

# Wind Power Forecasting – A Data-driven Method along with Gated Recurrent Neural Network

Adam Kisvari<sup>a</sup>, Zi Lin<sup>b</sup>, Xiaolei Liu<sup>a1</sup>

<sup>a</sup>James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom

<sup>b</sup> Department of Mechanical & Construction Engineering, Northumbria University, Newcastle, NE1 8ST, United Kingdom

## Abstract

Effective wind power prediction will facilitate the world’s long-term goal in sustainable development. However, a drawback of wind as an energy source lies in its high variability, resulting in a challenging study in wind power forecasting. To solve this issue, a novel data-driven approach is proposed for wind power forecasting by integrating data pre-processing & re-sampling, anomalies detection & treatment, feature engineering, and hyperparameter tuning based on gated recurrent deep learning models, which is systematically presented for the first time. Besides, a novel deep learning neural network of Gated Recurrent Unit (GRU) is successfully developed and critically compared with the algorithm of Long Short-term Memory (LSTM). Initially, twelve features were engineered into the predictive model, which are wind speeds at four different heights, generator temperature, and gearbox temperature. The simulation results showed that, in terms of wind power forecasting, the proposed approach can capture a high degree of accuracy at lower computational costs. It can also be concluded that GRU outperformed LSTM in predictive accuracy under all observed tests, which provided faster training process and less sensitivity to noise in the used Supervisory Control and Data Acquisition (SCADA) datasets.

**Keywords:** Wind power forecasting; SCADA data; Feature engineering; Deep learning; Offshore wind turbines.

## ABBREVIATION:

AdaGrad	Adaptive Gradient Algorithm
Adam	Adaptive Moment Estimation
ANN	Artificial Neural Network
AR	Autoregressive
ARMA	Autoregressive Moving Average
CEC	Constant Error Carousel

---

<sup>1</sup> Corresponding author, E-mail: [Xiaolei.Liu@glasgow.ac.uk](mailto:Xiaolei.Liu@glasgow.ac.uk) (XL)

31	CV	Cross-Validation
32	DT	Decision Tree
33	ET	Extra Trees
34	GB	Gradient Boost
35	GRNN	Gated Recurrent Neural Networks
36	GRU	Gated Recurrent Unit
37	IEC	International Electrotechnical Commission
38	IF	Isolation Forest
39	KNN	K-Nearest Neighbours
40	LSTM	Long Short-term Memory
41	MSE	Mean Square Error
42	NAG	Nesterov's Accelerated Gradient
43	NWP	Numerical Weather Predictions
44	ORE	Offshore Renewable Energy
45	PMG	Permanent Magnet Synchronous Generator
46	RF	Random Forest
47	RFE	Recursive Feature Elimination
48	RMSProp	Root Mean Square Propagation
49	RNN	Recurrent Neural Networks
50	SCADA	Supervisory Control And Data Acquisition
51	SGD	Stochastic Gradient Descent
52	SVM	Support Vector Machine
53	SVR	Support Vector Regressor

## 54 **1. Introduction**

55 In the past few decades, a growing emphasis has been placed on sustainable developments of natural resources and slowing  
56 down climate change which triggered revolutions in the energy sector. This led to a surge in interest for integrating carbon-free  
57 electrical energy production into energy portfolios. As part of this transition, wind power is considered an appealing alternative  
58 to replace conventional energy resources, mainly fossil fuel power plants. Although the integration of wind power offers great  
59 potential, it also faces great operational and planning challenges due to the intermittent nature of the wind resource [1], which

60 can result in financial losses to both grid operators and consumers. Several studies have been conducted in the areas of  
61 aerodynamic optimization of wind turbines [2], blade shapes [3], power curves [4], and optimizing of wind turbine position in  
62 a wind farm [5]. An essential part of effective integration of wind energy lies in the accurate forecasting of wind energy  
63 production, which is crucial to all stakeholders for avoiding overproduction by coordinating energy supply and demand [6] as  
64 well as enabling maintenance to be scheduled under power predictions [7].

### 65 *1.1 Motivation and incitement*

66 Even countries with the most advance renewable energy sectors, such as Scotland or Germany, face difficulties in fully  
67 relying on renewable sources. Today, grid operators are forced to resort to conventional power stations under certain weather  
68 conditions, which then need to quickly drop their output if the conditions change to avoid wasting power or overloading the  
69 grid, which may result in failures. These adjustments, however, bear significant costs as it was estimated that German  
70 consumers had to pay about \$553 million to cover the costs of compensating utility firms for adjustments to their inputs in 2016  
71 [8]. One of the solutions is to use available weather data as well as historical turbine data to predict wind power [9] ahead of  
72 actual generation. This is crucial as it not only relieves pressure on grid operators and reduces the output required from  
73 conventional power stations, but also due to the higher value of energy sources that can be scheduled in advance.

### 74 *1.2 Literature review*

75 Wind power forecasting models are by most scholars categorized as statistical and physical models. Both methods are  
76 capable of predicting wind power generation effectively, but they are profoundly different in approach [10]. Physical models  
77 use mathematical expressions to model highly complex and nonlinear dynamics of the atmospheric flow to produce numerical  
78 weather predictions (NWP). The obtained NWPs are adapted to local flow conditions and then used as inputs in the wind  
79 power forecasting systems [11]. On the other hand, statistical methods rely on relevant historical data to predict future power  
80 generation, traditionally using models such as autoregressive (AR) or autoregressive moving average model (ARMA). In recent  
81 years, the wealth of data supplied by the built-in Supervisory Control And Data Acquisition (SCADA) systems have given rise  
82 to excessively large and complex datasets, which exceed the capabilities of traditional prediction methods and therefore have  
83 been processed using machine learning techniques such as artificial neural networks (ANNs) and support vector machine  
84 (SVM) [10].

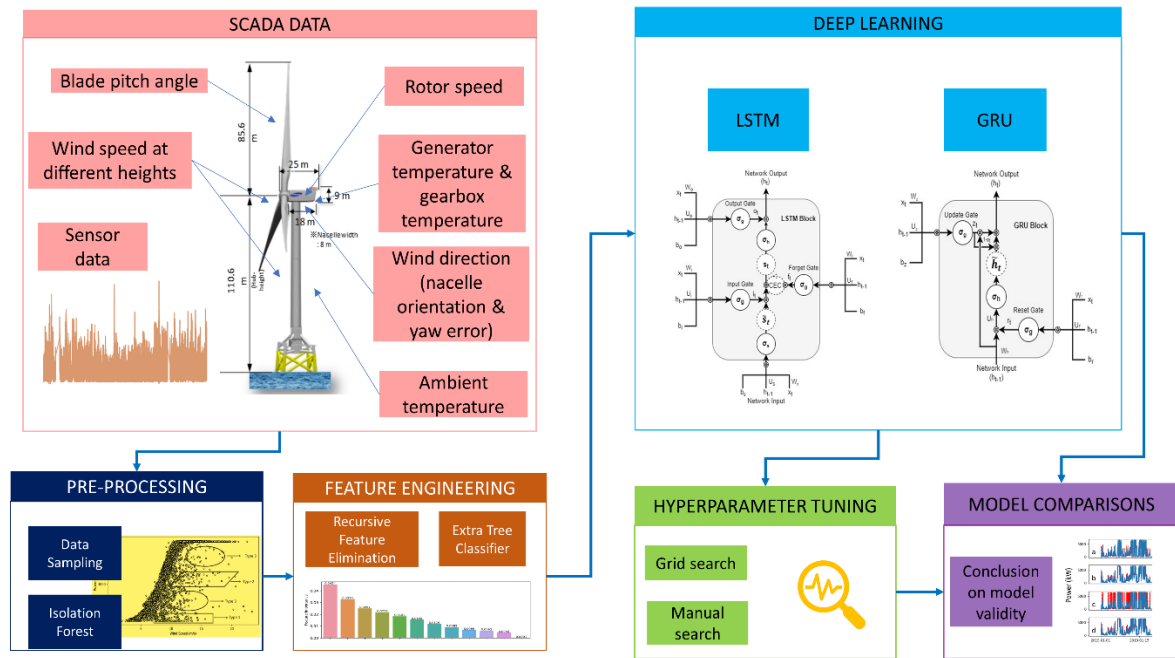
85 In recent years, ANNs emerged as one of the most commonly used machine learning algorithms in the field of wind power  
86 forecasting [12]. ANNs are complex structures that attempt to resemble the structure of the human brain based on a set of  
87 replicated processing units called neurons, which are interlinked and pass information via weighted connections that are  
88 adjusted during the training process. Developments in initialization algorithms and neuron activation functions enhanced the

89 capabilities of ANN and made it possible to solve complex non-linear problems by training models consisting of a large number  
90 of hidden layers, which is often referred to as “deep learning” [12]. The increasing complexity of wind turbine systems and the  
91 ensuing demand for improvements in reliability [13], maintenance [14], investments [15], and forecasting [16] prompted rising  
92 adoption of deep learning [17] in the wind energy sector.

93 Recurrent Neural Network (RNN) is a class of ANNs, in which the connection between its neurons form loops, allowing  
94 information to persist. This means it is capable of handling non-linear dependencies between past time series values and the  
95 estimate of values to be predicted via the inherent dynamic memory created by recurrent connections in the hidden layers.  
96 Despite its superiority over conventional ANNs, RNN suffers from a phenomenon referred to as vanishing or exploding  
97 gradients caused by error signals flowing backwards, which leads to oscillating weights or loss of long-term dependencies due  
98 to the rapid decay (vanishing) or increase (exploding) in the norm of gradient during training [18]. Amongst the numerous  
99 methods proposed to address vanishing and exploding gradients, the introduction of gating mechanisms to control the flow of  
100 information between layers has shown promising results and practical applications. Notable examples of RNN architectures  
101 adopting this principle are Gated Recurrent Unit (GRU) introduced by Cho et al. [19] and Long-short Term Memory (LSTM)  
102 proposed by Hochreiter et al. [20]

### 103 *1.3 Objective and methodology*

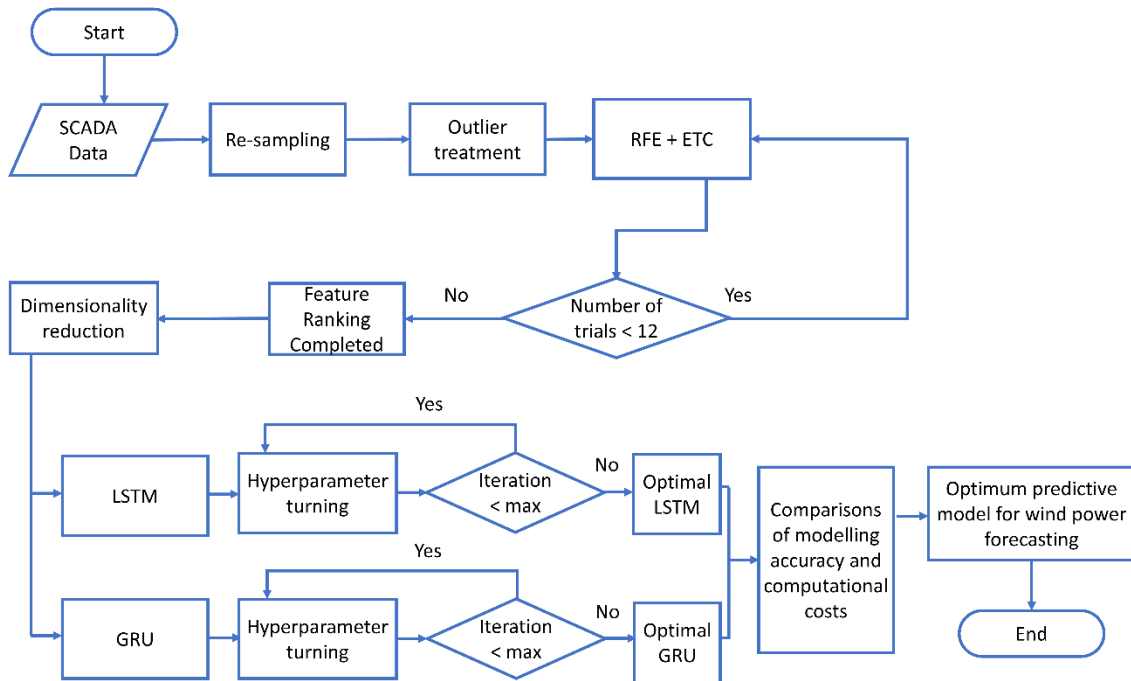
104 The major objective of this study is to explore the use of state of art machine learning techniques to construct and optimize  
105 deep learning models based on Gated Recurrent Neural Networks (GRNNs), namely GRU and LSTM, to predict wind power  
106 outputs from historical turbine data collected from the target wind turbine, a 7MW Offshore wind turbine situated in  
107 Levenmouth, Scotland. This study applies advanced data filtering, feature engineering, and model optimizing to deliver  
108 improvements in terms of predictive accuracy, generalization ability as well as computational performance for wind power  
109 prediction models. The methodology of this study and the used machine learning algorithm processing flowchart is summarised  
110 in **Fig. 1** and **Fig. 2**, respectively.



111

112

Fig. 1 – Diagram of applied methodology.



113

114

Fig. 2 – Machine learning algorithm processing flowchart.

115 *1.4 Contribution and paper organization*

116 The key contributions of this paper to the current knowledge gaps can be summarised as follows:

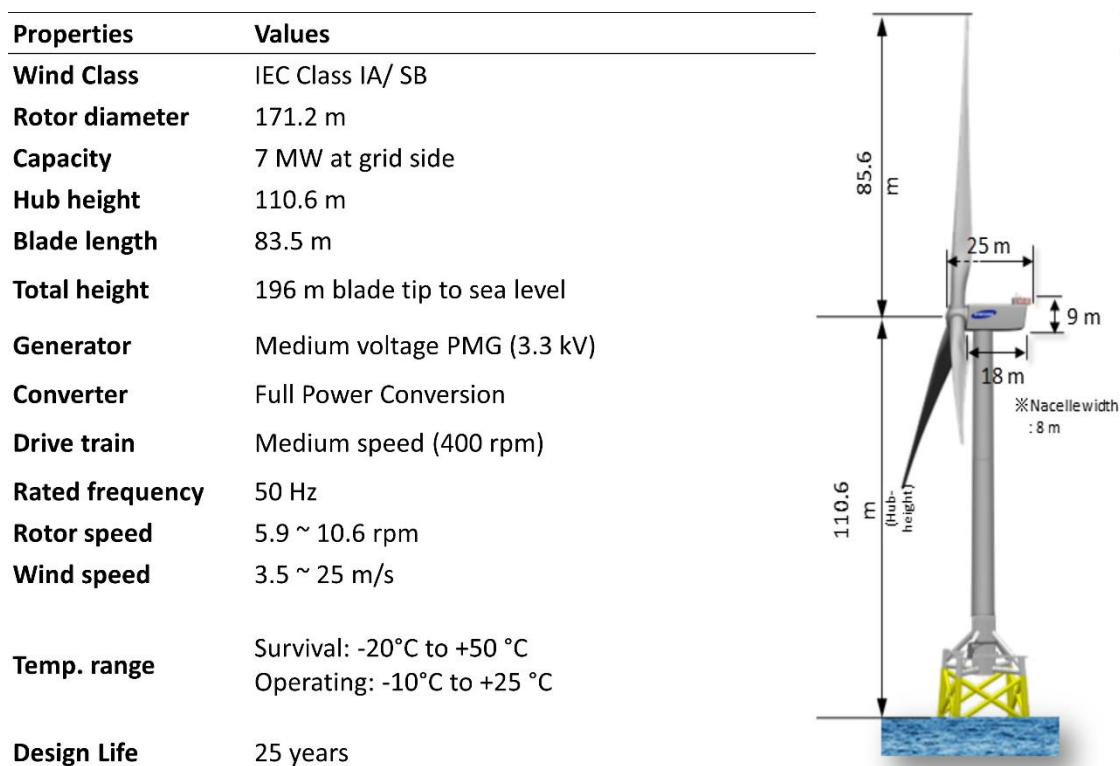
- 117     ▪ Existing studies on wind power forecasting using neural networks have mainly been based on mid-fidelity methods,  
118     such as LSTM, for which the entire variability of actual wind power may not be fully realised. Furthermore, the nature  
119     of wind has the feature of stochastic distributions and high variability. In recent years, GRNNs have been proven to  
120     be superior to traditional ANNs and vanilla RNNs for long-input time series sequences, which implies its great  
121     potential for wind power forecasting. There has been an increasing amount of investigations of GRNN in other fields,  
122     such as speech recognition [20] and traffic flow prediction [21]. However, to date, no such comparison has been made  
123     in the field of wind power forecasting. Therefore, in this paper, a novel deep learning method, using GRU, has been  
124     applied in predicting the power output for an offshore wind turbine and its validity in wind power forecasting has been  
125     comprehensively assessed by comparisons with the LSTM.
- 126     ▪ In modern wind turbines, several essential components, such as yaw-control system, pitch-control system, generator,  
127     gearbox, and rotor, can strongly impact power generation. However, these integral features within wind energy  
128     conversion systems were not widely studied in previous literature. In this paper, feature engineering was carried out  
129     by Recursive Feature Elimination (RFE) along with Extra Trees Classifier (ETC) in wind power prediction. The  
130     benefits of these methods are bi-fold by determining not only the explained variance of individual variables but also  
131     the optimal number of features to use to maintain a balance between computational cost and predictive accuracy. The  
132     application of RFE and ETC ensure effective feature selection by removing bias that arises from the varying  
133     contribution of individual variables to the explained variance as the pool of features is reduced.
- 134     ▪ In this study, Isolation Forest (IF) was used to detect and remove outliers in the target SCADA database, before feeding  
135     it to deep learning models for offshore wind power forecasting. IF is an outlier detection algorithm that is  
136     fundamentally different from its alternatives, applying explicit isolation of outliers rather than profiling normal data  
137     points through the use of density and distance measures. In the absence of any distributional assumptions, IF ensures  
138     efficient and effective operation with datasets of high-dimensionality, which makes it highly suitable for wind power  
139     application and enhance models by reducing computation time and costs [20].

140     The remainder of this paper is organized as follows. Section 2 provides a detailed description of the target wind turbine as  
141     well as the used SCADA datasets, including how the dataset was treated in pre-processing, resampling, and outlier detection.  
142     Section 3 introduces how features were engineered through RFE & ETC to identify the optimal subset of features to be used in  
143     the designed deep learning model. Section 4 introduces the theoretical background of GRU and LSTM. Section 5 presents the  
144     key observations and simulation results attained from final wind power predictive models, which were trained using GRNN.  
145     Section 6 concludes this study by summarizing key findings and contributions of this paper.

146 **2. SCADA data pre-processing**

147 *2.1 Target wind turbine*

148 The target wind turbine is a 7 MW demonstration offshore wind turbine situated in Levenmouth, Fife, Scotland, UK. It is  
 149 a three-bladed upwind turbine mounted on a jacket support structure with a total height of 196 m, from the blade tip to the sea  
 150 level. **Fig. 3** shows the configuration and major parameters of the wind turbine, which has a rotor diameter of 171.2 m and a  
 151 hub height of 110.6 m. In terms of operating regions, the designed cut-in, rated and cut-out speeds are 3.5, 10.9 and 25 m/s,  
 152 respectively. The wind turbine is based upon a Permanent Magnet Generator (PMG) that is driven via a medium speed (400  
 153 rpm) and connects to a full-power converter, allowing the wind turbine to achieve the maximum power coefficient at a wide  
 154 range of wind speeds. The target wind turbine is owned by Offshore Renewable Energy (ORE) Catapult [21].



155 **Fig. 3** - Schematic and main properties of Levenmouth offshore wind turbine [22].

156 *2.2 Data description*

158 The investigated SCADA datasets were recorded over a nine-month period from 1<sup>st</sup> July 2018 to 31<sup>st</sup> March 2019. The  
 159 time-series data signals were collected by the built-in SCADA system at 1 Hz (1-second intervals), generating 574 data points  
 160 at any given timestamp. The collected dataset was split into six-month training and three-month testing/validation datasets in  
 161 the modelling phase. Before processing the datasets, an initial data selection was conducted to limit the size of the applied

162 dataset by excluding redundant variables to manage computation costs. At this stage, data units were selected to ensure a high  
 163 degree of explained variance for the target variable (active power). This was achieved through the representation of:

- 164     ▪ independent inputs (i.e. meteorological factors), including wind speeds at various heights, wind direction  
 165         represented by a combination of nacelle orientation & yaw error, and ambient temperature;
- 166     ▪ aerodynamic factors affecting wind energy capture, such as average blade pitch angle;
- 167     ▪ key parameters in mechanical power transmission systems, such as instantaneous & averaged rotor speeds,  
 168         generator temperature and gearbox temperature.

169     Based on the above criteria, the following 12 features were selected at the initial stage: wind speed at the hub height of  
 170 110.6 m, wind speeds at heights of 25 m, 67 m and 110 m, respectively, generator temperature, gearbox temperature, nacelle  
 171 orientation, yaw error, average blade pitch angle, instantaneous and averaged rotor speed, and ambient temperature. The  
 172 statistical description of count, mean, percentile and standard deviation of selected features were presented in **Table 1**.

173                                   **Table 1** – Statistical descriptions of the raw SCADA datasets.

	Count	Mean	Standard deviation	Minimum	25%	Median	75%	Maximum
Wind speed (25 m), m/s	2.32E+07	7.44E+00	3.98E+00	-3.32E-02	4.63E+00	6.83E+00	9.71E+00	4.32E+01
Wind speed (67 m), m/s	2.32E+07	-7.95E+11	2.71E+15	-9.22E+18	4.83E+00	7.00E+00	9.79E+00	4.31E+01
Wind speed (110 m), m/s	2.32E+07	7.48E+00	3.84E+00	-1.50E+01	4.82E+00	6.97E+00	9.69E+00	4.16E+01
Wind speed (110.6 m), m/s	2.32E+07	7.48E+00	3.84E+00	-1.50E+01	4.82E+00	6.97E+00	9.69E+00	4.16E+01
Generator temperature, ° C	2.32E+07	5.04E+01	2.24E+01	-6.01E+01	3.26E+01	4.55E+01	6.27E+01	1.29E+02
Gearbox temperature, ° C	2.32E+07	-1.19E+12	3.32E+15	-9.22E+18	4.72E+01	5.20E+01	5.53E+01	1.27E+04
Nacelle orientation, °	2.32E+07	2.11E+02	7.23E+01	7.10E-04	1.80E+02	2.32E+02	2.54E+02	3.60E+02
Measured yaw error, °	2.32E+07	-1.53E-02	4.53E-01	-3.14E+00	-1.21E-01	3.12E-03	1.27E-01	3.18E+00
Average blade pitch angle, °	2.32E+07	3.90E+01	3.74E+04	-1.00E+03	-1.56E-01	8.43E-01	8.91E+01	1.27E+08
Instantaneous rotor speed, rpm	2.32E+07	5.09E+00	4.49E+00	-1.86E+00	1.50E-02	5.33E+00	9.15E+00	5.01E+03
Averaged rotor speed, rpm	2.32E+07	5.33E-01	4.71E-01	-5.90E-02	1.59E-03	5.57E-01	9.44E-01	5.12E+02
Ambient temperature, ° C	2.32E+07	1.09E+01	4.36E+00	0.00E+00	7.60E+00	1.05E+01	1.42E+01	2.55E+01

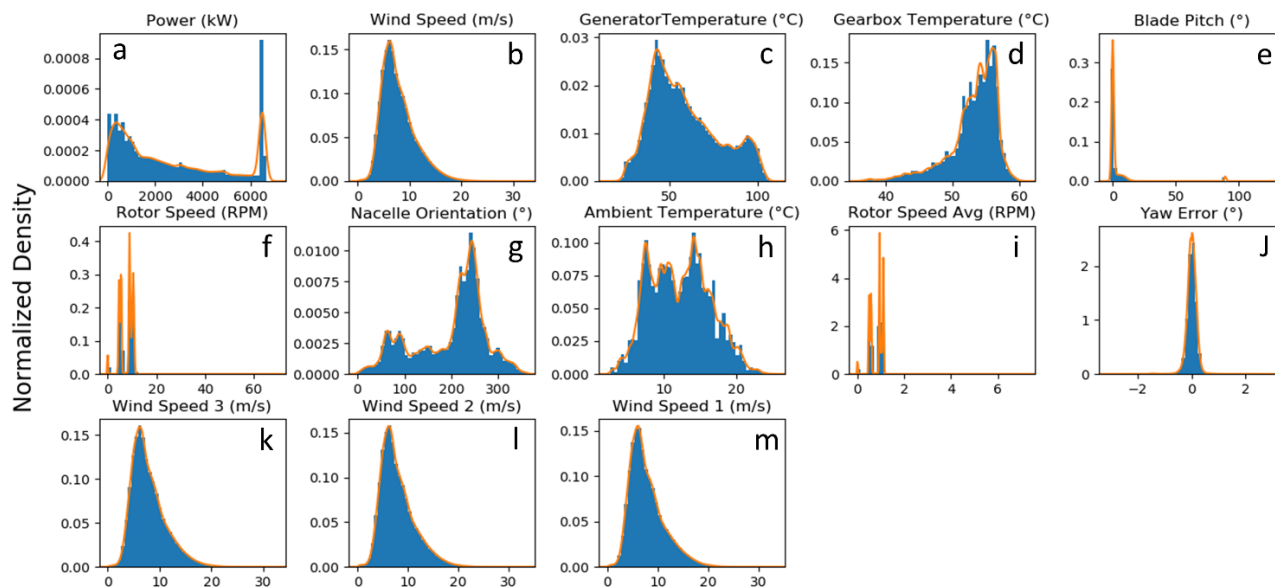
174  
 175 *2.3 Obvious outlier removal*

176     Closer examination of individual parameters highlights certain obvious errors in the SCADA dataset. Although physically  
 177 possible, negative values of active power, wind speed and highly negative blade pitch angles (defined as below  $-10^\circ$ ) carry no  
 178 practical meaning in wind power generation in general and have thus been removed along with the corresponding parameters  
 179 belonging to the same timestamp. Similarly, timestamps with missing values have also been removed from the time series to  
 180 avoid their negative influence on the predictive models. Such erroneous data points are often the results of sensor malfunction,  
 181 system processing errors or even sensor degradation, which make it essential to pre-process SCADA data before using them to  
 182 build models [23].

183     **Fig. 4** presents the data distribution of selected input and output features after the obvious outlier detection and removal.  
 184 It can be noted that the mean and median of wind speed at hub height are 7.86 and 7.17 m/s (see **Fig. 4b**), respectively, which



185 is lower than the rated wind speed (10.9 m/s). This implies that the wind turbine spends the majority of its operating time below  
 186 the rated power (7 MW), which is well illustrated in **Fig. 4a**. The mean of the average blade pitch angle was measured to be  
 187 3.42° (see **Fig. 4e**), while the mean of Nacelle orientation, which is representative of the prevailing wind conditions, is measured  
 188 to be 197.85° (see **Fig. 4g**). The mean ambient temperature for the investigated period is measured to be 12.2°C, with minimum  
 189 and maximum values of 2.2 and 25.5°C (see **Fig. 4h**), respectively. The scatterings of wind speed (see **Fig. 4b, k, l and m**),  
 190 ambient temperature (see **Fig. 4h**), yaw error (see **Fig. 4j**), generator temperature (see **Fig. 4c**), and gearbox temperature (see  
 191 **Fig. 4d**) can be considered a normal distribution, whereas the scattering of nacelle orientation showed a bimodal distribution  
 192 (see **Fig. 4g**), which indicated that the local wind conditions can be split into two dominant wind directions.

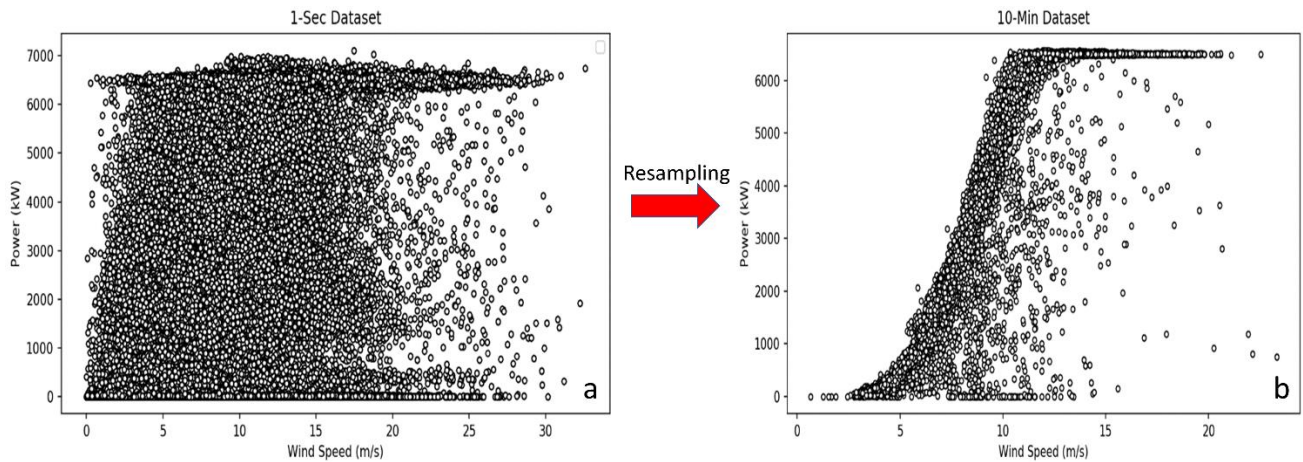


193  
 194 **Fig. 4** - Histograms of selected input and output parameters after obvious outlier detection.

195 *2.4 Data re-sampling*

196 One of the key challenges preventing wind energy from increasing its penetration in energy markets arises from the strong  
 197 volatility of wind caused by turbulence. To account for the effects of turbulence, aerodynamic models typically characterize  
 198 wind flow using a combination of steady-flow mean wind speed and a variation factor describing the fluctuations caused by  
 199 the embedded turbulent eddies (i.e. turbulence intensity). The effect of turbulence in the case of horizontal axis wind turbines  
 200 is bi-fold, causing the wind hitting the swept blade rotors to rapidly vary both in terms of speed and direction within a three-  
 201 dimensional space. This presents a significant issue, whereby wind speed measurements taken by the installed anemometers  
 202 are not necessarily coherent with the speed of wind flow hitting rotor blades, resulting in reduced correlations between the  
 203 measured wind speed and the power output, which present itself as scatters in the power curve. This effect could be curbed by  
 204 averaging the obtained data samples over an appropriate averaging period dependent on the size of the actual turbine [24]. The

205 international standard for power performance measurements of electricity producing wind turbines (IEC 61400-12-1) stipulates  
 206 an averaging time of 10 minutes for large wind turbines [25], which coincides with the averaging time standards of most  
 207 meteorological institutions and the wind power spectral gap. To this end, it is of key importance to tailor available input data  
 208 to the overall needs of the forecasting model through high-frequency data acquisition and, where required, appropriate  
 209 averaging. In this study, the original dataset that was collected at 1 Hz frequency was averaged over 10 minutes averaging  
 210 periods following IEC 61400-12-1. **Fig. 5a** and **Fig. 5b** displays the wind power curves constructed from the original 1-sec and  
 211 the resampled 10-min SCADA datasets, respectively. It can be noted that, due to the stochastic nature of wind, both wind power  
 212 curves presented a certain degree of scattering, which is particularly prominent in the 10-min power curve and is caused by the  
 213 non-linear and multidisciplinary dynamics associated with offshore wind turbine systems [26]. **Fig. 5b** (10-min SCADA  
 214 dataset) presented a smoother sigmoidal shaped power curve. Therefore, the resampled SCADA dataset with a sampling rate  
 215 of 10-min was used for this study to limit the impact of turbulence and noise on the overall turbine performance.



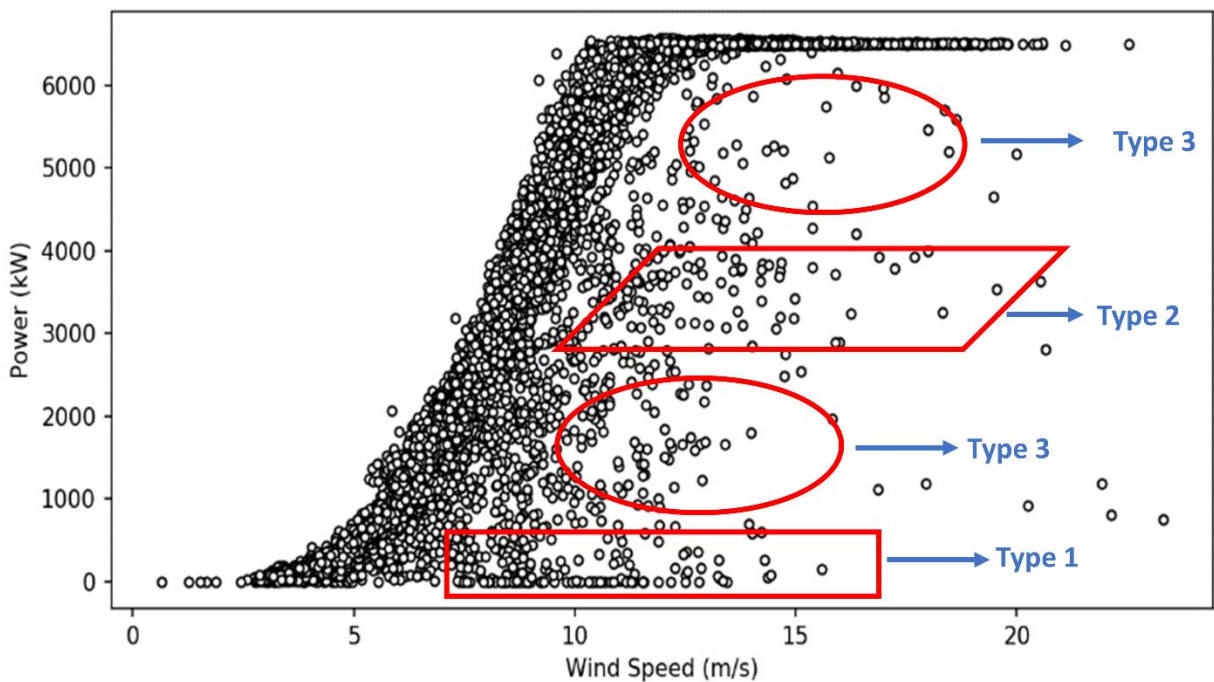
216  
 217 **Fig. 5** - Wind power curve under 1-second (a) and 10-minute (b) sampling rates.

218 *2.5 Anomalies detection and treatment*

219 SCADA datasets often contain erroneous data points, which may be caused by several reasons, including maintenance,  
 220 operational planning, breakdown and even sensor degradation. These erroneous data are detrimental to the performance of wind  
 221 power prediction models and therefore need to be removed using appropriate outlier detection methods. Closer inspection of  
 222 the wind turbine power (see **Fig. 6**) highlights three common types of anomalies present in the available wind turbine SCADA  
 223 dataset:

- 224 ■ Type 1 anomalies are represented in the scatter plot by a horizontally dense data cluster, whereby the wind speed is  
 225 larger than the cut-in speed (3.5 m/s), but the generated power is zero. This type of anomalies is normally the result  
 226 of turbine downtime [27], which can be cross-referenced using operating logs [21].

- 227
- 228 ■ Type 2 anomalies are represented by a dense data cluster that falls below the ideal power curve of the wind turbine. This type of anomalies can be caused by wind curtailment, whereby the power output of the turbine is artificially constrained by its operator below its operating capacity. Wind curtailment can be imposed by wind farm operators for several reasons, including lack of demand at given times, difficulties in storing large capacity wind power and finally the unstable nature of electric energy generated by wind turbines at times of volatile wind conditions.
  - 232 ■ Type 3 anomalies are randomly distributed around the curve and are normally caused by sensor malfunction, degradation or noise during signal processing [28,29]. It can also be noted that a fraction of Type 2 and 3 anomalies can also be described by the dispersion created due to incoherent wind speed measurements taken as a result of turbulence.
- 235



236

237 **Fig. 6** – Observed anomalies along wind power curve under 10-minute sampling rates.

238 Given the paramount importance of wind power curves as a wind turbine performance metric, the outliers pose significant

239 challenges in its vital applications. In this study, the IF algorithm is used to detect and remove various outliers from the 10-min

240 SCADA dataset, which has been considered as one of the most effective algorithms for novelty and outlier detection in wind

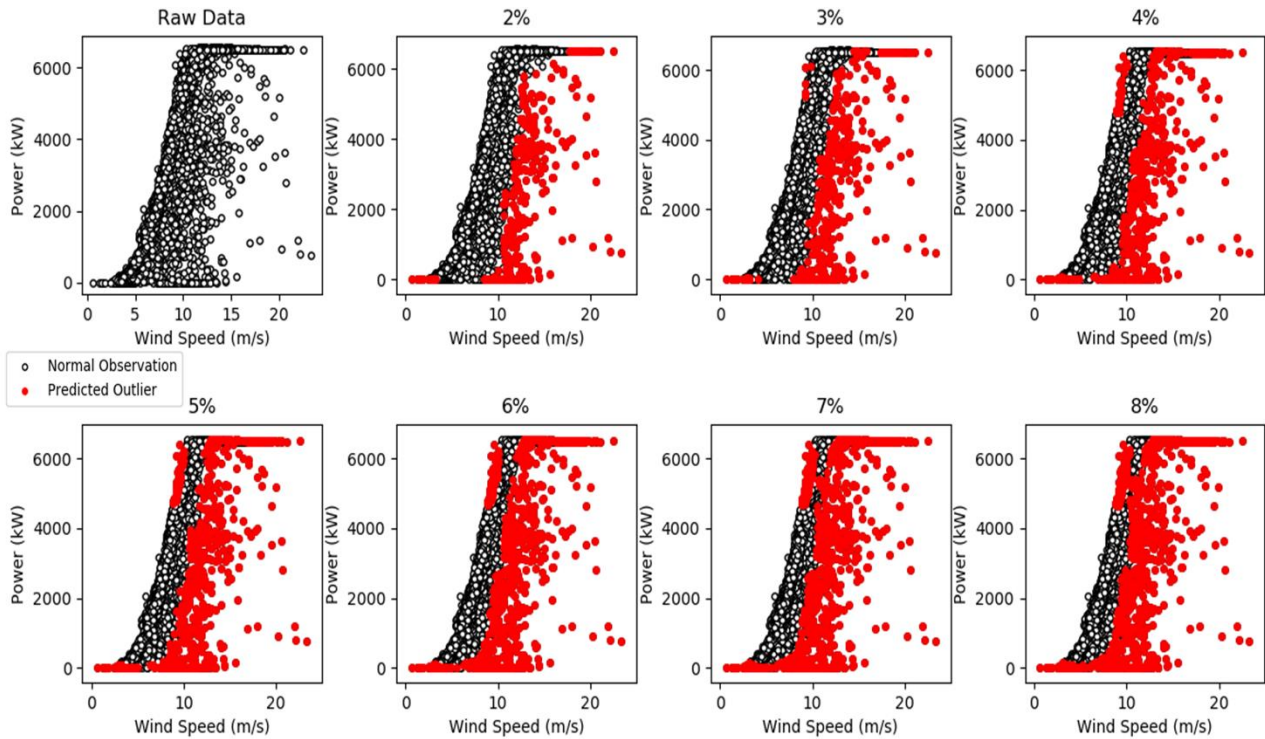
241 power prediction [21,30]. IF is an ensemble learning method based on a binary tree structure, consisting of a set of isolation

242 trees. It works by isolating all instances in a given dataset through iterative partitioning to achieve a random tree structure. In

243 this context, the number of splitting required to isolate an instance corresponds to its path length from the root node to the

244 terminating node, which is averaged over a number of trees. The results of the iterative application of different contamination

245 ratios (2 - 8%) through IF are presented in **Fig. 7**. In the current study, 4% contamination ratio was determined to be most  
246 suitable for the given task as it best represents the ideal shape of the wind power curve, taking into account the cut-in, rated and  
247 cut-off wind speeds of the target wind turbine, whilst preserving a wide range of wind speeds.



248

249

**Fig. 7** – Outlier detection and treatment along with isolation forest.

250

### 3. Feature engineering

251

Feature engineering aims to transform raw data, herein time series, into an optimal subset of features that best represent  
252 the underlying concept of the given dataset. In this study, a combination of two algorithms was used, namely RFE and ETC.

253

#### 3.1 RFE with Cross Validation

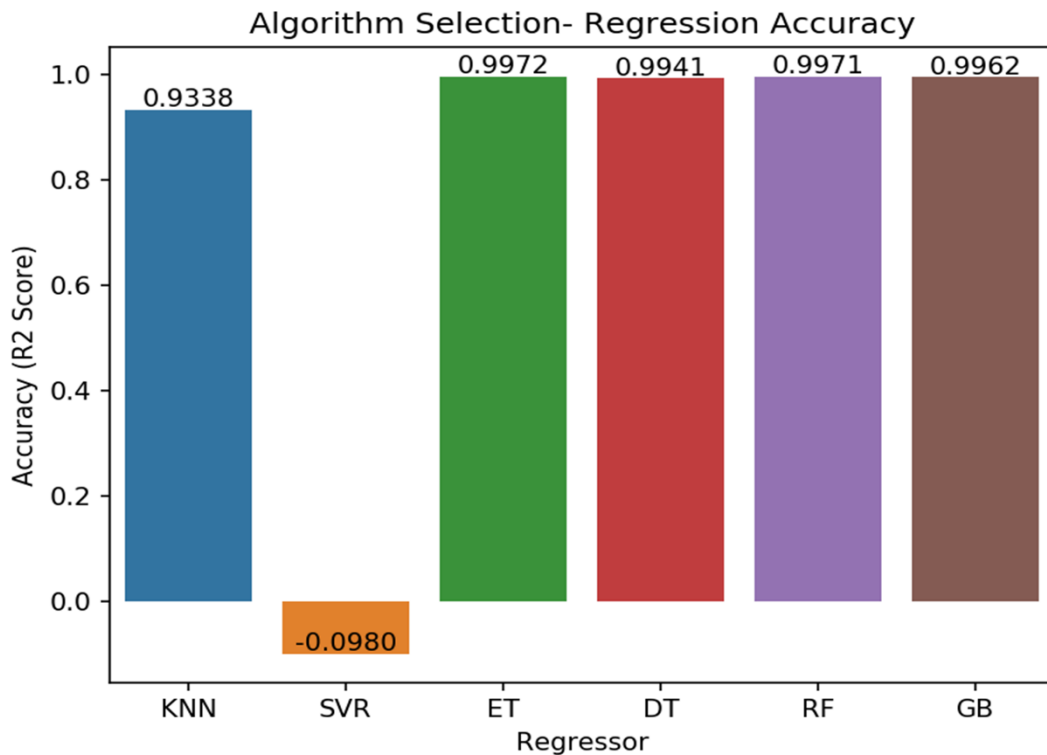
254

The RFE works by recursively removing features in a stepwise manner based on their feature importance and a measure  
255 of their relevance to the overall output until a specified number of features is attained. At each recursion, it uses model accuracy  
256 to eliminate a feature or a group of features that contributes least to predicting the desired output. The final ranking of the  
257 features is compiled based on the inverse order of their elimination [31]. Given that the current optimal number of features is  
258 not known, RFE was used in conjunction with cross validation to evaluate the performance of the model at each stepwise  
259 elimination stage against the validation data.

260

##### 3.1.1 Algorithm identification

261 An estimator algorithm needs to be trained through RFE to obtain feature importance coefficients for each variable, which  
262 can be used to rank and recursively eliminate features. To ensure a high degree of accuracy, six estimator algorithms were  
263 evaluated based on their performance on the given SCADA dataset. The six algorithms are K-Nearest Neighbours (KNN),  
264 Support Vector Regressor (SVR), Extra Tree (ET), Decision Tree (DT), Random Forest (RF), and Gradient Boost (GB),  
265 respectively. As presented in **Fig. 8**, it is clear that SVR is unsuitable for the current task. However, all other alternatives are  
266 comparable in terms of their performances. Amongst all options, ET Regressor (also referred to as Extremely Randomized  
267 Trees) showed marginally superior performance and was thus chosen as the estimator algorithm for the current RFE process.  
268 The ET algorithm is similar to other tree-based algorithms and works by building an ensemble of unpruned decisions or  
269 regression trees, depending on applications as a classifier or a regressor. As opposed to other tree-based methods, ET splits  
270 nodes by selecting cut-points fully at random and grows trees using the entire learning sample instead of bootstrap replicas  
271 [32].



272  
273 **Fig. 8** – Estimator algorithm selection of RFE.

274 *3.1.2 Recursive Feature Elimination (RFE)*

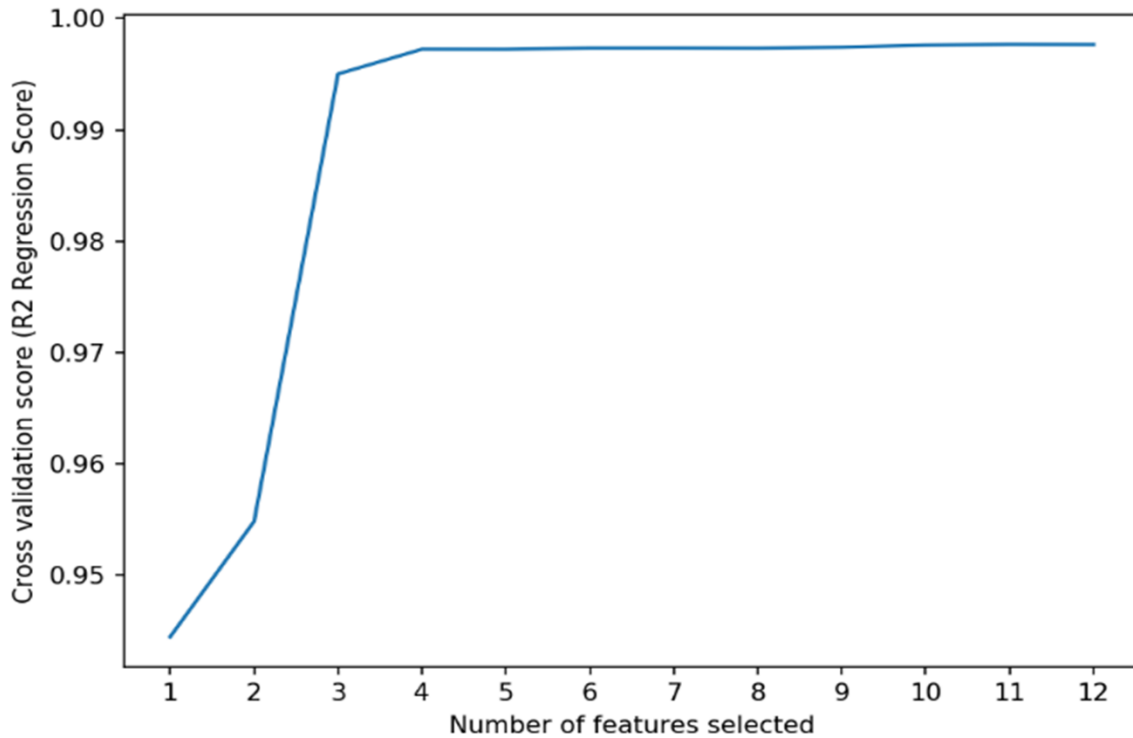
275 RFE was conducted by splitting the training dataset into the target variable (active power) and independent variables,  
276 which were fed into the model whilst applying a 10-fold cross validation using testing dataset. The R-squared (R2) statistical  
277 measure was used as the scoring function of the model due to its direct representation of the proportion of the target variable's

278 variance explained by the set of features, which simplifies the interpretation of the results. The R2 scoring function can be  
279 expressed as:

$$R^2 = 1 - \sum_{i=1}^N \frac{(y_i - \hat{y})^2}{(y_i - \bar{y})^2} \quad (1)$$

280 where  $N$  refers to the number of data points,  $y_i$  is the  $i^{\text{th}}$  actual value,  $\bar{y}$  is the mean value of  $y$  and  $\hat{y}$  is the predicted  
281 value of  $y$ .

282 As shown in **Fig. 9**, six parameters offered an ideal compromise between model accuracy and computation time. Using  
283 additional parameters would only enhance the cumulative explained variance marginally (<0.1%), whilst increasing the  
284 computational expense proportionally. It has been concluded that the six best features for the current task are wind speed at  
285 hub height, generator temperature, gearbox temperature, blade pitch angle, instantaneous rotor speed in RPM and nacelle  
286 orientation.

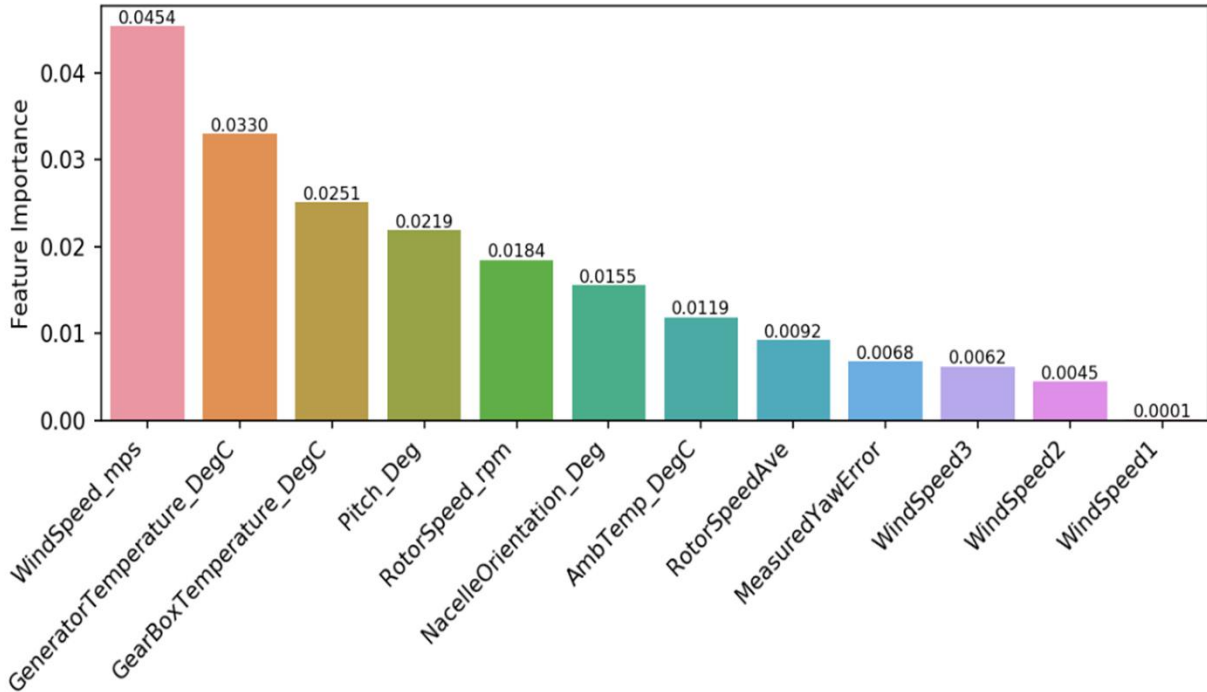


287  
288 **Fig. 9** – Cross validation score variation along with numbers of selected features.

### 289 3.2 Extra Tree Classifier (ETC)

290 To validate the findings from the RFE process, an ETC (also referred to as Extremely Randomized Trees Classifier) was  
291 implemented to compute the relative importance of features. ETC is an ensemble learning technique, which fits randomized  
292 decision trees onto various sub-samples of a given dataset to improve model accuracy and fit via averaging. As **Fig. 10**  
293 suggested, the six most significant features coincide with the findings from RFE, thus concluding its validity and confirming

294 the feature selection of wind speed at hub height, generator temperature, gearbox temperature, blade pitch angle, instantaneous  
295 rotor speed in RPM and nacelle orientation in the order of their significance.



296  
297 **Fig. 10** – Feature importance derivate from ETC.

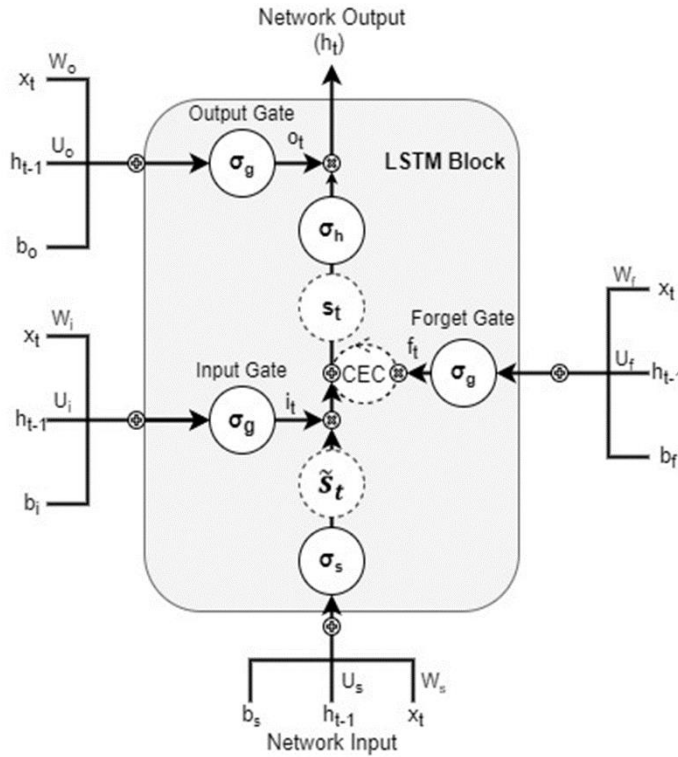
#### 298 **4. Deep learning configuration**

299 Whilst vanilla RNNs proved to be an advance from traditional ANNs, given their inherent dynamic memory, they still  
300 suffered a significant drawback from the unregulated backpropagation of error signals leading to vanishing or exploding  
301 gradients. GRNN solved this problem by using gating mechanisms which regulate the flow of information between layers and  
302 thus track long-term dependencies [33]. This characteristic is key to the wind power application given the high volatility of  
303 wind and the set of underlying physical factors, which influence its variance at different frequency ranges [34]. In this study,  
304 GRNN, in particular GRU and LSTM, is used and critically compared in wind power forecasting, using historical wind turbine  
305 data.

##### 306 *4.1 Long-Short Term Memory (LSTM)*

307 LSTM is built based on memory cells, which contains a recurrently self-connected linear unit, referred to as the Constant  
308 Error Carousel (CEC). CECs resolve the vanishing/exploding gradient problem as their local error back flow remains constant  
309 until the cell is exposed to new inputs or error signals. By introducing input and output gates, the CEC is protected from both  
310 forward flowing activation and backwards flowing error. Besides, a third forget gate is used to control the amount of  
311 information to forget from the historical data [20]. A typical structure of the LSTM unit is presented in **Fig. 11**. In practice,

312 LSTM [35] is capable of learning and remembering long-term dependencies, which makes it suitable for time-series forecasting  
 313 with long input sequences [36].



314  
 315 **Fig. 11 – LSTM Unit Structure.**

316 **Eq. (2) ~ Eq. (7)** summarized the computational process for any individual activation of the LSTM cell:  
 317 In **Eq. (2) ~ (4)**, input, forget and output gate activation vectors of  $i_t$ ,  $f_t$  and  $o_t$  were calculated through the assigned  
 318 weights of  $W_f$ ,  $W_i$ ,  $W_o$ ,  $U_f$ ,  $U_i$ ,  $U_o$  and the bias of  $b_f$ ,  $b_i$ ,  $b_o$  along with corresponding activation functions  $\sigma_l$ . Additionally,  $x_t$   
 319 is the input of neuron at time step  $t$  and  $h_{t-1}$  is the cell state vector for time step  $t - 1$ .

$$f_t = \sigma_l(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma_l(W_i x_t + U_i h_{t-1} + b_i) \quad (3)$$

$$o_t = \sigma_l(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

320 In **Eq. (5)**, the newly assessed value of state  $\tilde{s}_t$  is calculated in a similar mothed along with corresponding activation  
 321 functions  $\sigma_s$ .

$$\tilde{s}_t = \sigma_s(W_s x_t + U_s h_{t-1} + b_s) \quad (5)$$

322 In **Eq. (6)**, the cell state  $s_t$  is obtained from the previous cell state  $s_{t-1}$  and the newly assessed value of state  $\tilde{s}_t$ .

$$s_t = f_t \circ s_{t-1} + i_t \circ \tilde{s}_t \quad (6)$$



323 In **Eq. (7)**, the overall output  $h_t$  is generated from the Hadamard product ( $\odot$ ) of the output gate control signal  $o_t$  and the  
 324 cell state  $s_t$  of the LSTM unit across the activation function  $\sigma_{lh}$ .

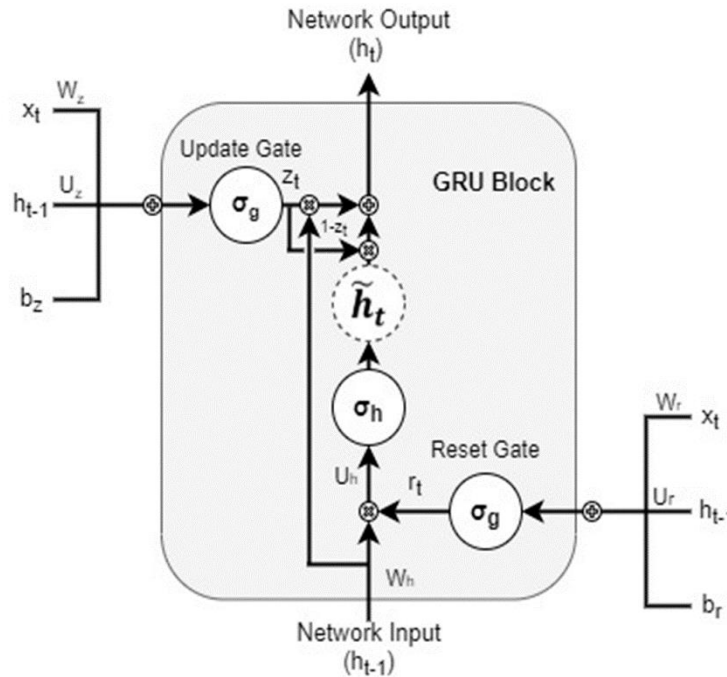
$$h_t = o_t \odot \sigma_{lh}(s_t) \quad (7)$$

325 Based on the above dependencies, the described functions can be deduced for input, forget and output gates [36] as:

- 326
- 327 ▪ Input gate ( $i_t$ ) controls the extent to which  $\tilde{s}_t$  (i.e. estimate of new cell state value) flows into the memory;
  - 328 ▪ Forget gate ( $f_t$ ) controls the extent to which  $s_{t-1}$  (i.e. previous state) is kept in the memory;
  - 329 ▪ Output gate ( $o_t$ ) controls the extent to which  $s_t$  (i.e. current state) contributes to the output ( $h_t$ ).

330 **4.2 Gated Recurrent Unit (GRU)**

331 GRU was firstly proposed by Cho et al. [19] as a more compact and simpler to implement hidden unit inspired by the  
 332 LSTM unit. GRUs [37] contain a reset and an update gate, which adaptively control how much each hidden unit remembers or  
 333 forgets during training without having separate memory cells. This means each hidden unit is able to adaptively capture  
 334 dependencies over different time scales, depending on the activity frequency of its gating mechanisms. For example, short-  
 335 term dependencies will be captured via frequent reset gate activity and long-term dependencies via frequent update gate activity  
 336 [19,38]. A classical structure of the GRU unit is presented in **Fig. 12**.



337  
 338 **Fig. 12 – GRU Unit Structure.**

339 **Eq. (8) ~ Eq. (11)** showed the governing equations of a GRU unit:

340 In **Eq. (8)** and **(9)**, the update gate  $z_t$  and the reset gate  $r_t$  were computed from the assigned weights of  $W_z, W_r, U_z, U_r$  and  
 341 the bias of  $b_z, b_r$  along with corresponding activation functions  $\sigma_g$ . In addition,  $x_t$  is the input of neuron at time step  $t$  and  $h_{t-1}$   
 342 is the cell state vector for time step  $t - 1$ .

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \quad (8)$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \quad (9)$$

343 Then, the obtained reset gate  $r_t$  is used to initiate a new memory content  $\tilde{h}_t$  in **Eq. (10)**. The Hadamard (elementwise)  
 344 product is calculated between  $U_h h_{t-1}$  and the reset gate  $r_t$ , which is operated to determine what information to eliminate from  
 345 previous time steps. Afterwards, the activity function of  $\sigma_{gh}$  is applied to produce the new cell state vector  $\tilde{h}_t$ .

$$\tilde{h}_t = \sigma_{gh}(W_h x_t + (r_t \circ U_h h_{t-1}) + b_h) \quad (10)$$

346 To end, the current cell state vector  $h_t$  is obtained through passing down the hold information to the next unit. To do so,  
 347 the update gate ( $z_t$ ) is involved in **Eq. (11)**:

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \quad (11)$$

348 The above relationships outline the exact nature of the operation for the two gates in GRU [36]:

- 349 ▪ Update gate ( $z_t$ ) controls how much of the previous hidden state  $h_{t-1}$  will be carried over to the current hidden state  
 350 (i.e. how much of the previous hidden state and output candidate of the current hidden state is to be used to calculate  
 351 the output  $h_t$ );
- 352 ▪ Reset gate ( $r_t$ ) controls how much of the previous hidden state  $h_{t-1}$  is to be used to compute the output candidate ( $\tilde{h}_t$ ).

### 353 4.3 Deep learning optimization

354 Model selection and optimization play a pivotal role in the design and implementation of any neural network given their  
 355 direct impact on the overall performance of predictive models. The evolution of deep learning neural networks has greatly  
 356 improved the overall accuracy of implemented models, which in turn increased their complexity. This, however, introduced  
 357 new challenges which arise from the great number of hyperparameters that are required to be optimized to maximize the  
 358 performance and minimize the training time. The key to overall success in this process lies in the trade-off between underfitting  
 359 and overfitting, which can be balanced using the optimal set of hyperparameters for a given dataset and the respective model.

360 In this study, grid search was used to tune hyperparameters to optimize the model performance taking into account both  
 361 GRU and LSTM units. Grid search works by implementing a given estimator and evaluating combinations from a grid of  
 362 parameters based on a user-defined set of metrics when fitting the estimator on a certain dataset. Cross validation is used to  
 363 evaluate and identify the combinations of hyperparameters that perform well across data points in each fold of the dataset. This  
 364 process aims to find the combination of hyperparameters that perform best on average across all folds, which will then be used  
 365 to train the given model. Furthermore, R2 score was used again to evaluate each hyperparameter combination. In this paper,

366 the type of model, number of hidden layers and neurons in each hidden layer were optimized using manual search conducted  
 367 by testing various network configurations, whereas other hyperparameters were tuned using the GridSearchCV algorithm,  
 368 including batch size, number of epochs, optimizer, activation function and kernel initializer. **Table 2** summarized the  
 369 hyperparameters considered during the grid search optimization.

370 **Table 2** - Hyperparameters optimization through grid search.

<i>Hyperparameter</i>	<i>Grid</i>	<i>Optimization</i>
Batch size	10, 20, 40, 60, 80, 100	20
Number of epochs	5, 10, 15, 20, 25	25
Optimizer	SGD, RMSProp, Adagrad, Adadelata, Adam, Adamax, Nadam	Nadam
Activation function	Sigmoid, tanh, ReLu, softmax, softplus, softsign, hard_sigmoid, linear	Softsign
Kernel initializer	uniform, lecun_uniform, normal, zero, glorot_normal, glorot_uniform, he_normal, he_uniform	he_uniform

371

372 *Batch size and number of epochs*

373 Batch size refers to the size of the data batch introduced to the network before the weights are updated, whereas the number  
 374 of epochs is the number of iterations completed over the entire dataset during training. Both hyperparameters have a significant  
 375 impact on the overall computational cost as well as the ability of the network to generalize well across unseen data domains.  
 376 Intuitively, the ideal scenario is to train the model using the smallest possible batch size and for as many iterations as long as  
 377 the model does not begin to overfit, which can be observed from the increase in testing/validation errors. Through grid search,  
 378 the ideal batch size and number of epochs were identified as 20 and 25, respectively.

379 *Optimizer*

380 The objective of any machine learning algorithm is to use inductive learning to learn general concepts from a training  
 381 dataset, where it is used to predict an output that is as close as possible to the actual output. This is achieved by using optimizers,  
 382 which iteratively update weight parameters (represented by  $W$  and  $U$  in **Eq. (2) ~ Eq. (5)** and **Eq. (8) ~ Eq. (10)**). It is used to  
 383 minimize the loss function, which represents the difference between predicted and actual values. Through grid search, Nesterov-  
 384 accelerated Adaptive Moment Estimation (Nadam) was identified as the ideal optimizer algorithm. Nadam is based on Adaptive  
 385 Moment Estimation (Adam), which is widely used given its computational efficiency, low memory requirement and superior  
 386 performance for a wide range of cases [39]. It differs in its use of Nesterov's Accelerated Gradient (NAG) in conjunction with  
 387 RMSprop (Root Mean Square Propagation) instead of AdaGrad (Adaptive Gradient Algorithm). The superiority of Nadam lies  
 388 in its use of NAG, which is able to achieve advanced step direction, compared to classical momentum by applying the  
 389 momentum vector to parameters before computing the gradient [40]. On the other hand, RMSProp adapts individual learning  
 390 rates based on the average of recent gradients for the weight, which is ideal for non-stationary datasets, such as wind turbine

391 power outputs [41]. In summary, Nadam outperforms other optimizers in the current scenario given that it combines the best  
392 properties of both RMSProp and NAG.

### 393 *Activation function*

394 Activation functions are mathematical functions (represented by  $\sigma(x)$  in **Eq. (2) ~ Eq. (5)**, **Eq. (7)**, and **Eq. (8) ~ Eq. (10)**)  
395 attached to neurons that define its output based on the calculated weighted sum of its inputs and the additional bias. Activation  
396 functions are key components for training and optimizing ANNs as they manipulate and propagate information through gradient  
397 processing, whilst introducing non-linearities. Through the grid search, the optimal activation function was found to be softsign  
398 [42]. Softsign is a non-linear activation function based on quadratic polynomial, which is often considered as an alternative to  
399 the classic hyperbolic tanh function given their similarities. Softsign and its derivative can be expressed as:

$$f(x) = \left( \frac{x}{|x| + 1} \right) \quad (12)$$

$$f'(x) = \left( \frac{f(x)}{(1 + x)^2} \right) \quad (13)$$

400 where  $x$  and  $|x|$  represent the input and its absolute value, respectively. Softsign, similar to tanh, ranges between 1 and -1  
401 and its output is centred at 0, which improve the networks back-propagation capability. Smoother asymptotes resulting from  
402 its polynomial convergence mean softsign does not saturate easily and is able to be trained faster [42].

### 403 *Kernel initializer*

404 In this study, the he\_uniform variance scaling initializer was used to initialize the weights of inter-neural connections based  
405 on its superior performance in the grid search. He\_uniform draws values from a uniform distribution bounded by a limit defined  
406 as:

$$limit = \pm \sqrt{\frac{6}{fan\_in}} \quad (14)$$

407 where  $fan\_in$  denotes the number of input units in the weight tensor [43].

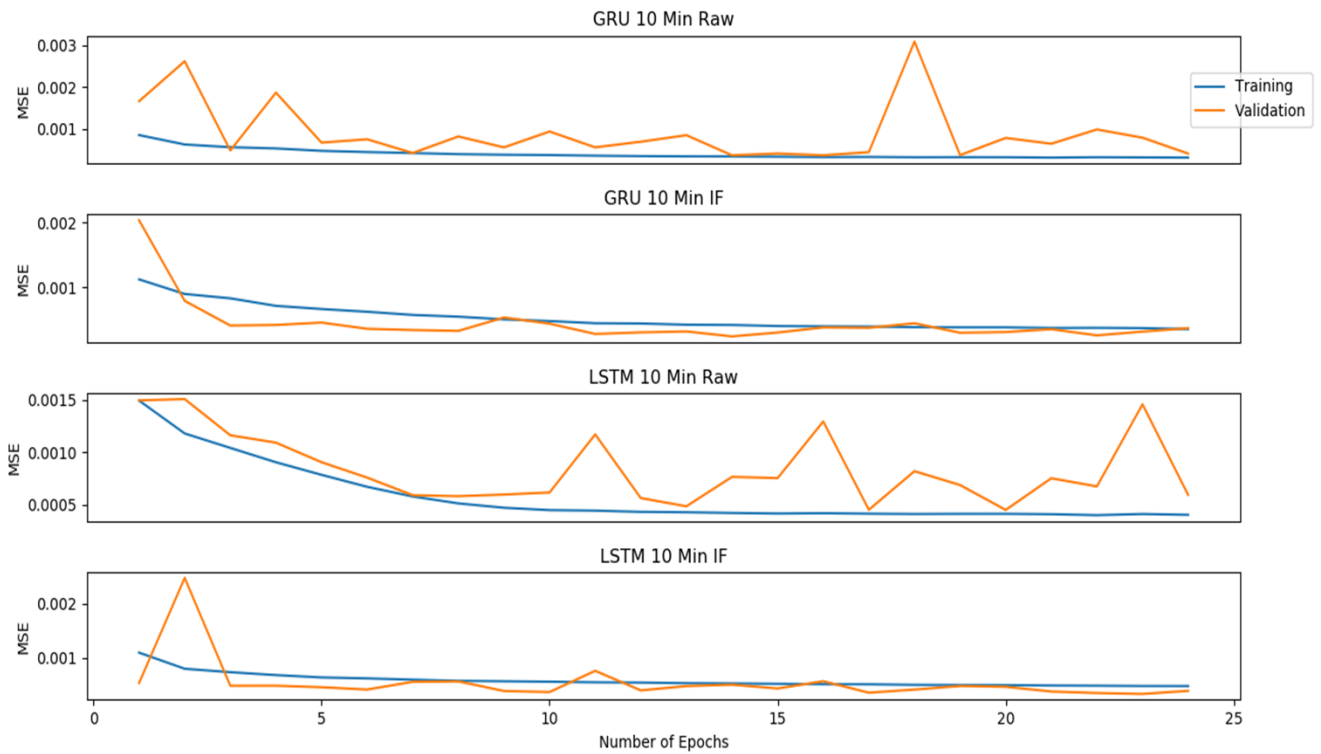
## 408 **5. Results and discussions**

### 409 *5.1 Performance evaluation*

410 This section presents the results and key observations attained from the final output of wind power prediction models  
411 trained using GRU and LSTM. The models were trained using selected input features (hub height wind speed, generator  
412 temperature, gearbox temperature, blade pitch angle, instantaneous rotor speed (RPM), nacelle orientation) and the desired  
413 output (active power). The training phase of the deep learning neural networks was conducted by feeding it with a training  
414 dataset, consisting of both input and output data. Afterwards, the model is presented with testing/validation data based on which

415 it made predictions for the output (active power). The predictive accuracy of the model is evaluated by using the loss function  
416 of Mean Square Error (MSE).

417 Both GRU and LSTM neural networks were hyperparameter tuned using grid search to ensure their optimal performance  
418 and implementation under identical architectures. Both models have been trained and validated using identical training and  
419 testing/validation datasets, which have been subjected to the same methods of sampling and filtering. **Fig. 13** showed the MSE  
420 profiles of the constructed deep learning predictive models along training and validation loops. It suggested that the use of IF  
421 filtering improved and accelerated the convergence of both predictive models, presenting quicker stabilizations of these models.  
422 The deep learning models trained using raw datasets did not converge and stabilize within the designated 25 epoch training  
423 period, implying significant training and validation losses. **Fig. 13** also clearly showed that GRU initializes at lower errors and  
424 later demonstrates quicker and more effective stabilization of losses, which serves as a sign of its robustness. Overall, all filtered  
425 configurations stabilized within 16-17 epochs, indicating that the networks were sufficiently ‘deep’ and optimized to converge  
426 efficiently under relatively short training time.



427  
428 **Fig. 13** – Convergence of training and validation loops in the deep learning models of GRU and LSTM.

429 *5.2 Model benchmarking*

430 **Table 3** showed the summary of modelling accuracies attained through the constructed GRU and LSTM. In terms of  
431 accuracy, it can be seen that GRU outperformed LSTM in each individual test. Their performance was comparable after filtering

432 with the recorded discrepancy in accuracy being 1.32%. With regards to training time, it has been observed that GRU trains on  
 433 average 38% faster compared to LSTM, which is credited to its simpler structure and fewer parameters as mentioned in section  
 434 4.2. The low accuracy of the LSTM model trained using the raw dataset indicates the algorithm’s sensitivity to noise, which  
 435 makes it underperform in wind power forecasting.

436

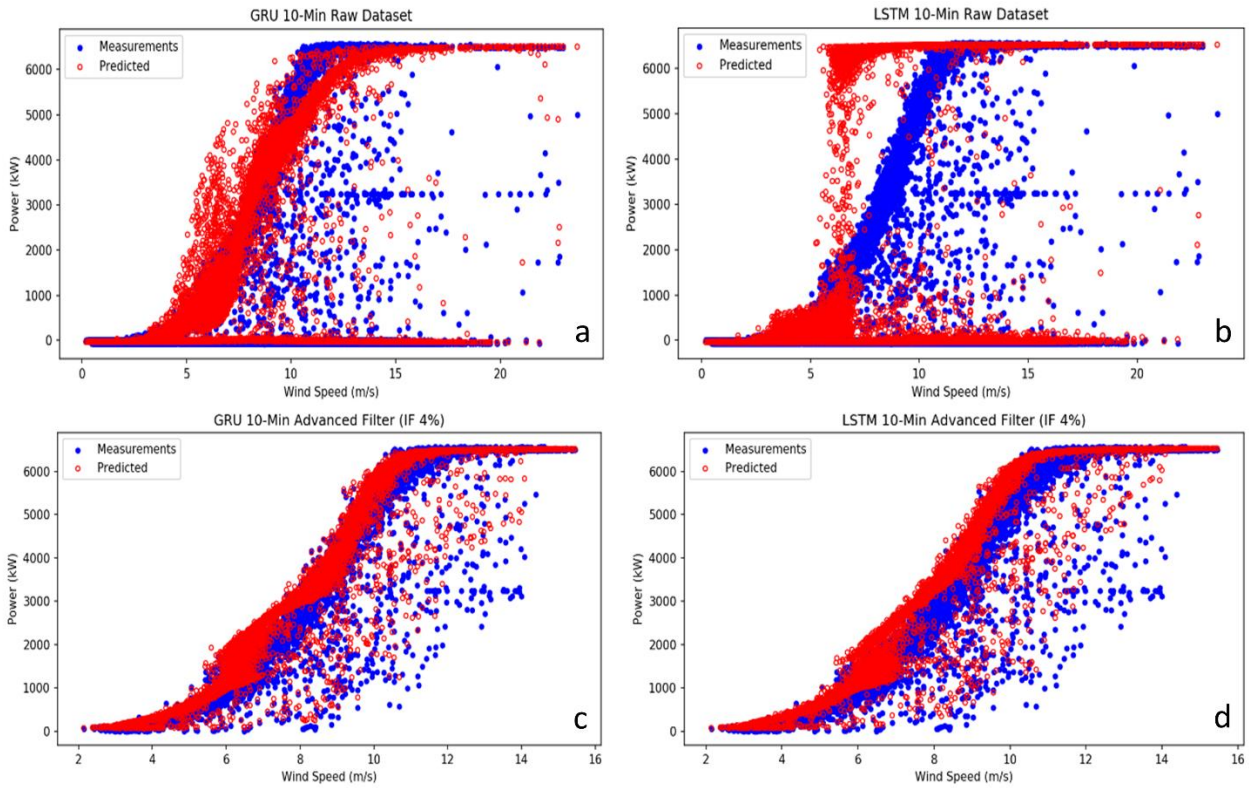
437

**Table 3** - Model performance evaluation of GRU and LSTM.

	Raw dataset		Dataset after Outlier filtering (IF)	
	GRU	LSTM	GRU	LSTM
MSE	0.01014	0.07096	0.003532	0.005272
Accuracy (%)	89.93	73.36	94.06	92.74
Training time (s)	131.29	207.54	96.25	159.48

438

439 **Fig. 14** showed the measured and the predicted wind power curves obtained from each individual GRU and LSTM deep  
 440 learning models. As can be seen, the proposed method of IF filtering is highly effective, as these models predicted the shape of  
 441 wind power curves ((**Fig. 14c** and **Fig. 14d**)) significantly more closely than the raw dataset (**Fig. 14a** and **Fig. 14b**). This  
 442 underlies improvement in the model’s ability to generalize well to unseen data as a result of removing certain noises presented  
 443 in the dataset. Again, GRU provided better adaptability to the sigmoidal shape of the wind power curve, which is advantageous  
 444 to the overall performance of the neural network modelling.



445

446

**Fig. 14** – Comparisons of measured and predicted wind power curves from GRU and LSTM deep learning models.

447

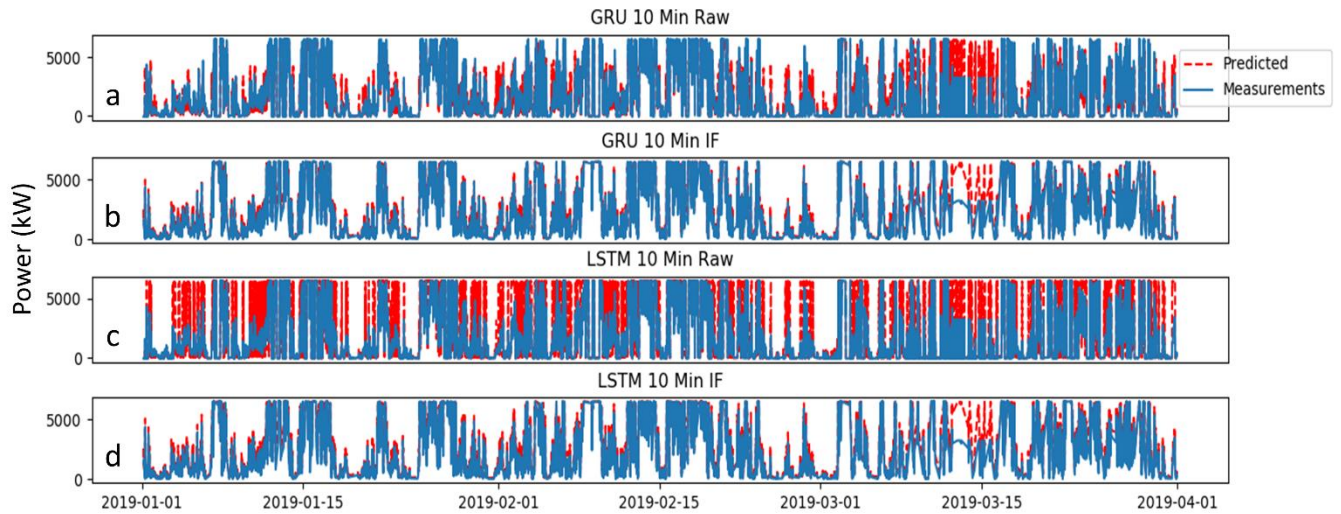
448

As shown in **Fig. 15**, the applied filtering techniques reduced the prediction errors significantly compared to the raw training dataset. The time-series analysis shows that in the raw data scenario, GRU responds better to the high-fluctuating nature of the signal, showing less sensitivity to the noise, compared to LSTM.

449

450

A common source of error in all models occurs around 15<sup>th</sup> March when the power output of the wind turbine is significantly curtailed for operational reasons.



451

452

**Fig. 15** – Comparisons of measured and predicted wind power over January ~ March 2019.

453 The investigations above evaluated several deep-learning-based wind power forecasting models to compare their predictive  
454 accuracy and training time. The use of Recursive Feature Elimination and grid-search-based hyperparameter optimization, both  
455 novelties in the field of offshore wind power prediction, has proven to have direct and positive impact on the performance of  
456 predictive models. The results also shown that the use of filtering techniques is essential to creating accurate wind power  
457 forecasting for offshore wind turbines due to the high-fluctuating and the noisy nature of the SCADA datasets. Both the  
458 accuracy and the training time of predictive models are enhanced significantly through the applications of outlier filters,  
459 reaching relatively high accuracy in all individual test cases.

### 460 *5.3 Summary*

#### 461 *5.3.1 Resampling and outlier detection*

462 The results above indicated that filtering data and removing erroneous measurements are imperative for monitoring and  
463 assessing a wind turbine's performance, as these seriously skewed power outputs. By cleaning outliers and removing anomalous  
464 values, such as negative powers arising from sensor malfunction and null power caused by turbine downtime, the value of the  
465 mean wind power output increased by 1 MW, which is more representative of the actual operational performance. Moreover,  
466 it has been shown that reducing the sampling rate through periodical averaging does filter out some of the noise and better  
467 reveals the shape of the power curve, providing a comprehensive performance assessment, as it prevented the skewed statistical  
468 distribution of the raw datasets.

#### 469 *5.3.2 Qualitative comparison between GRU and LSTM*

470 It is clear that GRU and LSTM share certain key similarities but operate in significantly different ways. Both of them have  
471 an additive characteristic, whereby new content is added on top of historical information from previous activations as opposed  
472 to hidden units found in traditional recurrent neural networks, which always replace the content of its units in the absence of  
473 memory. In this case, the new state is the product of the previous hidden state and the input. The additive characteristic of GRU  
474 and LSTM makes them superior to traditional vanilla RNNs, as it ensures information deemed important (by the forget gate in  
475 the case of LSTM or update gate for GRU) is propagated instead of being replaced and it also creates links across multiple  
476 temporal steps to allow errors to be back-propagated. This, in practice, minimizes the effects of vanishing or exploding gradients  
477 and ensure the tracking of long-term dependencies [38,44].

478 However, arising from their different gating mechanisms, GRU and LSTM have inherently different characteristics in  
479 terms of:

- 480 ▪ Cell State Exposure: LSTM controls the exposure of its cell state and memory content using its output gate, whereas  
481 GRU exposes its entire cell state;



482       ▪ Gate Control: In LSTM, input and forget gates work independently, which means that the amount of new information  
483           added via the input gate is controlled independently from the forget gate. In contrast, GRU controls the amount of  
484           information retained from the previous activation but is not able to independently control the addition of new  
485           information via candidate activation.

486       As discussed by Chung et al. [38] and Bahdanau et al. [45], the superiority of GRU and LSTM over traditional vanilla is  
487       evident. Also, as proven by the results in section 5.1 and 5.2, GRU's simpler cell structure, and subsequently fewer training  
488       parameters, result in shorter training time and the ability to train with fewer samples in wind power forecasting.

## 489 **6. Conclusions**

490       In this study, wind power prediction was explored in-depth by using historical turbine data collected from the target 7 MW  
491       Samsung offshore wind turbine situated in Levenmouth, Fife, Scotland, where a wide breadth of machine learning techniques  
492       was employed to build optimized predictive models using GRU and LSTM deep learning neural networks. This was achieved  
493       in several stages defined by the adopted methodology, which involved pre-processing raw database to ensure high-quality  
494       datasets, applying IF filter to minimize the number of erroneous measurements and identifying the optimal subset of features  
495       to best represent the underlying concept of the used datasets. To maximize performance, both GRU and LSTM deep learning  
496       models were hyperparameter tuned via a combination of manual and grid search. In this paper, the developed wind power  
497       forecasting approach is independent of turbine properties, and therefore can be applied for any types of wind turbine or wind  
498       farms. To sum up, the following conclusions have been reached:

499       ▪ Before input features were used for training in GRU and LSTM deep learning models, advanced data filtering  
500           algorithm IF was applied to input features of the current study. When training with filtered data, deep learning  
501           predictive models have an outstanding performance in wind power forecasting. IF filtering enhanced the performance  
502           of both GRU and LSTM in terms of accuracy, achieving over 92% for both cases. When combining with IF, the gated  
503           recurrent deep learning neural network displayed its full advantages.

504       ▪ The adoption of feature dimension reductions resulted in a cut of six features in the selected SCADA datasets, which  
505           have been validated and confirmed by both RFE and ETC. The other six more significant features have been identified  
506           as wind speed at hub height, generator temperature, gearbox temperature, blade pitch angle, instantaneous rotor speed  
507           and nacelle orientation in the order of their significance.

508       ▪ The approach developed in this paper has the advantage of high degree of accuracies while retaining low  
509           computational costs. The proposed GRU deep learning neural network can reach a higher forecasting accuracy and

510 lower training time compared with LSTM. The internal design of GRU offers a simpler cell structure and subsequently  
511 requires fewer training parameters in deep learning models of wind power forecasting. It can be concluded that GRU  
512 outperformed LSTM in predictive accuracy under all observed tests, whilst training 38% faster and showing  
513 robustness as well as less sensitivity to noise in the SCADA datasets.

#### 514 **Acknowledgement**

515 This research was funded by the EPSRC Doctoral Training Partnership (EP/R513222/1). The authors also thank the Offshore  
516 Renewable Energy (ORE) Catapult for provisions of the SCADA database.

#### 517 **References**

- 518 [1] J. Jung, R.P. Broadwater, Current status and future advances for wind speed and power forecasting,  
519 *Renewable and Sustainable Energy Reviews*. 31 (2014) 762–777.  
520 <https://doi.org/10.1016/j.rser.2013.12.054>.
- 521 [2] M. Yin, Z. Yang, Y. Xu, J. Liu, L. Zhou, Y. Zou, Aerodynamic optimization for variable-speed wind turbines  
522 based on wind energy capture efficiency, *Applied Energy*. 221 (2018) 508–521.  
523 <https://doi.org/10.1016/j.apenergy.2018.03.078>.
- 524 [3] C.M. Chan, H.L. Bai, D.Q. He, Blade shape optimization of the Savonius wind turbine using a genetic  
525 algorithm, *Applied Energy*. 213 (2018) 148–157. <https://doi.org/10.1016/j.apenergy.2018.01.029>.
- 526 [4] L.C. Pagnini, M. Burlando, M.P. Repetto, Experimental power curve of small-size wind turbines in turbulent  
527 urban environment, *Applied Energy*. 154 (2015) 112–121. <https://doi.org/10.1016/j.apenergy.2015.04.117>.
- 528 [5] X. Gao, H. Yang, L. Lu, Optimization of wind turbine layout position in a wind farm using a newly-  
529 developed two-dimensional wake model, *Applied Energy*. 174 (2016) 192–200.  
530 <https://doi.org/10.1016/j.apenergy.2016.04.098>.
- 531 [6] J.H. Kim, W.B. Powell, Optimal energy commitments with storage and intermittent supply, *Operations*  
532 *Research*. 59 (2011) 1347–1360. <https://doi.org/10.1287/opre.1110.0971>.
- 533 [7] C.Q.G. Munoz, F.P.G. Marquez, B. Lev, A. Arcos, New pipe notch detection and location method for short  
534 distances employing ultrasonic guided waves, *Acta Acustica United with Acustica*. 103 (2017) 772–781.  
535 <https://doi.org/10.3813/AAA.919106>.
- 536 [8] Q. Schiermeier, And now for the energy forecast: Germany works to predict wind and solar power  
537 generation, *Nature*. 535 (2016).
- 538 [9] S. Hanifi, X. Liu, Z. Lin, S. Lotfian, A Critical Review of Wind Power Forecasting Methods—Past, Present  
539 and Future, *Energies*. 13 (2020) 3764. <https://doi.org/10.3390/en13153764>.
- 540 [10] C. Li, S. Lin, F. Xu, D. Liu, J. Liu, Short-term wind power prediction based on data mining technology and  
541 improved support vector machine method: A case study in Northwest China, *Journal of Cleaner*  
542 *Production*. 205 (2018) 909–922. <https://doi.org/10.1016/j.jclepro.2018.09.143>.
- 543 [11] M. Lange, U. Focken, *Physical Approach to Short-Term Wind Power Prediction*, Springer, 2006.
- 544 [12] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, G. Nenadic, Machine  
545 learning methods for wind turbine condition monitoring: A review, *Renewable Energy*. 133 (2019) 620–  
546 635. <https://doi.org/10.1016/j.renene.2018.10.047>.
- 547 [13] J.M. Pinar Pérez, F.P. García Márquez, A. Tobias, M. Papaelias, Wind turbine reliability analysis,  
548 *Renewable and Sustainable Energy Reviews*. 23 (2013) 463–472.  
549 <https://doi.org/10.1016/j.rser.2013.03.018>.
- 550 [14] A.P. Marugán, F.P. García Márquez, J.M. Pinar Pérez, Optimal maintenance management of offshore  
551 wind farms, *Energies*. 9 (2016) 1–20. <https://doi.org/10.3390/en9010046>.
- 552 [15] A. Pliego Marugán, F.P. García Márquez, B. Lev, Optimal decision-making via binary decision diagrams

- 553 for investments under a risky environment, *International Journal of Production Research*. 55 (2017) 5271–  
554 5286. <https://doi.org/10.1080/00207543.2017.1308570>.
- 555 [16] A.M. Foley, P.G. Leahy, A. Marvuglia, E.J. McKeogh, Current methods and advances in forecasting of  
556 wind power generation, *Renewable Energy*. 37 (2012) 1–8. <https://doi.org/10.1016/j.renene.2011.05.033>.
- 557 [17] Z. Lin, X. Liu, L. Lao, H. Liu, Prediction of two-phase flow patterns in upward inclined pipes via deep  
558 learning, *Energy*. 210 (2020) 118541. <https://doi.org/10.1016/j.energy.2020.118541>.
- 559 [18] S.R. Moreno, R.G. da Silva, V.C. Mariani, L. dos S. Coelho, Multi-step wind speed forecasting based on  
560 hybrid multi-stage decomposition model and long short-term memory neural network, *Energy Conversion  
561 and Management*. 213 (2020) 112869. <https://doi.org/https://doi.org/10.1016/j.enconman.2020.112869>.
- 562 [19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning  
563 phrase representations using RNN encoder-decoder for statistical machine translation, *EMNLP 2014 -  
564 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference.  
565 (2014) 1724–1734*. <https://doi.org/10.3115/v1/d14-1179>.
- 566 [20] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation*. 9 (1997).  
567 <https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735>.
- 568 [21] Z. Lin, X. Liu, M. Collu, Wind power prediction based on high-frequency SCADA data along with isolation  
569 forest and deep learning neural networks, *International Journal of Electrical Power and Energy Systems*.  
570 118 (2020) 105835. <https://doi.org/10.1016/j.ijepes.2020.105835>.
- 571 [22] J. Serret, C. Rodriguez, T. Tezdogan, T. Stratford, P. Thies, Code comparison of a NREL-fast model of the  
572 levenmouth wind turbine with the GH bladed commissioning results, in: *Proceedings of the International  
573 Conference on Offshore Mechanics and Arctic Engineering - OMAE, 2018*.  
574 <https://doi.org/10.1115/OMAE2018-77495>.
- 575 [23] L. Ziegler, E. Gonzalez, T. Rubert, U. Smolka, J.J. Melero, Lifetime extension of onshore wind turbines : A  
576 review covering Germany, Spain, Denmark, and the UK, *Renewable and Sustainable Energy Reviews*. 82  
577 (2018) 1261–1271. <https://doi.org/10.1016/j.rser.2017.09.100>.
- 578 [24] E. Roslan, H. Mohamed, L. Chan, M.R. Isa, Effect of averaging period on wind resource assessment for  
579 wind turbine installation project at UNITEN, in: *AIP Conference Proceedings, 2018*.  
580 <https://doi.org/10.1063/1.5066898>.
- 581 [25] M. Jafarian, A.M. Ranjbar, Fuzzy modeling techniques and artificial neural networks to estimate annual  
582 energy output of a wind turbine, *Renewable Energy*. 86 (2010) 014501.  
583 <https://doi.org/10.1016/j.renene.2015.08.039>.
- 584 [26] A.E. Saleh, M.S. Moustafa, K.M. Abo-Al-Ez, A.A. Abdullah, A hybrid neuro-fuzzy power prediction system  
585 for wind energy generation, *International Journal of Electrical Power and Energy Systems*. 74 (2016) 384–  
586 395. <https://doi.org/10.1016/j.ijepes.2015.07.039>.
- 587 [27] Y. Zhu, C. Zhu, C. Song, Y. Li, X. Chen, B. Yong, Improvement of reliability and wind power generation  
588 based on wind turbine real-time condition assessment, *International Journal of Electrical Power and  
589 Energy Systems*. 113 (2019) 344–354. <https://doi.org/10.1016/j.ijepes.2019.05.027>.
- 590 [28] A. Lahouar, J. Ben Hadj Slama, Hour-ahead wind power forecast based on random forests, *Renewable  
591 Energy*. 109 (2017) 529–541. <https://doi.org/10.1016/j.renene.2017.03.064>.
- 592 [29] T. Yuan, Z. Sun, S. Ma, Gearbox fault prediction of wind turbines based on a stacking model and change-  
593 point detection, *Energies*. 12 (2019). <https://doi.org/10.3390/en12224224>.
- 594 [30] Z. Lin, X. Liu, Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data  
595 and deep learning neural network, *Energy*. (2020).  
596 <https://doi.org/https://doi.org/10.1016/j.energy.2020.117693>.
- 597 [31] P.M. Granitto, C. Furlanello, F. Biasioli, F. Gasperi, Recursive feature elimination with random forest for  
598 PTR-MS analysis of agroindustrial products, *Chemometrics and Intelligent Laboratory Systems*. 83 (2006)  
599 83–90. <https://doi.org/10.1016/j.chemolab.2006.01.007>.
- 600 [32] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Machine Learning*. 63 (2006) 3–42.  
601 <https://doi.org/10.1007/s10994-006-6226-1>.

- 602 [33] Y. Jung, J. Jung, B. Kim, S.U. Han, Long short-term memory recurrent neural network for modeling  
603 temporal patterns in long-term power forecasting for solar PV facilities: Case study of South Korea, *Journal*  
604 *of Cleaner Production*. 250 (2020) 119476. <https://doi.org/10.1016/j.jclepro.2019.119476>.
- 605 [34] I.V.D. Hoven, Power Spectrum Of Horizontal Wind Speed In The Frequency Range From 0.0007 To 900  
606 Cycles Per Hour, *Journal of Meteorology*. 14 (1957). [https://doi.org/http://doi.org/10.1175/1520-](https://doi.org/http://doi.org/10.1175/1520-0469(1957)014<0160:psohws>2.0.co;2)  
607 [0469\(1957\)014<0160:psohws>2.0.co;2](https://doi.org/http://doi.org/10.1175/1520-0469(1957)014<0160:psohws>2.0.co;2).
- 608 [35] G. Memarzadeh, F. Keynia, A new short-term wind speed forecasting method based on fine-tuned LSTM  
609 neural network and optimal input sets, *Energy Conversion and Management*. 213 (2020) 112824.  
610 <https://doi.org/10.1016/j.enconman.2020.112824>.
- 611 [36] C.A. Martín, J.M. Torres, R.M. Aguilar, S. Diaz, Using deep learning to predict sentiments: Case study in  
612 tourism, *Complexity*. 2018 (2018). <https://doi.org/10.1155/2018/7408431>.
- 613 [37] Z. Peng, S. Peng, L. Fu, B. Lu, J. Tang, K. Wang, W. Li, A novel deep learning ensemble model with data  
614 denoising for short-term wind speed forecasting, *Energy Conversion and Management*. 207 (2020)  
615 112524. <https://doi.org/10.1016/j.enconman.2020.112524>.
- 616 [38] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on  
617 Sequence Modeling, *ArXiv*. (2014) 1–9. <http://arxiv.org/abs/1412.3555>.
- 618 [39] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, 3rd International Conference on  
619 Learning Representations, ICLR 2015 - Conference Track Proceedings. (2015) 1–15.
- 620 [40] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep  
621 learning, in: *International Conference on Machine Learning*, 2013: pp. 1139–1147.
- 622 [41] S. Ruder, An overview of gradient descent optimization algorithms, *ArXiv*. (2016) 1–14.  
623 <http://arxiv.org/abs/1609.04747>.
- 624 [42] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation Functions: Comparison of trends in Practice  
625 and Research for Deep Learning, *ArXiv*. (2018) 1–20. <http://arxiv.org/abs/1811.03378>.
- 626 [43] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on  
627 imagenet classification, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015:  
628 pp. 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>.
- 629 [44] Y. Bengio, P. Simard, P. Frasconi, Learning Long-Term Dependencies with Gradient Descent is Difficult,  
630 *IEEE Transactions on Neural Networks*. 5 (1994) 157–166. <https://doi.org/10.1109/72.279181>.
- 631 [45] D. Bahdanau, K.H. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in:  
632 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings,  
633 2015: pp. 1–15.
- 634