IoT-Based Data Driven Prediction of Offshore Wind Power in a Short-Term Interval Span

Muhammad Khalid[1](✉), Mir Bilal Khan[2], Imam Dad[3], and Shayhaq Fateh[4]

[1] Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany
`Khalid.csd.uob@gmail.com`
[2] University of Hertfordshire Collage Lane, Hatfield AL10 9AB, UK
[3] University of Balochistan, Saryab Road, Quetta 87300, Pakistan
[4] University of Gwadar Balochistan, Gwadar, Pakistan

Abstract. Wind energy is becoming one of the most important suppliers of renewable energy but due to its reliance on weather conditions it is highly inconsistent and its integration into electricity grids is a challenge. In this research we present a comparative analysis of the performance of several prominent data mining techniques in prediction of wind energy generation. Data from the Big Data Challenge Bremen 2018 was used for short term forecasting. Of basic models, a decision tree produced the best performing model. It performed marginally better than SGD, OLS, LASSO and Bayesian ridge regression. Whereas, SVM, nearest neighbor and Gaussian NB performed very poorly. A further analysis using ensemble methods was performed where a Gradient Boosting was the best model. Further improvements of the IoT model are performed and limitations of this are discussed in detail.

Keywords: Big data · Wind power · SGD · SVM

1 Introduction

Wind energy is a major contributor towards renewable energy. Europe is a leader in producing wind power, lowering its bids on wind power more and more. This decrease comes from installing offshore wind parks, which can produce more with lower costs. The McKinsey institute is calling offshore wind even “the next big thing”. In their research it is stated that European countries offer record low bids starting with The Netherlands who offered 54.50 euro per megawatt-hour, which was then lowered by the Danish by 49.90 euro per megawatt-hour. The bid then further decreased, beyond expectations, in the 2017 German action (de Pee et al. 2017). Other parts of the world have also noticed this and have made plans accordingly, such as the five-year offshore plan of China. Similar plans are made by Korea, Poland, Taiwan and the United States (de Pee et al. 2017).

Although offshore wind energy has many advantages, such as it can be located near densely populated coastlines, it has its challenges too. Besides being reliable on many

W. Hussain and M. A. Jan (Eds.): IoTaaS 2021, LNICST 421, pp. 233–243, 2022.
https://doi.org/10.1007/978-3-030-95987-6_17

factors outside the industry, like interest rates and steel prices, the most important factor is stabilizing the grid. Wind energy is very irregular since it relies heavily on weather conditions, therefore integration into the electricity grid becomes a challenge (de Pee et al. 2017).

Research in forecasting offshore wind energy is a relative new area but with the ever-increasing use of renewable energy it is becoming more important. Several different methods have been used for forecasting wind power in wind parks. In the past mostly physics-based modeling has been used to understand the nature of forecasting wind power. However, since many data is being produced about the turbines and weather conditions, a data driven approach could predict wind energy more precisely. Research is done in different time frames using different models. Models such as ARMA (Hussain et al. 2021) make no use of data that could contribute to the wind power but focus on a moving average, such a model was used in (Zhao et al. 2016). A common approach however is using meteorological data with a time series. In (Soman et al. 2010) a numeric weather prediction (NWP) model is compared to statistical approaches, artificial neural network (ANN) and also a combination of these. The results show that different methods work better in different time frames and therefore one should test several methods on data to find a specific fit for the specific time resolution (Hussain et al. 2015; Hussain et al. 2018; Hussain and Sohaib 2019). Landberg (Landberg 1999) used a different meterological model as NWP, a high resolution limited area model together with a wind atlas analysis and application approach, to predict up to 36 h ahead. Although these models make good predictions, neural networks outperform these in general. Data driven research using neural networks has therefore become very popular, for example in (Mohandes et al. 1998) and long term prediction in (Barbounis et al. 2006). Other popular models include fuzzy models that have proven to be rather useful in very short-term prediction. Damousis et al. (Damousis et al. 2004) use an improved fuzzy model for prediction between 0.5 and 2 h ahead. An example of more complex models is by Zhao et al. (Zhao et al. 2016) a extreme learning machine was used together with backward time series analysis to forecast ultra-short-term wind power of a time frame of up to 6 h. In some of the recent approaches Hussain et al. (Hussain et al. 2021; Hussain et al. 2021; Hussain et al. 2022) proposed a new prediction model by introducing ordered weighted averaging (OWA) operator in prediction model (Hussain et al. 2021a; Hussain et al. 2021b). The proposed approach has the capability to handle complex nonlinear prediction (Hussain et al. 2022).

All the models discussed are considered to be rather complex. This is also a representation of the research available today. However, using more simple models would increase insights that could be taken from the analysis. Also, research done on specifically offshore wind parks is sparse.

In this research data from a offshore wind park is used in order to predict short term wind energy. This research is done by competing in the competition “Big Data Challenge Bremen 2018” made available by the University of Bremen (Ciodaro et al. 2019). This challenge lasted the entire month of March 2018 and one could hand in proposed solutions 15 times. In the following sections the data and model will be described, then the results will be presented followed by a extensive discussion.

2 Model and Data

In this section the data used will be described as well as the preprocessing steps and the set up for training and finding the best model. All tasks were conducted using the Python 3 environment and making use of its data mining packages numpy, pandas, and Scikit learn.

2.1 Description of Dataset

The data used for predicting wind power was obtained from the Big Data Challenge Bremen (Ciodaro et al. 2019). This data included a training set with the power included and a testing set without the power. The goal was therefore to predict the wind power of the test set. The dimensions of the training data are (52508, 19) and the dimensions of the testing set are (17668, 19), which corresponds roughly to 75% training data and 25% testing data. This data is time series and the training set was 15 minutely data of January 2016 till June 2017, the test set had the remaining months of 2017.

The variables that are provided in the dataset are several wind measurements, installed capacity and an interpolation indicator. The target variable wind power was measured every 15 min and was also expected to be predicted in this time interval. However, the wind measurements were only hourly, and to determine the missing values for every 15 min a linear interpolation was already performed. The variable "Interpolation" distinguishes the "real" values from the interpolated ones, having a 0 for every real value. All provided variables are shown in Table 1 with a description.

Table 1. Features with their description

Feature	Description
Datum	Start date of a 15 min interval (YYYY-MM-DD hh:mm:ss) in local time ME(S)Z
Windgeschwindigkeit 48 M, Windgeschwindigkeit 100 M, Windgeschwindigkeit 152 M	The wind speed at heights: 48 m, 100 m and 152 m (measured in m/s)
Windgeschwindigkeit 100 MP10, Windgeschwindigkeit 100 MP90	Probabilistic wind speeds at 100 m (percentile 10–90) in the 15-min section (measured in m/s)
Interpoliert	The wind measurements are only available for the full hours and were therefore interpolated for the 15-min intervals. This field identifies such interpolated values
Verfügbare Kapazität	The maximum possible power of the wind farm at the current time, measured in kW
Output	The produced wind energy (measured in kWh/h)

2.2 Preprocessing

The data does not require a big amount of preprocessing since there are no missing values and all variables (except interpolation) are numerical. The preprocessing that was performed included feature scaling, making sure that big values are not necessarily more important than small ones. This was necessary because for example the variable of installed capacity has very big values compared to the other variables. Further, since the data was linearly interpolated, a cubic interpolation was performed. The result of using linear interpolation is shown in Fig. 1, left the original data of Windgeschwindigkeit48M and on the right a black-dotted line of the linear interpolation. To make the effect of interpolation clearer, Fig. 2 shows Windgeschwindigkeit48M zoomed in for the first 10 values. The left figure shows the linear interpolation and the right the cubic interpolation. From these figures it is not clear whether cubic interpolation will have a big effect on the prediction of the wind power. The results of using this will be shown in the next section.

Since the date variable as it is, is not usable in a regression, new features were made. From this date variable, the following new features were extracted: hour number (indicating which hour of the day it is), day number (Indicating which day of the year it is), week number (indicating which week of the year it is) and month number (indicating which month of the year it is).

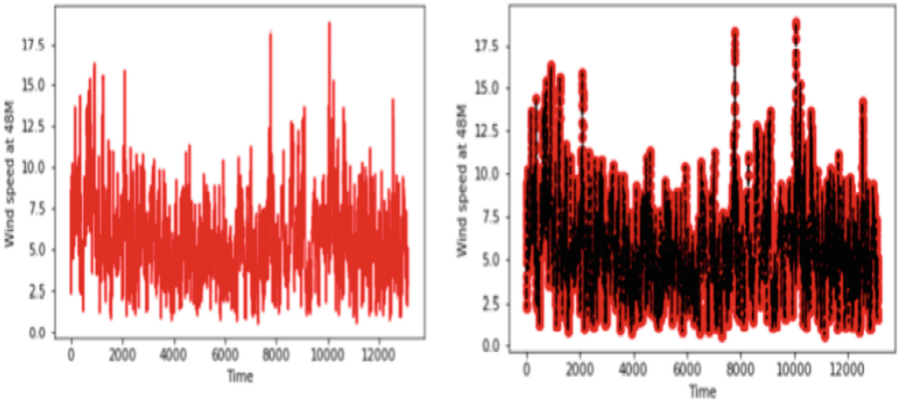

Fig. 1. Original (left) and linear interpolated (right) of Windgeschwindigkeit48M

2.3 Process of Modeling

The next step is using the data to predict the wind power of the test dataset. Since the provided test set does not have the target variable of wind power, a extra test set is created. A new training and test set were obtained by splitting the provided training dataset into a separate training and test set, according to the dimensions of the original split that is 75% training and 25% testing. These test and training sets were made keeping in mind that the data is time series and therefore no random sampling was involved. The method making a new test and training set was used because there was a limited number of submissions

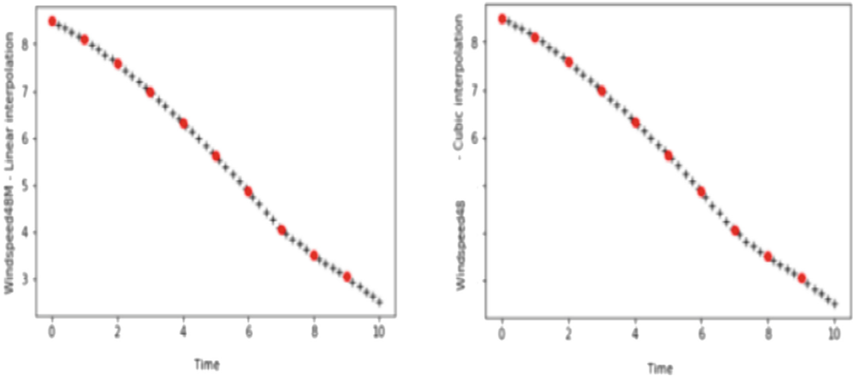

Fig. 2. First ten values of Windgeschwindigkeit48M using linear (left) and cubic (right)

for the competition and with this it was easier to see directly what models were working better. The process of finding the right model was done by simply applying several basic models to the data and then improving the model that turned out to be best.

3 Results

In this section the results of finding the best classifier will be shown. The results of using (and not using) interpolation, outlier removal, different loss functions and tuning parameters will be shown.

3.1 Finding the Model

First of all, the data was applied to a number of "basic" classifiers, in order to have an idea of the data and find what kind of classifier would work best. Table 2 shows these results, where "score" is the score provided by scikit learn and corresponds to R2, where 1 is the best value. From Table 2 it can be seen that a Decision tree works best in this case, it has a score of 0.7768. This means that a Decision tree model explains about 77.7% variability of the data around the mean.

In order to improve this score, the data was applied to several ensemble regressors that are mostly based on decision trees and would therefore improve the result. Table 3 shows that this is indeed the case, all ensemble methods improve the simple decision tree model by roughly 10%. Although the scores lie very close, Gradient Boosting (GBM) is chosen as the best classifier with a score of 0.8730. Meaning that the explained variability of the data increased by 9.62%, from decision tree to GBM.

3.2 Improving the Model

With regard to further improving the model, the parameters need to be tuned. The tuning is done by a so called GridSearchCV using cross validation. Before the tuning starts, it was found that using a loss function of "huber" gives better results than the default

Table 2. Results of applying different basic regressors

Regressor	Score
Decision tree	0.7768
SGD	0.7564
Baysian ridge	0.7525
OLS	0.7439
Lasso	0.7689
SVM	0.0468
Nearest neighbour centroid	0.0002
GaussianNB	0.0026

Table 3. Results of applying different ensemble regressors

Regressor	Score
Random forrest	0.8684
Extra Treesregressor	0.8711
Adaboost	0.8676
Bagging	0.8680
GBM	0.8730

used in scikit learn, which was least squares regression. Using this the score improved by 1.71% and was used in all further improvements.

As mentioned before the tuning was done by using a GridSearchCV. The cross validation in such a grid search was done by TimeSeriesSplit provided by scikit learn. This method is especially for time series data, such that samples are made with regards to the time series. Using a random state, the scores of the model without tuning and with tuning can be seen in Table 4. This table indicates that tuning the parameters increases the model score slightly by almost 1%, also one can observe that the mean increases and the standard deviation decreases indicating that the model improves.

The resulting model from the grid search has the following parameters: maximum depth of 5, minimal samples split of 700, minimal samples leaf of 60, subsample 0.75 and maximum number of features 9. Further decreasing the learning rate and increasing number of features does not improve the model any more. Initially the optimal number of features was 50, but the best result was found to be 100 with a learning rate of 0.05. The exact meaning of all of these parameters will not be discussed here, but can be found in the documentation of Scikit learn.

In order to further improve the results, cubic interpolation was applied to the data. Originally, a linear interpolation was used but this could indicate that not all data was correct. The results of this are also shown in Table 4. The importance of all features

Table 4. Results of tuning parameters, showing R^2 score and cross validation scores for linear and cubic interpolation

Models	R2	CV: Mean	CV: std	CV: Min	CV: Max
No tuning	0.8890	0.8567	0.0155	0.8406	0.8717
With Tuning	0.8899	0.8576	0.0147	0.8429	0.8723
With cubic interpolation					
No tuning	0.8893	0.8563	0.0148	0.8415	0.8711
With tuning	0.8847	0.8562	0.0137	0.8425	0.8687

according to this grid search was also plotted in Fig. 3 for linear interpolation and in Fig. 4 for cubic interpolation. The most interesting from these graphs is that the importance changes slightly with different interpolation, indicating that these models are actually different. Looking however back into the results in Table 4: Although the no-tuning model of cubic interpolation is slightly better than the no-tuning linear interpolation, it shows that linear interpolation is after all still the best option since tuning really improves the model.

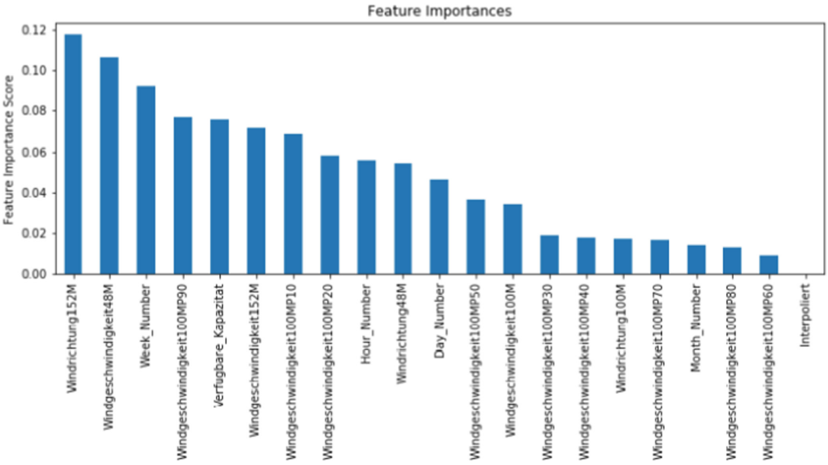

Fig. 3. Feature Importance linear interpolation

In order to eliminate effects of overfitting it was decided to test all of these models by handing them in on the challenge website. The result was that all the tuning and cubic interpolation actually decreased the performance of the model on the challenge data, which was not expected. The final technique applied to improve the score was removing outliers. However, removing outliers did not improve the score and also not the challenge score. Therefore, the final model that gave the best result on the test data for the challenge was a "simple" Gradient Boosting Regressor with Huber loss. In the next section these results will be discussed in more detail.

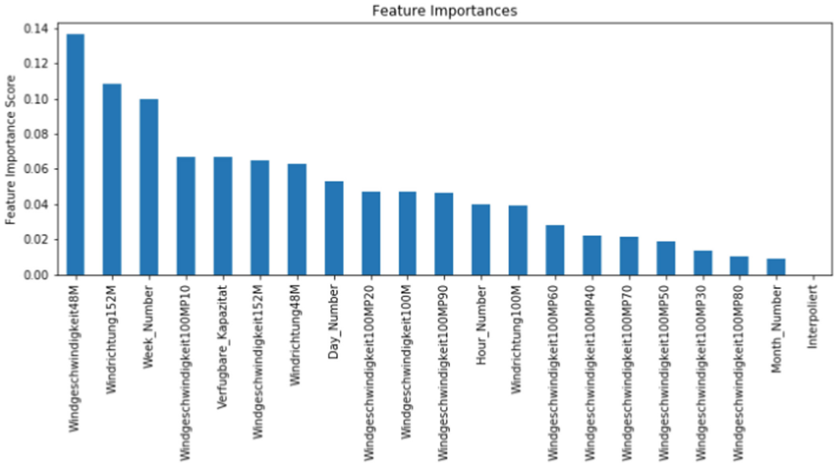

Fig. 4. Feature Importance cubic interpolation

4 Discussion

The idea behind a challenge such as this is not only to arrive at the best results, but to understand why certain methods seem to work significantly better than others. In this context, there were certain interesting observations made in the course of this project. They are as follows:

1. Conventionally, interpolation and outlier removal are considered the important approaches to take on immediately when dealing with numerical data. However, here this seems to be rather tricky. As it is shown above (Fig. 2), the data is generated through interpolation itself and the intervals are too small for further interpolation to make any significant difference. Hence, attempting linear or cubic interpolation did not seem to have any remarkable increase in model performance. Further, outlier removal is usually irrelevant to time series data, as every prediction is dependent on the data points that came before. Therefore, there are no outliers as such, and yet again, another conventionally popular approach seemed to be quite useless. The idea behind attempting outlier removal, however, was to prevent overfitting.
2. Another interesting observation has been that using Huber loss function instead of squared error loss seems to improve the model. A possible explanation for this points to the first observation again; due to lack of outliers Huber loss function works very effectively in comparison to a mean- squared loss.
3. The observation that may take one for surprise is the varied behavior of tuning parameters. In the model devised from the training data, tuning seemed to work rather effectively, prompting its use on the test data. However, due to overfitting, it had quite the reverse effect on the test data.
4. The final observation is that when linear and cubic interpolation was applied on the nine final features, (Fig. 3 and Fig. 4), it did seem make a difference considering that it caused certain features to change. However, the difference did not seem to improve model performance much.

The challenges faced in this project seemed to be rather different from conventional challenges that one faces while handling data. Pre-processing data to eliminate irrelevant information and handle missing values is usually a rather tricky and complex part of data analysis. However, the dataset provided for this project could be termed rather'clean'. Moreover, conventional methods seemed to adversely effect the results, which made the modeling rather tricky. Therefore, it must be considered that despite some methods churning out excellent results for some datasets, uninformed use of such methods may prove detrimental to model performance.

5 Conclusion

Due to limited time and a spirit of competition, only a number of methods were used. However, there may be more scope of improvement given that numerical, time-series data has always been very popular in research. Some methods that are used for such datasets have been discussed in this section.

Several classifiers such as Naive Bayes and Support Vector machines have enjoyed a lot of popularity in the field. Classifiers such as Naive Bayes have an advantage here as they may be straightforwardly extended into a semi-supervised learning algorithm, while SVM needs to be altered to transductive SVMs (Joachims 1999) in order to be fed into such an algorithm. There is remarkable research going on in the field of active learning (Hosein 2018) that also allows us to do so. The main idea behind active learning is that is a learning algorithm may choose the data it wants to learn from, it may perform better with conventional methods with substantially less data for training. Multiple stochastic and machine learning prediction methods [13–14] are applied to predict the behavior of the dataset. The approaches alert the service provider in case of any violation to mitigate the risk of violation and to take immediate remedial action.

The data handled in this project is numerical by large. Hence, without delving into assumptions and hypotheses, it may be simply pointed out that Decision Tree and SGD appear to work best for the model, in comparison to other basic regressors. Decision Trees have been considered somewhat reliable, its big advantage being the ability to find odd interactions. To use a simple example to explain this, in the prediction of voting behavior, if income only matters for people who are highly educated, trees can find that. In a linear regression, all interactions are made visible, which may turn out to be quite resource-consuming for large data sets; even with four variables and two-way interactions, there are six possible interactions, all of which are not mostly relevant.

Based on the type of data is provided, conventional validation or cross-validation may be applied. Conventional validation primarily involves partitioning the data set into two sets of seventy percent for training and thirty percent for test. This works with reliable accuracy when the data set is considerably large. However, when the data set in comparably smaller, conventional validation may not be very helpful. In recent studies, cross-validation has been proven to perform far better for data sets with poorly populated values. The basic idea is to not use the entire data set while training the learner, some of the data is removed before the training begins. After the learner has been trained, the data that was'left out' can be used to test the performance of the devised learner. However, it is interesting to note that cross-validation would be rather futile for this study as the dataset is time-series data, which cannot be sampled randomly. This, however, is the broad explanation for a whole class of model evaluation methods called cross-validation (Schneider 1997); the details, however, is beyond the scope of this paper.

An interesting approach to analyzing this data would be to further further interpolate between the intervals and generate more data. Given the time, it would be interesting the observe if the new model would be any different from the given model. The quality of interpolation, however, may be measured from two different viewpoints; prediction and characterization (Caruso 1999). Prediction implies that good-quality interpolation is when prediction error of an unknown point is minimized. Characterization states that the surface that results from good interpolation must, in most ways, resemble the original surface. However, it is not necessarily a trade-off.

Logically, the next approach would be to analyze the time-series data for trends; in this case, seasonal, etc. Autocorrelation and partial autocorrelation seem to be rather popular methods of doing so. Autocorrelation, sometimes known as ‘serial correlation’, is the correlation of a time series with its own past and future values. This helps identify patterns in long term data. The given dataset would be particularly relevant for using such methods as the data is generated in short intervals, over a long period of time.

Another possible approach would be to derive the features using Principal Component Analysis instead of using ‘real’ features. PCA is essentially a dimension-reduction tool that is used to reduce a large set of data to a significantly smaller set, while preserving the valuable information. This method may prove rather efficient when handling ‘big’ data.

Using neural networks and machine learning also seems like obvious ways of improving a given model. Given the dataset, a method that should work efficiently is Multilater Perceptron. An MLP can be viewed as a logistic regression classifier where, at first, the input is transformed using a learned non-linear transformation. This transformation projects the input data into a space where it becomes linearly separable, while this intermediate layer is referred to as a hidden layer. A single hidden layer is sufficient to make MLPs a universal approximator. However, there are substantial benefits to using many such hidden layers, i.e., the very premise of deep learning (Raza et al. 2021); this however, is largely beyond the scope of this paper.

Any, or a combination, of the above-mentioned approaches could further improve the performance of the model. With the increasing use of renewable energy (and therefore the demand of understanding such energies) and the many possibilities to further increase a forecasting model, there is a high motivation and potential for future research in this area.

References

de Pee, A., Küster, F., Schlosser, A.: Winds of change? Why offshore wind might be the next big thing (2017)

Barbounis, T.G., Theocharis, J.B., Alexiadis, M.C., Dokopoulos, P.S.: Long-term wind speed and power forecasting using local recurrent neural network models. IEEE Trans. Energy Convers. **21**, 273–284 (2006). https://doi.org/10.1109/TEC.2005.847954

Caruso, C.: Interpolation methods comparison. Comput. Math. Appl. **35**(12), 109–126 (1999)

Ciodaro, G., Cosin, D., Florent, R.: Project report bremen big data challenge-edition 2019 (2019)

Damousis, I.G., Alexiadis, M.C., Theocharis, J.B., Dokopoulos, P.S.: A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. IEEE Trans. Energy Convers. **19**, 352–361 (2004). https://doi.org/10.1109/TEC.2003.821865

Hussain, W., Alkalbani, A.M., Gao, H.: Forecasting with machine learning techniques. Forecasting **3**(4), 868-869 (2021)

Hussain, W., Gao, H., Raza, M.R., Rabhi, F.A., Merigo, J.M.: Assessing cloud QoS predictions using OWA in neural network methods. Neural Comput. Appl., 1–18 (2022)

Hussain, W., Hussain, F.K., Hussain, O.K.: Towards soft computing approaches for formulating viable service level agreements in cloud. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) ICONIP 2015. LNCS, vol. 9492, pp. 639–646. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26561-2_75

Hussain, W., Hussain, F.K., Saberi, M., Hussain, O.K., Chang, E.: Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs. Futur. Gener. Comput. Syst. **89**, 464–477 (2018)

Hussain, W., Merigo, J.M., Gao, H., Alkalbani, A.M., Rabhi, F.A.: Integrated AHP-IOWA, POWA framework for ideal cloud provider selection and optimum resource management. IEEE Trans. Serv. Comput. (2021a). https://doi.org/10.1109/TSC.2021a.3124885

Hussain, W., Merigó, J. M., Raza, M.R.: Predictive intelligence using ANFIS-induced OWAWA for complex stock market prediction. Int. J. Intell. Syst. (2021b). https://doi.org/10.1002/int.22732

Hussain, W., Merigó, J.M., Raza, M.R., Gao, H.: A new QoS prediction model using hybrid IOWA-ANFIS with fuzzy c-means, subtractive clustering and grid partitioning. Inf. Sci. **584**, 280–300 (2022)

Hussain, W., Sohaib, O.: Analysing cloud QoS prediction approaches and its control parameters: considering overall accuracy and freshness of a dataset. IEEE Access **7**, 82649–82671 (2019). https://doi.org/10.1109/ACCESS.2019.2923706

Schneider, J.: Cross validation (1997)

Landberg, L.: Short-term prediction of the power production from wind farms. J. Wind Eng. Ind. Aerodyn. **80**, 207–220 (1999). https://doi.org/10.1016/S0167-6105(98)00192-5

Mohandes, M.A., Rehman, S., Halawani, T.O.: A neural networks approach for wind speed prediction. Renew. Energy **13**, 345–354 (1998). https://doi.org/10.1016/S0960-1481(98)00001-9

Hosein, S.: Active learning: Curious AI algorithms (2018)

Raza, M.R., Hussain, W., Merigó, J.M.: Cloud sentiment accuracy comparison using RNN, LSTM and GRU. In: 2021 Innovations in Intelligent Systems and Applications Conference (ASYU) (2021)

Soman, S.S., Zareipour, H., Malik, O., Mandal, P.: A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium 2010, Arlington, TX, USA, pp. 1–8. IEEE (2010)

Joachims, T.: Transductive inference for text classification using support vector machines (1999)

Zhao, Y., Ye, L., Li, Z., Song, X., Lang, Y., Su, J.: A novel bidirectional mechanism based on time series model for wind power forecasting. Appl. Energy **177**, 793–803 (2016). https://doi.org/10.1016/j.apenergy.2016.03.096