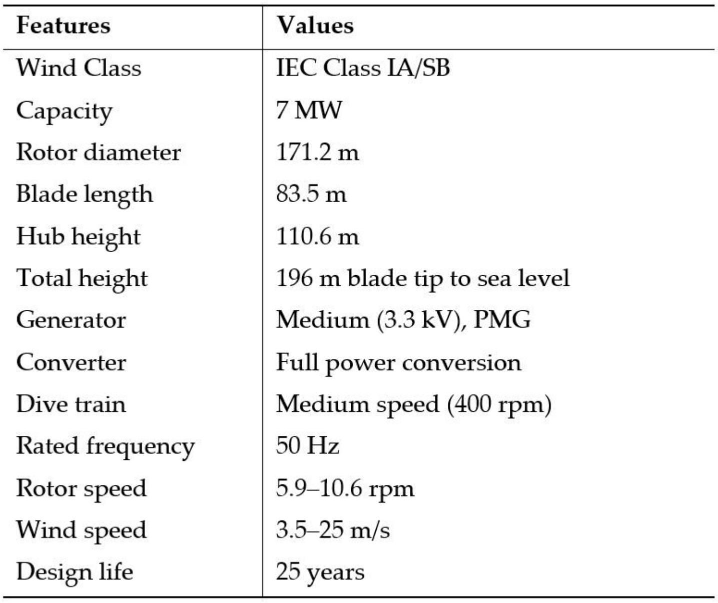
Chapter 3 - Data Pre-processing

Pre-processing in the context of wind power prediction refers to the steps involved in preparing and transforming raw wind data before feeding it into a predictive model. The goal is to improve the quality and relevance of the data, remove noise or outliers, and extract features that can improve the accuracy of wind power forecasts. Pre-processing techniques vary depending on the characteristics of the data and the requirements of the prediction model. A description of the wind power data used in this thesis is presented in this section, followed by a discussion of the pre-processing methods employed.

3.1 Wind power dataset

The source SCADA data used in this research are measured at a 1 Hz frequency from the Levenmouth Demonstration Turbine (LDT). This offshore wind turbine is located just 50 m from the coast at Leven, a seaside town in Fife, Scotland [73]. The wind turbine was acquired by the Offshore Renewable Energy (ORE) Catapult in 2015, while its construction was completed by Samsung in October 2013 [74].

ORE Catapult's wind turbine is a three-bladed upwind turbine installed on a jacket structure [42]. The turbine is rated to work at 7 MW, but to decrease the noise, it is limited to operating at the highest power of 6.5 MW [73]. The turbine's rotor diameter is 171.2 m, and its hub height is 110.6 m. Each blade of the turbine measures 83.5 m and weighs 30 tons. The defined cut-in speed is 3.5 m/s, which means its electricity generation will start when wind speeds reach this value. The turbine will shut down if the wind blows too hard (roughly 25 m/s) to prevent equipment damage. Its operating temperature is between −10 °C to +25 °C, and it has been designed to work for 25 years [75]. Figure 3.1 shows the configuration and main parameters of the LDT.



Features	Values
Wind Class	IEC Class IA/SB
Capacity	7 MW
Rotor diameter	171.2 m
Blade length	83.5 m
Hub height	110.6 m
Total height	196 m blade tip to sea level
Generator	Medium (3.3 kV), PMG
Converter	Full power conversion
Dive train	Medium speed (400 rpm)
Rated frequency	50 Hz
Rotor speed	5.9–10.6 rpm
Wind speed	3.5–25 m/s
Design life	25 years

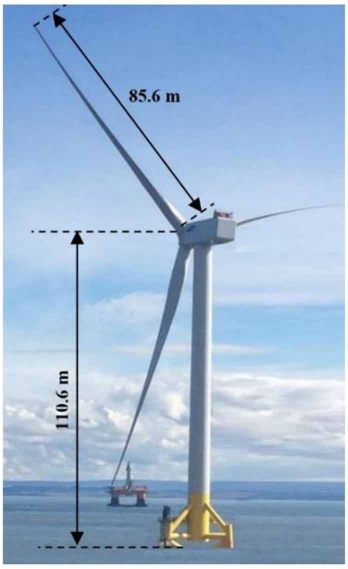
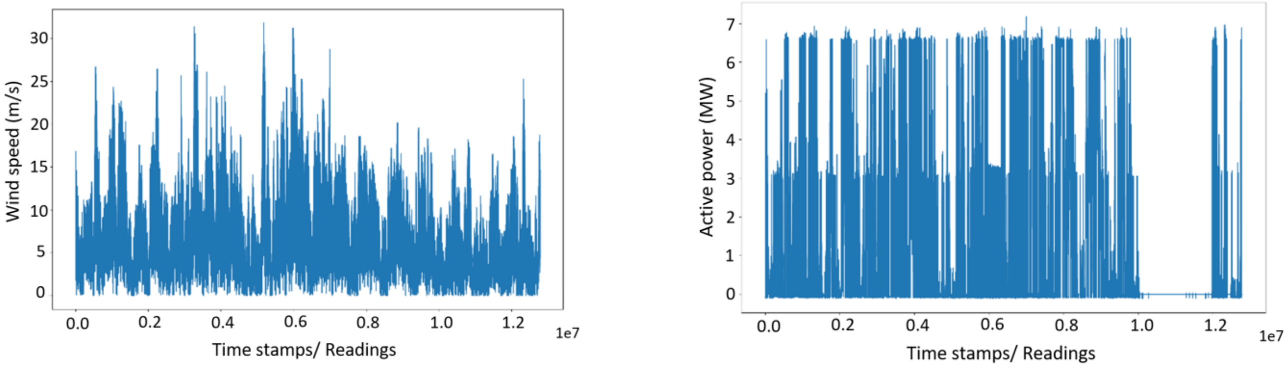


Figure 3.1. Main parameters and schematic of Levenmouth wind turbine [35].

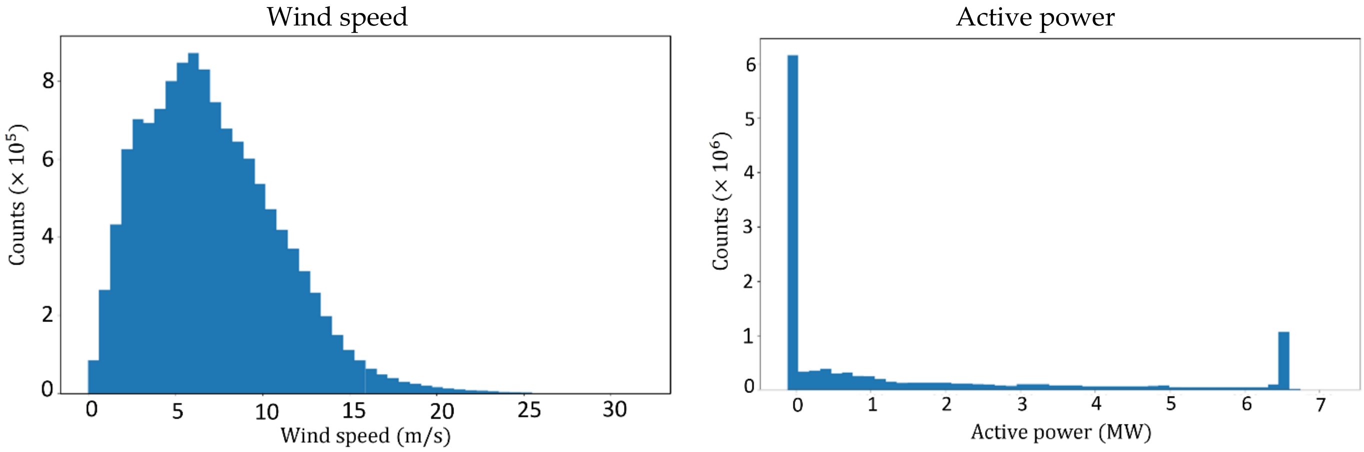
3.2 Feature Selection

This study used the SCADA datasets for five months, from 1st of January 2019 to 31th of May 2019, at a 1 Hz frequency (with one-second intervals). Each timestamp in the time-series data includes 574 different observations, including the generated power, wind speed at different levels, blade pitch angle, nacelle orientation, etc. At the beginning of the data processing, a feature selection was carried out to decrease the size of the dataset to reduce the computation time by excluding unnecessary variables. All variables except the time stamp, wind speed, and active power were removed at this stage, which were not necessary in the ARIMA and univariate LSTM and CNN forecasting methods. Keeping the wind speed variable was vital, as it verified the accuracy of the generated power. For example, failure to generate power when high wind speeds were recorded was recognised as a stop in power generation due to reasons such as maintenance. After removing the redundant information, observations of wind speed and active power were plotted as shown in Figure 3.2a and 3.2b. The histograms of this dataset for wind speed and active power are presented in Figure 3.3a, 3.3b, and Table 3.1 shows their statistical descriptions.



(**a**) (**b**)

Figure 3.2. Wind speed observations (a), wind active power observations (b).


Wind speed

Active power

(**a**) (**b**)

Figure 3.3. Histogram of wind speed (a) and active power (b).

Table 3.1. Statistical descriptions of the SCADA datasets.

	Active Power, kW	**Wind Speed, m/s**
Count	1.0×10^{7}	1.0×10^{7}
Mean	1.8×10^{3}	7.6
Standard deviation	2.3×10^{3}	3.9
Minimum	-1.2×10^{2}	-3.3×10^{-2}
25% of values	−6	4.7
Medium	5.9×10^{2}	7.1
75% of values	3.2×10^{3}	1
Maximum	7.2×10^{3}	3.2

3.3 Obvious Outlier Removal

An initial assessment of Figure 3.2b highlights that a large part of the recorded generated power at the end of this time-series (May 2019) is equal to zero. Usually, the generated power of a turbine can be zero when no wind is blowing. However, the evaluation of Figure 3.2a shows a continuous wind blowing with fluctuations similar to previous months. Therefore, it is speculated that the turbine was out of action during this period. Based on this assumption, it was decided that this month (May 2019) should be removed from the dataset. The time-series after this omission was reduced to four months, from 1 January 2019 to 30 April 2019. A closer look at the active power, as shown in Figure 3.4, revealed another error in the SCADA data, namely the existence of negative values. Negative values do not represent any practical meaning in wind power generation. Shen et al. [76] believe that these values represent time stamps when turbine blades do not rotate, but the turbine's control system requires electricity [76]. These values need to be eliminated along with the corresponding parameters of the same timestamp for better forecasting results [42]. Since the elimination of these negative values disrupts the time continuity of the time-series, and can possibly lead to errors in wind power prediction, at this stage it was decided to create and assess three types of datasets based on different actions against negative values.

The three pre-processing methods against the negative values are:

1) Total elimination of negative values without any substitution;
2) Replacement of negative values with the average amount of power in the whole 4-month period;
3) Replacement of negative values with positive values of power at the nearest timestamp.

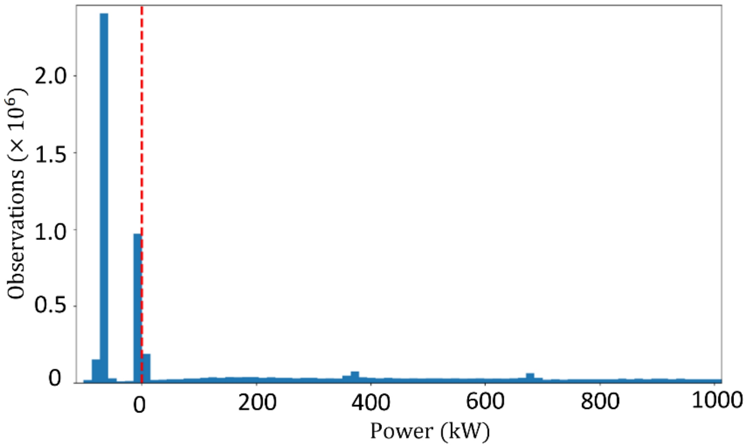


Figure 3.4. Wind power observations (only power values under 1000 kW are shown). The dotted red line indicates the power value of zero (the boundary of negative/positive values).

3.4 Resampling

The effect of wind turbulence as one of the obstacles to increasing the wind energy penetration into the energy market is more significant in horizontal axis wind turbines [43]. This is because the wind speed and direction change rapidly after hitting swept blade rotors. Therefore, the wind speed measurements by installed anemometers are not equal to the speed of the wind flow hitting the turbine blades [42]. These differences, which lead to a decrease in the correlation between the measured wind speed and the output power, and then scattering of the power curve, can be resolved by averaging the samples of both wind power and wind speed in a reasonable average period [42]. The SCADA data for this study was recorded with a 1 Hz frequency; as a result, it was possible to create multiple averaged sets for removing the mentioned obstacle. The maximum sampling rate used for wind speed and power forecasting in the literature is 10 min [7]. This is equivalent to an average time that the international standard for power performance measurements of electricity-producing wind turbines (IEC 61400-12-1) establish a max sample rate of 10 min for large wind turbines [77]. Based on the IEC 61400-12-1 and the reviewed literature, the data presented here was averaged for each 10 min of data collection. Figure 3.5a and 3.5b show the wind power curves for the original and 10 min resampled data.

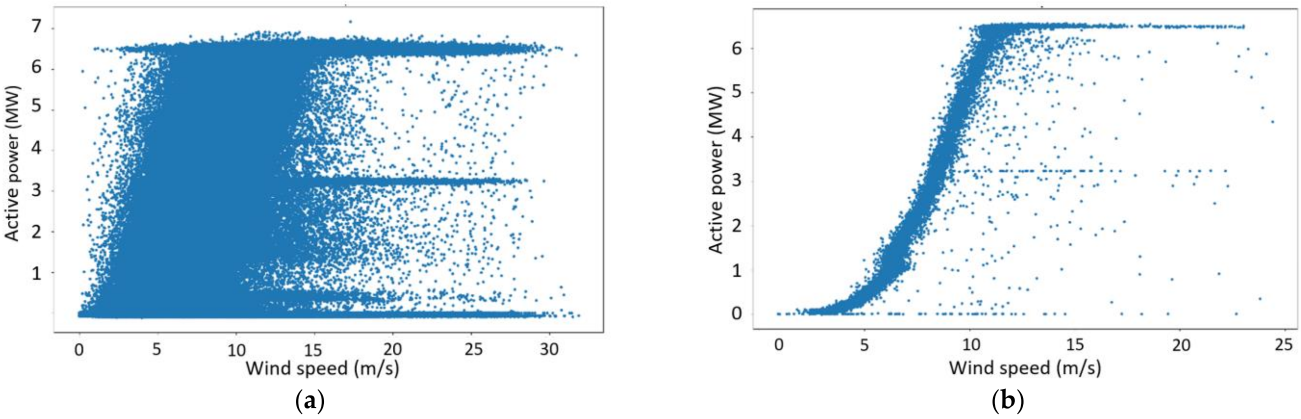


Figure 3.5. Wind power curves: (a) original 1s data, and (b) 10 min resampled data.

3.5 Outlier Detection and Treatment

Outliers in a dataset are specific data points that are different or far from most other regular data points, often due to variability in the data or measurement errors [78]. These anomalies can arise from various sources, including sensor malfunctions, data entry errors, or rare events. Undetected or improperly treated anomalies can adversely affect wind power forecasting applications, leading to biases with high prediction errors [78]. Proper identification and management of outliers are crucial to ensuring the robustness and accuracy of forecasting models.

There are various reasons for having outliers among wind turbine and wind farm measurements, including wind turbine downtime [76], data transmission, processing or management failure [79], data acquisition failure [80], electromagnetic disturbance [76], wind turbine control system fault (such as the pitch control system fault) [81], damage of the blades or the existence of ice or dust [82], shading effect of neighbouring turbines, and fluctuation of air density [83] to name a few.

Figure 3.6 shows four different types of anomalies in the current SCADA data. Category A points have negative, zero, or low values of generated power during speeds larger than the cut-in speed [42]. The leading causes of these outliers are incorrect wind power measurements, wind turbine failure, and unexpected maintenance. Wind speed sensors and communication errors cause category B outliers. The mid-curve outliers (category C) represent power values lower than ideal—this is caused by the down-rating of the wind turbine and data acquisition. Outliers in category D are scattered irregular points due to faulty sensors exacerbated during harsh weather conditions [76].

There are different methods for anomaly detection in machine learning, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Isolation Forest (IF), local outlier factor, and Elliptic Envelope (EE). In this study, three common methods for wind power forecasting are investigated. EE outlier detection method assumes a multivariate Gaussian distribution underlines most of the data. Observations that deviate significantly from this assumed Gaussian distribution are considered outliers [43]. IF, which is an unsupervised learning algorithm, recognises anomalies by isolating them in the data. This algorithm works based on two main features of anomalies, that they are few and different. It operates by recursively partitioning the data space using randomly selected features and split values. Anomalies, which are few and different, require fewer partitions to be isolated compared to normal data points. This results in a shorter path length in the tree structure used by the Isolation Forest, effectively distinguishing anomalies from regular data points. Its efficiency and capability to handle high-dimensional data make Isolation Forest a robust method for detecting outliers. The One-Class Support Vector Machine (OCSVM) is a common unsupervised learning algorithm for outlier detection, assuming rare anomalies create a boundary for most data, and considering data points out of the boundary as outliers [84]. OCSVM works by mapping input data into a high-dimensional feature space using a kernel function and then finding the optimal hyperplane that best separates the normal data points from the origin. The algorithm maximizes the margin between the data points and the hyperplane, effectively creating a decision boundary. Data points lying outside this boundary are identified as anomalies. This method is particularly effective for datasets with complex distributions, where the definition of normal and anomalous data is not straightforward. This method of outlier detection and treatment is selected as the third method.

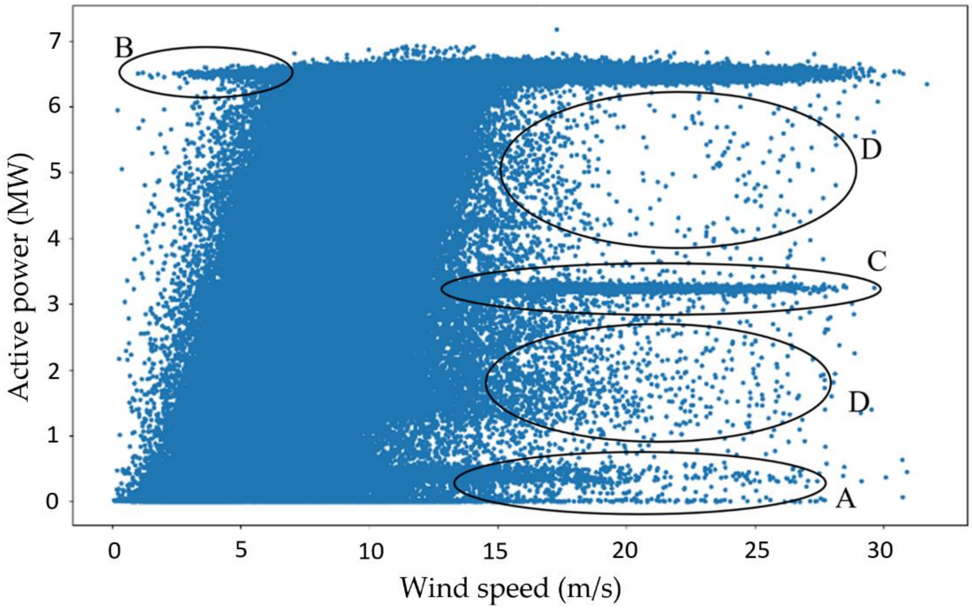


Figure 3.6. Observed anomalies coupled with the power curve of the 1 Hz original data. (A) Low power output in high wind speeds in turbine failure cases; (B) Outliers due to the wind speed sensor and communication errors; (C) Power outputs less than the rated power as a result of the turbine's down-rating; (D) Scattered outliers caused by sensor malfunctions or noise in signal processing.

3.6 Prediction models

In this work, various forecasting methods have been used in different sections including Linear Regression (LR), Random Forest (RF), ARIMA, Feed Forward Neural Network (FFNN) to deep learning methods including CNN and LSTM. In this section, the three methods of ARIMA, LSTM and CNN are explained in detail.

3.6.1 ARIMA

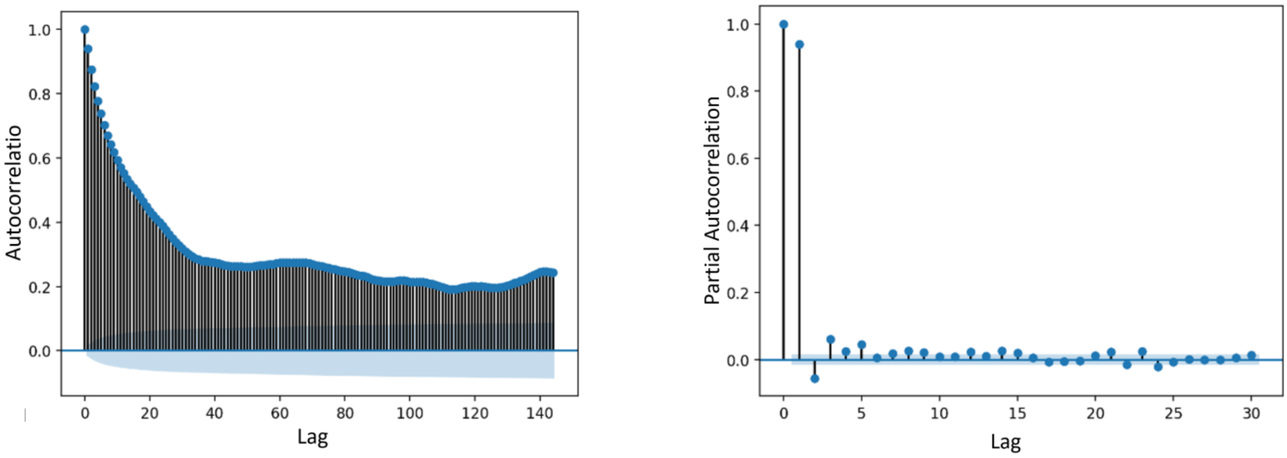
The standard approach of the Box–Jenkins method [85] was used for the ARIMA model development. The ARIMA model is a widely used set of statistical models for analysing and predicting time-series data [86]. This model can be expressed as [87]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (17)$$

while ϕ_t and θ_t are coefficients, p, q, and d are the lag number of observations in the model, the order of moving average, and the degree of difference, respectively. Degree of difference (d) values greater than 0 imply that the data has been nonstationary but has become stationary after some degree of difference.

The ARIMA model combines the AR, moving average (MA), and the Integrated (I) components, where the Integrated component represents the substitution of the data with the difference between its values and the preceding values [88]. The forecasting accuracy of the ARIMA model depends on selecting the most appropriate combination of p, d, and q. Normally, for small data sets, the autocorrelation function (ACF) and partial autocorrelation function (PACF) can be used to determine which AR or MA component should be selected in the ARIMA model [89]. These two factors, which can be graphically plotted, are widely used elements in analysing and predicting time-series. They highlight the correlation between an observation and the observations' value at prior time steps.

The difference between the ACF and PACF lies in how they measure relationships between time series observations. ACF considers the correlation between observations at different time lags, considering all intervening correlations. In contrast, PACF measures the direct correlation between observations at two time steps, effectively removing the influence of the intermediate observations. This means that PACF isolates the impact of each lag by controlling for the correlations of the shorter lags, providing a clearer view of the direct relationships between observations separated by various lags. Figures 3.7a and 3.7b show the observations' ACF and PACF plots for wind power. An appropriate ARIMA model can be selected based on the explanations in Table 3.2 [90]. The value of d (degree of difference) depends on the number of differencing operations needed to transform the time series data into a stationary form. Stationarity implies that the statistical properties of the time series, such as the mean and variance, remain constant over time. If the original time series data exhibit trends or other non-stationary behaviours, differencing helps to remove these trends by subtracting the previous observation from the current observation. This process is repeated until the resulting time series shows no significant trends, indicating that stationarity has been achieved.



(**a**) (**b**)

Figure 3.7. ACF (a) and PACF (b) plots for generated power of LDT. The blue points represent the value of autocorrelation and partial autocorrelation of different time lags.

Table 3.2. ACF and PACF application for statistical model selection.

Model	Autocorrelation	Partial Autocorrelation
AR (*p*)	Tails off gradually	Cuts off after *p* lags
MA(*q*)	Cuts off after *q* lags	Tails off gradually
ARMA (*p*, *q*)	Tails off gradually	Tails off gradually

The ARIMA model forecasting steps after resampling and outlier treatment can be seen in Figure 3.8. The first step is assessing the stationarity of the time-series. Stationarity is one of the assumptions during time-series modelling, which shows the consistency of the summary statistics of the observations.

When a time-series is stationary, it means that the statistical properties of the time-series (such as mean, variance, and autocorrelation) do not change over time. This property can be violated by having any trend, seasonality, and other time-dependent structures. There are two main methods for the stationarity assessment of time-series: the visualisation approach and the augmented Dickey–Fuller (ADF) test. The visualisation method uses graphs to show whether the standard deviation changes over time. On the other hand, the ADF method is a statistical significance test used for the ARIMA model. It compares the p-value of the test statistic with the critical values to perform hypothesis testing. The null hypothesis (H0) is that the time series has a unit root (non-stationary), and the alternative hypothesis (H1) is that the time series does not have a unit root (stationary). This test clarifies the stationarity of the data at different levels of confidence by determining whether a unit root is present in the time series, thus indicating if differencing is required.

Regarding the data used in this study, due to the high number of observations and wide dispersion, checking stationarity through visualization is not that accurate. Therefore, in this study, the ADF method was used.

The ADF test's execution provides a p-value which, by comparing it with a threshold (such as 5% or 1%), can identify the stationarity of the data. In this step, nonstationary data in this step needs to be changed to stationary by methods such as differencing, which involves subtracting the previous observation from the current observation. After ensuring the time-series is stationary, a persistence method, which uses the most recent observation as the forecast for the next period, is created as a baseline. Then, through a detailed grid search, the best values of *p*, *d* and *q* hyperparameters for the ARIMA forecasting for each pre-processed data were found. The last step is ARIMA forecasting and comparing its error with the error of the persistence method.

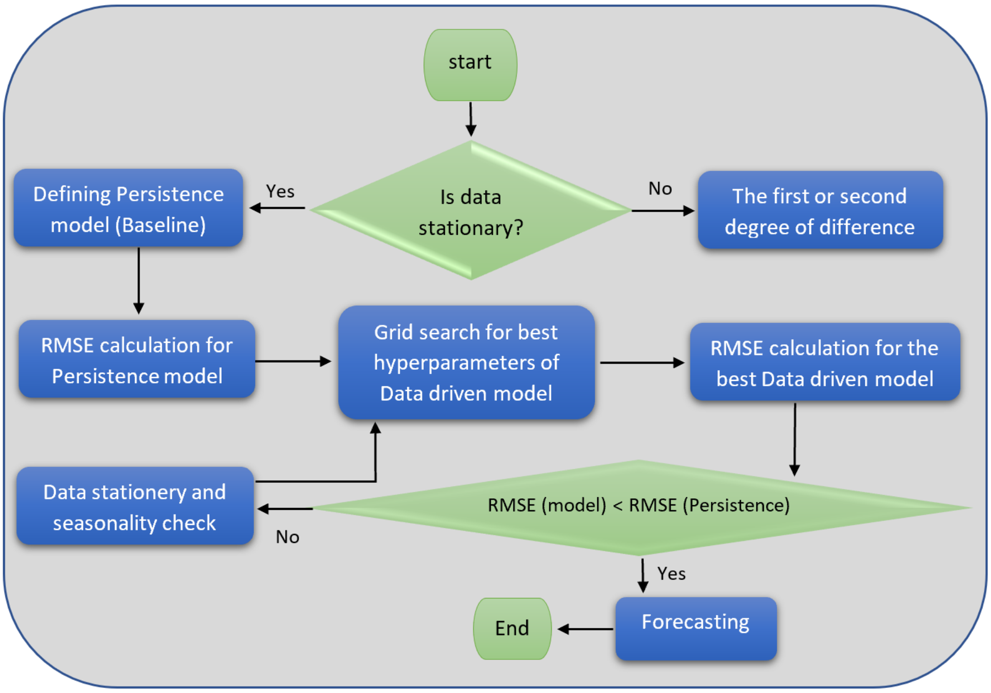

start
Is data stationary?
Yes
No
Defining Persistence model (Baseline)
The first or second degree of difference
RMSE calculation for Persistence model
Grid search for best hyperparameters of Data driven model
RMSE calculation for the best Data driven model
Data stationery and seasonality check
RMSE (model) < RMSE (Persistence)
No
Yes
Forecasting
End

Figure 3.8. Flowchart of ARIMA wind power forecasting models.

3.6.2 Long Short-Term Memory (LSTM)

Deep learning prediction models have transformed various fields by leveraging large datasets to make precise predictions. These models use complex neural network architectures to learn features and patterns from data. Recurrent Neural Networks (RNNs), a type of deep learning model, are particularly effective for tasks involving sequential data. RNN has a high ability to represent all dynamics, meaning they can effectively capture and model the complex temporal dependencies and patterns within sequences. However, its effectiveness is affected by the limitations of the learning process.

The main limitation of gradient-based methods that use backpropagation in RNN models is their path integral time-dependence on assigned weights. This means that the gradients calculated during training can become very small or very large as they are propagated back through many time steps. This issue, known as the vanishing or exploding gradient problem, can hinder the model's ability to learn long-term dependencies and effectively update the weights. As a result, RNNs can struggle to capture and retain information over long sequences, impacting their performance on tasks requiring memory of distant events [17]. When the time lag between the input signal and the target signal increases to more than 5–10 time steps, the normal RNN loses the learning ability, meaning it can no longer effectively capture and utilize the information from earlier time steps to predict future values. This occurs because the gradients used to update the network’s weights during backpropagation either vanish or explode. Vanishing gradients cause the model to stop learning as the weight updates become negligibly small, while exploding gradients cause the model to become unstable with excessively large weight updates. This error elimination raises the question of whether normal RNNs can show practical benefits for feed-forward networks. To address this problem, the LSTM has been developed based on memory cells. The LSTM consists of a recurrently attached linear unit known as the constant error carousel (CEC). CECs, by keeping the local error backflow constant —meaning they ensure that the error signal remains steady and does not diminish as it is propagated back through time—, mitigate the gradient's vanishing problem [42]. CECs can be trained by adjusting both the back-propagation over time and the real-time recurrent learning algorithm [91].

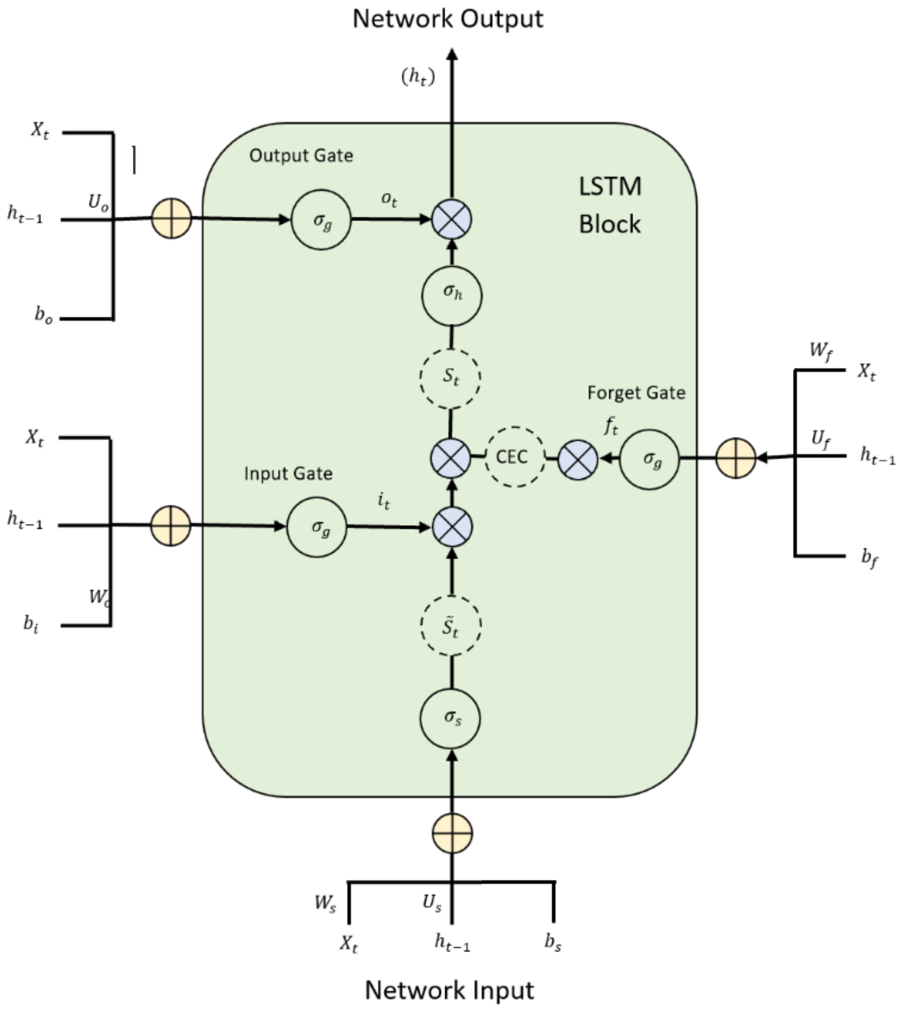


Figure 3.9. Typical structure of the LSTM.

Figure 3.9 shows the typical structure of the LSTM. As can be seen, there are three gate units in a basic LSTM cell, which includes the input, output, and forget gates. The gate activation vectors of i_t, o_t and f_t for input, output, and forget gates, respectively, are calculated in Equations (18)–(20).

$$i_t = \sigma_l(W_i x_t + U_i h_{t-1} + b_i) \tag{18}$$

$$o_t = \sigma_l(W_o x_t + U_o h_{t-1} + b_o) \tag{19}$$

$$f_t = \sigma_l\big(W_f x_t + U_f h_{t-1} + b_f\big) \tag{20}$$

In these equations, W_i, W_o, W_f U_i, U_o, and U_f represent the assigned weight matrices, and b_i, b_o, and b_f represent the biase vectors in conjunction with relevant activation functions σ_l. In addition, x_t is the neuron input at time step t, and the cell state vector at time step t − 1 is h_{t-1}. As shown in Equation (21), the next evaluated value of the state $\tilde{S}_t$ can be calculated based on the relevant activation function σ_s.

$$\tilde{S}_t = \sigma_s(W_s x_t + U_s h_{t-1} + b_s) \tag{21}$$

In Equation (22), the newly assessed value of $\tilde{S}_t$ and the prior cell state S_{t-1} are used to calculate the cell state S_t, which by itself will be used with the output gate control signal o_t and the activation function σ_{lh} to obtain the overall output h_t according to Equation (23).

$$S_t = f_t \circ S_{t-1} + i_t \circ \tilde{S}_t \tag{22}$$

$$h_t = o_t \circ \sigma_{lh}\,(S_t) \tag{23}$$

As evident from Equations (22) and (23), the output h_t is dependent on the state S_t of the LSTM cell and the activation function σ_{lh} that is usually tanh (x). The state S_t depends on the state of the prior step S_{t-1} as well as the new value of the state $\tilde{S}_t$.

The function of the LSTM model can be concluded as:

- Input gate (i_t) controls the extent to which $\tilde{S}_t$ flows into the memory.
- Output gate (o_t) regulates the extent to which S_t gives to the output (ht).
- Forget gate (f_t) controls the extent to which S_{t-1} (i.e., previous state) is kept in the memory.

Specifying the best LSTM model for wind power forecasting requires the determination of the neural network's best combination of hyperparameters. LSTMs have five main hyperparameters, which consists of including the number of lag observations as inputs of the model, the quantity of LSTM units for the hidden layer, the model exposure frequency to the entire training dataset, the number of samples inside an epoch in each weight updating, and finally, the used difference order for making nonstationary data stationary.

3.6.3 Convolutional Neural Network (CNN)

CNN is a particular type of NNs that uses a mathematical function called convolution instead of general matrix multiplication. The basic structure of a CNN model is shown in figure 3.10 This deep learning model like any feed-forward neural network (FFNN) consist of three main layers including input layer, hidden layers, and output layer. The hidden layer is where the convolution function is performed based on a dot product of the convolution kernel with the matrix of its input layer. The convolution process can be presented as follows:

$$h_{ij}^k = f\big((W^k \otimes x)_{ij} + b_k\big) \tag{24}$$

where f, x and b represent the activation function, the vector of input series and the vector of bias, respectively and W^k denotes the connected kernel weights to the kth feature map. In this equation, $\otimes$ symbol indicates the convolution function, and the common activation function f is the Rectified Linear Unit (ReLU). Following the convolution operation on the input matrix of each convolution layer, a feature map is created as the input of the next layer.

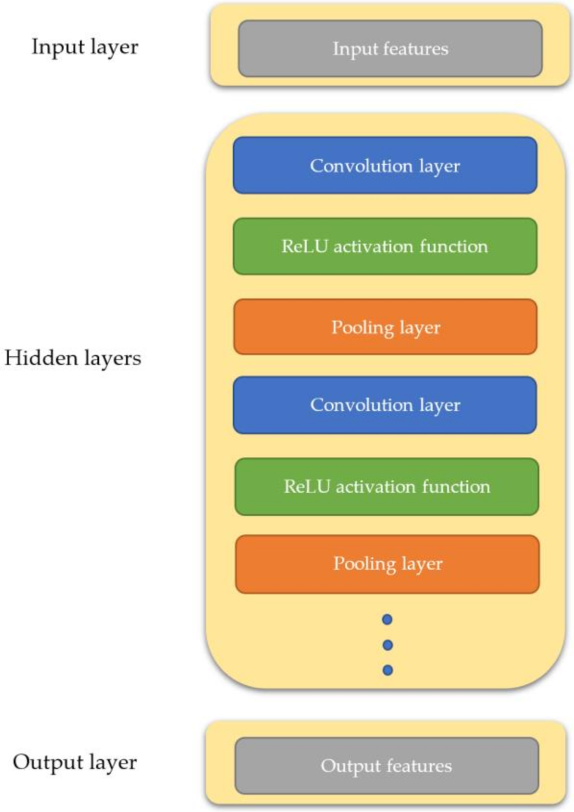


Figure 3.10. Basic architecture of a CNN model

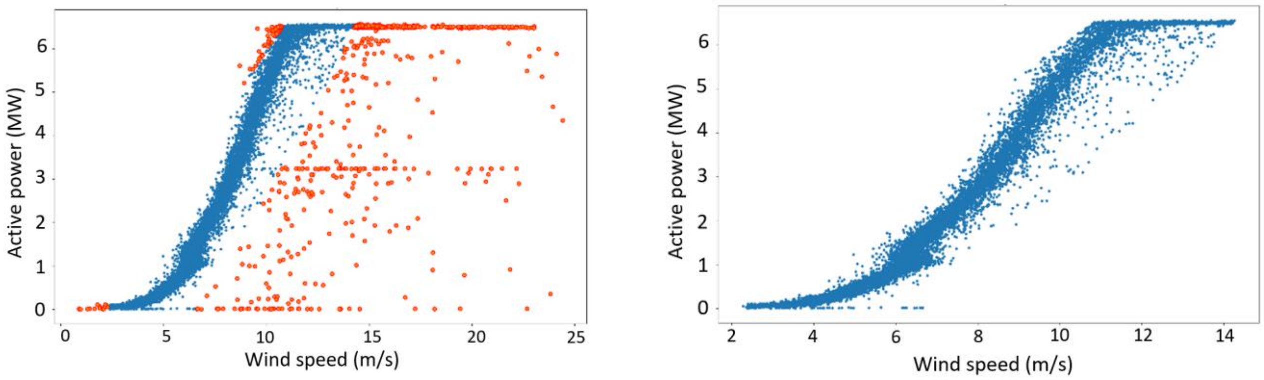
The convolution layer in a CNN is followed by the pooling layer, which performs a non-linear down-sampling operation. The pooling layer reduces the spatial dimensions (width and height) of the data representation, while maintaining the depth. This non-linear down-sampling can take different forms, such as max pooling, which selects the maximum value from each region of the feature map, or average pooling, which computes the average value of each region.

The main purpose of applying the pooling layer is to gradually decrease the size of the data representation, thereby reducing the number of parameters and computations in the network. This helps in controlling overfitting by making the model more generalized and less sensitive to the exact position of features in the input data. By down-sampling the feature maps, the pooling layer also makes the network invariant to small translations and distortions, contributing to the robustness of the CNN.

CNN can effectively extract the hidden non-linear dependencies of time series through automatic creation of filters [92]. As a result, it has been widely used for time series based predictions [31], [93], [94]. The CNN model in this study consists of two convolutional layers. It can also have up to four fully connected dense layers. The channel number of each convolution layer and the quantity of the dense layers is suggested by the hyperparameter optimisation algorithm. Similarly, these hyperparameters were also optimized for the LSTM model, ensuring that both network architectures are well-tuned for the specific task at hand. The impact of pre-processing on wind power forecasting accuracy is examined using ARIMA and LSTM models to determine the most efficient way to deal with outliers. The ARIMA model is applied as a forecasting model because of its short response time and ability to capture the correlations in time series. On the other hand, the LSTM is used to take advantage of its ability to learn non-static features from nonlinear sequential data automatically. These two models have been used many times in past research to predict wind power or speed.

3.7 Results for comparison of different pre-processing methods

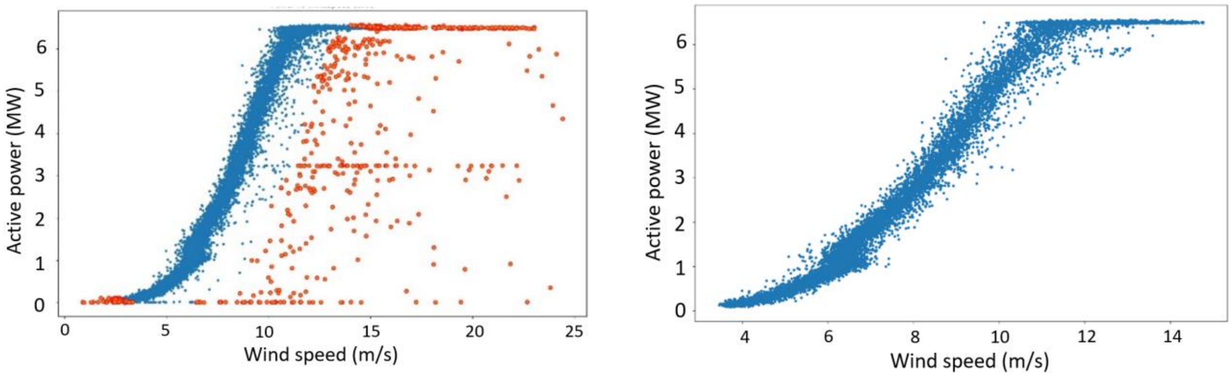
This research employs packages and subroutines written in Python to implement the proposed pre-processing algorithms. A PC with Intel Core i5–7300 32.6GHz CPU and 8 GB RAM (without any GPU processing) is used to run the experiments. Three outlier detection methods, described in the previous sections, are used to detect and remove the outliers of the resampled dataset. The results of these treatments are provided in Figures 3.11 to 3.13. As can be seen, there are differences between different methods in detecting observations as outliers. For example, OCSVM (Figure 3.13) is not able to detect and remove the mid-curve outliers (category C in Figure 3.6) representing the power values lower than ideal.



(a)

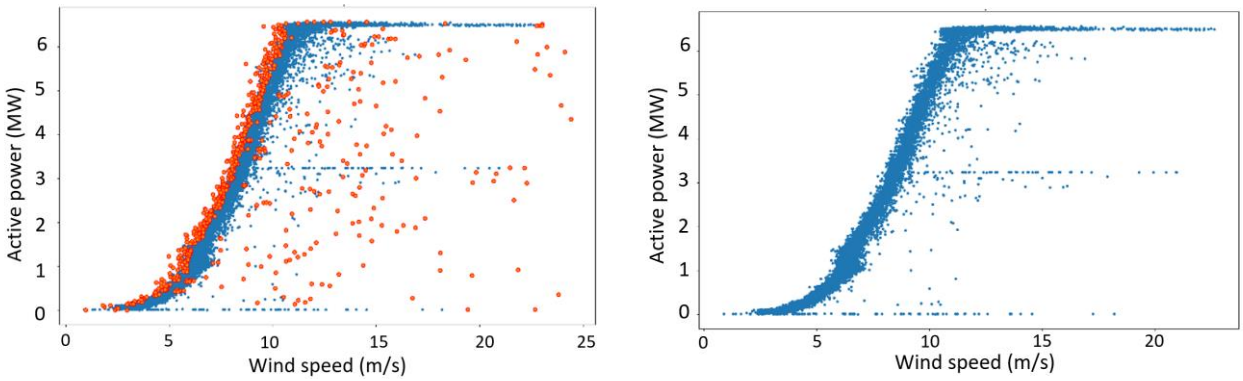
(b)

Figure 3.11. Elliptic envelope application for outlier detection and treatment (a) before outlier treatment and (b) after outlier treatment



(a) (b)

Figure 3.12**.** Isolation Forest application for outlier detection and treatment (a) before outlier treatment and (b) after outlier treatment application for outlier detection and treatment



(a) (b)

Figure 3.13. OCSVM application for outlier detection and treatment (a) before outlier treatment and (b) after outlier treatment

Six different pre-processing methods based on applying three different outlier detection methods and three approaches against the negative power values are considered (Table 3.3). Different cases of pre-processed data are fed to two commonly used forecasting models, ARIMA and LSTM. The grid search method, which is a systematic way of working through multiple combinations of parameter tunes, is applied for the hyperparameter tuning. It involves specifying a range of values for each hyperparameter and exhaustively evaluating every possible combination to identify the optimal set. Table 3.4 shows the selected hyperparameters for the ARIMA and LSTM models. As discussed in previous sections, the considered hyperparameters for the ARIMA model are ppp (order of the autoregressive model), qqq (order of the moving average model), and ddd (degree of differencing). For the LSTM model, the assessed hyperparameters include the number of lag observations, number of LSTM units, exposure frequency, number of samples per epoch, and differencing order. As expected, the values of the hyperparameters vary depending on the different employed pre-processing methods (Table 3.3).

Table 3.3. best ARIMA and LSTM hyperparameters resulted from the grid search

Case data	Pre-processing approach	ARIMA hyperparameters	LSTM hyperparameters
Case 1	Negatives replaced by mean *; Outliers not removed	(2, 0, 1)	(3, 100, 100, 150, 0)
Case 2	Negatives replaced by nearest positive value; Outliers not removed	(1, 1, 1)	(6, 100, 150, 150, 0)
Case 3	Negatives removed; Outliers not removed	(1, 0, 2)	(3, 100, 150, 150, 0)
Case 4	Negatives removed; Outliers removed by EE method	(1, 1, 3)	(3, 100, 100, 150, 0)
Case 5	Negatives removed; Outliers removed by IF method	(3, 1, 1)	(3, 150, 150, 150, 0)
Case 6	Negatives removed; Outliers removed by OCSVM method	(1, 1, 3)	(3, 150, 150, 150, 0)

*: mean value has been calculated after removing negative values

After selecting the best ARIMA and LSTM prediction methods, both models were trained by the first 95% of the dataset (as training data) to make predictions for the remaining 5% of the dataset. The predicted values were compared with the measured values to determine the RMSE of each forecasting process. Table 3.4 provides the RMSE values of ARIMA, LSTM, and persistence methods.

Table 3.4. RMSE values of Persistence, ARIMA and LSTM models for six different treated case data

	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6
Negative values	Mean[1]	Nearest Positive[2]	Removed	Removed	Removed	Removed
Outliers	Not removed	Not removed	Not removed	EE removed	IF removed	OCSVM removed
Persistence	636.3	720.5	830.5	512.7	**509**	566
ARIMA	622.8	713	813	505.3	**503.3**	559
LSTM	626	695.8	785	501.5	497.5	550.6

Comparing the RMSE values of all three models from Table 3.4 for case data 1, 2, and 3 clarifies that the complete elimination of negative values (without any replacement) will lead to worse forecasting. The highest RMSE value of case 3 means that removing the negative values will decrease the forecasting accuracy. One of the reasons for this performance drop can be attributed to the creation of discontinuities in the dataset.

[1] Replaced by mean value calculated after removing negative values

[2] Replaced by nearest positive value

Regarding the best specific value to be used instead of negatives, a comparison of case data 1 and 2 proves that replacing negative values with average wind power values has a better impact than replacing them with the nearest (neighbour) positive value. Replacing negative values with average values can lead to 15% forecasting improvement for ARIMA and 11% improvement for the LSTM models.

The results from Table 3.4 also highlight the importance of dealing with outliers in wind power forecasting. Cases 4, 5, and 6, representing the outlier removed data, show significant enhancement of accuracy compared to the other cases without any action against the anomalies. Comparing the error levels of case data 3 with cases 4, 5, and 6 (for both ARIMA and LSTM models), shows 30% to 38% forecasting improvement by elimination of the outliers either by IF, EE, or one-class SVM outlier detection methods.

Assessment of RMSE values of cases 4, 5, and 6 shows that IF and EE outlier detection methods outperform the OCSVM method. An elliptic envelope can improve forecasting performance by up to 9.61% and 8.92% compared to OCSVM for ARIMA and LSTM methods, respectively. This performance enhancement can reach 9.96% and 9.64% for ARIMA and LSTM by applying the isolation forest.

As shown in Table 3.4, ARIMA and LSTM methods for all treated case have better performances compared to the persistence methods. This is understandable if one remembers that in the persistence method only one preceding step data is used for forecasting, whilst ARIMA and LSTM models consider a more extensive range of prior data.

It is also clear that LSTM performs better than ARIMA for almost all approaches against negative values and outliers. This is believed to be due to the fact that LSTMs are better equipped to learn both short-term and long-term correlations. Additionally, LSTMs can capture the nonlinear dependencies between features more effectively due to their architecture, which allows them to model complex, non-linear relationships and interactions in the data.

Chapter 4 - Hyperparameter optimisation of Deep Learning Models

4.1 Introduction

Over the last decade, various methods have been developed to forecast wind power [7]. One of the most common prediction methods in this field, are machine learning (ML) algorithms. ML algorithms are mathematical functions that represent the relationship between different aspects of data. These algorithms, which spend a significant amount of time training from data, must be configured before training by setting some variables known as parameters and hyperparameters [95].

Parameters, such as the weight at each neuron in a Neural Network (NN), are intrinsic to the model equation and can be determined while the algorithm is being trained. But the hyperparameters, in contrast, are not directly learned by the learning algorithm. They are specified outside of the training procedure and their role is to control the capacity of the models and increase their flexibility to fit the data. This means that hyperparameters, like learning rate, number of layers, or number of units in each layer, can be adjusted to allow the model to better capture the underlying patterns in the data, handle varying levels of complexity, and adapt to different types of datasets. A correct choice of hyperparameters is important to prevent overfitting and improve the generalization of the algorithm.

Hyperparameters have a large impact on the performance of the learning algorithms and their values vary depending on the specific problem domain where the algorithm is used. As a result, they need to be optimised for each dataset [62]. The process of finding the best hyperparameters for a given dataset is called hyperparameter optimisation or hyperparameter tuning. Hyperparameter tuning consists of defining the hyperparameter space, a method for sampling candidate hyperparameters, and a metric that is required to be minimised or maximised. Since it is not possible to define a formula to find the hyperparameters, different combinations of hyperparameters need to be tried and the model performance evaluated at each stage. But the critical step is to determine how many different hyperparameter combinations are going to be tested. Intuitively, the higher the number of hyperparameter combinations, the greater the chance of obtaining a better performing model. At the same time, this leads to greater computational cost, because we will end up training a large number of models simultaneously. It is also important to determine which hyperparameters have a greater effect on the performance of the ML models. [95].

The hyperparameters of simple ML models, such as linear models or tree-based algorithms, can be estimated manually through iterative trial and error. This approach can be very challenging for users who do not have enough professional background and practical experience. To overcome the disadvantages of manual search, automatic search algorithms such as grid search and random search have been proposed.

Grid search does an exhaustive search through training the ML model with all possible combinations of hyperparameters. Then, after evaluating the performance of the model based on a predefined metric, it identifies hyperparameters that achieve the best performance [96]. This search method is used extensively in the literature. For example in the developed wind speed prediction model by Zhou et al. [97], the hyperparameters of support vector regression (SVR) were selected by grid search. In another research, Kisvari et al. [42] applied the grid search to tune the hyperparameters of the Gated Recurrent Unit (GRU) and LSTM models. Although this method automatically handles the optimisation process, for more complex models such as NNs, it quickly loses its efficiency by increasing the number of hyperparameters and widening the range of their values [96]. Because in these cases the training of the models becomes very costly both in time and required computing facilities, trying all combinations of hyperparameters is not an option.

Random search, on the other hand, tries random combinations of a range of values to increase the efficiency in a high-dimensional space. Bergstra et al. [98] showed that random search has a much higher efficiency, especially in a high-dimensional space. Nonetheless, in the random selection of combinations, there is the possibility of not considering the best combination, especially in complex models [61]. In both random and grid search methods, all the candidate points are generated upfront and evaluated in parallel, and once all evaluations are done, the best hyperparameters will be selected. Selecting and evaluating candidates without considering the past evaluated hyperparameters leads to the inefficiency of these methods, because considerable time is spent evaluating inappropriate hyperparameters. As a result, there is a demand for smarter tunning methods that achieve higher accuracy and efficiency by considering the results of the previously assessed hyperparameters.

Sequential model-based optimisation (SMBO) which is also known as Bayesian optimisation, is an effective algorithm for solving the optimisation problem of functions with high-dimensional space [95]. In sequential search, several hyperparameters are selected and after evaluation of their quality, where to sample next will be decided. SMBO methods have been used in the literature for hyperparameter optimisation of ML models. For example, Masum et al. [62] used Bayesian optimisation to estimate the best hyperparameters for deep neural networks for detecting network intrusion. In the field of wind power forecasting, Zha et al. [63] utilised a Tree-structured Parzen Estimator (TPE) algorithm to estimate the best hyperparameters of the temporal convolution network (TCN), but they did not compare the optimisation performance of the applied algorithm with other optimisation methods. In another study, the TPE search method is used to optimise the LSTM model for wind power prediction [3]. Although in comparing the accuracy and efficiency of the proposed method with the conventional grid search method, the authors proved the better optimization performance of the TPE in terms of its ability to search a wider range of hyperparameters simultaneously, they did not investigate other smart and advanced hyperparameter tuning methods.

Three hyperparameter optimisation algorithms including Scikit-opt, Hyperopt, and Optuna are investigated and their performance in the hyperparameter optimisation of deep learning models are assessed.

These algorithms are the most common and optimal algorithms used in similar researches. CNN and LSTM deep learning methods are selected for time series-based predictions. These models can solve various prediction issues with short-term and long-term dependencies and it is useful to study ways by which their performance can be improved. The accuracy and calculation time of CNN and LSTM models are compared on power data from a real offshore wind turbine. This comparison of advanced hyperparameter optimisation techniques has not been investigated in the literature.

As previously mentioned, in sequential search methods, several hyperparameters are selected and after evaluating their quality, the next sampling location is decided. The goal of sequential search is to make fewer evaluations of the models with various hyperparameters and evaluate only those that have the most promising candidate hyperparameter. The trade-off here is between less ML model training time and the time to estimate where to sample next. Hence, this model is the preferred option when the evaluation procedure (training the model and evaluating its performance) takes longer than the process of evaluating where to sample next.

Sequential models integrate sample information with previous information about the unknown function to achieve posterior information about the function distribution. This posterior information is then used to determine the location of the optimal performance. Bayesian optimization is an effective sequential method that can solve the optimization problem of unknown functions because it efficiently balances exploration and exploitation. It uses prior knowledge and updates its beliefs based on new data, which allows it to make informed decisions about where to sample next. This results in a more efficient search for the global optimum, especially when dealing with complex, high-dimensional, or expensive-to-evaluate functions.

4.2 Bayesian optimisation

Bayesian optimisation is a sequential tool for optimisation of functions that do not presume any functional forms [99]. In Bayesian optimisation, contrary to the grid or random search, the results of past evaluations are employed for building a probabilistic model that maps hyperparameters to the probability of a score in the objective function. In this way, the optimisation process is more efficient as the next set of hyperparameters are selected in an informed manner. This efficiency comes from considering promising hyperparameters in past results which make fewer calls to the objective function. During optimisation, the aim is finding the maximum value of an unknown objective function *f*:

$$x^* = \arg\max_{x \in \chi} f(x) \tag{25}$$

where χ is the search space of hyperparameters, x.

In Bayesian optimisation *f* is treated as a random function and a prior is placed over it. The prior captures the behaviour of the objective function *f*, meaning it represents our initial beliefs about the function's properties before any data is observed. These properties might include smoothness, periodicity, or other structural characteristics inferred from domain knowledge or assumptions. Following the collection of the function evaluations, the prior is updated to build the posterior distribution over the objective function. The posterior distribution is then used to construct an acquisition function to determine the next query point.

As can be seen in this optimisation process two functions are required, a function that acts as the prior of the optimisation functions, and a posterior function which is an acquisition function that determines where to sample next. The prior function can be estimated by Gaussian processes or other algorithms such as TPE or RF. On the other hand, EI or Upper Confidence Bound (UCB) can be used as the acquisition functions.

Some of these functions and their performance are discussed in the following sections.

4.2.1 Gaussian process (GP)

The Gaussian process *f(x)*, as a popular probabilistic model with strong analytic tractability, is used to describe a distribution over functions. A Gaussian process presumes that each finite subset of a group of random variables follows a multivariate normal distribution with a mean and a covariance function:

$$m(x) = E[f(x)] \tag{26}$$

$$k(x, x') = E\big[\big(f(x) - m(x)\big)\big(f(x') - m(x')\big)\big] \tag{27}$$

based on these functions, the Gaussian process can be written as:

$$f(x) = \mathcal{GP}\big(m(x), k(x, x')\big) \tag{28}$$

Usually, the mean function of the Gaussian process is assumed to be zero but for calculation of the covariance function *k*, the widely used exponential square function defined by equation (29) can be:

$$k(x_i, x_j) = \exp\left(-\frac{1}{2}\|x_i - x_j\|^2\right) \tag{29}$$

where x_i and x_j depict the *i*th and *j*th samples (pairs of input values and their corresponding function values), respectively. When x_i and x_j move away from each other, which shows a weak correlation, the value of the covariance function (*k*) approaches 0. Otherwise, there is a strong correlation and the value of *k* approaches 1.

To determine the posterior distribution of f(x), certain steps must be taken. First, a sample of *t* observations is taken as the training set $D_{1:t} = \{x_n, f_n\}_{n=1}^{t}, f_n = f(x_n)$. Based on the assumption that the values of the function f are drawn in line with the multivariate normal distribution $f \sim N(0, \mathbf{K})$, where

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_t) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_t, x_1) & k(x_t, x_2) & \dots & k(x_t, x_t) \end{bmatrix} \tag{30}$$

while all elements inside the matrix **K** are calculated using equation (29) which represents the degree of correlation between two samples. Then, the new sample $f_{t+1} = f(x_{t+1})$ can be calculated based on the function *f*. Considering the assumption of the Gaussian process, the value of the function f_{t+1} plus $\mathbf{f}_{1:t}$ in the training set follows the *t+1* dimensional normal distribution:

$$\begin{bmatrix}\mathbf{f}_{1:t}\\ f_{t+1}\end{bmatrix} \sim N\left(0, \begin{bmatrix}\mathbf{K} & \boldsymbol{k}\\ \boldsymbol{k}^T & k(X_{t+1}, X_{t+1})\end{bmatrix}\right) \tag{31}$$

where $\mathbf{f}_{1:t} = [f_1, f_2, \ldots, f_t]^T$ and:

$$\mathbf{k} = [k(X_{t+1}, X_1)k(X_{t+1}, X_2)\ldots k(X_{t+1}, X_t)] \tag{32}$$

afterwards, based on the principle of joint Gaussian distribution, the prediction of the target $f_{t+1} \sim N(\mu_{t+1}, \sigma_{t+1}^2)$ can be given by:

$$\mu_{t+1}(X_{t+1}) = \mathbf{k}^T\mathbf{K}^{-1}\mathbf{f}_{1:t} \tag{33}$$

$$\sigma_{t+1}^2(X_{t+1}) = -\mathbf{k}^T\mathbf{K}^{-1} + k(X_{t+1}, X_{t+1}) \tag{34}$$

Finally, the posterior distribution of the objective function over all possible values of $f_{t+1}$is obtained and the entire regression model based on the Gaussian process is completed.

4.2.2 Acquisition function

After finding the posterior distribution of the objective function, the next challenge is to know how to search for the next points which derivates the maximum of the function *f*. In Bayesian optimisation, the acquisition function is used to achieve this goal. Assuming that the high value of the acquisition function corresponds to the large value for the objective function *f*, maximizing the acquisition function will maximise the objective function f.

The common acquisition functions that are used in Bayesian optimisation include the probability of improvement (PI) and EI functions. Function PI tries to search around the current optimum sample to find points that may exceed the current optimum value. This function can be described as:

$$PI(x) = \phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) \tag{35}$$

where ϕ represents the cumulative distribution function of the standard Gaussian distribution. μ(x) is the predicted mean of the objective function at point x, *f(x+)* is the value of the current best observed point (the highest known value of the objective function) and *σ(x)* is the predicted standard deviation of the objective function at point x.

As evident, the main drawback of the PI acquisition function is that it selects samples only from regions close to the current optimal solution (exploration). Therefore, the possible better points that are far away from the local optimal points may not be investigated [96].

To solve the problem of falling into the local optimum solution, the EI acquisition function is employed [100]. The EI function, while exploring the region of the current optimum value, calculates the expected improvement of the new point. The expectation here refers to the mean of the potential improvements, weighted by their probabilities. If the improvement value is less than the desired value after running the algorithm, it will be assumed that the current optimal point is the best solution available in that area, and therefore the algorithm searches for the optimal point in other parts of the domain (exploration).

The difference between the function value at the new selected point and the current optimal value is called the degree of improvement *I*. If the value of the function at the new point is lower than the current optimal value, the improvement function is considered 0:

$$I(x) = max\{0, f_{t+1}(x) - f(x^+)\} \tag{36}$$

where *t+1* refers to the next time step or iteration in the optimization process, indicating the function value at the newly selected point. Based on the assumption that the distribution of the function value at the new sampling point ($f_{t+1}(x)$) obeys the normal distribution with mean $\mu(x)$ and standard deviation $\sigma^2(x)$, the random variable *I* obeys the normal distribution too with the mean $\mu(x) - f(x^+)$ and standard deviation $\sigma^2(x)$. Equation (37) shows the probability density of *I*:

$$f(I) = \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{\left(\mu(x)-f(x^+)-I\right)^2}{2\sigma^2(x)}\right), I \geq 0. \tag{37}$$

Now, the EI can be defined as:

$$E(I) = \int_{\infty}^{\infty} I\, f(I) \mathrm{d}I = \int_{I=0}^{I=\infty} I\, \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{(\mu(x) - f(x^+) - I)^2}{2\sigma^2(x)}\right) \mathrm{d}I = \sigma(x)[Z\phi(Z) + \varphi(Z)] \tag{38}$$

where φ is the probability density function of the standard normal distribution and:

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)} \tag{39}$$

As can be seen from these equations (36-39) the EI can be calculated from σ, μ and the current optimal point $f(x^+)$.

In this study, the EI function is used as the acquisition function for all hyperparameter optimization techniques since it can address the exploration and exploitation trade-off. It does this by balancing the potential for discovering new, optimal points (exploration) with the need to refine known high-performing areas (exploitation). The EI function calculates the expected improvement at each candidate point, which incorporates both the predicted mean and uncertainty (variance) of the model. This means that points with high uncertainty and those expected to perform well are both considered, ensuring a thorough search of the parameter space without getting stuck in local optima. [62]. The EI function is proven to have a strong theoretical guarantee [101] and empirical effectiveness [102].

4.2.3 Tree-structured Parzen Estimator (TPE)

In cases of hyperparameter optimisation with higher dimensions and a small fitness evaluation budget, an alternative to the GP approach is required. In GP, the aim was to approximate f(x) or the probability of score given the hyperparameter ($P(y|x)$). This is done by using marginal probabilities, including the probability of each hyperparameter and the probability of the hyperparameters given the score, *p(x/y)*:

$$P(y|x) = \frac{P(x|y) \times P(y)}{P(x)} \tag{40}$$

where y is the score f(x) and x represent the hyperparameters.

But in TPE, instead of approximating the left side of equation (40), the probability of hyperparameters given the score that is obtained when sampling some of the values of the hyperparameters, $P(x|y)$ is approximated. This conditional probability ($P(x|y)$) is approximated using two different functions: function $\ell(x)$ for the cases where the performance is smaller than a certain value of performance, and the function $g(x)$ for cases where the performance is bigger than the certain value of performance:

$$P(x|y) \simeq \begin{cases} \ell(x) & \text{if } \ y < y^* \\ g(x) & \text{if } \ y \geq y^* \end{cases} \tag{41}$$

these two densities of *l(x)* and *g(x)* will then be used in the EI function, and after some derivations will end up at equation (42). This expected improvement can determine where to sample the next for hyperparameters.

$$EI_{y^*}(x) \propto \left(\gamma + \frac{g(x)}{\ell(x)}(1-\gamma)\right)^{-1} \tag{42}$$

The benefits of TPE include its ability to efficiently handle high-dimensional hyperparameter spaces and small evaluation budgets by focusing on regions of the search space that are more promising. TPE dynamically balances exploration and exploitation by modelling the probability of hyperparameters given their performance, allowing it to concentrate sampling on hyperparameter values that are more likely to yield better performance. This results in more efficient optimization compared to traditional GP-based methods, especially in complex and high-dimensional settings. Additionally, TPE's ability to separately model good and bad performance regions help in better navigating the hyperparameter space and avoiding poor-performing areas.

4.3 Optimisation algorithms

4.3.1 Scikit optimise (GP-EI)

Scikit optimise is an open-source Python package that can perform various forms of Bayesian optimisation. It implements several search algorithms including Bayesian optimisation with gaussian processes (through the *gp_minimize* function), Bayesian optimisation with random forest (through the *forest_minimize* function), and Bayesian optimisation with Gradient Boosting Trees (through the *gbrt_minimize* function).

To perform the optimisation, the first step is to define the objective function that needs to be minimised. The objective function usually takes the ML model and the hyperparameters and outputs a performance metric.

Scikit optimise comes with a built-in module to create hyperparameter spaces to sample from. The samples can be integers, reals, and categories. In addition, a variety of acquisition functions are available to choose from including the EI, PI and Lower Confidence Bound (LCB).

It is important to mention that the Lower Confidence Bound (LCB) acquisition function is a strategy used in Bayesian optimization to balance exploration and exploitation. LCB selects the next sampling point by considering both the mean prediction and the uncertainty of the prediction. It typically involves a trade-off parameter that adjusts the emphasis on exploring uncertain areas versus exploiting areas with known good performance. This helps in efficiently searching the hyperparameter space by ensuring that regions with high uncertainty are adequately explored, potentially discovering better hyperparameters.

In this project the Scikit optimise is implemented to use Bayesian optimisation with GP as the surrogate. To this end, the gp_minimize package is imported and the RMSEs of two prediction models (CNN and LSTM) were defined as the objective function. Then the hyperparameter space is defined as well as the initial number of points at which to evaluate the objective function before starting to guide the Bayesian optimisation search. As mentioned earlier in this study the EI is used due to its capability of exploration and exploitation to guide the search in Bayesian optimisation.

4.3.2 Optuna (TPE-EI)

As a second tuning method in this study, the Optuna optimisation package with its define-by-run design is employed. Optuna is a recently developed optimisation algorithm [103] that has been successfully used to tune the hyperparameters of LSTM model for wind power prediction [3]. The selection of Optuna as one of the advanced optimisation algorithms in this research is based on three specific features, define-by-run context, efficient sampling, and ease of setup. The define-by-run context enables the dynamic building of the search space. This means that the search space can be constructed and modified on-the-fly during the optimization process, allowing for more flexible and adaptive tuning strategies. Unlike traditional static search space definitions, define-by-run allows for conditional parameter sampling and complex search space structures that can adapt based on intermediate results and feedback during the optimization run. This dynamic nature facilitates a more tailored and efficient exploration of the hyperparameter space, ultimately leading to better optimization outcomes.

In the optimisation process, different hyperparameter combinations are considered as input to maximise a function and respond a validation score as an output. The objective function formulates the search space dynamically through interacting with the trial object [104]. This means that the search space can be adjusted during the optimization process based on the outcomes of previous trials. Efficient sampling of Optuna allows handling of both types of sampling, relational sampling to benefit the parameters correlations, and independent sampling to consider each sample separately [104]. Relational sampling considers the dependencies and interactions between different hyperparameters. Another advantage of Optuna is its easy setup, which allows it to be used for a variety of tasks, from lightweight experiments performed through interactive interfaces to heavyweight distributed computing [103]. There are various features in Optuna that need to be set up. Sampler, as the hyperparameter search algorithm, can be selected among different options including the basic ones such as grid search, random search and CMA-ES. CMA-ES (Covariance Matrix Adaptation Evolution Strategy) is an advanced evolutionary algorithm that adapts the covariance matrix of the sampling distribution to guide the search towards optimal solutions, making it particularly effective for complex, high-dimensional optimization problems. In this work the TPE algorithm is used to consider each hyperparameter independently. The next option is the pruner that stops testing trials that do not offer promising performances. The default pruner which is used in this study is the Median Pruner, which stops trials that perform worse than the median of completed trials. However, other pruners such as the Percentile Pruner, which stops trials that fall below a certain performance percentile; Successive Halving, which repeatedly halves the number of trials by discarding the lower-performing half; Hyperband, which allocates resources dynamically based on performance; and Threshold Pruner, which stops trials that do not meet a predefined performance threshold, are also available. It is also necessary to set the direction of the tuning process, which indicates whether we want to minimise or maximise the objective function. As mentioned earlier, the aim of this study is to minimise the RMSE value of the wind power forecasting. As a result, the minimise direction is selected. Further details of the optimisation by this package can be found in [103].

4.3.3 Hyperopt (Annealing-EI)

Hyperopt is a python library that enables implementing Bayesian optimisation. Hyperopt has already been applied for hyperparameter optimisation of deep neural networks and convolutional neural networks [61]. Hyperopt provides three search algorithms: random search (*rand.suggest*), annealing (which is another sequential mode-based optimisation with the gaussian process alternative in order to be able to sample nested hyper parameters), and TPE. In this thesis, the annealing search method is selected to compare its performance with the TPE search method used in the Optuna package and the GP search method used in the Scikit optimise algorithm.

4.4 Results of comparison of different hyperparameter optimisation methods

The same SCADA data used in chapter 3 is employed here to carry out the comparison of the hyper parameter optimisation techniques. During the pre-processing stage, the negative wind power values within the dataset were replaced with zeros based on the recommendations issued in [76]. In addition to diminishing the negative impact of wind turbulence on the correlation between the generated wind power and measured wind speed, the time series resolution averaged 10 minutes in coordination with the approved average time by the international standard for power performance measurements of electricity-producing wind turbines (IEC 61400-12-1) [77]. The dataset was then divided into two parts, the first 90% for training, and the rest 10% for testing both CNN and LSTM deep learning models, which were optimised by various hyperparameter optimisation techniques.

The search spaces for hyperparameters of the CNN and LSTM models for all three optimisation methods are determined according to Table 4.1. Considering the mentioned ranges of values for the different hyperparameters, this results in more than 2000 million hyperparameter combinations for CNN models, and more than 38 million combinations for LSTM models.

Table 4.1. search space for hyperparameters of the CNN and LSTM models

CNN Model		LSTM Model	
Hyperparameter	Search Space	Hyperparameter	Search Space
Units of first conv. Layer	30, 31, …, 130	Units (neurons) of first LSTM Layer	10, …, 100
Units of second conv. Layer	30, 31, …, 130	Number of dense layers	1, 2, 3, 4
Number of dense layers	1, 2, 3, 4	Units (neurons) in dense layer	10, 11, …, 100
Units (neurons) in dense layer	10, 11, …, 100	Activation function	Sigmoid, tanh, ReLU
Activation function	Sigmoid, tanh, ReLU	Loss function	MAE, MSE
Loss function	MAE, MSE	Optimiser	ADAM, Adadelta
Optimiser	ADAM, Adadelta	Epochs	50, 51, …, 150
Epochs	50, 51, …, 100		

To carry out all the simulations and experiments, the Python programming language with Scikit-opt, Optuna, Hyperopt packages, and Scikit-learn libraries are employed on a PC with Intel Core™ i7–11850H 2.5GHz CPU and 16GB RAM (without GPU processing). In addition, for better investigation of the performance of the various hyper parameter optimisation and forecasting models, similar parameters were set for all selected models; for example, the selected time lag (input layer length) was set at 10 in all simulations.

4.4.1 Optimisation by Scikit-optimise algorithm

As the first optimisation method in this research, the Scikit optimise is implemented. To utilise optimisation in this algorithm, except for defining the hyperparameter space, other parameters must be defined. These parameters include the objective function, the initial number of points at which to evaluate the objective function before starting to guide the Bayesian optimisation search, the acquisition function, and the number of times that we want to sample the hyper parameter space subsequently. The Gaussian process is selected as the search method and EI as the acquisition function. The RMSE values of the wind power predictions by forecasting models which were built with selected hyperparameters are considered as the objective function. Scikit optimise algorithm for tuning the CNN model was launched for 300 tests. The best ten hyperparameter combinations with the least RMSE values are obtained according to Table 4.2 while the entire optimisation process took approximately 455 minutes. As can be seen in Table 4.2, the minimum RMSE value of 533.74 is obtained for trial number 204 with shown values for different hyperparameters.

Table 4.2. Best hyperparameter combinations of CNN model found by Scikit-Optimise

Rate	**Trial number**	Units of first conv. layer	Units of second conv. layer	No. of dense layers	No. of dense units	Activation function	Loss function	optimiser	Epochs	RMSE
1	204	87	89	4	100	ReLU	MAE	ADAM	104	553.74
2	219	128	12	4	100	ReLU	MAE	ADAM	101	560.29
3	210	128	12	4	100	ReLU	MAE	ADAM	104	562.66
4	257	128	12	1	100	ReLU	MAE	ADAM	132	566.65
5	217	128	12	4	100	ReLU	MAE	ADAM	101	569.20
6	88	12	12	4	100	ReLU	MAE	ADAM	87	569.57
7	230	128	31	2	100	ReLU	MAE	ADAM	98	572.18
8	138	12	91	4	100	ReLU	MAE	ADAM	89	572.47
9	162	12	104	4	100	ReLU	MAE	ADAM	88	572.52
10	78	128	128	4	100	ReLU	MAE	ADAM	90	572.57

The Scikit optimise optimisation method with similar settings is also used to tune the LSTM prediction model. In this case, the entire tuning process for 300 trials took approximately 1740 minutes and a minimum RMSE value of 549.07 was obtained for trial number 146 with shown values for different hyperparameters. Table 4.3 shows the best ten hyperparameter combinations for the LSTM model that lead to the lowest RMSE values.

As can be seen from Tables 4.2 and 4.3, the values of some of the hyperparameters such as the activation function, loss function, and optimiser are equal for all proposed hyperparameter combinations. This shows that the optimisation algorithm has quickly reached a level of certainty regarding these parameters. On the other hand, for other hyperparameters such as the units of convolution layers, small changes can be seen until reaching the optimum value.

Table 4.3. Best hyperparameter combinations of LSTM model found by Scikit-Optimise

Rate	**Trial number**	Units of first LSTM layer	dense layers	Units of dense layer	Activation function	Loss function	optimiser	Epochs	RMSE
1	146	61	1	68	ReLU	MSE	ADAM	150	549.07
2	177	27	1	10	ReLU	MSE	ADAM	150	549.37
3	260	42	1	59	ReLU	MSE	ADAM	150	549.74
4	26	12	4	100	ReLU	MAE	ADAM	149	549.94
5	298	100	4	10	ReLU	MSE	ADAM	74	551.75
6	85	54	1	29	ReLU	MSE	ADAM	150	551.99
7	39	100	1	100	ReLU	MSE	ADAM	150	552.31
8	279	100	4	100	ReLU	MSE	ADAM	58	552.48
9	203	50	1	67	ReLU	MSE	ADAM	150	553.02
10	219	75	1	42	ReLU	MAE	ADAM	50	554.33

4.4.2 Optimisation by Optuna

As the second tuning method in this study, the Optuna optimisation package with its define-by-run design is investigated. There are some features in the Optuna optimisation package that need to be tuned. Sampler, as the hyperparameter search algorithm, can be selected among different methods including the basic ones such as grid search, random search and CMA-ES. In this study the default search algorithm, TPE, is used. TPE can model the underlying objective function and its uncertainty. This modelling allows it to make informed decisions, meaning it uses existing data to predict and select the most promising hyperparameters for future trials, thus enhancing the efficiency and effectiveness of the search process. The next option is the pruner that stops testing trials that are not offering promising performances. The median pruner performs when the trial's best intermediate result is poorer than the median of the intermediate results of former trials. It is also necessary to set the direction of the tuning process, which indicates whether we want to minimize or maximize the objective function. As previously mentioned, the aim of this study is to minimize the RMSE value of the wind power forecasting. More details of the optimisation process of this package can be found in [103].

The same hyperparameter spaces considered for scikit optimise, are used again. For the CNN model, the minimum RMSE value of 553.66 is obtained for trial number 269 after about 492 minutes and regarding the LSTM model, the minimum RMSE value of 538.14 is obtained for trial number 91 after about 456 minutes. These shows that for both CNN and LSTM models, the best hyperparameter combinations can be found in a similar amount of time. Tables 4.4 and 4.5 show the best ten hyperparameter combinations for CNN and LSTM models, found by the Optuna algorithm, respectively.

Table 4.4. Best hyperparameter combinations of CNN model found by Optuna

Rate	**Trial number**	Units of first conv. layer	Units of second conv. layer	No. of dense layers	No. of dense units	Activation function	Loss function	optimiser	Epochs	RMSE
1	269	108	80	3	45	ReLU	MAE	ADAM	86	553.66
2	203	113	82	3	44	ReLU	MAE	ADAM	84	560.9
3	97	72	32	3	47	ReLU	MSE	ADAM	95	561.73
4	175	119	66	3	43	ReLU	MAE	ADAM	78	567.36
5	228	119	78	3	39	ReLU	MAE	ADAM	84	570.45
6	205	114	81	3	44	ReLU	MAE	ADAM	83	572.98
7	227	119	77	3	40	ReLU	MAE	ADAM	84	573.19
8	156	71	69	3	43	ReLU	MAE	ADAM	77	573.36
9	77	83	62	3	46	ReLU	MAE	ADAM	94	574.49
10	160	81	66	3	38	ReLU	MAE	ADAM	81	575.24

Table 4.5. Best hyperparameter combinations of LSTM model found by Optuna

Rate	**Trial number**	Units of LSTM layer	No. of dense layers	No. of dense units	Activation function	Loss function	optimiser	Epochs	RMSE
1	91	45	3	60	ReLU	MSE	ADAM	128	538.14
2	123	43	2	61	ReLU	MSE	ADAM	124	541.04
3	81	34	2	73	ReLU	MSE	ADAM	125	542.27
4	272	46	2	59	ReLU	MSE	ADAM	120	542.34
5	255	42	2	59	ReLU	MSE	ADAM	108	542.39
6	69	37	1	54	ReLU	MSE	ADAM	104	543.49
7	82	37	2	73	ReLU	MSE	ADAM	127	545.09
8	248	50	2	63	ReLU	MSE	ADAM	105	545.44
9	241	42	2	61	ReLU	MSE	ADAM	103	546.29
10	258	52	2	63	ReLU	MSE	ADAM	123	547.53

As evident from Tables 4.4 and 4.5, the values of two hyperparameters, namely the Activation function and optimiser, are equal for all the ten best combinations of CNN and LSTM models. While the best loss function selected in the ten most accurate hyperparameter combinations of the LSTM model is the mean square error (MSE), the results of experiments show that the value of this hyperparameter in the CNN model can be both the MSE and the mean absolute error (MAE).

4.4.3 Hyperopt

Hyperopt is the third hyperparameter optimisation algorithm employed in this research. In the Hyperopt algorithm, first the configuration space is defined and then the *fmin* driver is used to determine the direction of the optimisation. Hyperopt provides three search algorithms including the random search (*rand.suggest*), annealing, and the TPE. Annealing is another GP alternative SMBO model with the capability of sampling the nested hyper parameters. In this study the annealing search method is selected to compare its performance with the TPE search method in the Optuna algorithm and the GP search method in Scikit optimise.

The best ten hyperparameter combinations of the CNN and LSTM models found by Hyperopt are shown in Tables 4.6 and 4.7. For the CNN model, the minimum RMSE value of 556.57 was obtained for trial number 203 after approximately 368 minutes. Regarding the LSTM model, the minimum RMSE value of 532.59 was obtained for trial number 236 after approximately 892 minutes.

Table 4.6. Best hyperparameter combinations of CNN model found by Hyperopt

Rate	**Trial number**	Units of first conv. layer	Units of second conv. layer	No. of dense layers	No. of dense units	Activation function	Loss function	optimiser	Epochs	RMSE
1	203	105	33	2	21	ReLU	MSE	ADAM	80	556.57
2	237	105	33	2	58	ReLU	MAE	ADAM	80	567.44
3	270	105	33	2	21	ReLU	MSE	ADAM	80	567.88
4	204	105	33	2	58	ReLU	MAE	ADAM	80	569.32
5	200	105	33	2	21	ReLU	MAE	ADAM	80	571.45
6	257	105	33	2	21	ReLU	MAE	ADAM	80	571.72
7	172	105	33	2	21	ReLU	MAE	ADAM	80	571.85
8	285	105	34	2	14	ReLU	MAE	ADAM	80	572.39
9	220	105	33	2	21	ReLU	MAE	ADAM	80	573.22
10	66	105	33	1	21	ReLU	MAE	ADAM	80	574.06

Table 4.7. Best hyperparameter combinations of LSTM model found by Hyperopt

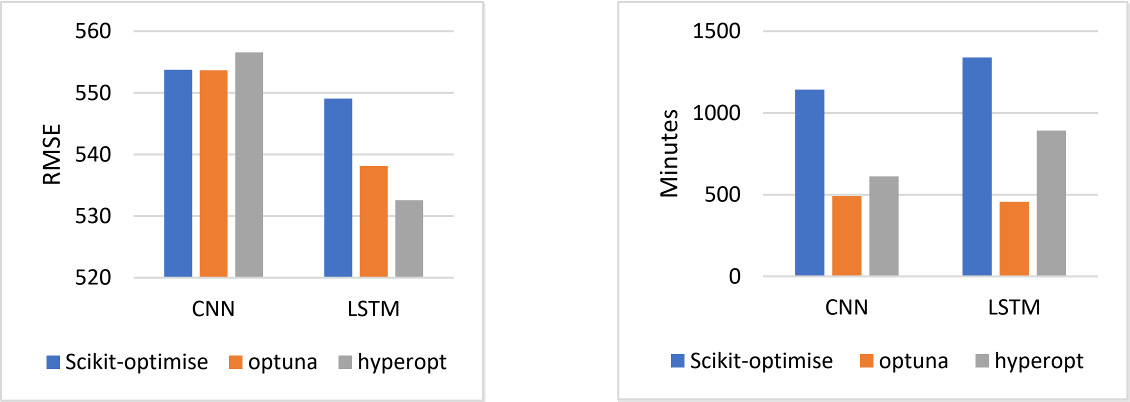
Rate	**Trial number**	Units of LSTM layer	No. of dense layers	No. of dense units	Activation function	Loss function	optimiser	Epochs	RMSE
1	236	52	2	60	ReLU	MSE	ADAM	99	532.59
2	96	52	2	60	ReLU	MSE	ADAM	99	541.23
3	195	52	2	60	ReLU	MSE	ADAM	99	541.76
4	237	52	2	97	ReLU	MSE	ADAM	99	541.83
5	35	52	2	60	ReLU	MSE	ADAM	99	542.32
6	244	52	2	60	ReLU	MSE	ADAM	99	543.59
7	266	52	2	60	ReLU	MSE	ADAM	99	544.18
8	14	52	2	60	ReLU	MSE	ADAM	66	546.58
9	154	66	2	60	ReLU	MSE	ADAM	99	547.82
10	213	52	1	60	ReLU	MSE	ADAM	99	547.92

4.4.4 Comparison of the hyperparameter optimisation algorithms

By comparing the prediction accuracy of prediction models whose structures (hyperparameters) are proposed by optimisation methods, the tuning methods can be compared. The processing speed is another factor in determining the performance of the optimisation methods. Table 4.8 shows the performance of the prediction models tuned by the three optimisation techniques, both in processing time and accuracy.

Table 4.8. The RMSE and processing time of the prediction models tuned in different method

Optimisation methods		RMSE (kW)		Calculation time (minutes)	
Algorithm	Search method	CNN	LSTM	CNN	LSTM
Scikit-Opt	GP	553.74	539.64	455	600
Optuna	TPE	553.66	538.14	492	456
Hyperopt	Annealing	556.57	532.59	368	892



(a) (b)

Figure 4.1. RMSE (a) and calculation time (b) of the CNN and LSTM tuned by various optimisation methods

Considering the calculated RMSE values in Table 4.8 and figure 4.1(a), the first point that can be recognised is that the LSTM model is a better choice than the CNN for short term wind power prediction. This is believed to be due to the capability of the LSTM model in learning both long-term and short-term dependencies. On the other hand, while the prediction accuracy of different structures of the CNN model, proposed by different optimisation methods are very similar, the LSTM model optimised by the Hyperopt algorithm based on the annealing search method results in the highest accuracy. In terms of prediction accuracy, the Optuna optimisation method ranks second after the Hyperopt algorithm.

Based on the comparison of the required time for calculations/simulations in each optimisation in Table 4.8 and the figure 4.1(b), it can be concluded that the Optuna optimisation algorithm using the TPE search method and EI acquisition function, has the best efficiency for both CNN and LSTM models. Figure 4.2 shows the prediction results of three individual LSTM prediction models tuned by the various optimisation methods. As can be in this figure, when the wind power generation encounters abrupt changes, the LSTM model optimised by Hyperopt and Optuna algorithms have better prediction performance than the LSTM model optimised by Scikit optimise.

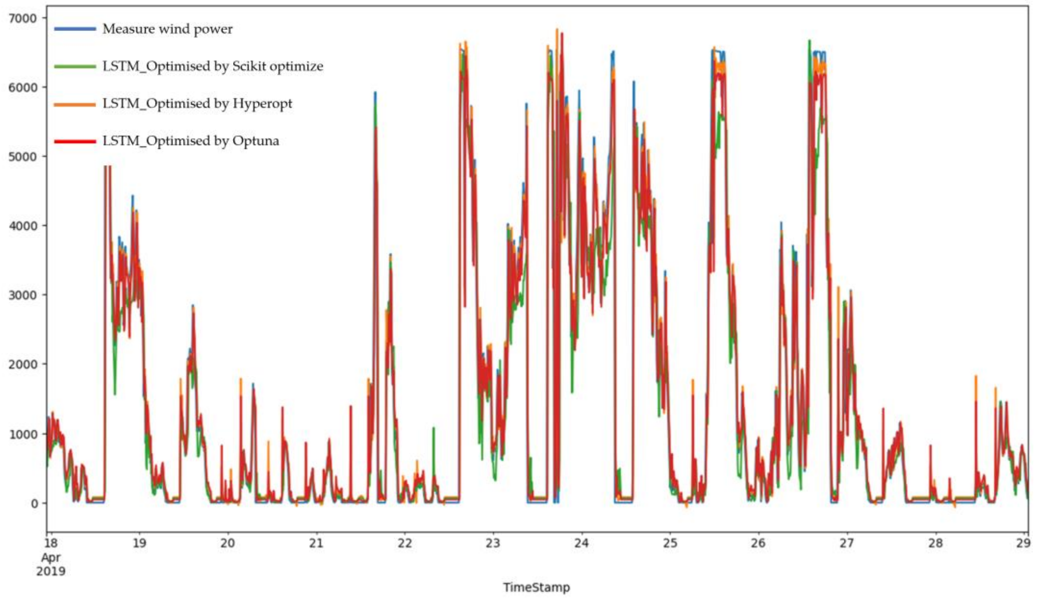


Figure 4.2. Wind power predictions of LSTM models optimised by three different optimisation algorithms

4.4.5 Randomness impact on the optimisation performance

As previously mentioned earlier, most ANNs have randomness in their training process. Randomness can have different reasons, the most important of which is the random initialisation of weights and biases in NNs. Different initialisation of weights causes changes in the training results that make the networks unstable and unreliable. An unstable network is one whose performance varies significantly with each training run due to the different starting points in the parameter space. This instability can lead to inconsistent and unpredictable performance, making it challenging to assess the true capability of the model. As a result, to improve the performance of the prediction models and achieve consistent results, it is necessary to predict this randomness. In this context, predicting randomness means controlling or accounting for the inherent variability in the training process. In ML, for prediction of randomness the seed concept is used. Seed allows for the prediction of the randomness and makes the results reproducible [105]. Defining a specific seed value before training ensures consistent results of ANNs.

In this study, to investigate the impact of random initialisation on the accuracy of deep learning models, an experiment is carried out consisting of one hundred trials of CNN model predictions with different seeds but constant hyperparameters. As can be seen in the figure 4.3 and Table 4.9, the RMSE values of the predictions vary from 570.4 to 631.7 with an average value of 595.9 and standard deviation of 13.78. It is evident that the changes in random initialisation, while all other variables are constant, do have an impact on the prediction accuracy of the forecasting models.

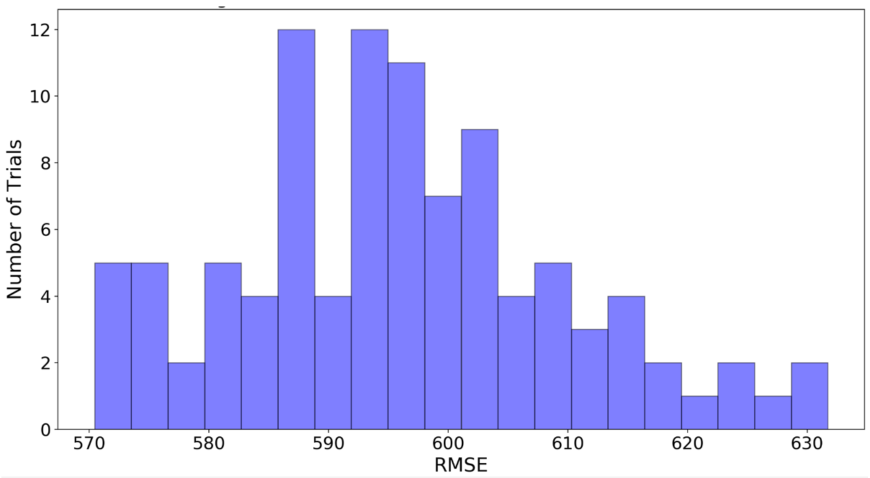


Figure 4.3. Histogram of RMSE values of CNN model with different seeds

Table 4.9. The statistical description of RMSE prediction results with different seeds

RMSEs of the CNN prediction model	
Counts of trials	100
Average of RMSEs	595.96
Standard deviation	13.78
Minimum RMSE	570.45
Maximum RMSE	631.73

This means that without considering a wider range of random seeds, it is difficult to trust the performance of a specific deep learning model. In other words, various initialisations are required to see the robustness of the prediction model. In order to investigate the impact of the randomness feature of NNs on the accuracy of the deep learning models built by the best hyperparameter combinations, 10 different random seeds [123, 951, 375, 435, 599, 54, 602, 325, 691, 36] are used for the best found hyperparameter combinations of the CNN and LSTM deep leaning models to find the structure which is less sensitive to the randomness.

Table 4.10 shows the RMSE values related to the predictions made by the CNN models based on the 10 best hyperparameter combinations and 10 different random seeds. For a better analysis of the test results presented in Table 4.10, the RMSE values are plotted in figure 4.4. In addition, the distribution of RMSE values for each hyperparameter combination are shown in figure 4.5.

Table 4.10. RMSE values of different hyperparameter combinations of CNN model with different seeds

Seeds		**H.C 1**	**H.C 2**	**H.C 3**	**H.C 4**	**H.C 5**	**H.C 6**	**H.C 7**	**H.C 8**	**H.C 9**	**H.C 10**
R.S 1	123	582.05	592.06	621.04	596.58	587.33	594.24	597.20	623.10	584.03	605.28
R.S 2	951	581.68	598.74	590.73	596.52	581.34	612.38	612.04	582.82	581.25	599.65
R.S 3	375	590.17	619.76	594.03	617.20	579.50	582.82	607.01	566.11	588.17	575.05
R.S 4	435	569.66	596.19	595.14	614.82	586.07	636.65	610.35	585.86	602.05	625.39
R.S 5	599	597.98	592.18	623.40	590.01	593.39	573.66	596.64	574.51	572.42	620.09
R.S 6	54	597.95	564.98	682.69	616.43	584.56	582.11	592.07	575.23	588.85	596.86
R.S 7	602	596.23	613.62	610.71	601.07	607.69	595.51	606.02	591.19	583.16	602.35
R.S 8	325	587.22	594.58	617.99	621.27	629.48	584.02	604.10	596.17	587.38	584.62
R.S 9	691	582.61	591.89	636.24	578.85	592.06	595.31	617.61	587.37	581.86	590.43
R.S 10	36	598.92	592.02	609.06	591.16	592.17	602.31	583.57	604.86	595.76	593.70
Averages		**588.45**	**595.60**	**618.10**	**602.39**	**593.36**	**595.90**	**602.66**	**588.72**	**586.49**	**599.34**
STD		**9.60**	**14.56**	**26.94**	**14.26**	**14.97**	**18.18**	**10.25**	**16.48**	**8.17**	**15.19**

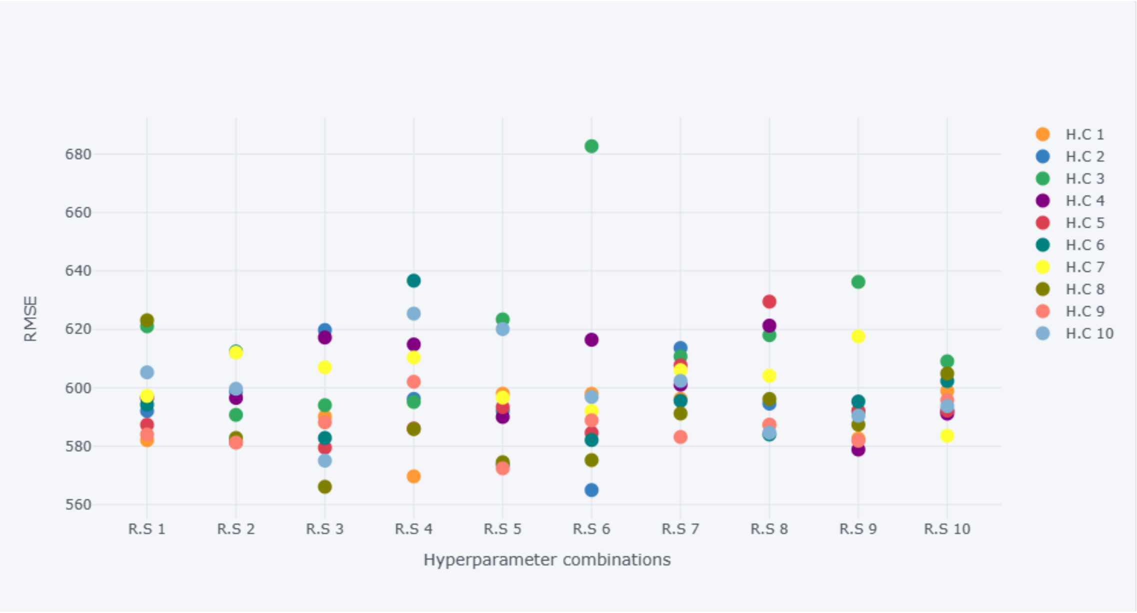


Figure 4.4. Comparison of RMSE Across Different Hyperparameter Combinations and Seeds for CNN Model

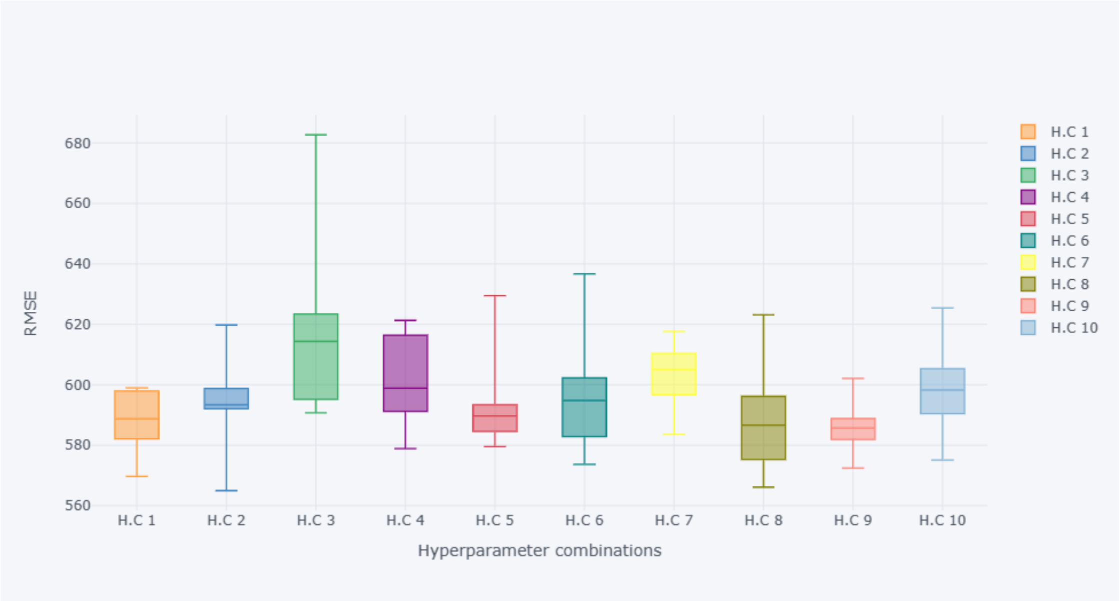


Figure 4.5. Average and standard deviations of RMSE values for different hyperparameter combination of CNN model

As evident from Table 4.10 and figures 4.4 and 4.5, considering all the set seed values, combination number 9 has the best performance, as it has the least average and standard deviation values of RMSEs. In other words, a CNN model built based on the hyperparameter of combination number 9 will be less sensitive to the randomness feature of NNs. A similar experiment/simulation was carried out for the LSTM prediction model. In a similar manner, Table 4.11 shows the RMSE values related to the predictions made by the LSTM models based on the 10 best hyperparameter combinations and 10 different random seeds. Figure 4.6 shows the RMSE values for different hyperparameter combinations and figure 4.7 demonstrates distribution of RMSE values for each hyperparameter combination.

Table 4.11. RMSE values of different hyperparameter combinations of LSTM model with different seeds

Seeds		**H.C 1**	**H.C 2**	**H.C 3**	**H.C 4**	**H.C 5**	**H.C 6**	**H.C 7**	**H.C 8**	**H.C 9**	**H.C 10**
R.S 1	123	565.41	670.06	553.7	560.58	580.36	558.4	549.45	577.62	681.99	574.64
R.S 2	951	565.62	558.31	562.39	591.9	567.04	550.72	567.53	555.90	556.53	573.81
R.S 3	375	565.64	571.43	565.93	565.04	571.15	590.16	556.83	577.39	556.23	586.87
R.S 4	435	545.13	562.01	569.52	562.27	566.86	587.41	581.36	572.37	571.77	576.19
R.S 5	599	575.46	573.69	564.41	569.04	575.63	573.44	578.02	563.22	601.12	700.64
R.S 6	54	713.16	559.72	547.07	573.1	571.66	563.3	595.56	576.98	560.43	581.04
R.S 7	602	585.11	582.99	574.58	590.25	583.1	568.11	590.65	558.49	559.33	572.58
R.S 8	325	560.84	550.14	580.14	563.89	599.04	569.42	558.06	587.30	595.30	589.38
R.S 9	691	563.77	566.52	565.97	563.51	546.47	568.94	568.75	554.34	573.16	568.38
R.S 10	36	572.29	574.24	587.55	550.10	562.65	581.54	576.65	576.86	559.29	614.30
Averages		**581.24**	**581.24**	**567.13**	**568.97**	**572.4**	**571.14**	**572.29**	**570.05**	**581.52**	**593.78**
STD		**47.49**	**47.49**	**11.85**	**13.07**	**13.85**	**12.46**	**14.95**	**11.23**	**38.77**	**39.78**

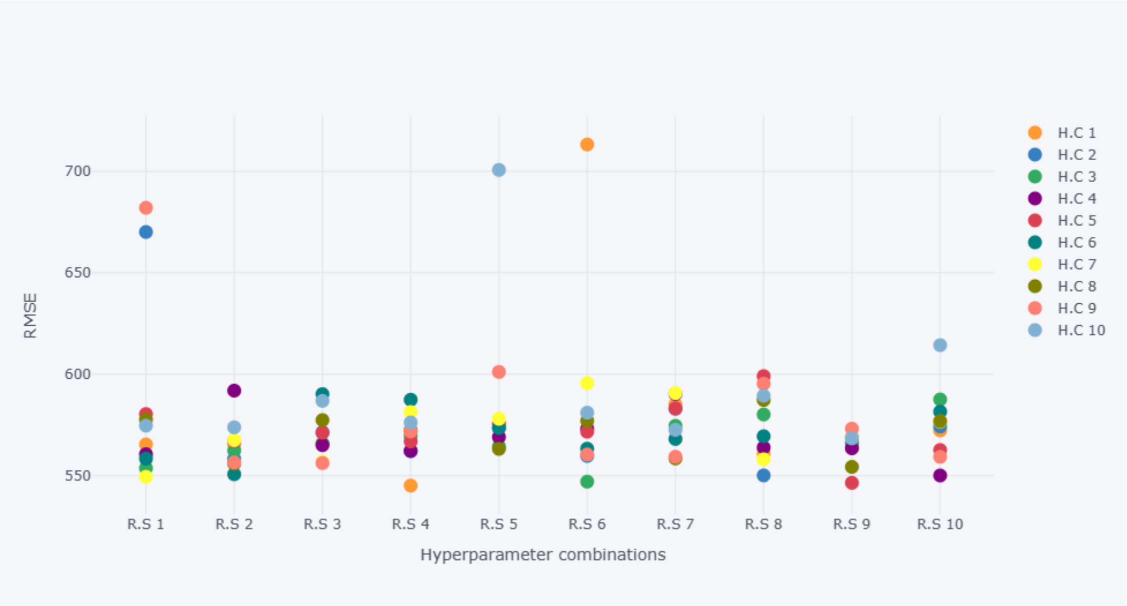


Figure 4.6. Comparison of RMSE Across Different Hyperparameter Combinations and Seeds for LSTM Model

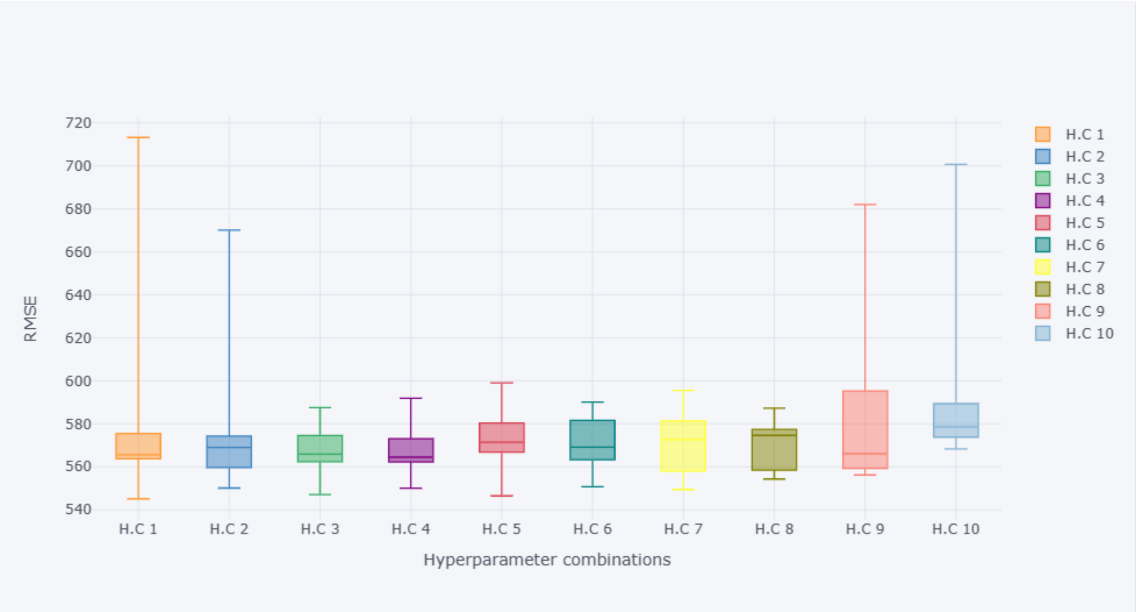


Figure 4.7. Average and standard deviations of RMSE values for different hyperparameter combination of LSTM model

As noticed from Table 4.11 and figures 4.6 and 4.7, considering all the set seed values, combination number 3 has the best performance, as it has the least average and standard deviation values of RMSEs. In other words, an LSTM model based on the hyperparameter of combination number 3 will be less sensitive to the randomness feature of NNs. To assess the robustness of the LSTM model based on the third hyperparameter combinations, an experiment is caried out in which two LSTM models based on the hyperparameter combination sets 1 and 3 were utilised for wind power prediction. In this experiment, no seed value is set to see the performance of the proposed structures. As can be seen from figure 4.8, the forecasting model, which was built with the third hyperparameter combination, is more accurate. Better forecasting performance is clearly visible especially in higher wind power values, such as for example between hours 15:00 and 18:00 on 26th of April (figure 4.9) and during times when the wind power production experiences sudden changes.

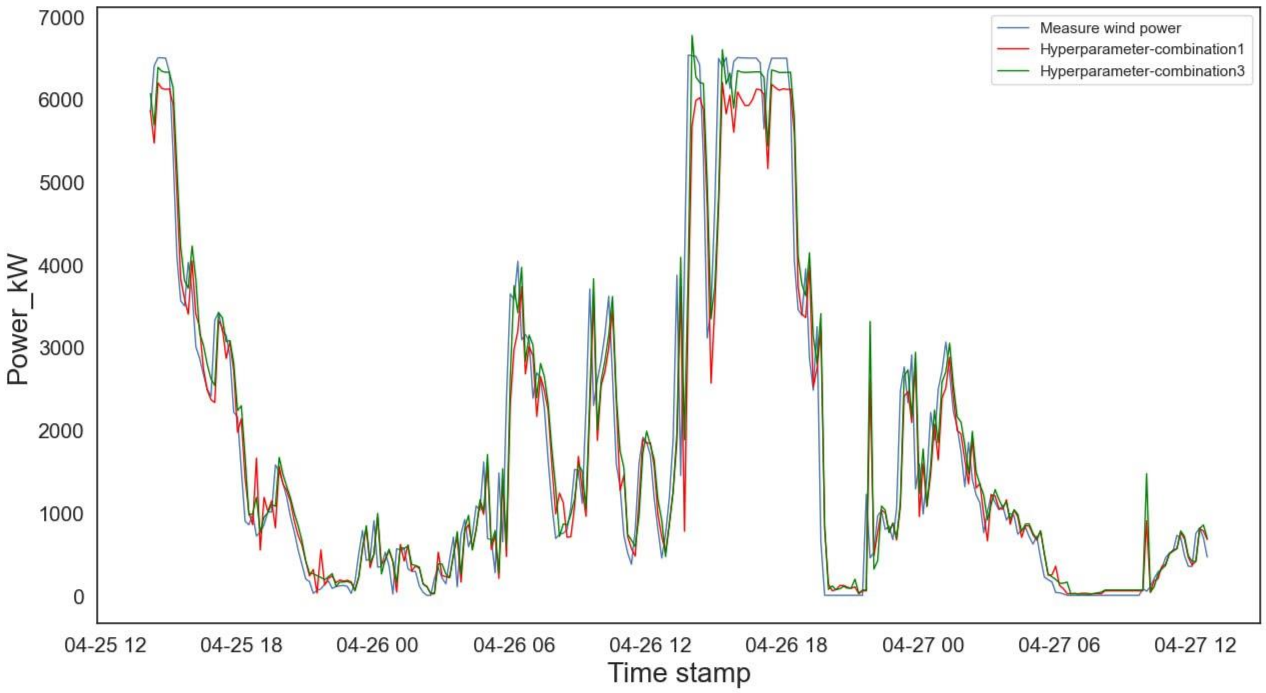


Figure 4.8. wind power prediction of LSTM model built by two different hyperparameter combinations

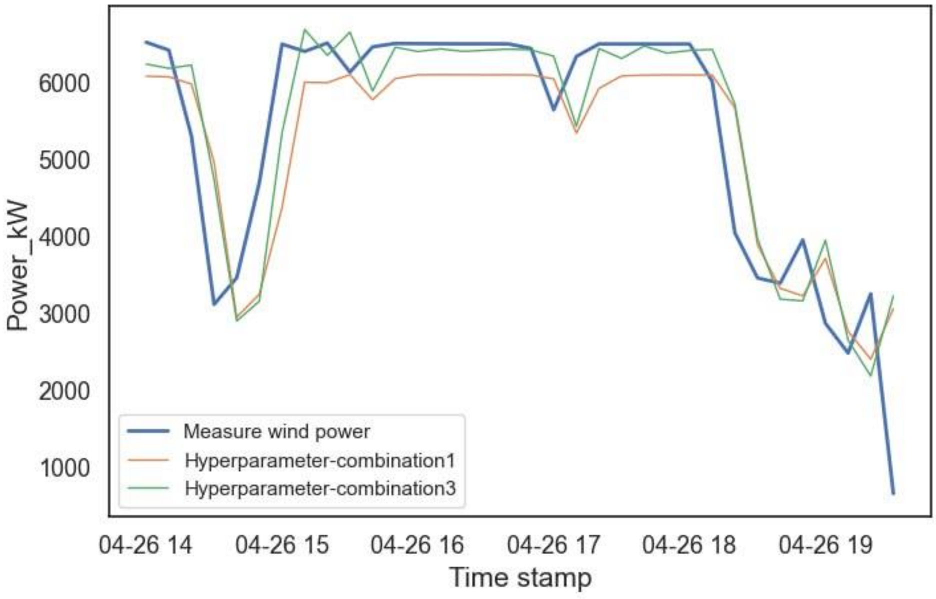


Figure 4.9. wind power prediction of LSTM model built by two different hyperparameter combinations between hours 14:00 and 19:00 of 26th of April

Chapter 5 - WPD-CNN-LSTM prediction model

5.1 Introduction

In this chapter, a novel hybrid forecasting model is presented for 10-minutes-ahead wind power forecasting of an offshore wind turbine in Scotland. The proposed model is based on applying WPD, optimised CNN and LSTM models without inputing future weather forecasting data. The proposed model is applicable when a complex non-linear relationship between the variables exists in the wind power time series data. The originality (novelty) of this part of the research lies in investigating the combined performance of WPD and sequential model-based optimised (SMBO) CNN and LSTM models for short-term offshore wind power forecasting. WPD can decompose the original time series data into different sub-series with different frequencies. CNN and LSTM with different performance in learning short-term and long-term dependencies can extract linear and non-linear relations from historical wind power data to make an accurate prediction. As discussed in chapter 2, hybrid wind power prediction methods combine different methods to utilise their unique merits and improve the overall prediction accuracy. Table 5.1 presents the improvement of prediction performance of some proposed hybrid methods in recent years. What is clear is that although combining different methods improves overall performance, on the other hand it complicates the model and increases the required computation time. Hence, it is vital to obtain a balance between accuracy and efficiency.

Table 5.1. Hybrid wind power prediction models

Combined model	Year	Inputs	Accuracy improvement	Features
ICEEMDAN, MOMFO, Wavelet NN [39]	2020	wind power	62.38% improvement in MAPE compared to wavelet NN	A robust hybrid method with appropriate accuracy and stability
LSTM, GMM [17]	2019	wind speed	Up to 4.96% RMSE improvement over traditional methods	LSTM used for prediction and GMM for uncertainty description
CNN, RBFNN, DGF [31]	2019	wind power	accurate than traditional models for 24 h-ahead wind power prediction	the novel double Gaussian function (DGF) employed for RBFNN
GA, LSTM [40]	2021	wind power	Up to 30% accuracy improvement compared to existing methods	GA application for LSTM window size and neurons number optimisation
BPNN, RBFNN, LSSVM [46]	2017	wind speed & direction, temp.	Significant improvement in accuracy.	Pearson correlation coefficient (PCC) is used to improve the mapping accuracy
MODA, ELNN [41]	2018	Wind speed	43.96% MAPE reductions compared to comparison models	MODA application for ELNN optimisation
IF, GRU, LSTM [42]	2021	wind power	IF filtering improved forecasting performance by over 92%.	A robust model with less sensitivity to noise in SCADA data

In chapter 4 it was discussed that the optimisation of hyperparameters can enhance prediction accuracy. It was also highlighted that instead of manual tuning, grid search, or random search that are relatively inefficient, SMBO methods can be used. It was demonstrated that these methods, by systematically evaluating hyperparameters that show greater potential based on recent research findings, can more efficiently converge on optimal hyperparameters. This approach not only reduces the time required to find the best hyperparameters but also improves the overall performance of the model by focusing computational resources on the most promising configurations [61]. Using these methods can increase the accuracy and efficiency of the prediction models.

Another method that can effectively improve the accuracy of power prediction methods is signal processing methods such as data decomposition, data noise removal, or data feature selection. All decomposition-based forecasting models published in the literature use the same framework. In this framework, the original non-stationary time series is decomposed into stationary sub-series. Independent forecasting models are then used to predict each sub-series. Finally, all predictions are added together to form the final forecast. Independent forecasting of each sub-series can efficiently enhance the prediction accuracy [64].

Su et al. [65] decomposed the wind speed data into four low-frequency and four high-frequency components by WPD. Then the four high-frequency components were decomposed into 60 intrinsic mode functions (IMFs) through ensemble empirical mode decomposition (EEMD). These components were then fed to individual LSTM models with yaw error and rotor speed data. The power prediction results of the proposed approach showed an improvement in accuracy. However, the effect of the direct application of the wind power dataset for prediction was not investigated. Zu et al. [66] used WPD to decompose wind power time series into three levels. The gained sub-series were fed to a gated recurrent unit (GRU), and the predictions were reconstructed to obtain the results. Experimental results showed that the proposed WPD-GRU-SELU model has a higher prediction accuracy than other recurrent neural network (RNN) models. In another research, Mujeeb et al. [67] combined wavelet packet transform (WPT) and deep convolutional neural network (DCNN) to predict the day-ahead hourly wind power of ISO New England's wind farm, however, the authors did not attempt to forecast the sub-series with different independent methods. In addition to WPD, other wavelet transform methods have recently been used in the field of wind power prediction. For example, Azimi et al. [68], with a combination of the K-means clustering method with discrete wavelet transform (DWT) and multilayer perceptron neural network (MLPNN), improved the wind power forecasting accuracy of the National Renewable Energy Laboratory (NREL). Shi et al. [69] employed variational mode decomposition (VMD) and LSTM to provide hourly predictions of day-ahead wind power of a Chinese wind farm. In another study, Liu et al. [70] combined empirical mode decomposition (EMD), LSTM, and Elman neural network (ENN) to develop a hybrid model, obtaining lower RMSE values in multi-step wind speed predictions compared to each model used individually. To obtain better forecasting results, some researchers use error correction mechanisms through application of the double decomposition methods. Double decomposition refers to a process that involves breaking down the data into two separate components to address different aspects of variability or error. For example, Ma et al. [71] used this decomposition approach with the LSTM model and proved the better prediction performance of the proposed model compared to models without double decomposition.

5.2 Methodology

The framework of the proposed WPD-LSTM-CNN model is demonstrated in figure 5.1, and the entire process is depicted in the following steps:

Step 1) The raw SCADA data of an offshore wind turbine are pre-processed by removing the obvious outliers, imputing the missing data, and resampling. The detail of the pre-processing part is described in Chapter 3, section 3.3 and 3.4.

Step 2) WPD decomposes the pre-processed wind power time series into several approximations and detail coefficients (sub-series). The detail of the decomposition method is described in Chapter 2, section 2.4.6.

Step 3) The optimised CNN with surrogate optimisation method is used to predict the high-frequency sub-layers obtained from the WPD. Details and structure of this method are provided in Chapter 2, section 2.2.

Step 4) The LSTM is tuned and employed for predicting the low-frequency sub-layer with details described in Chapter 2, section 2.3.

Step 5) After the prediction of each sub-layer, the final forecasting result is generated by summing all the predictions of the sub-layers. The result is compared with models including RF, FFNN, CNN, LSTM, WPD-LSTM and WPD-CNN to evaluate the forecasting performance of the developed method.

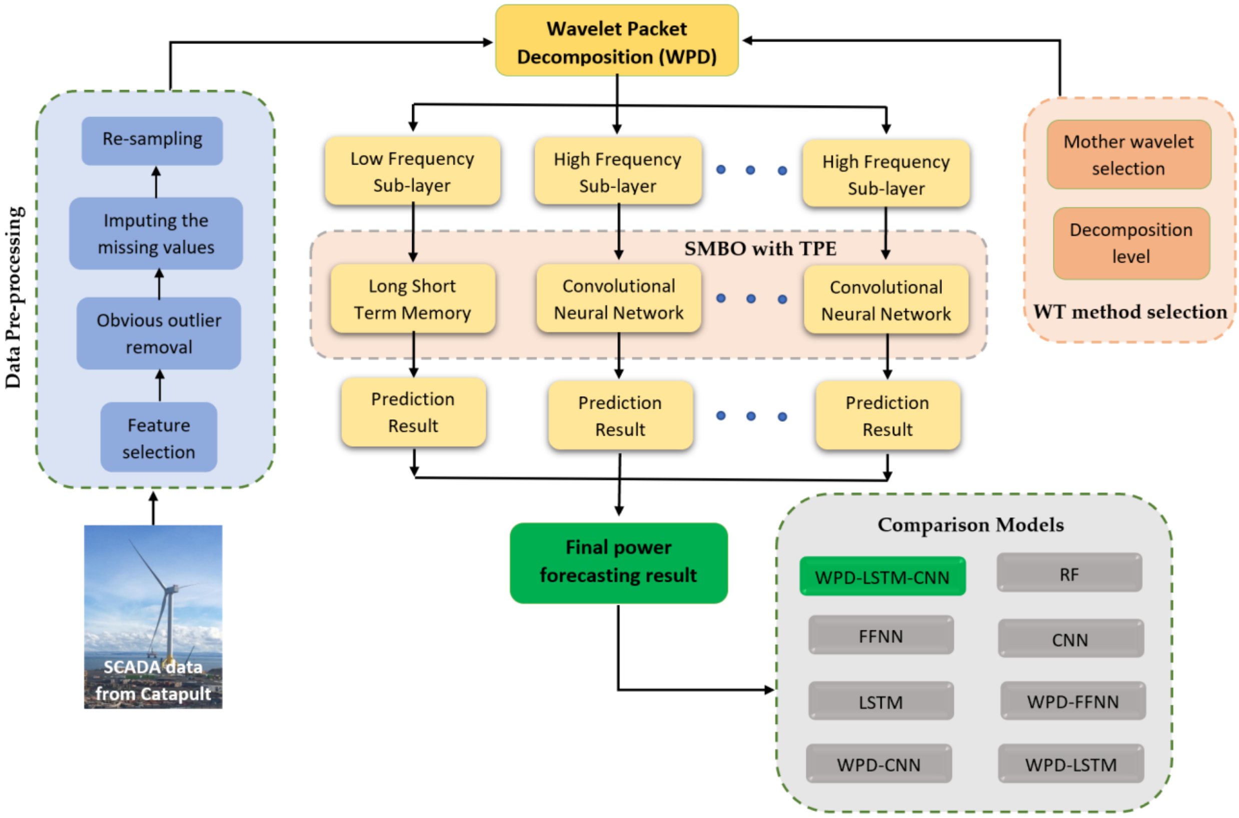


Figure 5.1. Diagram of the applied methodology

5.2.1 Wavelet Packet Decomposition (WPD)

As a signal processing method, WPD is an efficient mathematical solution for decomposing signals into approximation and detail components with different time frequencies [106]. Decomposing signals into approximation and detail components means breaking down the original signal into parts that capture different aspects of the signal's information: approximation components capture the overall trend (low-frequency components), while detail components capture finer, transient features (high-frequency components). WPD uses low-pass and high-pass filters to achieve this decomposition. The approximate coefficient, obtained by applying a low-pass filter, contains the low-frequency part of the signal and represents the long-term dependencies, essentially summarizing the overall trend of the signal. On the other hand, the detail coefficients, gained by applying a high-pass filter, include high-frequency components and depict the short-term dependencies, capturing the signal's fine details and rapid changes [107]. In contrast to the wavelet decomposition process in which only the approximation coefficients are decomposed, in WPD, the detail coefficients can also be decomposed [93]. As a result, it can contribute to higher accuracies in signal analysis than normal wavelet transforms methods. In addition, through decomposition by WPD, the high-frequency component of the signal can have a better resolution [108].

There are two types of WPDs: discrete wavelet transform and continuous wavelet transform. Continuous wavelet transform for a signal *f(t)* can be described as:

$$CWT_f(a,b) = \langle f(t), \Psi_{a,b}(t)\rangle = \int_{-\infty}^{+\infty} f(t)\Psi * ((t-b)/a)/\sqrt{a}dt \tag{43}$$

where $\Psi(t)$ denotes the selected mother wavelet function, a and b are the scale and translation coefficients, respectively, and * indicates the complex conjugate. The scale and translation coefficient in discrete wavelet transform can be explained by:

$$\begin{cases} a = 2^j \\ b = k2^j \end{cases} \tag{44}$$

where j and k are scale and translation factors, respectively, the following equations can illustrate the decomposition process of WPD:

$$\begin{cases} P_j^{2i-1}(t) = HP_{j-1}^i(t) \\ P_j^{2i}(t) = GP_{j-1}^i(t) \end{cases} \tag{45}$$

where *t* is the time index, P_j^i represents the *i*-wavelet packet for level *j* and *H* and *G* are the low- and high-pass filters.

and the reconstruction process can be described as follows:

$$P_j^i(t) = H * P_{j+1}^{2i-1}(t) + G * P_{j+1}^{2i}(t) \tag{46}$$

The reconstruction process involves combining the decomposed components to recreate the original signal. Specifically, the approximate and detail coefficients obtained from the decomposition process are passed through their respective inverse filters (conjugates of *H* and *G*) and then summed. This process effectively rebuilds the original signal from its decomposed parts, ensuring that both the low-frequency (long-term trends) and high-frequency (short-term details) information are accurately preserved and integrated.

The performance of WPD is highly dependent on the selected mother wavelet and the chosen level of decomposition. According to the literature, the normal decomposition level is in the range of 2 to 4 for time series prediction models [109]. In this work, the 2-level framework of WPD is employed with the schematic diagram shown in figure 5.2.

In addition, various mother wavelets are examined due to the impact of the mother wavelet on the decomposition performance and prediction accuracy. In this study, sixteen wavelets from four widely used wavelet families in the literature (Daubechies, Haar, Sym, and Coif) were selected [106, 107, 109], and their performance in prediction improvement of forecasting models, including LR, RF, FFNN and LSTM are assessed.

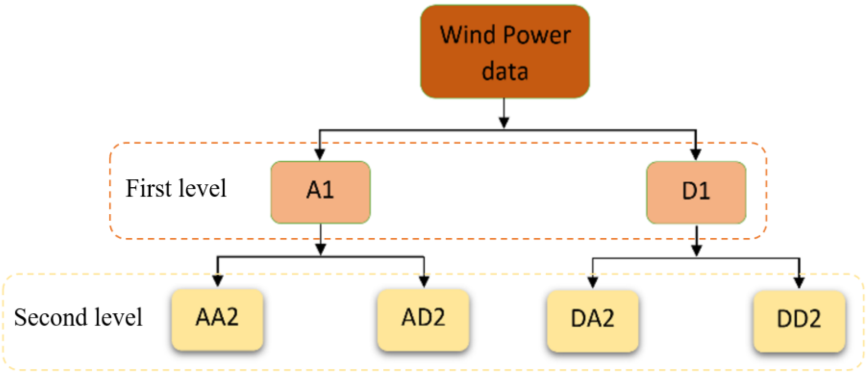


Figure 5.2. Schematic diagram of WPD with two layers.

Table 5.2 and Figure 5.3 show the RMSE values of wind power forecasts based on the application of different mother wavelets. The forecasting models used are linear regression, random forest, feed-forward neural network (NN), and LSTM. These models were trained using 90 percent of the data as training data, and the RMSE of predictions is shown for the remaining 10 percent of the data. As can be seen, the Daubechies wavelets of order 5 have the best performance (least prediction error) and are therefore selected as the mother wavelet in this study. These four different models were chosen to leverage their unique strengths: linear regression for its simplicity and interpretability, random forest for its robustness and ability to handle non-linear relationships, feed-forward NN for its capability to model complex patterns, and LSTM for its proficiency in capturing temporal dependencies in sequential data.

The corresponding decomposition result of the application of WPD for the wind power time series used in this research is shown in figure 5.4. The upper graph in the figure represents the wind power time-series before decomposition and the next four graphs show four sub-series obtained after decomposition.

Table 5.2. RMSE values of wind power forecast based on the application of different mother wavelets

Model	**db1**	**db2**	**db3**	**db4**	**db5**	**db6**	**db7**	**db8**	**db9**	**db10**	**haar**	**sym2**	**sym5**	**sym8**	**coif1**	**coif5**
LR	795	788	783	784	712	799	804	804	799	795	774	788	801	803	786	803
RF	799	755	735	722	687	710	714	713	711	704	722	757	730	734	729	719
FFNN	822	773	757	732	686	737	730	753	750	764	754	774	765	768	757	761
LSTM	812	764	746	722	690	723	717	733	730	728	746	765	747	751	743	740

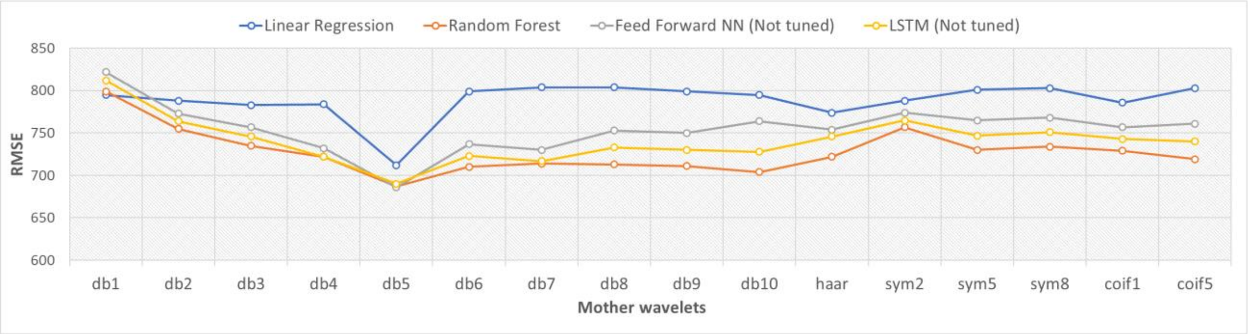


Figure 5.3. Prediction performance of forecasting models based on decomposition with different mother wavelets

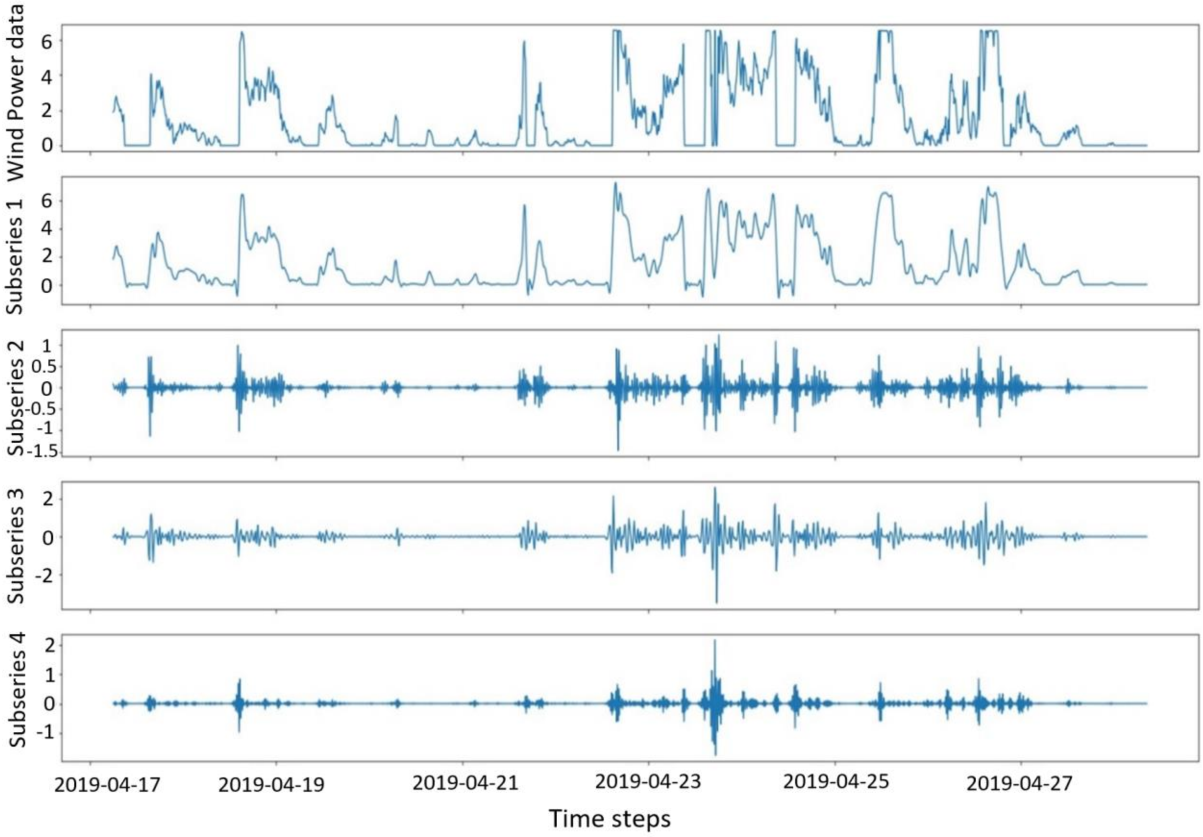


Figure 5.4. Wavelet packet decomposition result of wind power time series

To improve the decomposition, leading to more accurate predictions, this research employs the single branch reconstruction method. This technique is crucial during the process of reconstructing each component of the final decomposition back to the original level. The single branch reconstruction method operates by considering the values of other components at the same level as zero while reconstructing each individual component. This means that when one component is being reconstructed, all other components are temporarily disregarded (set to zero). By isolating each component in this way, the reconstruction process can more accurately capture the individual characteristics of each component, reducing interference from other components and thereby enhancing the overall prediction accuracy. This method ensures that each component's unique contribution to the original signal or data is more precisely understood and accounted for, ultimately leading to better predictive performance [93].

High-frequency components of wind power time series have short-term dependencies, while low-frequency components have long-term dependencies. Detection of short-term dependencies in high-frequency components is possible with fully connected layers, and calculations are performed faster than LSTM recurrent layer [92]. Therefore, this thesis selects the CNN model to predict the high-frequency sublayers. On the other hand, the LSTM recurrent layer, which can better sequentially process the temporal data with long-term dependencies, is used for predicting the low-frequency sub-series.

5.3 Results of comparison of the developed model with other prediction models

In order to investigate the prediction performance of the proposed method, the SCADA data of the Leven mouth Demonstration Turbine (LDT) which was processed in Chapter 3 and 4 is divided into four equal experimental parts, each including 4200 samples. Each model is trained using the training set (samples 1-3800) to learn the patterns and relationships within the data. The training process involves optimizing the model parameters to minimize prediction errors on the training set. Once trained, the models' performances are evaluated on the testing set (samples 3801-4200) to assess their ability to generalize and accurately predict unseen data. Figure 5.5 shows these four power time series, and Table 5.3 provides their statistical descriptions. For comparison, seven wind power forecasting models, including the RF, FFNN, CNN model, LSTM model, WPD-FFNN model, WPD-CNN model, WPD-LSTM model, and the proposed WPD-CNN-LSTM model were selected.

To better investigate the forecasting performance of the various models, all selected models have similar parameters; for example, the selected time lag (input layer length) was set at 10 for all of them. The values of MAE, MSE, RMSE and R-square of all prediction models for the four datasets are shown in Tables 5.4 and 5.5.

Table 5.3. Statistical descriptions of the wind power data

Data	Mean	Min	Max	Standard Derivation
Dataset 1	1794.2	0	6566.9	2167.8
Dataset 2	2029.7	0	6567.5	2316.3
Dataset 3	1372.4	0	6569.6	2040.3
Dataset 4	2121.3	0	6551.2	2349.9

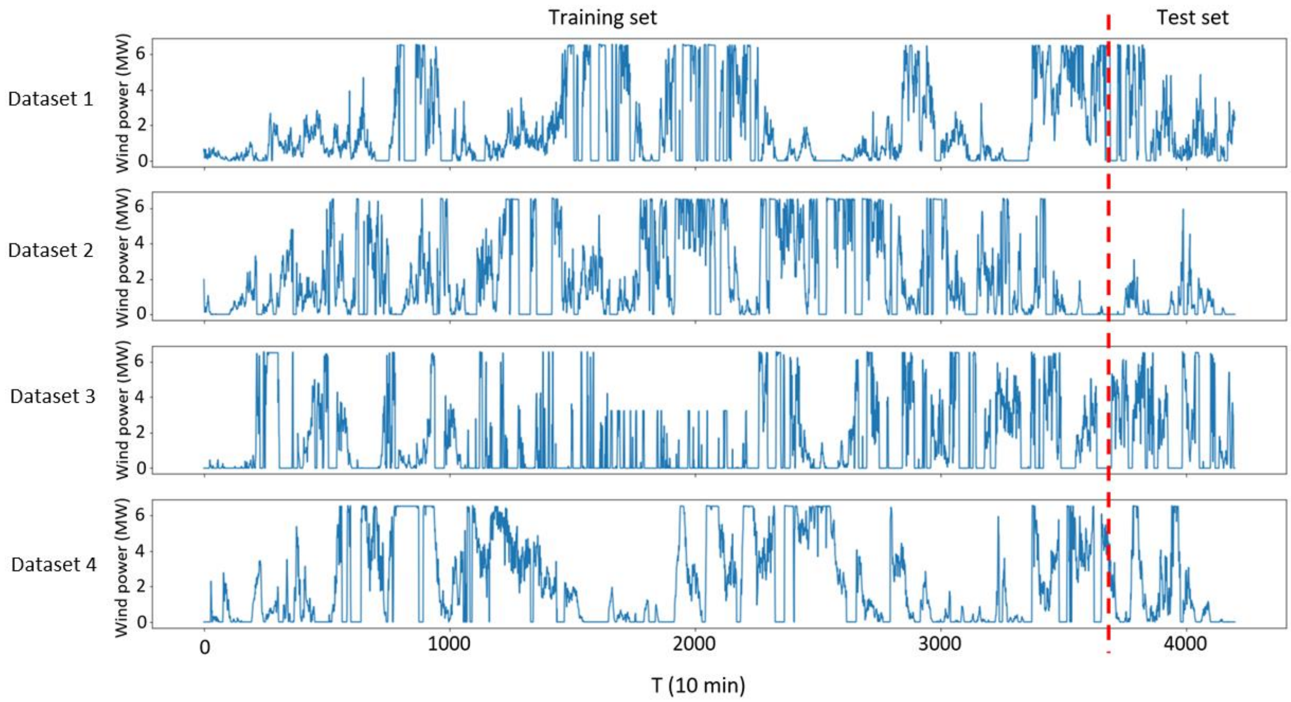


Figure 5.5. Four sets of 10-min averaged wind power time series

Table 5.4. Performance comparison between WPD-CNN-LSTM and other models for datasets 1 and 2

Comparison models	**Dataset 1**				**Dataset 2**			
	MSE	**RMSE**	**MAE**	**R-square**	**MSE**	**RMSE**	**MAE**	**R-square**
RF	350741.8	592.2	405.8	0.836	177039.3	420.8	190.5	0.73
FFNN	329902.1	574.3	402	0.846	140227.3	374.4	158.9	0.786
CNN	309210.7	556.1	392.1	0.856	144140.7	379.6	167.1	0.78
LSTM	300599.8	548.3	386.2	0.858	142455.1	377.4	208.1	0.783
WPD-FFNN	28240.2	168	118.5	0.987	9741.4	98.6	48.3	0.985
WPD-CNN	16890.39	129.9	99.88	0.8937	6970.04	83.5	44.66	0.891
WPD-LSTM	16844.8	129.7	95.3	0.992	6348.4	79.6	46.1	0.99
WPD-CNN-LSTM	15354.9	123.9	90.8	0.993	6336.4	79.6	40.6	0.99

Figure 5.6 shows the prediction results of all forecasting models for the last day of the dataset 1, and figure 5.7 shows the same forecast for only two hours of the last day.

Table 5.5. Performance comparison between WPD-CNN-LSTM and other models for the datasets 3 and 4

Comparison models	**Dataset 3**				**Dataset 4**			
	MSE	**RMSE**	**MAE**	**R-square**	**MSE**	**RMSE**	**MAE**	**R-square**
RF	1303094	1141.5	636.5	0.741	287230.2	535.9	300.8	0.914
FFNN	1297934.5	1139.2	667.3	0.741	254659.1	504.6	269.8	0.923
CNN	1249351.3	1117.7	658.3	0.751	241560.7	491.5	269.1	0.928
LSTM	1181880.9	1087.1	631	0.769	235092.6	484.9	272.6	0.929
WPD-FFNN	66242.6	257.3	172.2	0.986	16465.3	128.3	72.1	0.995
WPD-CNN	44450.1	210.8	148.2	0.991	13064.04	114.3	71.39	0.8964
WPD-LSTM	46707.87	216.1	161.37	0.8919	11220.9	105.9	64.7	0.997
WPD-CNN-LSTM	42461.7	206.1	146.7	0.991	11876.4	108.9	64.9	0.996

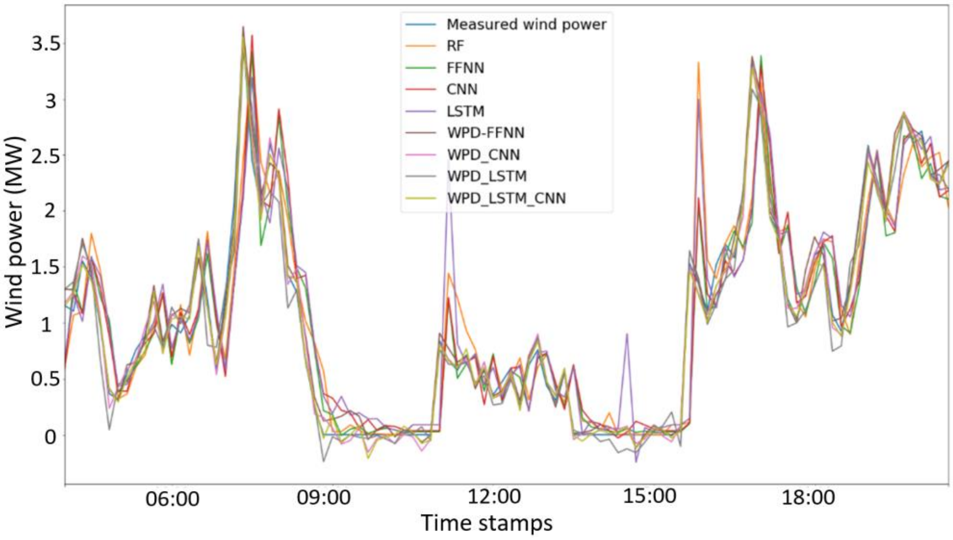


Figure 5.6. Forecasting results of the involved models for dataset 1

As can be seen from Tables 5.4 and 5.5 and figures 5.6 and 5.7, when the wind power generation encounters abrupt changes, the methods that use WPD to decompose the data have a better prediction performance than the other methods. Figure 5.8-5.11 show the forecasting results of the proposed WPD-LSTM-CNN model for datasets 1 to 4, respectively.

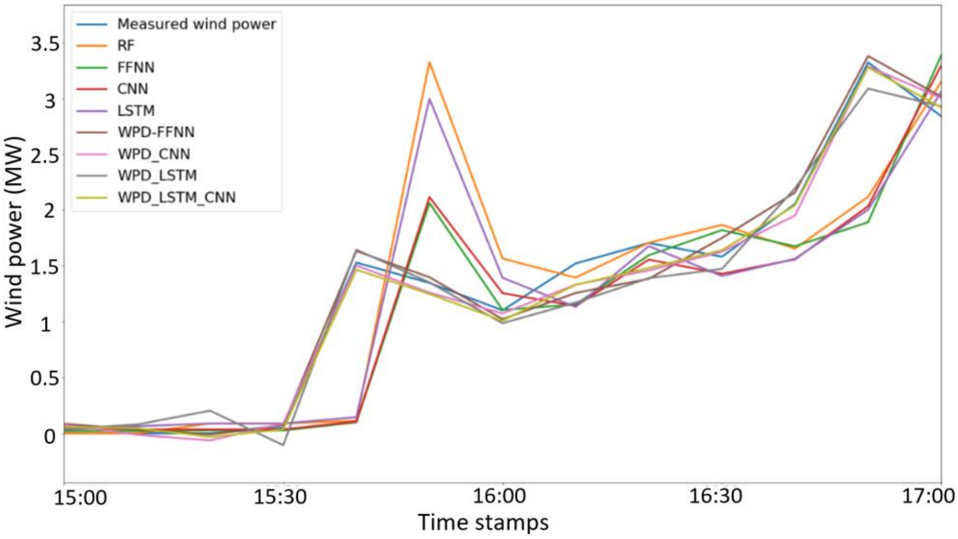


Figure 5.7. Forecasting results of the involved models for two hours of dataset 1

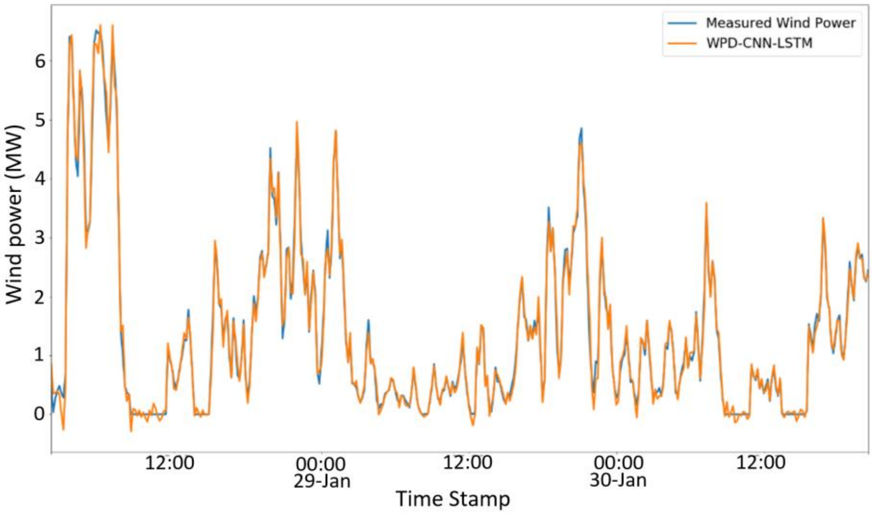


Figure 5.8. wind power forecasting with the proposed WPD-LSTM-CNN model for data set #1

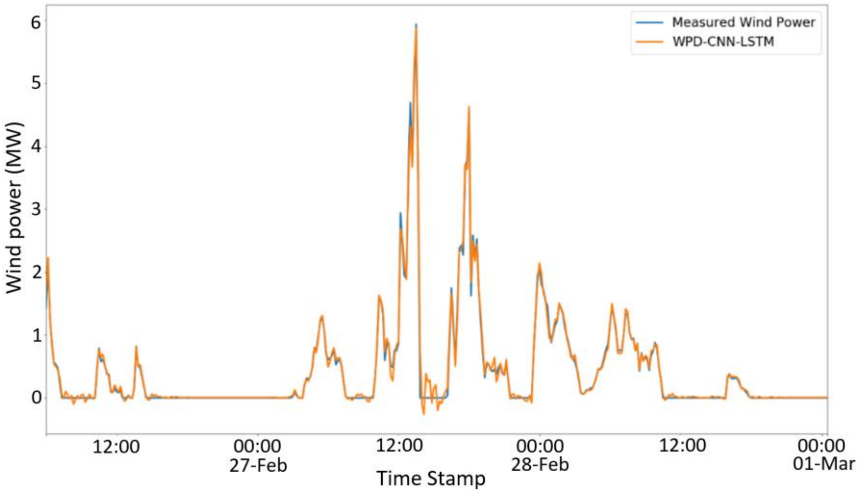


Figure 5.9. wind power forecasting with the proposed WPD-LSTM-CNN model for data set #2

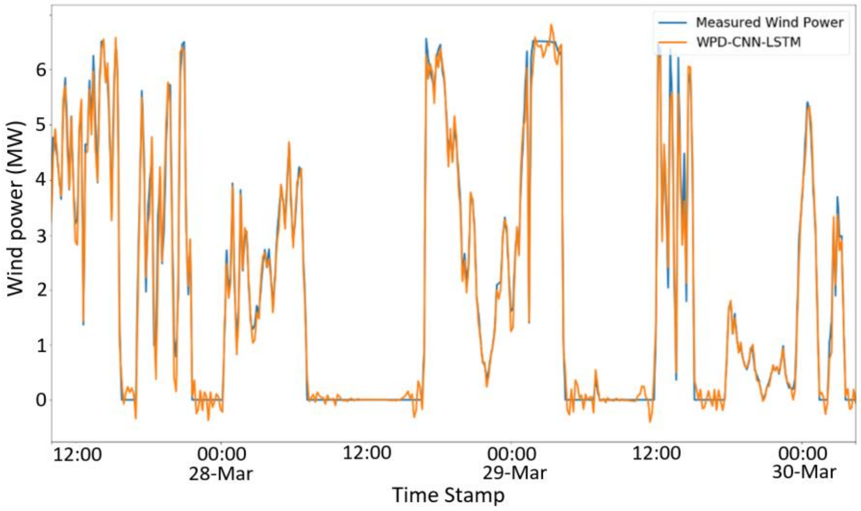


Figure 5.10. wind power forecasting with the proposed WPD-LSTM-CNN model for data set #3

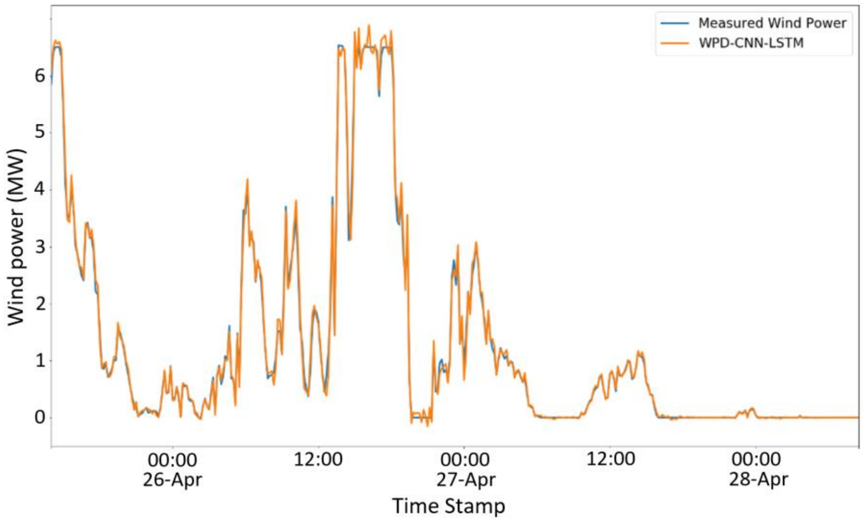


Figure 5.11. wind power forecasting with the proposed WPD-LSTM-CNN model for data set #4

Based on figures 5.8 to 5.11 and Tables 5.4 and 5.5, the following conclusions are drawn:

1) Comparing the prediction performance of the FFNN, CNN and LSTM models with WPD-FFNN, WPD-CNN and WPD-LSTM models, respectively, the significant impact of WPD on improving the prediction capability is evident;
2) The developed WPD-CNN-LSTM model has the highest prediction precision among all the models which are considered here;
3) A combination of two optimised deep learning methods, CNN, and LSTM, for the prediction of different sub-series increases the forecasting accuracy compared to forecasting all sub-series with only one of them;

For further assessment of the forecasting performance of the proposed hybrid model, the $P_{MSE,}$ $P_{RMSE,}$ P_{MAE} of the trial tests are used to provide a comparative analysis between the WPD-LSTM-CNN model and other forecasting models. Table 5.6 provides a comparative analysis between the proposed model and other involved forecasting models for the four experimental tests, respectively.

Table 5.6. Promoting percentages of the involved forecasting models by the WPD-CNN-LSTM model

Promoting percentages	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Dataset 4**	**Average**
P_{RMSE} (%)					
RF	79.08%	81.08%	81.94%	80.24%	80.59%
FFNN	78.43%	78.74%	81.91%	79.01%	79.52%
CNN	77.72%	79.03%	81.56%	78.45%	79.19%
LSTM	77.40%	78.91%	81.04%	78.16%	78.88%
WPD-FFNN	26.25%	19.27%	19.90%	17.46%	20.72%
WPD-CNN	4.62%	4.67%	2.23%	7.35%	4.72%
WPD-LSTM	4.47%	0.09%	4.63%	2.75%	2.99%
P_{MSE} (%)					
RF	95.62%	96.42%	96.74%	96.09%	96.22%
FFNN	95.35%	95.48%	96.73%	95.59%	95.79%
CNN	95.03%	95.60%	96.60%	95.35%	95.65%
LSTM	94.89%	95.55%	96.41%	95.23%	95.52%
WPD-FFNN	45.63%	34.95%	35.90%	31.85%	37.08%
WPD-CNN	9.09%	9.09%	4.47%	14.11%	9.19%
WPD-LSTM	8.84%	0.19%	9.09%	5.52%	5.91%
P_{MAE} (%)					
RF	77.62%	78.69%	76.95%	78.49%	77.94%
FFNN	77.41%	74.45%	78.02%	76.02%	76.47%
CNN	76.84%	75.70%	77.72%	75.96%	76.55%
LSTM	76.49%	80.49%	76.75%	76.27%	77.50%
WPD-FFNN	23.38%	15.94%	14.81%	10.26%	16.10%
WPD-CNN	9.09%	9.09%	1.01%	9.38%	7.14%
WPD-LSTM	4.72%	11.93%	9.09%	0.31%	6.51%

Based on the reported promoting percentages in Table 5.6, it can be recognised that:

1) The developed WPD-LSTM-CNN is the most accurate short-term forecasting model for wind power time series among all evaluated models;

2) The WPD-LSTM-CNN model outperforms all forecasting models based on the use of non-decomposed data. For example, in experimental dataset 1, the RMSE value of the WPD-LSTM-CNN model, compared to models RF, FFNN, CNN and LSTM, was reduced by 79.08%, 78.43%, 77.72% and 77.4%, respectively, and the MAE value of the proposed model for the same experimental dataset, compared to models RF, FFNN, CNN and LSTM was reduced by 77.62%, 77.41%, 76.84% and 76.49%, respectively;
3) Comparing the proposed model with other decomposed-based models, as shown in Table 5.6, the WPD-LSTM-CNN model outperforms the WPD-FFNN model. For example, in experimental dataset 2, the RMSE values of the developed model are reduced by 19.27%, the MSE value is reduced by 34.95%, and the MAE level is reduced by 15.94% compared to the WPD-FFNN;
4) The developed model can also outperform the WPD-CNN model. As can be seen from the evaluation criteria of the experimental dataset 3, for example, the RMSE, MSE and MAE values of the WPD-LSTM-CNN model, compared to the WPD-CNN, are reduced by 2.23%, 4.47% and 1.01%, respectively.
5) The WPD-LSTM-CNN can also outperform the WPD-LSTM model. For example, in experimental dataset 4, the RMSE, MSE and MAE values of the WPD-CNN-LSTM model, compared to the WPD-LSTM, are decreased by 2.75%, 5.52% and 0.31%, respectively.

Chapter 6 - Multi-step ahead prediction

6.1 Introduction

One-step ahead prediction of wind power has been investigated in several studies, but the results of these investigations do not provide sufficient assurance for the controllability of wind power systems. Up until this point in the thesis, we have also focused on one-step-ahead predictions. However, by increasing the prediction horizon and analysis of power production for the coming hours, the penetration increment of wind energy into the power grid will be facilitated [110].

Accurate multi-step ahead wind power prediction is a growing field of interest as it brings several advantages [7]. For example, by accurately predicting the generated power of the day-ahead, an efficient energy trading strategy can be scheduled [111]. In addition, accurate multi-step ahead forecasting leads to improved system stability, meaning that the power grid can better handle fluctuations in wind power generation, reducing the risk of outages and ensuring a consistent supply of electricity. This improved stability allows for better planning for the operation and maintenance of wind turbines. Furthermore, it enables a balance between production and demand [112]. Multi-step ahead forecasting tasks are more challenging compared to one-step ahead forecasting since they must deal with several complications, including the accumulation of prediction errors of previous steps, lack of information, and the reduction of accuracy. Therefore, regardless of the selected model for forecasting, choosing the appropriate modelling strategy for multi-step ahead forecasting is an important issue for research and practical applications [113].

The most used strategies in the literature for multi-step ahead forecasting of wind speed or wind power are categorised into three main approaches: the recursive or iterated, direct, and multiple-input multiple-output (MIMO) strategies. Table 6.1 shows several applications of these strategies, conducted in recent years for multi-step ahead wind speed/power forecasting. The application of these strategies can be seen in several research conducted in the past few years in Table 6.1.

In the recursive strategy, a one-step ahead prediction model is used to predict one step ahead. Then the predicted values are used as input data to predict the next step and this process continues until the desired horizon is reached. Using previous predictions to predict the next step makes this strategy susceptible to error accumulation [125].

Table 6.1. Application of different strategies for multi-step ahead wind power/speed forecasting

Literature	**Year**	**Predicted feature**	**Simulation strategy**	**Highlights of the prediction process**
Y. Fu et al. [114]	2021	Wind power	Recursive	1 step to 4 step prediction, based on RNN, LSTM or GRU
J. Wang et al. [115]	2018	Wind speed	Recursive	Application of VMD and Kullback-Leibler divergence to extract optimal features.
Y. Huang et al. [116]	2019	Wind speed	Recursive	An integration of EEMD, LSTM and GRNN
Z. Liang et al. [117]	2016	Wind power	Recursive	One step and multi-step ahead prediction based on error forecast correction
H. Liu et al. [118]	2018	Wind speed	Recursive	A combination of Empirical wavelet transforms and RNNs
H. Liu et al. [119]	2010	Wind speed/power	Recursive	An integration of WT and LSTM
X. Shi et al. [69]	2018	Wind power	Recursive / direct	(Recursive and direct prediction based on VMD and LSTM
Y. Li et al. [120]	2018	Wind speed	Direct	An integration of EWT, LSTM, RELM and IEWT
H. Liu et al. [121]	2018	Wind speed	Direct	A combination of WPD, CNN and CNNLSTM
H. Liu et al. [122]	2018	Wind speed	Direct	A hybrid method based on WPD, EMD and ELM
Z. Wu et al. [123]	2010	Wind power	MIMO	Secondary decomposition and non-agnostic uncertainty sampling-active learning-sample selection strategy
L. Xiao et al. [110]	2017	Wind speed	MIMO	based on decomposing algorithms and modified neural networks
Meng et al. [124]	2016	Wind speed	MIMO	Up to 5 h prediction using WPD and NNs trained by crisscross optimisation algorithm

This strategy has been used in several studies to predict wind power or wind speed. Fu et al. [114] employed this strategy to create a hybrid model based on the application of LSTM and GRU to predict wind power for up to four steps (2-hours) ahead. The results of their experiments showed that the prediction error increases with the increment of the prediction horizon. In another study, Liang et al. [117] used the recursive strategy for one step and multi-step ahead wind power forecasting of a wind farm in China, based on application of support vector machine (SVM) and extreme learning machine (ELM). They showed the superior prediction performance of their proposed model compared with other predictive models. Nonetheless, it was found that the accuracy of all models decreases with the increment of forecasting steps. Liu et al. [119] applied the recursive strategy for five and ten steps (50 to 100 minutes) ahead predictions of both wind speed and wind power. The results of their experiments showed that the prediction error almost doubled with the increment of the prediction horizon from 5 to 10.

In contrast to the recursive strategy, which uses a single model for forecasting, in the direct strategy, a separate model is built for each forecasting horizon and only previous real observations are used as input data. Although the probability of accumulation of errors in this strategy is reduced, the need to create multiple models for each forecasting horizon reduces the efficiency of the strategy [113]. Li et al. [120] employed the direct strategy for one to five steps ahead wind speed forecasting of a wind farm using a hybrid model combining the Empirical Wavelet Transform (EWT), the LSTM, and the regularised extreme learning machine (RELM). Although in their research, the proposed method was compared with other methods, the effect of increasing the prediction horizon on the prediction accuracy was not investigated. In another project, Liu et al. [121] used the direct approach for one to three steps ahead wind speed prediction and noted the prediction error rise with increment of the prediction horizon but again, the effect of the selected strategy on the accuracy of multi-step ahead prediction was not investigated.

To address the drawbacks of previous approaches, the MIMO strategy was proposed by Taieb et al. [126]. In this strategy, unlike previous strategies, the forecast value in each single step, instead of a scalar quantity, is a vector of future values whose length equals the forecast horizon. The MIMO strategy is used in several studies. For example, Wu et al. [123] used the MIMO strategy in a hybrid method based on secondary decomposition-model selection for two and three steps ahead wind power forecasting of a wind turbine in Belgium. Xiao et al. [110] used the MIMO strategy in a wind speed forecasting model based on decomposition algorithms and NNs. It was found that with the increase of the forecasting horizon, the prediction error increased, as expected. The effect of the chosen strategy on the prediction performance was not investigated in this research either.

The three afore mentioned strategies (direct, MIMO, and recursive) have been used separately in the literature, and to the best of the author's knowledge, no researcher has used all three simultaneously, especially in the field of wind power or wind speed forecasting. Therefore, it has not been possible to determine which strategy is the best approach for wind power prediction. Hence, the main contribution of this chapter is to present a thorough comparison of the existing strategies for multi-step ahead wind power forecasting.

All experimental comparisons within this thesis are performed on two wind power time series datasets that were pre-processed with the same method. In addition, the LSTM deep learning model is selected as the prediction tool in all experiments, as the aim of this study is not to make a comparison of pre-processing or ML methods, but rather to demonstrate the impact of the forecasting strategy on the prediction performance for a given dataset and prediction algorithm.

Finally, the Optuna hyperparameter optimisation algorithm, is used in this section to tune the hyperparameters of the LSTM model. The Optuna optimisation algorithm, based on the TPE search method and EI acquisition function, has been successfully applied in previous chapters [3]. Using the Optuna hyperparameter optimization algorithm, enhances the performance of the prediction models by selecting an effective hyperparameter combination [3].

The proposed procedure is demonstrated in Figure 6.1, beginning with the selection of three needed features including the time stamps, wind power, and wind speed from raw SCADA data of two wind turbines, one in Scotland and the other in Turkey. Using the IF outlier detection method, the anomalies are removed with the aim of improving the prediction accuracy. After finishing the data pre-processing step, supervised training and test data are provided through three different strategies for multi-step ahead forecasting. Afterwards, LSTM models are tunned using the hyperparameter optimisation algorithm to reach the highest possible prediction accuracy in the defined domain of hyperparameters. Finally, the resulting prediction performances of the various strategies are compared to select the best strategy.

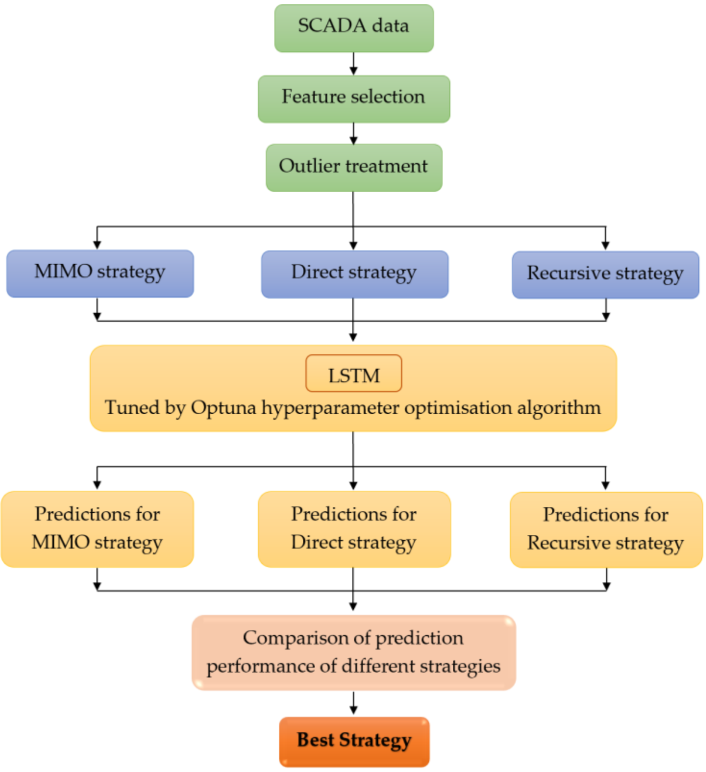

SCADA data
Feature selection
Outlier treatment
MIMO strategy
Direct strategy
Recursive strategy
LSTM
Tuned by Optuna hyperparameter optimisation algorithm
Predictions for MIMO strategy
Predictions for Direct strategy
Predictions for Recursive strategy
Comparison of prediction performance of different strategies
Best Strategy

Figure 6.1. Diagram of the applied methodology

6.2 Wind power dataset

In this section two real wind power datasets are used for the comparison between the different strategies of multi-step ahead wind power forecasting, namely an offshore wind turbine in Scotland and a wind turbine in Turkey [3, 42]. Figure 6.2 shows the power-speed curve of these two wind turbines, Table 6.2 demonstrates the statistical description, and figure 6.3 shows the histogram of their generated power. Two-month period from 1st March 2019 to 30th April 2019 of the first turbine in Scotland and a two-month period from 1st September 2018 to 31st October 2018 of a Turkish wind turbine is selected for training and testing all multi-step ahead forecasting models in all strategies.

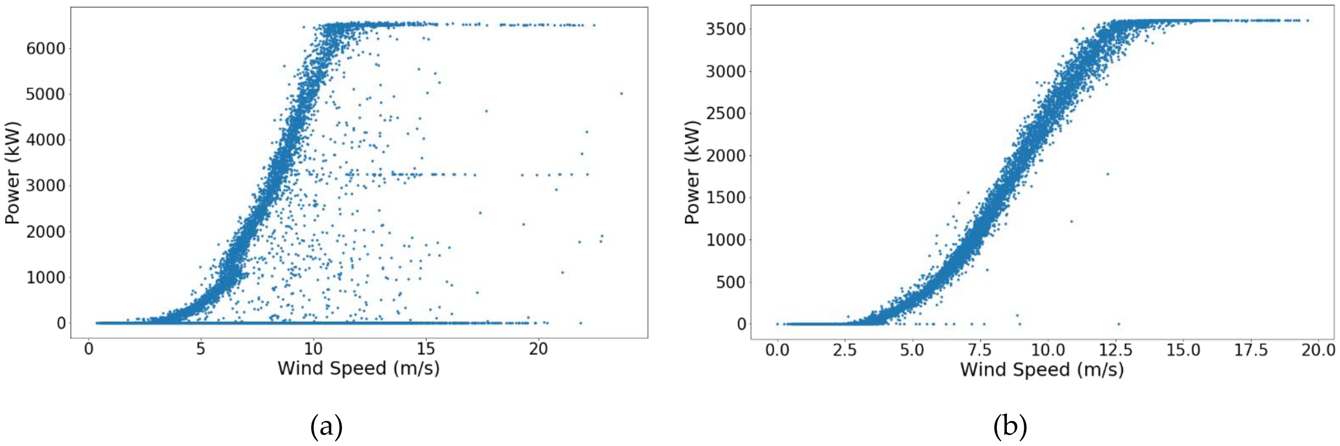
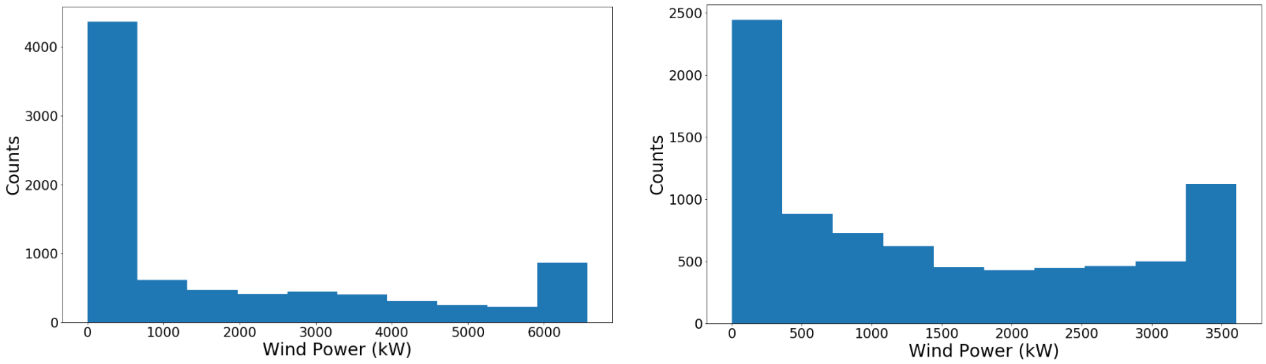


Figure 6.2. wind power-wind speed curve of (a) wind turbine 1 in Scotland and (b) wind turbine 2 in Turkey

Table 6.2. The statistical description of generated wind power of two wind turbines

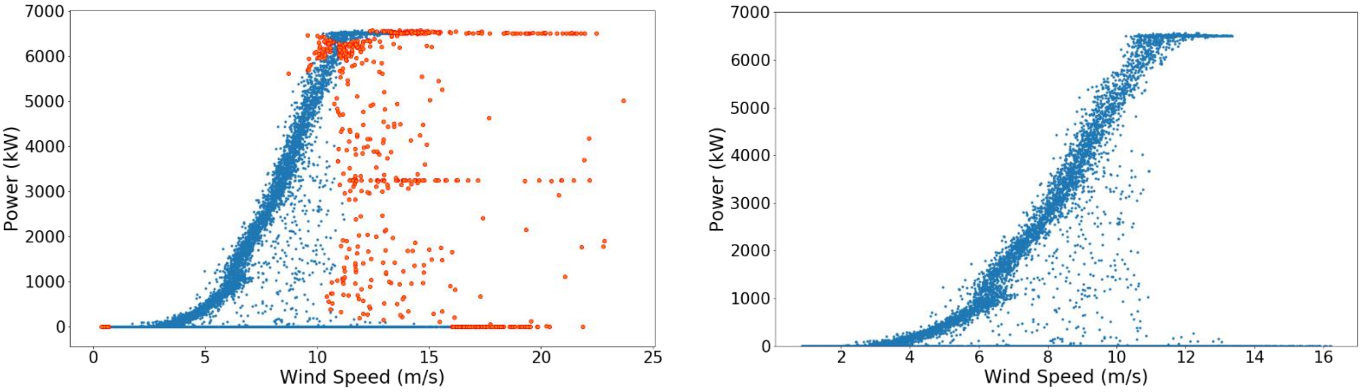
	Wind turbine 1 (Scotland), kW	**Wind turbine 2 (Turkey), kW**
Count	8352	8083
Mean	1756.87	1418.77
Standard deviation	2234.58	1263.72
Minimum	0	0
25%	0	211.39
Medium	470.88	1077.54
75%	3241.19	2574.52
Maximum	6569.58	3604.58



(a) (b)

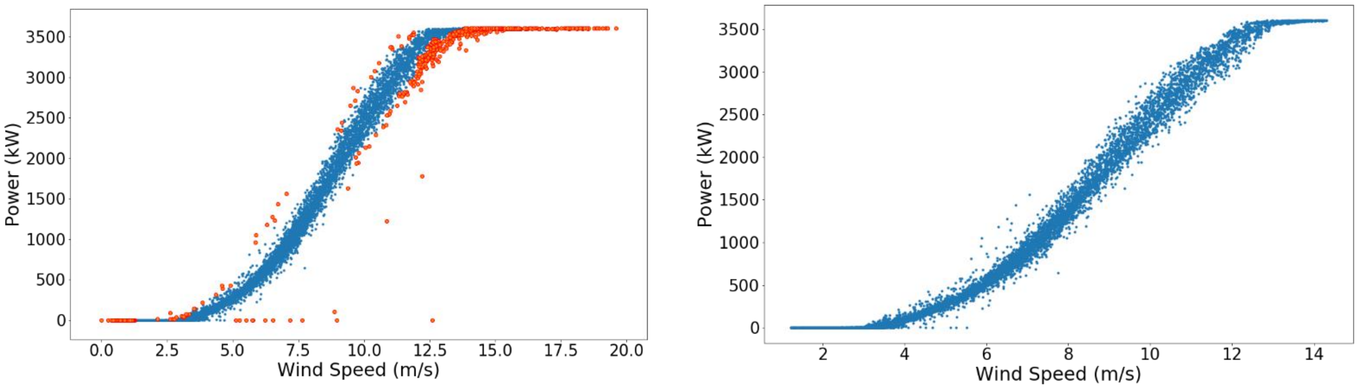
Figure 6.3. Histograms of wind power of (a) wind turbine 1 in Scotland, and (b) wind turbine 2 in Turkey

As evident from figure 6.2, both wind turbine datasets have some outliers. Regardless of the reason for these anomalies, it is necessary to remove them in order to have an accurate prediction. A variety of outlier detection methods are available such as GP based methods. The IF method has been used based on its satisfying performance in chapter 3 and similar work [3]. The detected outliers of the two wind turbines with the IF method are shown in Figures 6.4 and 6.5. These detected anomalies are removed from the datasets.



(a) (b)

Figure 6.4. Outlier detection and treatment for wind turbine 1 in Scotland, (a) detected outliers in red, (b) outlier removed data



(a) (b)

Figure 6.5. Outlier detection and treatment for wind turbine 2 in Turkey, (a) detected outliers in red, (b) outlier removed data

6.3 Strategies for multi-step ahead wind power forecasting

A multi-step ahead time series forecasting can be defined as a prediction of the next H>1 (y_{N+1}, …, y_{N+H}) values based on the N number of present and prior observations. In this study, three widely used strategies in multi-step ahead wind power forecasting including recursive, direct, and MIMO strategy are briefly discussed and then used for two steps to six steps ahead forecasting of two wind turbines in Europe.

Throughout this section, the notation f is used to denote the functional dependency between the past and future observations, T represents the number of past observations considered to predict future values, and w represents the error or noise of the prediction.

6.3.1 Method 1: recursive strategy

As previously mentioned, in the recurrent strategy, a one-step ahead prediction model is trained and used for one-step ahead prediction, i.e.,

$$\hat{y}_{t+1} = f(y_t, \dots, y_{t-T+1}) + w \tag{46}$$

The recursive strategy is shown schematically in figure 6.6. For the prediction of further steps, the first predicted step is considered as part of the input data to predict the next step. This process of using the predictions as part of the input data to predict the next step continues until the entire forecast horizon is achieved. Considering $\hat{f}$ as the one-step ahead prediction model, the predictions of each step can be defined as:

$$\hat{y}_{N+H} = \begin{cases} \hat{f}(y_N, \dots, y_{N-T+1}) & if\ H = 1 \\ \hat{f}(\hat{y}_{N+H-1}, \dots, \hat{y}_{N+1}, y_N, \dots, y_{N-T+H}) & if\ H\ \in \{2, \dots, T\} \\ \hat{f}(\hat{y}_{N+H-1}, \dots, \hat{y}_{N+H-T}) & if\ H\ \in \{T+1, \dots, H\} \end{cases} \tag{47}$$

The existing noise in the wind power time series and errors of predictions in each step decreases the performance of the recursive strategy in multi-step forecasting, especially when the forecasting horizon H exceeds the dimension of the past observations T considered to predict future values.

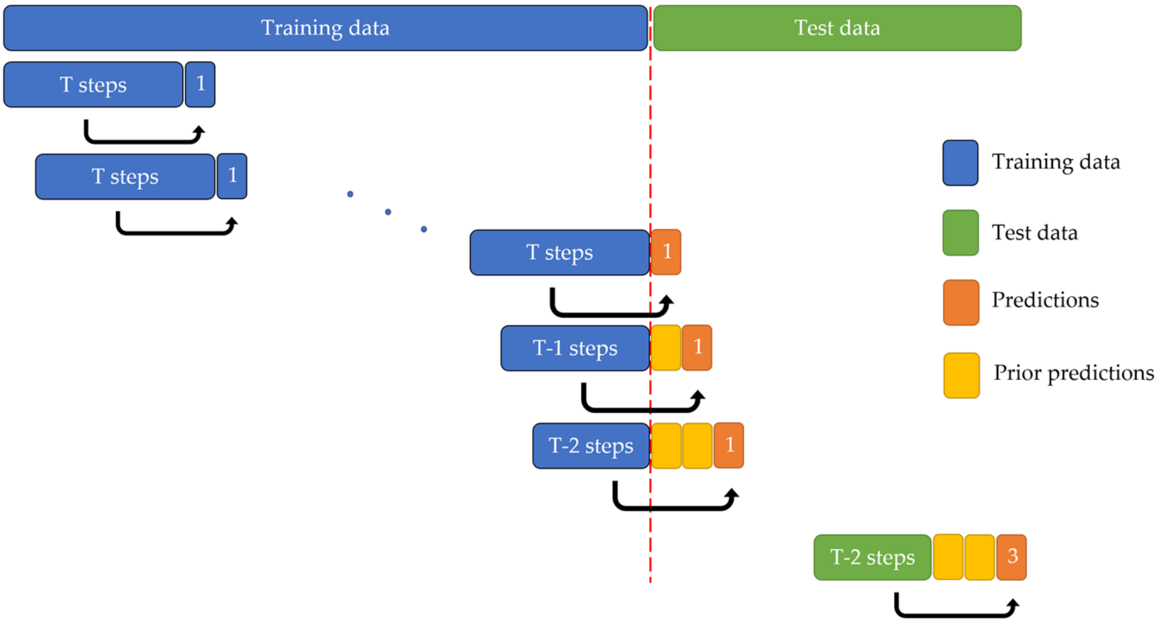


Figure 6.6. Schematic of recursive strategy for multi-step ahead forecasting (3-step ahead in this case)

6.3.2 Method 2: Direct strategy

In the direct strategy, for each forecast horizon, an independent model is required, and therefore, independently trained models are needed for the number of required forecasting horizons. The prediction of each step in this strategy can be obtained from the equation (48):

$$\hat{y}_{N+H} = \hat{f}_H(y_N, \dots, y_{N-T+1}) \tag{48}$$

where $\hat{f}_H$ denotes the prediction model for forecast horizon H. Figure 6.7 shows a schematic of the direct strategy for a 3-step ahead prediction. As can be seen in this figure, the prediction model is trained based on the dependencies between the input data length T and the measurements of the next third steps. The schematic shown in Figure 6.7 illustrates the trained predicting model, which can be used only for three-step ahead forecasting.

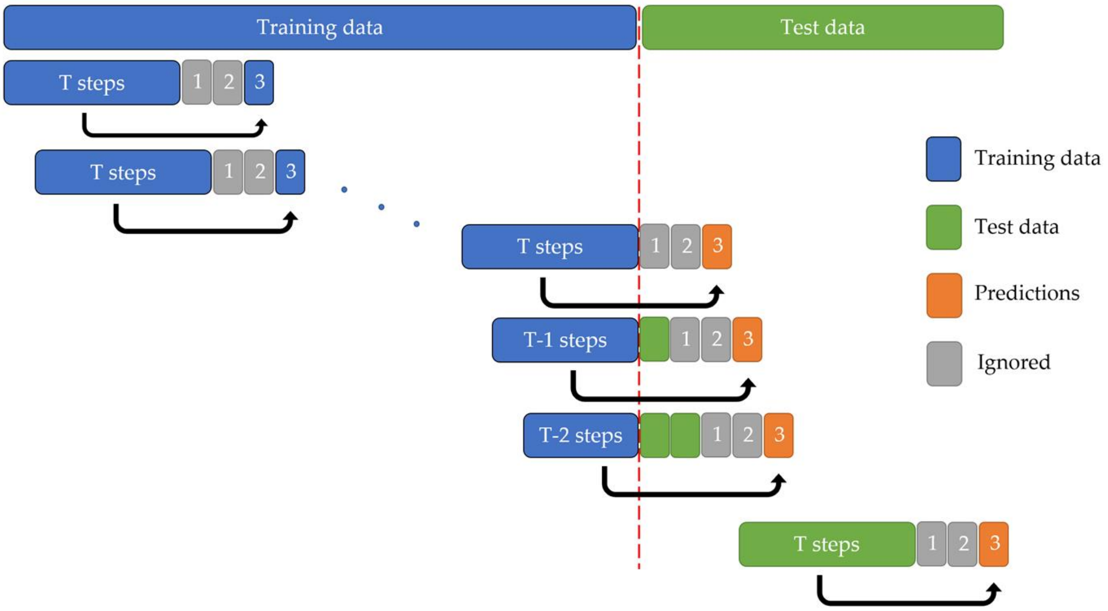


Figure 6.7. Schematic of direct strategy for multi-step ahead forecasting (3-step ahead in this case)

The main advantage of using the direct strategy is the non-accumulation of errors as a result of not using previous predictions to predict the next steps. However, training a model to predict the *H* step independently and without considering other steps, eliminates the possibility of considering simple dependencies such as linear correlations between input, target, and intermediate variables which can affect the prediction accuracy. In addition, a lot of time is spent on creating and training independent models for each forecast horizon, which leads to a decrease in the efficiency of the strategy.

6.3.3 Method 3: Multi-input multi-output (MIMO) strategy

In the MIMO strategy, as previously mentioned, the forecast value in each single step, is a vector instead of a scalar quantity. The vector of future values is returned in one step by a muti-output model $\hat{F}$ where:

$$[\hat{y}_{t+H}, \dots, y_{t+1}] = \hat{F}(y_N, \dots, y_{N-T+1}) \tag{49}$$

In contrast to the direct strategy, the MIMO strategy uses only one model for all forecast horizons and considers all the intermediate steps between the input data and the *H* step ahead. In this case, the stochastic dependencies between future values will be preserved. Figure 6.8 shows an example of using the MIMO strategy for a forecast with three output steps. In this strategy, after performing the prediction, step *H* will be selected as the desired prediction step.

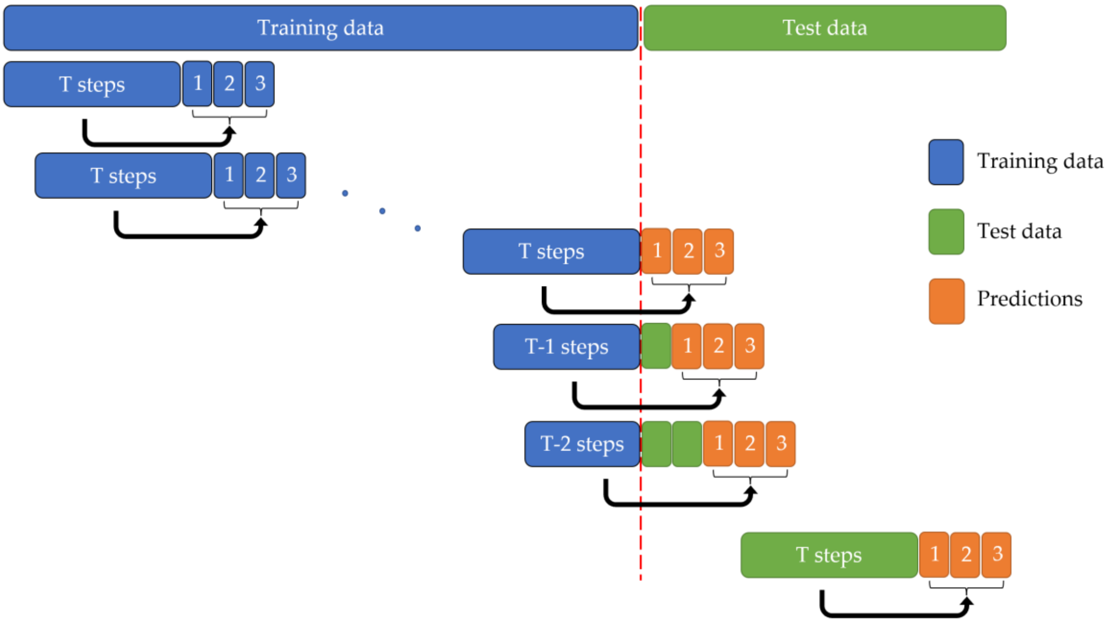


Figure 6.8. Schematic of MIMO strategy for multi-step ahead forecasting (3-step ahead in this case)

6.4 Results of comparison of different strategies for multi-step ahead prediction

6.4.1 Hyperparameter optimisation of different forecasting approaches

The search space of the hyperparameters of the LSTM model for all muti-step ahead predictions in all strategies is defined according to Table 6.3. Using the provided ranges of values for different hyperparameters, leads to approximately 28 million combinations of hyperparameters for the LSTM models.

Table 6.3. search space for hyperparameters of the LSTM models

Hyperparameter	Search Space
Units (neurons) of first LSTM Layer	10, …, 50
Number of dense layers	1, 2, 3, 4
Units (neurons) in dense layer	10, 11, …, 300
Activation function	Sigmoid, tanh, ReLU
Loss function	MAE, MSE
Optimiser	ADAM, Adadelta
Epochs	50, 51, …, 100

For simulation of all multi-step ahead prediction strategies, the Python programming language with a variety of packages and libraries including Numpy, Pandas, Matplotlib, and Scikit-learn are employed on a PC with Intel Core™ i7–11850H 2.5GHz CPU and 16GB RAM. Search and pruning process is performed by Optuna optimisation algorithm to find the best hyperparameters of LSTM models used in different strategies and different wind turbine datasets. The best combination of hyperparameters in this research are those that give the minimum value of RMSE. Tables 6.4 to 6.7 show the values of the hyperparameters for 2-step to 6-step ahead forecasting for both wind turbines.

Table 6.4. Best hyperparameter combinations of LSTM models in different 2-step ahead prediction strategies

Wind turbine	**Prediction Strategy**	LSTM units	dense layers	Dense layer units	Activation function	Loss function	optimiser	Epochs
WT1 (Scotland)	**MIMO**	26	4	211	ReLU	MAE	ADAM	86
	Direct	30	3	295	ReLU	MAE	ADAM	100
	Recursive	27	3	49	ReLU	MSE	ADAM	61
WT2 (Turkey)	**MIMO**	42	3	194	ReLU	MAE	ADAM	100
	Direct	30	3	125	ReLU	MAE	ADAM	61
	Recursive	18	4	283	ReLU	MSE	ADAM	87

Table 6.5. Best hyperparameter combinations of LSTM models in different 3-step ahead prediction strategies

Wind turbine	**Prediction Strategy**	LSTM units	dense layers	Dense layer units	Activation function	Loss function	optimiser	Epochs
WT1 (Scotland)	**MIMO**	17	2	179	ReLU	MAE	ADAM	79
	Direct	21	4	105	ReLU	MAE	ADAM	62
	Recursive	14	3	61	ReLU	MSE	ADAM	85
WT2 (Turkey)	**MIMO**	29	1	224	ReLU	MAE	ADAM	51
	Direct	43	4	149	ReLU	MAE	ADAM	99
	Recursive	29	3	79	ReLU	MSE	ADAM	95

Table 6.6. Best hyperparameter combinations of LSTM models in different 4-step ahead prediction strategies

Wind turbine	**Prediction Strategy**	LSTM units	dense layers	Dense layer units	Activation function	Loss function	optimiser	Epochs
WT1 (Scotland)	**MIMO**	27	2	136	ReLU	MAE	ADAM	77
	Direct	30	3	239	ReLU	MAE	ADAM	76
	Recursive	11	2	270	ReLU	MSE	ADAM	80
WT2 (Turkey)	**MIMO**	33	2	252	ReLU	MAE	ADAM	77
	Direct	28	3	135	ReLU	MAE	ADAM	68
	Recursive	46	1	73	ReLU	MSE	ADAM	100

Table 6.7. Best hyperparameter combinations of LSTM models in different 5-step ahead prediction strategies

Wind turbine	**Prediction Strategy**	LSTM units	dense layers	Dense layer units	Activation function	Loss function	optimiser	Epochs
WT1 (Scotland)	**MIMO**	15	3	116	ReLU	MAE	ADAM	50
	Direct	10	4	206	ReLU	MAE	ADAM	82
	Recursive	28	2	81	ReLU	MAE	ADAM	69
WT2 (Turkey)	**MIMO**	12	3	214	ReLU	MAE	ADAM	80
	Direct	17	2	290	ReLU	MAE	ADAM	92
	Recursive	34	4	282	ReLU	MSE	ADAM	82

Table 6.8. Best hyperparameter combinations of LSTM models in different 6-step ahead prediction strategies

Wind turbine	**Prediction Strategy**	LSTM units	dense layers	Dense layer units	Activation function	Loss function	optimiser	Epochs
WT1 (Scotland)	**MIMO**	20	2	178	ReLU	MAE	ADAM	57
	Direct	26	1	69	ReLU	MAE	ADAM	68
	Recursive	18	3	167	ReLU	MAE	ADAM	98
WT2 (Turkey)	**MIMO**	41	4	32	ReLU	MAE	ADAM	96
	Direct	14	4	34	ReLU	MAE	ADAM	99
	Recursive	40	2	289	ReLU	MAE	ADAM	99

6.4.2 Comparison of different multi-step ahead forecasting strategies

The prediction results of the multi-step ahead forecasting strategies are assessed and compared in order to identify the strategy which results into the highest accuracy for wind power forecasting. Five forecast horizons, from two-step to six-step ahead are selected. Since the resolution of both time series equals 10 minutes, the two-step to six-step ahead forecasts imply wind power prediction over the next 20 to 60 minutes. This range of time periods provide the possibility of fixing small defects, planning electricity distribution, and detecting wind turbine faults.

To evaluate the prediction performance of the employed hybrid wind power forecasting model based on IF, LSTM, and Optuna optimisation algorithm, for the datasets used in this research, a one-step ahead prediction is performed first. The results of these predictions for the two wind turbines are provided in Figures 6.9 and 6.10.

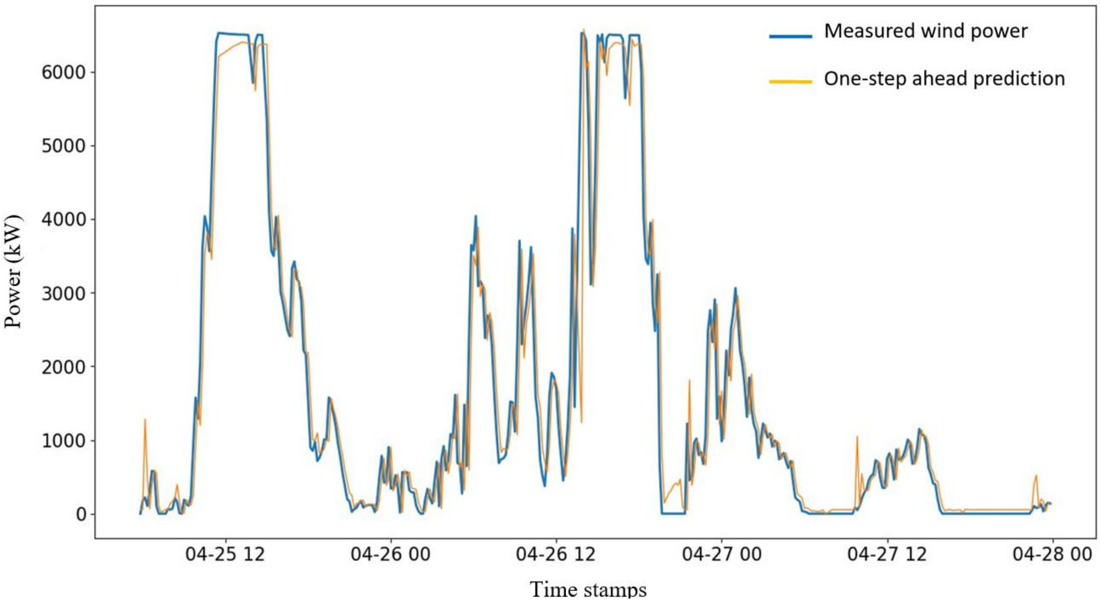


Figure 6.9. One-step ahead wind power predictions of Wind turbine1 (Scotland)

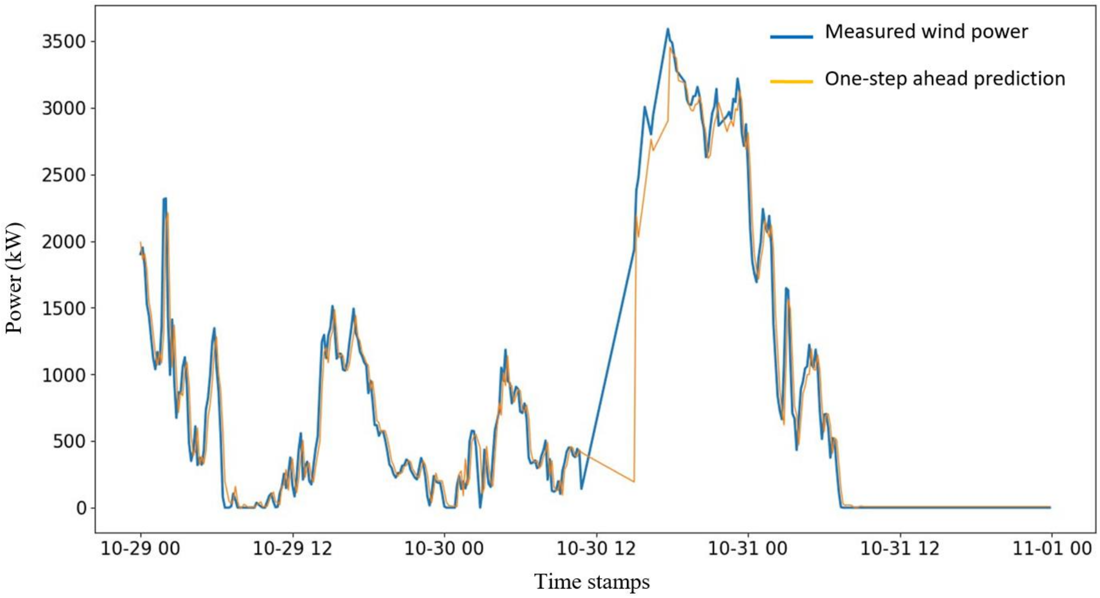


Figure 6.10. One-step ahead wind power predictions of Wind turbine 2 (Turkey)

The evaluation criteria of these forecasts including MSE, RMSE, MAE, and R-square are shown in Table 6.9. Comparing the evaluation parameters of the two turbines indicates the better performance of the employed hybrid model for the SCADA data of the second wind turbine, located in Turkey. Although it is not appropriate to compare the prediction performance of a model on two different datasets, two reasons can be given for the superior performance of the employed hybrid model for the SCADA data of the second wind turbine in Turkey. Firstly, as it was clear in Figures 6.4 and 6.5 in Section 6.2, the first time series dataset (wind turbine 1) contains more outliers than the second dataset (wind turbine 2). Even after the outlier treatment step using the IF outlier detection method, some of the outliers such as the zero power values at high wind speeds remain in the data and decrease the prediction accuracy. Secondly, the generated power range of wind turbine 1 is higher than that of wind turbine 2, as shown in Figure 6.3 and Table 6.2, and therefore, the minimal forecasting errors can have a negative impact on the values of obtained evaluation criteria.

Table 6.9. Prediction performance of LSTM model for **one**-step ahead prediction

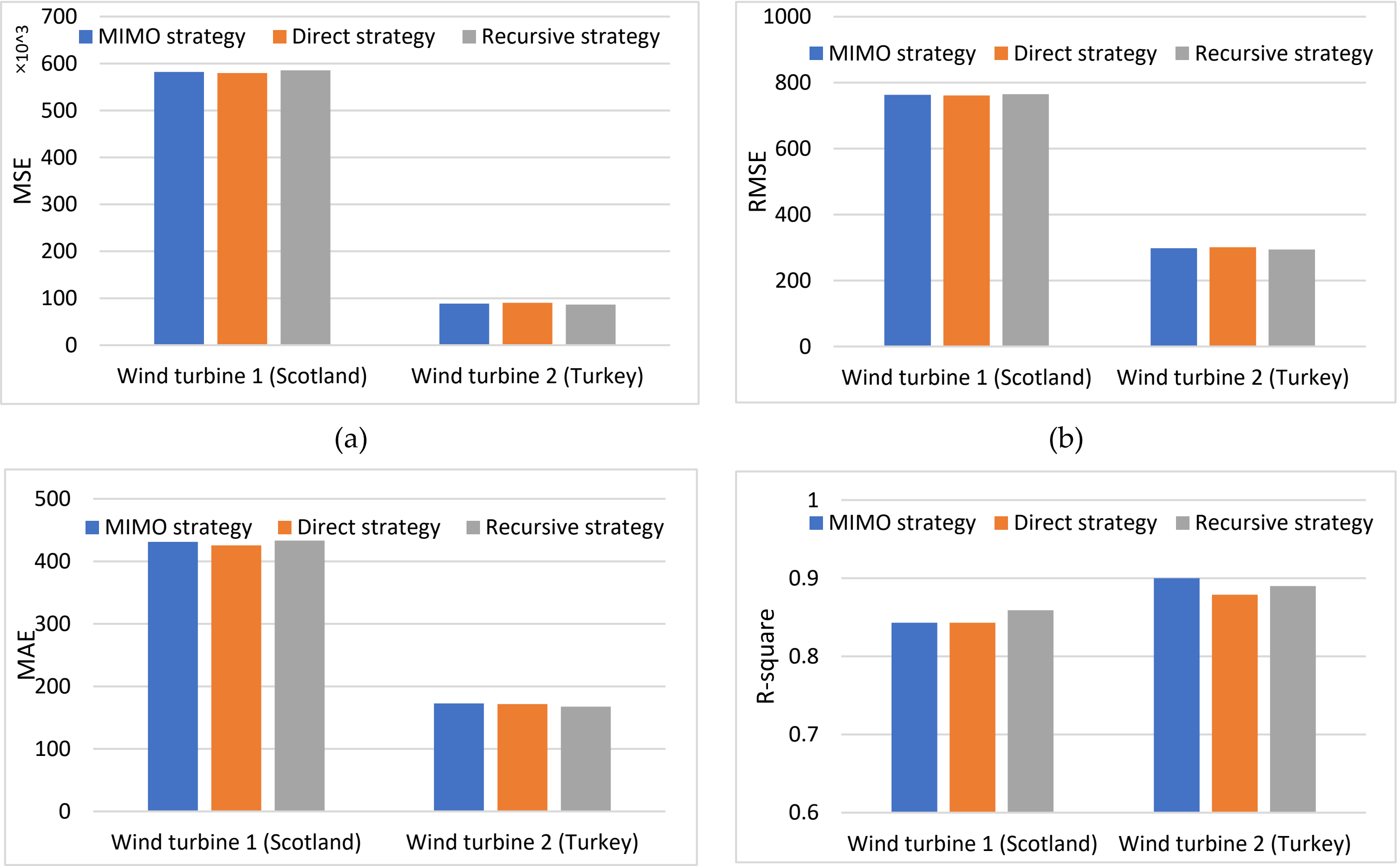
Evaluation criteria	**Wind turbine 1 (Scotland)**	**Wind turbine 2 (Turkey)**
MSE	309,130.89	36,922.7
RMSE	555.99	192.15
MAE	308.48	112.1
R-square	0.916	0.955

After the initial evaluation of the developed hybrid model in this research for one-step ahead wind power forecasting, different strategies of multi-step ahead forecasting were examined and compared for five forecast horizons: 2, 3, 4, 5 and 6-step ahead. First, for 2-step ahead forecasting, three LSTM models, one for each strategy are built based on the hyperparameter combinations found by the Optuna optimisation algorithm (see Table 32). The values of MSE, RMSE, MAE and R-square for these forecasts are shown in Table 6.10.

Table 6.10. Prediction performance of different strategies for 2-step ahead prediction

Evaluation criteria	**Wind turbine 1 (Scotland)**			**Wind turbine 2 (Turkey)**		
	MIMO strategy	**Direct strategy**	**Recursive strategy**	**MIMO strategy**	**Direct strategy**	**Recursive strategy**
MSE	582268.70	579772.77	585570.52	88726.07	90468.77	86415.44
RMSE	763.06	761.43	765.22	297.87	300.78	293.965
MAE	431.04	425.61	433.26	172.84	171.64	167.53
R-square	0.843	0.843	0.859	0.900	0.879	0.890

The prediction performance of different strategies for 2-step ahead wind power forecasting is presented in Figure 6.11. As evident from this figure and Table 6.10, in 2-step ahead forecasting, both wind turbines show almost identical predictions using the different strategies. This means that the application of different strategies to find dependencies between the previous steps and the two forward steps has no effect on improving the prediction accuracy.



(a) (b) (c) (d)

Figure 6.11. Comparison of prediction performance of different strategies for of 2-step ahead prediction, (a) MSE, (b) RMSE, (c) MAE, (d) R-square

A similar process was conducted for other forecasting horizons. Three LSTM models, one for each strategy are built based on the best hyperparameter combinations shown in Tables 6.5 to 6.8 for three-step to six-step ahead wind power forecasting.

The prediction results of different strategies for the third step of wind power forecasting are shown in Figure 6.12 and 6.13 for wind turbine 1 and 2, respectively. In addition, the values of MSE, RMSE, MAE and R-square for the predictions of this time horizon are shown in Table 6.11.

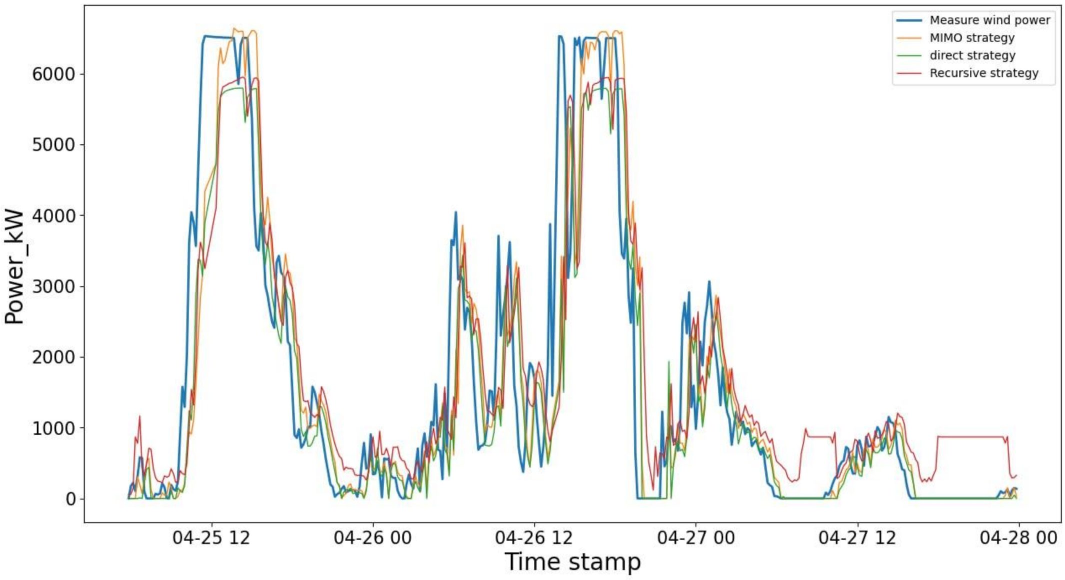


Figure 6.12. *3*-step ahead wind power predictions of WT1 (Scotland) in three different strategies

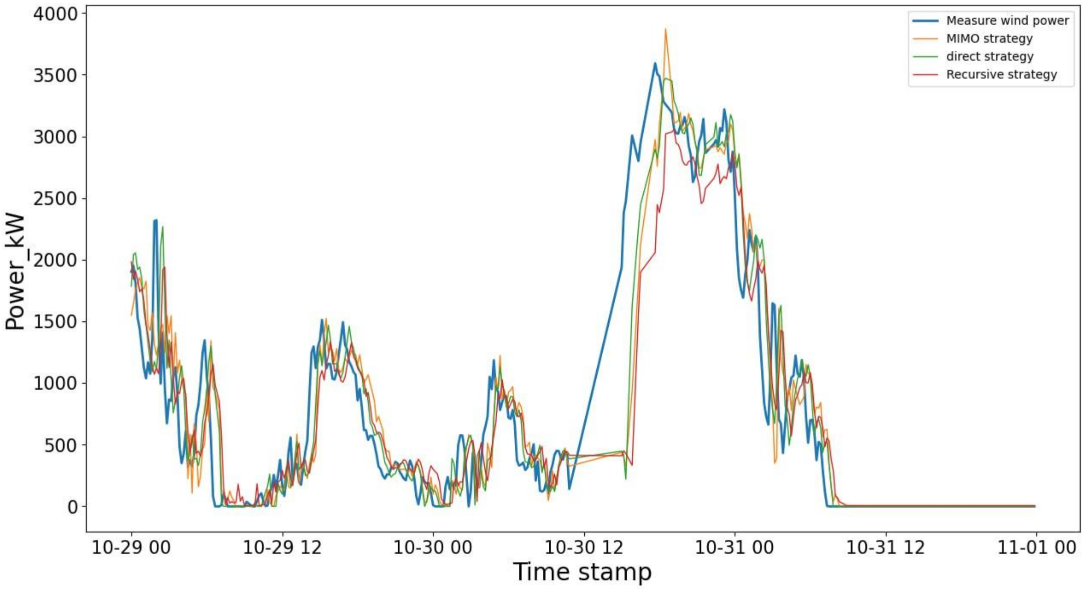
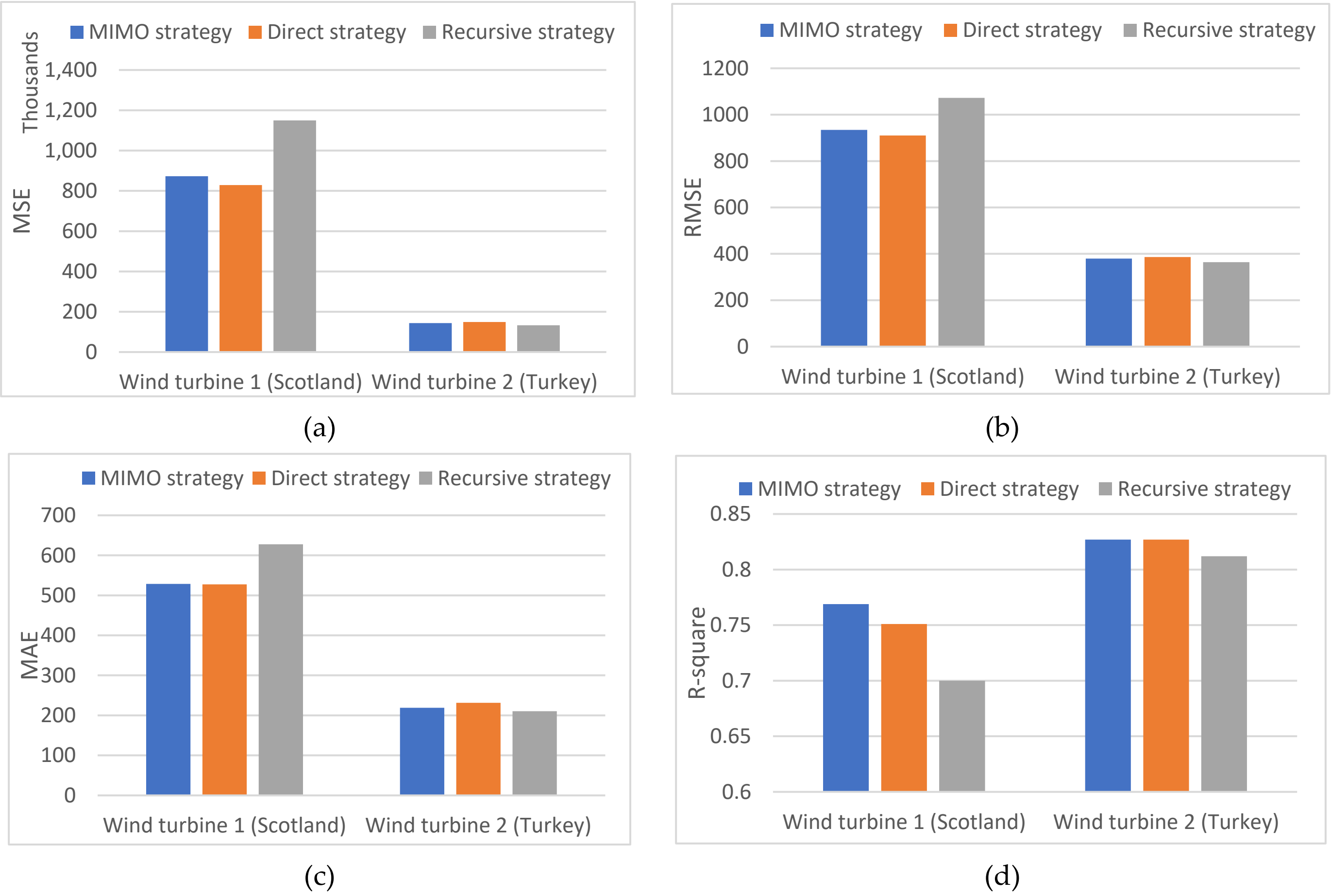


Figure 6.13. *3*-step ahead wind power predictions of WT2 (Turkey) in three different strategies

Table 6.11. Prediction performance of different strategies for 3-step ahead prediction

Evaluation criteria	**Wind turbine 1 (Scotland)**			**Wind turbine 2 (Turkey)**		
	MIMO strategy	**Direct strategy**	**Recursive strategy**	**MIMO strategy**	**Direct strategy**	**Recursive strategy**
MSE	873,059.66	828,668.33	1,149,838.2	144,129.9	149,329.8	132,684.3
RMSE	934.38	910.31	1,072.31	379.64	386.43	364.26
MAE	528.69	527.32	627.67	218.79	231.59	210.59
R-square	0.769	0.751	0.700	0.827	0.827	0.812

Figure 6.14 compares the prediction performance of different strategies in 3-step ahead wind power forecasting. As evident from this figure and Table 6.11, in wind turbine 1, the direct strategy results in the highest prediction performance compared to other strategies, while the recursive strategy has the weakest performance and higher prediction errors. On the other hand, in wind turbine 2, the recursive strategy proves to be the best prediction strategy, while the direct strategy has the lowest performance and shows higher prediction errors.



(a)

(b)

(c)

(d)

Figure 6.14. Comparison of prediction performance of different strategies for of 3-step ahead prediction, (a) MSE, (b) RMSE, (c) MAE, (d) R-square

Tables 6.12 and 6.13 show the results of the tests to determine forecasting performance for different strategies in forecast horizons 4 and 5. Both turbines showed the highest prediction error when using the recursive strategy for these two horizons.

Table 6.12. Prediction performance of different strategies for 4-step ahead prediction

Evaluation criteria	**Wind turbine 1 (Scotland)**			**Wind turbine 2 (Turkey)**		
	MIMO strategy	**Direct strategy**	**Recursive strategy**	**MIMO strategy**	**Direct strategy**	**Recursive strategy**
MSE	1,115,557.8	1,099,817.9	1,444,275.9	181,134	182,143.2	201,020.6
RMSE	1,056.19	1,048.72	1,201.78	425.59	426.78	448.35
MAE	635.51	624.81	739.17	252.88	252.69	262.28
R-square	0.657	0.676	0.576	0.777	0.767	0.626

Table 6.13. Prediction performance of different strategies for 5-step ahead prediction

Evaluation criteria	**Wind turbine 1 (Scotland)**			**Wind turbine 2 (Turkey)**		
	MIMO strategy	**Direct strategy**	**Recursive strategy**	**MIMO strategy**	**Direct strategy**	**Recursive strategy**
MSE	1,374,991.1	1,342,568.3	1473549.06	216,021.6	232,704.5	278,660.6
RMSE	1,172.6	1,158.69	1213.89	464.78	482.39	527.88
MAE	710.44	714.48	713.4	275.73	293.27	312.06
R-square	0.655	0.611	0.534	0.718	0.722	0.558

Similar experiments were conducted to investigate the performance of MIMO, direct and Recursive strategies for 6-step ahead forecasting. The same training and testing datasets are used, and LSTM models were built based on the hyperparameter combinations proposed by Optuna optimisation algorithm in Table 6.8. Then the 6-step ahead predictions are made in three different strategies. The results of these predictions are shown in Figure 6.15 and 6.16 for wind turbine 1 and 2, respectively. In addition, the values of MSE, RMSE, MAE and R-square of these predictions are shown in Table 6.14.

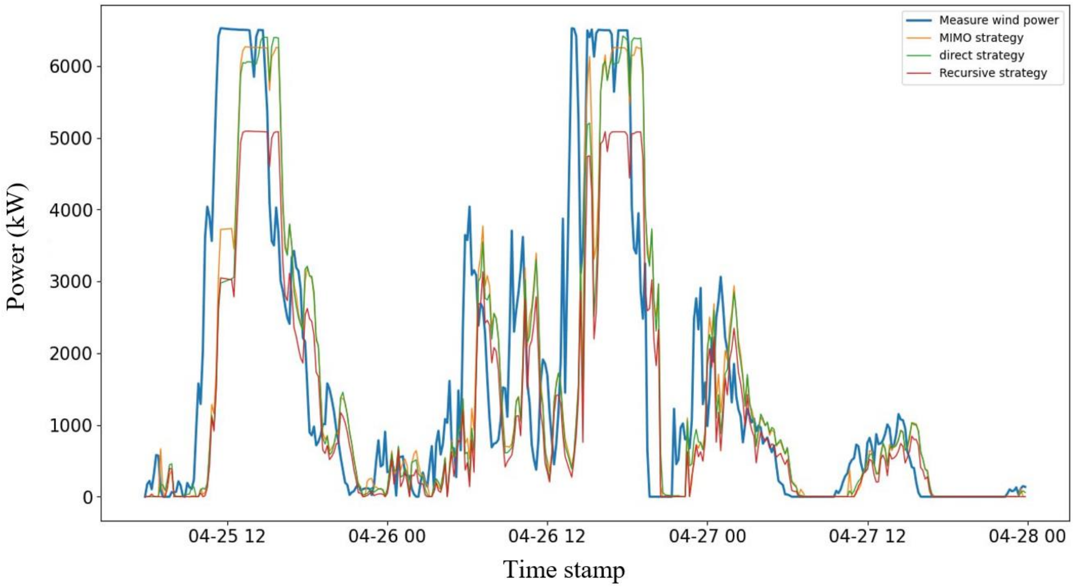


Figure 6.15. ***6***-step ahead wind power predictions of WT2 (Turkey) in three different strategies

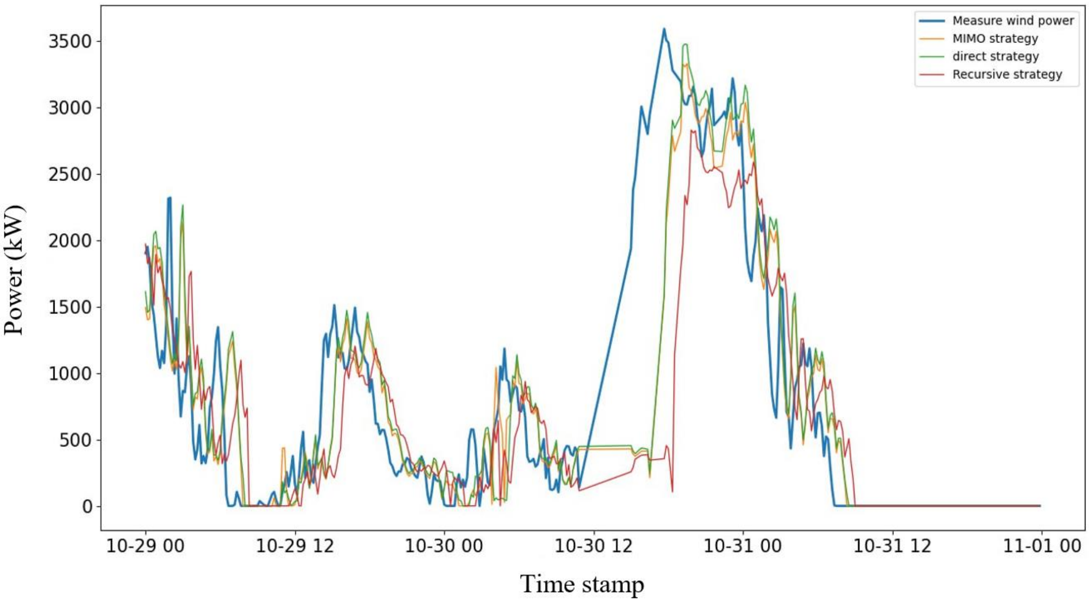
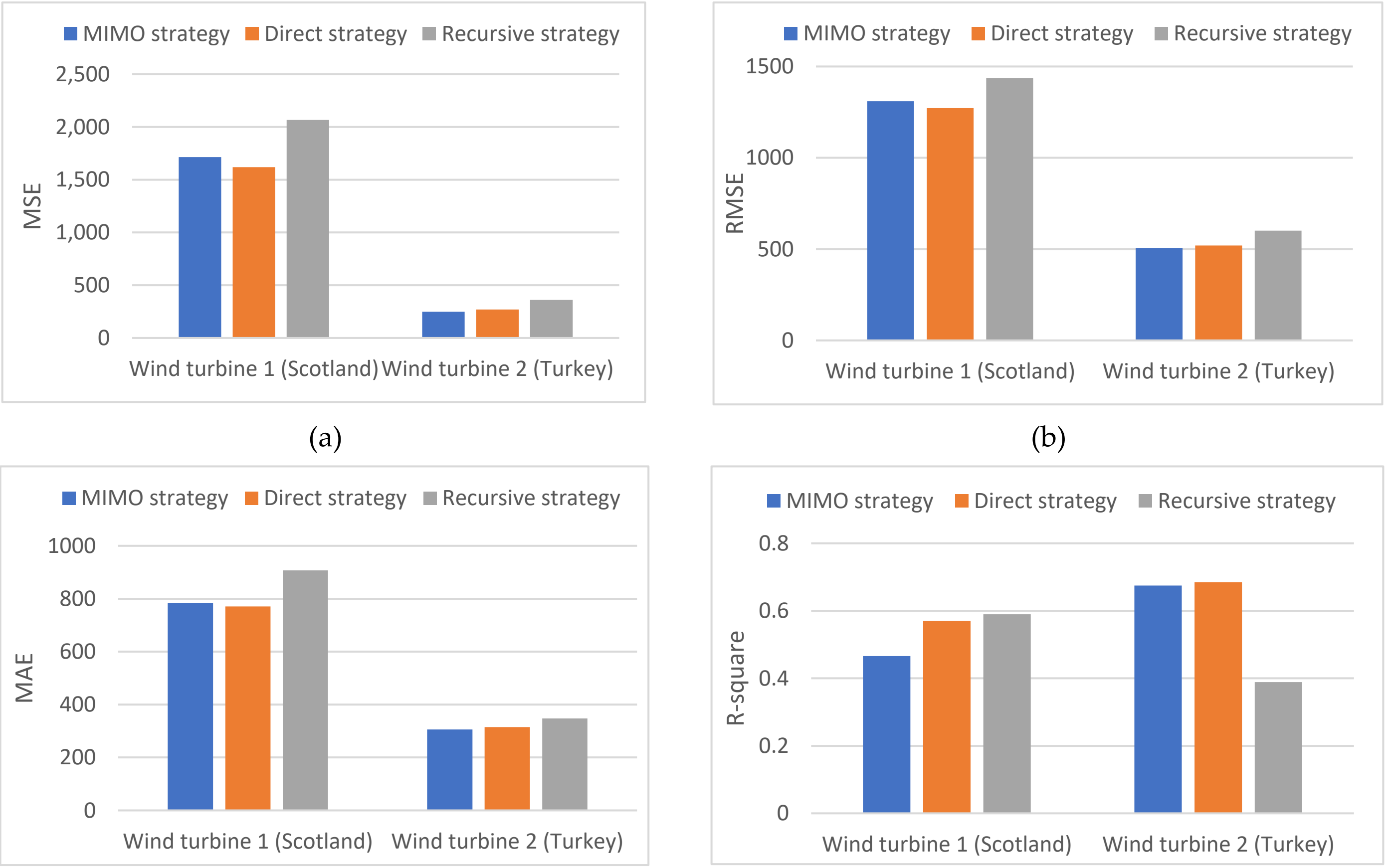


Figure 6.16. ***6***-step ahead wind power predictions of WT2 (Turkey) in three different strategies

Table 6.14. Prediction performance of different strategies for 6-step ahead prediction

Evaluation criteria	**Wind turbine 1 (Scotland)**			**Wind turbine 2 (Turkey)**		
	MIMO strategy	**Direct strategy**	**Recursive strategy**	**MIMO strategy**	**Direct strategy**	**Recursive strategy**
MSE	1,714,249	1,618,416.4	2,066,069	256,446.9	270,165.5	361,449.4
RMSE	1,309.29	1,272.17	1437.38	506.41	519.77	601.21
MAE	784.69	770.69	906.93	305.59	315.14	347.30
R-square	0.466	0.57	0.59	0.675	0.685	0.389

Figure 6.17 shows the 6-step ahead forecasting performance of the utilised models for the three different strategies. According to the figure, the recursive strategy gives the weakest results for both wind turbines. Similarly, to the previous forecasting horizon, this one also shows accumulating errors as a result of the application of predictions made previously. As for the most appropriate strategy, the experiments show that the direct strategy for wind turbine 1 and the MIMO strategy for wind turbine 2 perform the best.



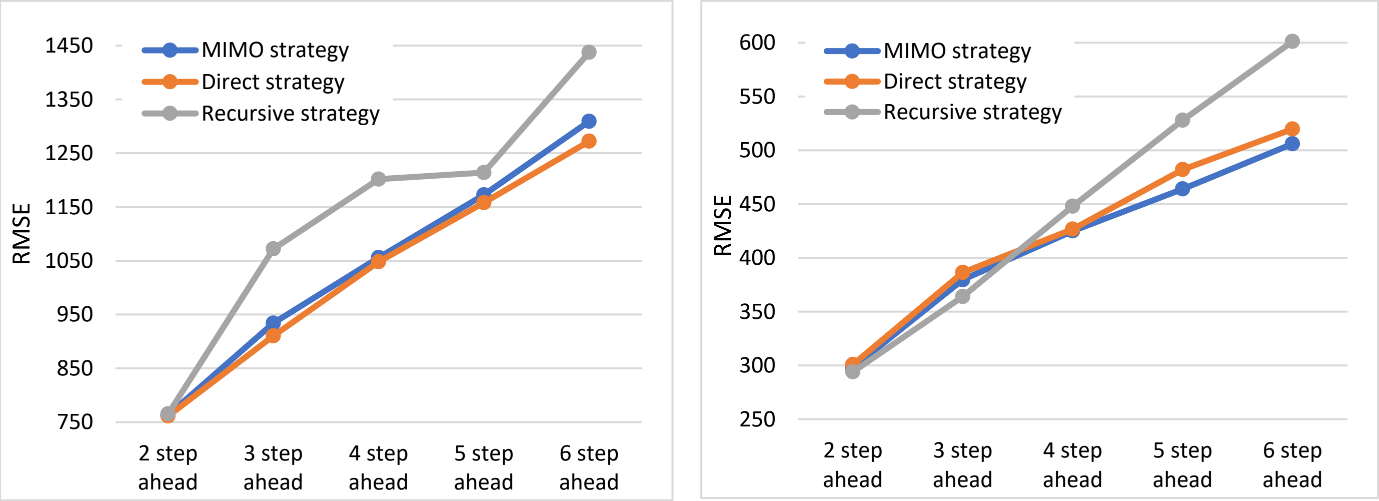
(a) (b)

(c) (d)

Figure 6.17. Comparison of prediction performance of different strategies for of 6-step ahead wind power forecasting, (a) MSE, (b) RMSE, (c) MAE, and (d) R-square

A more detailed analysis of the impact of different strategies in different forecasting horizons is shown in Figures 6.18(a) and 6.18(b) for turbines 1 and 2, respectively. According to the findings, even though different forecasting strategies do not have a significant impact on forecast accuracy in two-step ahead wind power forecasting, differences in forecasting performance can be observed as the forecasting horizon increases.

In the case of wind turbine 1, for all forecast horizons higher than two steps, wind power prediction using the recursive strategy results in the highest error. This is due to the negative effects of errors in wind power prediction for the previous steps. The negative effect becomes more pronounced when there are more outliers in the dataset. Apart from the recursive strategy, the direct strategy performs better than the MIMO strategy in wind turbine 1.



(a) (b)

Figure 6.18. multi-step ahead prediction performance of different strategies for wind turbine 1 in Scotland (a), and wind turbine 2 in Turkey (b)

Based on the comparison of different multi-step-ahead forecasting strategies for the second turbine (where the power time series contains fewer anomalies than the first turbine dataset), the recursive strategy performs the best in two and three-step ahead wind power forecasts. It means that using the predictions made in the previous steps as input data can improve the prediction accuracy for these two forecast horizons when the data is clean. However, when the forecasting horizon is extended to four, five, and six steps, the forecasting errors increase and the performance of multi-step ahead forecasting is reduced.

Compared to the first turbine, the second turbine performs differently under direct and MIMO strategies in terms of multi-step ahead predictions. In contrast to the first turbine, the application of the MIMO strategy for multi-step ahead forecasting on all forecast horizons has a superior performance compared to the direct option for the second turbine. Consequently, if the data is clean, multi-step ahead forecasting using the MIMO strategy, in which several inputs relate to several outputs, will improve the accuracy.

Chapter 7 - Conclusion, Future Trends and Open Issues

In this section, the conclusion of the thesis is discussed, as well as future trends for the thesis, in which a conclusion is drawn for all the cases that were examined.

7.1 Conclusion

In light of the importance of accurate wind power forecasting for wind energy management, increasing the penetration of wind power into the power grid, and improving maintenance efficiency, artificial intelligence-based solutions, are needed to improve the accuracy and efficiency of forecasting.

There are various parameters that must be considered to optimise forecasting methods. It begins with checking raw data for patterns and dependencies, identifying and removing outlier data, and then providing an appropriate ML algorithm, adjusting it, and sometimes combining multiple methods to increase accuracy, efficiency, and robustness.

Data received from different wind turbines/farms usually display different patterns and dependencies between variables. It is therefore necessary to examine them separately. For several reasons, this data contains outliers, which may contribute to errors in power predictions if not corrected. On the other hand, there are many ML methods that are being developed and it is challenging to identify a method that is capable of detecting and learning linear, non-linear, short-term and long-term dependencies in data. Once the model is developed, it should be able to predict power within a short timeframe. Accordingly, this thesis covers some applications of ML algorithms that are used to create a time series-base wind power forecasting model.

Based on the study of past research, it can be concluded that physical methods, which offer good performance at medium to long-term prediction, are complex and need considerable computing resources. On the other hand, statistical methods, which perform better in short to medium term periods, are easy to be modelled and inexpensive. Combination of these two major methods with their merits led to the promising hybrid methods. It was also revealed that wind speed, temperature, wind direction, relative humidity and air pressure were the most often used input features in reviewed studies. Also, the one-year period and the sampling rate of 10 min were the most common features used for input data.

Using SCADA data from an offshore wind turbine, the data was pre-processed to remove outliers and negative values to help determine the best pre-processing method. The results confirmed that replacing negative values with average power values has the most positive effect on prediction accuracy. Furthermore, comparisons between several data items showed a significant effect of outlier treatment methods on performance prediction. The results showed that the removal of outliers with the isolation forest method improves the prediction accuracy compared to the ellipse coverage and OCSVM methods.

In the next step of the research, the focus has been on fine-tuning the hyperparameters of ML models to increase the accuracy and efficiency of forecasting. As opposed to grid search or random search which are time-consuming and unreliable, three hyperparameter optimisation techniques, Scikit-opt, Hyperopt, and Optuna, are used to tune CNN and LSTM prediction models, two widely used deep learning models. The results showed that the Optuna optimisation technique using TPE search algorithm and EI acquisition function, has the highest efficiency for both CNN and LSTM models. Regarding the improvement of the prediction accuracy, it has been demonstrated that while for the CNN model, all the optimisation methods perform almost the same, the LSTM model optimised by the Hyperopt algorithm based on the annealing search method achieves the highest accuracy. In addition, in this research, the sensitivity of different structures of CNN and LSTM models to seed changes was investigated and the most resistant structure against randomness was selected for both models. The proposed models provides the highest level of accuracy, efficiency, and robustness for the utilised offshore wind power dataset. This level of performance for short term wind power forecast can be used for regulation actions, real-time grid operations, market clearing and wind turbine control systems.

A novel wind power forecasting method is proposed based on the combination of WPD, optimised LSTM and CNN models. In the developed WPD-LSTM-CNN model, first, the obvious outliers that diminish the prediction accuracy, are removed and the resolution of data averaged over 10 minutes in order to mitigate the influence of turbulence. After an assessment of various mother wavelets and selection of the db5 mother wavelet, resulting in the best performance, WPD is employed to decompose the pre-processed wind power time series into several sub-series with different frequencies. The appropriate decomposition of signals into several sub-series increases the stationary of data and thus makes the prediction models more efficient. Three tuned independent CNNs are employed for the prediction of the high-frequency sub-series, and one optimised LSTM model is adopted to complete the forecasting of the low-frequency sub-layer. For the optimisation of these deep learning models, the SMBO method as a formalisation of Bayesian optimisation, provided in the Optuna optimisation package, is used to reduce the dependence on computational resources.

For the prediction performance assessment of the proposed model, various forecasting models were employed, including the RF model, FFNN model, CNN model, LSTM model, WPD-FFNN model, WPD-CNN model, and WPD-LSTM model. Based on the prediction results for four different datasets it is observed that WPD, through extracting the hidden features of the signals, can effectively improve the prediction performance of the forecasting models. This improvement is more pronounced during time steps when the wind power encounters abrupt changes. Considering the four different datasets, using WPD improved the average accuracy of the FFNN, CNN and LSTM models, by 74.10%, 78.13% and 78.38%, respectively.

It is also observed that the optimised CNN and LSTM models have good performance in learning the short-term and long-term dependencies of the wind power time series. Using SMBO methods for hyper-parameter selection of these deep learning models, instead of commonly used methods such as grid and random search, increases the prediction accuracy and efficiency. Using the CNN model increases the forecasting accuracy by 6.56% and 1.57% on average compared to the RF and FFNN models, respectively. Likewise, the application of the LSTM model improves the prediction accuracy by an average of 8% and 3.05% compared to the RF and FFNN models, respectively.

Furthermore, the simultaneous application of the CNN and LSTM models to predict the approximation and detail components, instead of using only one of these models to predict both components, was shown to improve the prediction performance. The results of the simulations have shown that this approach leads to an improvement in accuracy of up to 11.93% and 14.11%, compared to the application of only the CNN and LSTM models, respectively.

Since forecasting wind power over several steps into the future is challenging due to decreasing dependency between past and future variables, several strategies were evaluated to identify the most useful strategy for wind power multi-step ahead prediction. Three strategies including the recursive, direct, and multi-input multi-output (MIMO) strategies are investigated on two wind turbine datasets of two turbines in Turkey and Scotland. For all strategies and forecast horizons considered, the followings are applied: (i) a hybrid prediction method based on application of the Isolation Forest for outlier detection and removal, (ii) long short-term memory (LSTM) for prediction, and (iii) a new hyperparameter optimisation algorithm for tuning of the LSTM model. Based on the results of the experiments, it was concluded that in two-step ahead (20 min) wind power forecasting, all different strategies come up with almost identical results. The multi-step ahead wind power prediction with the MIMO strategy performs best when the dataset contains no outlier data, and vice versa, when it contains outlier data, the direct strategy performs best in forecast horizons of more than two steps ahead. In the case of datasets containing outliers, for forecast horizons above two steps, wind power forecasting using the recursive strategy results in the highest error. The errors in previous wind energy forecasts, used to predict next steps accumulate, leading to a decrease in accuracy.

Given the comprehensive analysis and findings presented, my approach to deploying an optimal wind power forecasting solution would involve leveraging a hybrid method that integrates both physical and statistical models. Initially, I would pre-process the data to remove outliers using the IF method, ensuring the data's accuracy and consistency. Next, I would employ a combination of WPD to enhance data stationarity and deep learning models, specifically optimized CNN and LSTM networks. For hyperparameter optimization of the CNN and LSTM models, my suggestion would be to apply the Optuna algorithm, prioritizing the TPE search and EI acquisition function to fine-tune the models efficiently. For multi-step ahead predictions, I would adopt the MIMO strategy, especially when dealing with clean datasets, to mitigate error accumulation over extended forecast horizons.

7.2 Future trends in wind power forecasting

Wind power forecasting has great potential for advancement and improvement in the future. Continuous innovation of technology and data analysis techniques is expected to make wind power forecasting more accurate, reliable, and efficient.

Advances in data analysis techniques can improve the performance of forecasting models by combining large volumes of historical weather data, real-time meteorological observations, wind turbine sensor data, and grid data. In addition, advanced analytics can continuously update and refine forecasting models by assimilating real-time data through ingesting real-time data from weather satellites, radars and in-situ sensors. With the assimilation of real-time data, more accurate forecasting models are ensured, even with changes in weather conditions.

In addition, future research should comprehensively examine the data lifecycle in wind power forecasting models. It is crucial to determine optimal data periods and sampling rates, as older data may distort predictions by failing to accurately represent the current state of the wind farm. Further analysis is required to develop strategies for dynamically updating training datasets, ensuring that the most relevant and current data are utilized. This approach will enhance the accuracy and reliability of predictive models, ultimately improving wind power forecasting.

The Internet of Things (IoT) devices and sensors are becoming more prevalent in wind farms. These devices can collect real-time data on wind speed, direction, temperature, and other relevant parameters. By integrating this big data with forecasting models, operators can continuously update and refine predictions, improving the overall accuracy of wind power forecasts.

Further research is still needed to combine multiple forecasting methods to enhance wind force forecasting accuracy. Numerical weather forecasting models, statistical models, machine learning algorithms, and even human expertise can be combined. With the combination of these different approaches, wind power forecasts can benefit from different perspectives and provide more accurate predictions.

Future developments in ML algorithms will improve the performance of models in many ways. ML techniques, such as feature selection algorithms, can automatically identify the most relevant features from large sets of input data. This helps in selecting the key weather variables and other factors that have the most significant impact on wind power generation, leading to more accurate forecasts. Next-generation of ML algorithms, such as transformer models and graph neural networks (GNNs), will better capture the complex and nonlinear relationships between variables. More efficient learning and modelling of these intricate relationships will result in more accurate predictions.

ML algorithms excel at capturing nonlinear relationships between variables. In wind power forecasting, there are often complex interactions between weather parameters (e.g., wind speed, temperature, humidity) and power output. ML models, such as neural networks or support vector machines, can effectively learn and model these nonlinear relationships, resulting in more accurate predictions. It is also important understanding and leveraging the lifecycle of data is paramount. Future work should focus on optimizing each stage of the data lifecycle, from data collection and pre-processing to model training and real-time application. By enhancing the quality and consistency of data throughout its lifecycle, we can improve the accuracy and reliability of AI-driven forecasts.

In general, wind power forecasting holds great potential for the future. More accurate and reliable wind power predictions will be available as a result of advances in data analytics, weather modeling, computing technologies, and integration with energy systems. As a result of these improvements, wind farm operators, energy grid managers, and market participants will be able to better plan, optimise, and integrate wind power into a broader energy landscape.

It is also required to point out that in this thesis, the methodology used for splitting data into training and testing sets involves using the same testing set to both select hyperparameters and assess the model's performance on new data. This approach conflicts with the standard practice in the machine-learning community, which typically involves splitting data into training, testing, and validation sets. Training data is used for parameter optimization, testing data for hyperparameter selection, and validation data to assess model performance. Additionally, a 90%-10% split is used for training-testing purposes throughout the work, which goes against common practices where cross-validation is typically used to analyse model performance. Future work should address these issues by implementing a more robust data splitting strategy that includes separate validation sets and by employing cross-validation techniques. This will provide a fair assessment of model performance and ensure that hyperparameter selection does not bias the evaluation, leading to more reliable and accurate forecasting models.

References:

[1] Arinze U, Eng M, Mahmut G and Eng M 2020 Challenges of Wind Power Integration to the Power Grid 0–8

[2] DNV 2022 ENERGY TRANSITION OUTLOOK UK 2022 A national forecast to 2050

[3] Hanifi S, Lotfian S, Zare-behtash H and Cammarano A 2022 Offshore Wind Power Forecasting—A New Hyperparameter Optimisation Algorithm for Deep Learning Models *Energies* **15** 6919

[4] Wang X, Guo P and Huang X 2011 Energy Procedia A Review of Wind Power Forecasting Models *Energy Procedia* **12** 770–8

[5] Rogus R and Castro R 2020 Comparative Analysis of Wind Energy Generation Forecasts in Poland and Portugal and Their Influence on the Electricity Exchange Prices

[6] Garrigle E V M 2013 The value of accuracy in wind energy forecasts

[7] Hanifi S, Liu X, Lin Z and Lotfian S 2020 A Critical Review of Wind Power Forecasting Methods-Past, Present and Future *Energies* **13** 1–24

[8] Singh S, Bhatti T S and Kothari D P 2007 Wind power estimation using artificial neural network *J. Energy Eng.* **133** 46–52

[9] Sharma R and Singh D 2018 A Review of Wind Power and Wind Speed Forecasting *Rahul Sharma J. Eng. Res. Appl. www.ijera.com* **8** 1–09

[10] Wu Y and Hong J 2007 A literature review of wind forecasting technology in the world 504–9

[11] Soman S S, Zareipour H, Malik O and Mandal P 2010 A review of wind power and wind speed forecasting methods with different time horizons *North Am. Power Symp. 2010, NAPS 2010* 1–8

[12] Jung J and Broadwater R P 2014 Current status and future advances for wind speed and power forecasting *Renew. Sustain. Energy Rev.* **31** 762–77

[13] Chang W 2014 A Literature Review of Wind Forecasting Methods *J. power energy Eng.* 161–8

[14] De Giorgi M G, Ficarella A and Tarantino M 2011 Assessment of the benefits of numerical weather predictions in wind power forecasting based on statistical methods *Energy* **36** 3968–78

[15] Focken U, Lange M and Waldl H-P H-P 2001 Previento - A Wind Power Prediction System with an Innovative Upscaling Algorithm *Proc. Eur. Wind Energy Conf.* 1–4

[16] Felice M De, Alessandri A and Ruti P M 2013 Electricity demand forecasting over Italy : Potential benefits using numerical weather prediction models *Electr. Power Syst. Res.* **104** 71–9

[17] Zhang J, Yan J, Infield D, Liu Y and Lien F sang 2019 Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model *Appl. Energy* **241** 229–44

[18] Foley A M, Leahy P G, Marvuglia A and McKeogh E J 2012 Current methods and advances in forecasting of wind power generation *Renew. Energy* **37** 1–8

[19] Jung S and Kwon S D 2013 Weighted error functions in artificial neural networks for improved wind energy potential estimation *Appl. Energy* **111** 778–90

[20] Zhao Y, Ye L, Li Z, Song X, Lang Y and Su J 2016 A novel bidirectional mechanism based on time

series model for wind power forecasting *Appl. Energy* **177** 793–803

[21] Durán M J, Cros D and Riquelme J 2007 Short-term wind power forecast based on ARX models *J. Energy Eng.* **133** 172–80

[22] Gallego C, Pinson P, Madsen H, Costa A and Cuerva A 2011 Influence of local wind speed and direction on wind power dynamics - Application to offshore very short-term forecasting *Appl. Energy* **88** 4087–96

[23] Firat U, Engin S N, Sarcalar M and Ertuzum A B 2010 Wind Speed Forecasting Based on Second Order Blind Identification and Autoregressive Model *2010 Ninth Int. Conf. Mach. Learn. Appl.* 686–91

[24] Wu Y R and Zhao H S 2010 Optimisation maintenance of wind turbines using Markov decision processes *2010 Int. Conf. Power Syst. Technol. Technol. Innov. Mak. Power Grid Smarter, POWERCON2010* 1–6

[25] Lin Z and Liu X 2020 Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network *Energy* **201** 117693

[26] Lee K, Booth D and Alam P 2005 A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms **29** 1–16

[27] Marugán A P, Márquez F P G, Perez J M P and Ruiz-Hernández D 2018 A survey of artificial neural network in wind energy systems *Appl. Energy* **228** 1822–36

[28] Bilal B, Ndongo M, Adjallah K H, Sava A, Kebe C M F, Ndiaye P A and Sambou V 2018 Wind turbine power output prediction model design based on artificial neural networks and climatic spatiotemporal data *Proc. IEEE Int. Conf. Ind. Technol.* **2018-Febru** 1085–92

[29] Wang J, Yang W, Du P and Niu T 2018 A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm *Energy Convers. Manag.* **163** 134–50

[30] G. Sideratos D H 2012 Probabilistic Wind Power Forecasting Using Radial Basis Function Neural Networks *IEEE Trans. Power Syst.* **27** 1788–96

[31] Hong Y Y and Rioflorido C L P P 2019 A hybrid deep learning-based neural network for 24-h ahead wind power forecasting *Appl. Energy* **250** 530–9

[32] Pelletier F, Masson C and Tahan A 2016 Wind turbine power curve modelling using artificial neural network *Renew. Energy* **89** 207–14

[33] Sideratos G and Hatziargyriou N D 2007 An advanced statistical method for wind power forecasting *IEEE Trans. Power Syst.* **22** 258–65

[34] Jyothi M N and Rao P V R 2016 Very-short term wind power forecasting through Adaptive Wavelet Neural Network *2016 - Bienn. Int. Conf. Power Energy Syst. Towar. Sustain. Energy, PESTSE 2016* 1–6

[35] Xu L and Mao J 2016 Short-term wind power forecasting based on Elman neural network with particle swarm optimisation *Proc. 28th Chinese Control Decis. Conf. CCDC 2016* 2678–81

[36] Catalão J P S, Pousinho H M I, Member S and Mendes V M F 2009 An Artificial Neural Network Approach for Short-Term Wind Power Forecasting in Portugal *2009 15th Int. Conf. Intell. Syst. Appl. to*

Power Syst. 1–5

[37] Chang W 2013 Application of Back Propagation Neural Network for Wind Power Generation Forecasting *Int. J. Digit. Content Technol. its Appl.* **7** 502–9

[38] Carolin Mabel M and Fernandez E 2008 Analysis of wind power generation and prediction using ANN: A case study *Renew. Energy* **33** 986–92

[39] Du P, Wang J, Yang W and Niu T 2019 A novel hybrid model for short-term wind power forecasting *Appl. Soft Comput. J.* **80** 93–106

[40] Shahid F, Zameer A and Muneeb M 2021 A novel genetic LSTM model for wind power forecast *Energy* **223** 120069

[41] Wang J, Yang W, Du P and Li Y 2018 Research and application of a hybrid forecasting framework based on multi-objective optimisation for electrical power system *Energy* **148** 59–78

[42] Kisvari A, Lin Z and Liu X 2021 Wind power forecasting – A data-driven method along with gated recurrent neural network *Renew. Energy* **163** 1895–909

[43] Lin Z, Liu X and Collu M 2020 Electrical Power and Energy Systems Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks *Electr. Power Energy Syst.* **118** 105835

[44] Marcos J, Alexandre L, Saulo K G, Jairo R F and Mattos J G Z De 2017 A Meteorological – Statistic Model for Short-Term Wind Power Forecasting *J. Control. Autom. Electr. Syst.* **28** 679–91

[45] Shetty R P, Sathyabhama A, Srinivasa P P and Adarsh Rai A 2016 Optimised radial basis function neural network model for wind power prediction *Proc. - 2016 2nd Int. Conf. Cogn. Comput. Inf. Process. CCIP 2016* 1–6

[46] Liu J, Wang X and Lu Y 2017 A novel hybrid methodology for short-term wind power forecasting based on adaptive neuro-fuzzy inference system *Renew. Energy* **103** 620–9

[47] Zhao P, Wang J, Xia J, Dai Y, Sheng Y and Yue J 2012 Performance evaluation and accuracy enhancement of a day-ahead wind power forecasting system in China *Renew. Energy* **43** 234–41

[48] Kou P, Gao F and Guan X 2013 Sparse online warped Gaussian process for wind power probabilistic forecasting *Appl. Energy* **108** 410–28

[49] Giorgi M G De, Congedo P M, Malvoni M and Laforgia D 2015 Error analysis of hybrid photovoltaic power forecasting models : A case study of mediterranean climate *Energy Convers. Manag.* **100** 117–30

[50] Louka P, Galanis G, Siebert N, Kariniotakis G, Katsafados P, Pytharoulis I and Kallos G 2008 Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering *J. Wind Eng. Ind. Aerodyn.* **96** 2348–62

[51] Ziegler L, Gonzalez E, Rubert T, Smolka U and Melero J J 2018 Lifetime extension of onshore wind turbines: A review covering Germany, Spain, Denmark, and the UK *Renew. Sustain. Energy Rev.* **82** 1261–71

[52] Yang W, Court R and Jiang J 2013 Wind turbine condition monitoring by the approach of SCADA data analysis *Renew. Energy* **53** 365–76

[53] Manobel B, Sehnke F, Lazzús J A, Salfate I, Felder M and Montecinos S 2018 Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks *Renew. Energy* **125** 1015–20

[54] Vaccaro A, Mercogliano P, Schiano P and Villacci D 2011 An adaptive framework based on multi-model data fusion for one-day-ahead wind power forecasting *Electr. Power Syst. Res.* **81** 775–82

[55] Peng H, Liu F and Yang X 2013 A hybrid strategy of short term wind power prediction *Renew. Energy* **50** 590–5

[56] Lange M and Focken U 2008 New developments in wind energy forecasting *IEEE Power Energy Soc. 2008 Gen. Meet. Convers. Deliv. Electr. Energy 21st Century, PES* 1–8

[57] Lydia M, Kumar S S, Selvakumar A I and Prem Kumar G E 2014 A comprehensive review on wind turbine power curve modeling techniques *Renew. Sustain. Energy Rev.* **30** 452–60

[58] Lange B, Rohrig K, Ernst B, Schlögl F, Cali Ü, Jursa R and Moradi J 2006 Probabilistic Forecasts for Daily Power Production *Eurepean Wind Energy Conf.*

[59] Velázquez S, Carta J A and Matías J M 2011 Influence of the input layer signals of ANNs on wind power estimation for a target site: A case study *Renew. Sustain. Energy Rev.* **15** 1556–66

[60] Seyedzadeh S, Pour Rahimian F, Rastogi P and Glesk I 2019 Tuning machine learning models for prediction of building energy loads *Sustain. Cities Soc.* **47** 101484

[61] Bergstra J, Bardenet R, Bengio Y and Kégl B 2011 Algorithms for hyper-parameter optimisation *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011* 1–9

[62] Masum M, Shahriar H, Haddad H, Faruk M J H, Valero M, Khan M A, Rahman M A, Adnan M I, Cuzzocrea A and Wu F 2021 Bayesian Hyperparameter Optimisation for Deep Neural Network-Based Network Intrusion Detection *Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021* 5413–9

[63] Zha W, Liu J, Li Y and Liang Y 2022 Ultra-short-term power forecast method for the wind farm based on feature selection and temporal convolution network *ISA Trans.*

[64] Liu H and Chen C 2019 Data processing strategies in wind energy forecasting models and applications: A comprehensive review *Appl. Energy* **249** 392–408

[65] Su Y, Yu J, Tan M, Wu Z, Xiao Z and Hu J 2019 A LSTM Based Wind Power Forecasting Method Considering Wind Frequency Components and the Wind Turbine States *2019 22nd Int. Conf. Electr. Mach. Syst. ICEMS 2019* 19–24

[66] Zu X R and Song R X 2018 Short-term Wind Power Prediction Method Based on Wavelet Packet Decomposition and Improved GRU *J. Phys. Conf. Ser.* **1087**

[67] Mujeeb S, Alghamdi T A, Ullah S, Fatima A, Javaid N and Saba T 2019 Exploiting deep learning for wind power forecasting based on big data analytics *Appl. Sci.* **9**

[68] Azimi R, Ghofrani M and Ghayekhloo M 2016 A hybrid wind power forecasting model based on data mining and wavelets analysis *Energy Convers. Manag.* **127** 208–25

[69] Shi X, Lei X, Huang Q, Huang S, Ren K and Hu Y 2018 Hourly day-ahead wind power prediction using the hybrid model of variational model decomposition and long short-term memory *Energies* **11** 1–20

[70] Liu H, Mi X wei and Li Y fei 2018 Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network *Energy Convers. Manag.* **156** 498–514

[71] Ma Z, Chen H, Wang J, Yang X, Yan R, Jia J and Xu W 2020 Application of hybrid model based on double decomposition, error correction and deep learning in short-term wind speed prediction *Energy Convers. Manag.* **205** 112345

[72] Al-Yahyai S, Charabi Y, Al-Badi A and Gastli A 2012 Nested ensemble NWP approach for wind energy assessment *Renew. Energy* **37** 150–60

[73] Dethlefs N 2020 Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines 1693–710

[74] Anon Platform for Operational Data (POD); https://pod.ore.catapult.org.uk/

[75] Serret J, Rodriguez C, Tezdogan T, Stratford T and Thies P 2018 Code comparison of a NREL-fast model of the levenmouth wind turbine with the GH bladed commissioning results *Proc. Int. Conf. Offshore Mech. Arct. Eng. - OMAE* **10**

[76] Shen X, Fu X and Zhou C 2019 A Combined Algorithm for Cleaning Abnormal Data of Wind Turbine Power Curve Based on Change Point Grouping Algorithm and Quartile Algorithm *IEEE Trans. Sustain. Energy* **10** 46–54

[77] Jafarian M and Ranjbar A M 2010 Fuzzy modeling techniques and artificial neural networks to estimate annual energy output of a wind turbine *Renew. Energy* **35** 2008–14

[78] Zou M and Djokic S Z 2020 A review of approaches for the detection and treatment of outliers in processing wind turbine and wind farm measurements *Energies* **13**

[79] Sainz E, Llombart A and Guerrero J J 2009 Robust filtering for the characterisation of wind turbines: Improving its operation and maintenance *Energy Convers. Manag.* **50** 2136–47

[80] Villanueva D and Feijóo A 2016 Normal-based model for true power curves of wind turbines *IEEE Trans. Sustain. Energy* **7** 1005–11

[81] Taslimi-Renani E, Modiri-Delshad M, Elias M F M and Rahim N A 2016 Development of an enhanced parametric model for wind turbine power curve *Appl. Energy* **177** 544–52

[82] Zhao Y, Ye L, Wang W, Sun H, Ju Y and Tang Y 2018 Data-driven correction approach to refine power curve of wind farm under wind curtailment *IEEE Trans. Sustain. Energy* **9** 95–105

[83] Schlechtingen M, Santos I F and Achiche S 2013 Using data-mining approaches for wind turbine power curve monitoring: A comparative study *IEEE Trans. Sustain. Energy* **4** 671–9

[84] Tran L, Fan L and Shahabi C 2019 Outlier Detection in Non-stationary Data Streams *ACM Int. Conf. Proceeding Ser.* 25–36

[85] Box G, Jenkins G, Reinsel G and Ljung G 2015 *Time series analysis: forecasting and control* (John Wiley & Sons)

[86] Erdem E and Shi J 2011 ARMA based approaches for forecasting the tuple of wind speed and direction *Appl. Energy* **88** 1405–14

[87] Ho S L, Xie M and Goh T N 2002 A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction *Comput. Ind. Eng.* **42** 371–5

[88] Nair K R 2017 Forecasting of wind speed using ANN , ARIMA and Hybrid Models 170–5

[89] Liu H, Tian H qi and Li Y fei 2012 Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction *Appl. Energy* **98** 415–24

[90] Kavasseri R G and Seetharaman K 2009 Day-ahead wind speed forecasting using f-ARIMA models *Renew. Energy* **34** 1388–93

[91] López E and Valle C 2018 Wind Power Forecasting Based on Echo State Networks and Long Short-Term Memory 1–22

[92] Oehmcke S, Zielinski O and Kramer O 2018 Input quality aware convolutional LSTM networks for virtual marine sensors *Neurocomputing* **275** 2603–15

[93] Huang X, Li Q, Tai Y, Chen Z, Zhang J, Shi J, Gao B and Liu W 2021 Hybrid deep neural model for hourly solar irradiance forecasting *Renew. Energy* **171** 1041–60

[94] Liu H, Mi X and Li Y 2018 Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM *Energy Convers. Manag.* **159** 54–64

[95] Bergstra J, Komer B, Eliasmith C, Yamins D and Cox D D 2015 Hyperopt: A Python library for model selection and hyperparameter optimisation *Comput. Sci. Discov.* **8**

[96] Wu J, Chen X Y, Zhang H, Xiong L D, Lei H and Deng S H 2019 Hyperparameter optimisation for machine learning models based on Bayesian optimisation *J. Electron. Sci. Technol.* **17** 26–40

[97] Zhou J, Liu H, Xu Y and Jiang W 2018 A hybrid framework for short term multi-step wind speed forecasting based on variational model decomposition and convolutional neural network *Energies* **11**

[98] Bergstra J and Bengio Y 2012 Random search for hyper-parameter optimisation *J. Mach. Learn. Res.* **13** 281–305

[99] Putatunda S and Rama K 2018 A comparative analysis of hyperopt as against other approaches for hyper-parameter optimisation of XGBoost *ACM Int. Conf. Proceeding Ser.* 6–10

[100] Mockus J, Tiesis V and Zilinskas A 1978 The application of Bayesian methods for seeking the extremum *Towar. Glob. Optim.* **2**

[101] Bull A D 2011 Convergence Rates of Efficient Global Optimisation Algorithms *J. Mach. Learn. Res.* **12** 2879–904

[102] Snoek J, Larochelle H and Adams R P 2012 Practical Bayesian Optimisation of Machine Learning Algorithms *Adv. Neural Inf. Process. Syst.* **25** 1–9

[103] Akiba T, Sano S, Yanase T, Ohta T and Koyama M 2019 Optuna: A Next-generation Hyperparameter Optimisation Framework *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 2623–31

[104] Srinivas P and Katarya R 2022 hyOPTXg: OPTUNA hyper-parameter optimisation framework for predicting cardiovascular disease using XGBoost *Biomed. Signal Process. Control* **73** 103456

[105] Peng H and Bai X 2018 Artificial neural network-based machine learning approach to improve orbit

prediction accuracy *J. Spacecr. Rockets* **55** 1248–60

[106] Li P, Zhou K, Lu X and Yang S 2020 A hybrid deep learning model for short-term PV power forecasting *Appl. Energy* **259** 114216

[107] Gupta A, Kumar A and Boopathi K 2021 Intraday wind power forecasting employing feedback mechanism *Electr. Power Syst. Res.* **201** 107518

[108] Wang J and Wang J 2021 A New Hybrid Forecasting Model Based on SW-LSTM and Wavelet Packet Decomposition : A Case Study of Oil Futures Prices **2021**

[109] Amjady N and Keynia F 2009 Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm *Energy* **34** 46–57

[110] Xiao L, Qian F and Shao W 2017 Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm *Energy Convers. Manag.* **143** 410–30

[111] Jorgensen K L and Shaker H R 2020 Wind Power Forecasting Using Machine Learning: State of the Art, Trends and Challenges *2020 8th Int. Conf. Smart Energy Grid Eng. SEGE 2020* 44–50

[112] Ela E, Kirby B, Navid N and Smith J C 2012 Effective ancillary services market designs on high wind power penetration systems *IEEE Power Energy Soc. Gen. Meet.* 1–8

[113] Bao Y, Xiong T and Hu Z 2014 Multi-step-ahead time series prediction using multiple-output support vector regression *Neurocomputing* **129** 482–93

[114] Fu Y and Hu W 2021 Multi-step Ahead Wind Power Forecasting Based on Recurrent Neural Networks 217–22

[115] Wang J and Li Y 2018 Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy *Appl. Energy* **230** 429–43

[116] Huang Y, Yang L, Liu S and Wang G 2019 Multi-step wind speed forecasting based on ensemble empirical mode decomposition, long short term memory network and error correction strategy *Energies* **12** 1–22

[117] Liang Z, Liang J, Wang C, Dong X and Miao X 2016 Short-term wind power combined forecasting based on error forecast correction *Energy Convers. Manag.* **119** 215–26

[118] Liu H, Mi X and Li Y 2018 Wind speed forecasting method based on deep learning strategy using empirical wavelet transform , long short term memory neural network and Elman neural network *Energy Convers. Manag.* **156** 498–514

[119] Liu H, Tian H Q, Chen C and Li Y fei 2010 A hybrid statistical method to predict wind speed and wind power *Renew. Energy* **35** 1857–61

[120] Li Y, Wu H and Liu H 2018 Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction *Energy Convers. Manag.* **167** 203–19

[121] Liu H, Mi X and Li Y 2018 Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network *Energy Convers. Manag.* **166** 120–31

[122] Liu H, Mi X and Li Y 2018 An experimental investigation of three new hybrid wind speed forecasting models using multi-decomposing strategy and ELM algorithm *Renew. Energy* **123** 694–705

[123] Wu Z, Xia X, Xiao L and Liu Y 2020 Combined model with secondary decomposition-model selection and sample selection for multi-step wind power forecasting *Appl. Energy* **261** 114345

[124] Meng A, Ge J, Yin H and Chen S 2016 Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimisation algorithm *Energy Convers. Manag.* **114** 75–88

[125] Ben Taieb S, Bontempi G, Atiya A F and Sorjamaa A 2012 A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition *Expert Syst. Appl.* **39** 7067–83

[126] Taieb S Ben, Bontempi G, Sorjamaa A and Lendasse A 2009 Long-term prediction of time series by combining direct and MIMO strategies *Proc. Int. Jt. Conf. Neural Networks* 3054–61