DESIGN AND IMPLEMENTATION OF A MACHINE LEARNING-BASED WIND TURBINE CONTROL SYSTEM

Prabhavathi K[1*], *P. B. Edwin* Prabhakar[2], *Arunadevi* M [3]and *Shanthi* D[4]

[1]Assistant Professor.Bannari Amman Institute of Technology, Sathyamangalam.638402,India
[2] New Prince Shri Bhavani College Of Engineering and
Technology, Approved by AICTE, Affilated To Anna University ,India
[3]Assistant Professor, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai – 127
[4]Assistant Professor, Prince Dr.K.Vasudevan College of Engineering and Technology, Chennai – 127

Abstract. The Machine Learning-Based Wind Turbine Control System (MLBWTCS) is a new technology that uses machine learning algorithms to optimize the performance of wind turbines. The system collects data from sensors installed on the wind turbine to monitor various variables such as wind speed, blade pitch angle, generator torque, and power output. The data collected is preprocessed and fed into a machine learning model, which predicts the optimal settings for the turbine operations. The predictions are then used to control the operations of the wind turbine in real-time. The MLBWTCS has been shown to improve the efficiency and reliability of wind turbines, resulting in increased power generation and reduced maintenance costs. This paper presents a detailed description of the design and implementation of the MLBWTCS, including data collection, preprocessing, feature selection and machine learning model selection.

Keywords: Machine learning, wind turbine control system, pre-processing, feature selection

1. Introduction

Wind-energy conversion systems consist of a windmill, a gearbox, and a generator, as demonstrated in Figure 1. To establish this nonlinear model, the aerodynamics, tower, drive train, and the generator should be considered [1][14]. In the following subsections, these subsystems are discussed separately. Finally, a nonlinear state-space model is established.

*Correspondingauthor:prabhavathik@bitsathy.ac.in

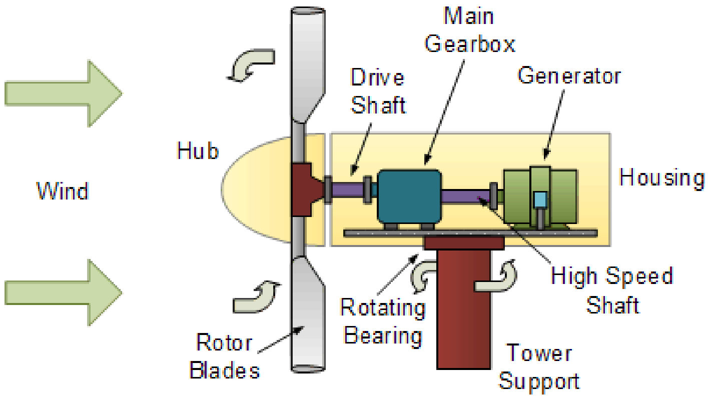

Figure1. Structure of Wind Turbine

A wind turbine extracts the kinetic energy from the wind by slowing the wind down, and transferring this energy into the spinning shaft so it is important to have a good design. The available power in the wind that is available for harvesting depends on both the wind speed and the area that is swept by the rotating turbine blades[2-4][18].

So therefore, the faster the wind speed or the larger the rotor blades the more energy can be extracted from the wind. So research can say that wind turbine power production depends on the interaction between the rotor blades and the wind and it is this interaction that is important for a wind turbine design[5][12][19].

2. Related Work

An adaptive wind turbine power curve model based on k-nearest neighbor (KNN) regression was proposed. The KNN model was trained using historical wind turbine data and used to predict the optimal power output settings for the turbine. The results showed that the KNN model could improve the accuracy of power output predictions compared to traditional models[6-8].

A recent study reviewed the use of machine learning techniques in wind turbine condition monitoring and fault diagnosis[9][11]. The review showed that machine learning algorithms, such as artificial neural networks and support vector machines, could be used to detect and diagnose faults in wind turbines at an early stage.

An intelligent wind turbine control system using a fuzzy-neural network was proposed. The system used historical wind turbine data to train the fuzzy-neural network, which was used to predict the optimal settings for wind turbine operations. The results showed that the system could improve the power output and efficiency of wind turbines[10][15].

An intelligent control system for wind turbines based on neural network and fuzzy inference studied. The system used historical wind turbine data to train the neural network, which was used to predict the optimal settings for wind turbine operations. The fuzzy inference system was used to adjust the settings based on real-time wind conditions[13][16-17]. The results showed that the system could improve the power output and efficiency of wind turbines.

3. Research methodology

The Machine Learning-Based Wind Turbine Control System (MLBWTCS) is a new technology that uses machine learning algorithms to optimize the performance of wind turbines. The design and implementation of a Machine Learning-Based Wind Turbine Control System (MLBWTCS) involves Proposed MLBWTC Several steps:

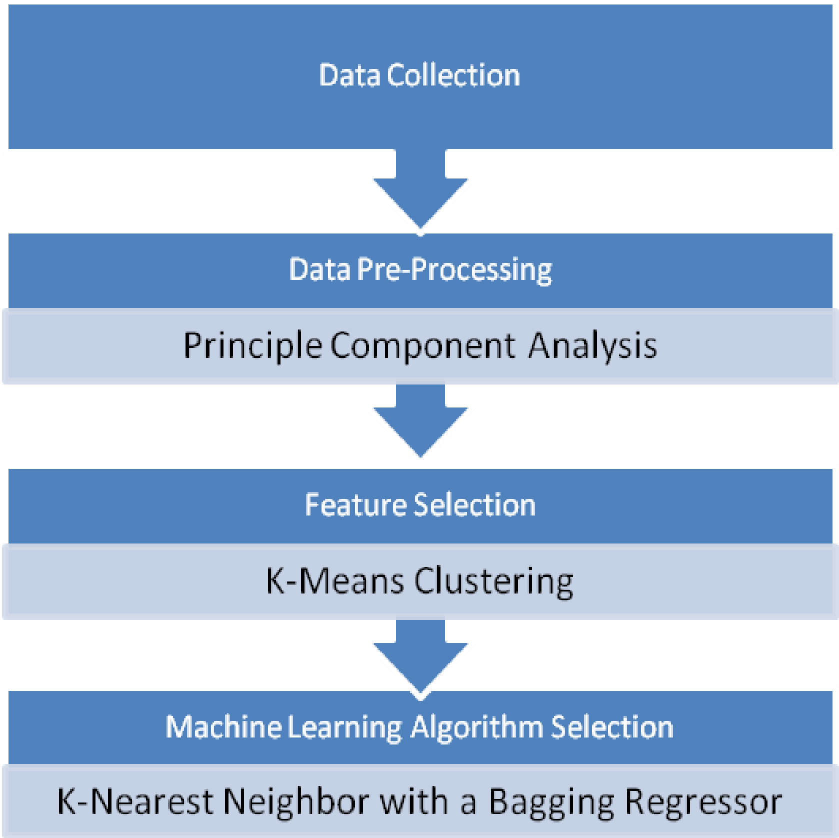

Figure 2.Machine Learning-Based Wind Turbine Control System (MLBWTCS)

Data Collection

The data collected by the sensors should be stored in a secure and organized to facilitate analysis. This may involve storing the data on a server or cloud-based platform.

3.1 Data pre-processing using Principal component analysis (PCA)

Principal component analysis (PCA) is also used in this study to determine the important features. PCA is a mathematical tool used to represent the variation of features in a dataset by using a small number of factors. The 2D or 3D projection of samples is shown by setting the axes (principal components, PCs) as the factors. Principle components are constructed in such a sequence, where the first principle component (PC1) holds on to the largest possible variance in the dataset. The second principle component (PC2) holds the largest possible variance among all the remaining combinations, given that PC2 is not correlated with PC1. The subsequent principle components are designed in a similar way.

3.2 Feature Selection using K-means clustering

K-means clustering can be used for feature selection in Wind Turbine Control System by identifying the most important features that contribute to the performance of the system. The following are some steps for using k-means clustering for feature selection:

1. Identify the features: First, identify the set of features that will be used for the analysis. This may include variables such as wind speed, wind direction, blade pitch angle, generator torque, and power output.

2. Normalize the features: Normalize the features to ensure that they are on the same scale. This is important for ensuring that the clustering algorithm gives equal weight to all features.

3. Determine the number of clusters: Decide on the number of clusters to use for the analysis. This will depend on the specific research question and the complexity of the dataset.

4. Perform k-means clustering: Use k-means clustering to cluster the data points based on the selected features. The algorithm will group the data points into clusters based on their similarity.

5. Evaluate the clustering: Evaluate the quality of the clustering by examining the within-cluster sum of squares (WCSS) and the silhouette score. WCSS measures the sum of the squared distances between each data point and its assigned Centroid. The silhouette score measures the similarity of each data point to its assigned cluster compared to other clusters.

6. Select the features: Finally, select the most important features based on their contribution to the clustering. Features that are highly correlated with the clustering can be selected for use in the Wind Turbine Control System.

Overall, the use of k-means clustering for feature selection can help identify the most important variables that contribute to the performance of the Wind Turbine Control System. This can help improve the accuracy and efficiency of the system by reducing the number of variables that need to be considered.

3.3 K-Nearest Neighbor with a Bagging Regressor

K-Nearest Neighbor (KNN) with a Bagging Regressor is a machine learning approach that can be used for Wind Turbine Control System. The KNN algorithm is a non-parametric method used for regression and classification tasks. It works by finding the k-nearest neighbors of a data point and using their target values to predict the value of the new data point. The Bagging Regressor is an ensemble method that combines multiple KNN models to improve the overall performance and reduce overfitting.

K-nearest neighbor (KNN) is a supervised machine learning model that deduces a function from a training sample dataset. KNN is simple and easy to understand and can be applied to regression and classification problems but it has a major drawback that will be discussed in a later section. Each sample in the dataset has an input vector and a desired output value. After the model is trained using the training dataset, the trained model will be used to determine the output for any given dataset.

Once the distance from the points in the training set has been measured, the model will look for the new point that gives the nearest distance between k nearest points. The value of k will be used to determine the number of points being measured during training. Hence, it is crucial to determine the value of k. A large k will reduce the noise and minimize the prediction loss, but will increase the computational cost and time if a large training dataset is used; however, a small k will simplify the prediction process and reduce computational cost. Hence, the computational time for KNN will become shorter as the size of the dataset grows. Validation error is used to determine the value of k to be used in a KNN regressor.

Next, a bagging tree (BT) regressor is used to improve on the KNN regressor. In a BT regressor, multiple data subsets, Di, are constructed from the training dataset from the KNN regressor by sampling randomly with replacement and without pruning. This is the bootstrap method. These bootstraps will eventually be used to construct a single regression tree. All individual trees are then combined in an ensemble. Hence, this method is also called the tree ensemble method. The predicted outcome will be averaged over all the trees, as shown in Figure 7. Therefore, a BT regressor helped to improve the accuracy of the trained model by reducing the variance or errors.

4. Evaluation Criteria

1. R-squared (R^2) score

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

Where y_i is the observed outcome, $\bar{y}$ is the mean if the observed outcome, $\hat{y}_i$ is the predicted outcome and N is the number of observed outcomes.

No of Datasets	KNN	FNN	Proposed MLBWTCS
100	25	45	75
200	35	55	80
300	45	65	85
400	55	75	95

Table 1.Comparison table for R-squared (R^2) score

The Comparison table 1 of R-squared (R^2) score demonstrates the different values of existing KNN, FNN and proposed MLBWTCS. While comparing the Existing algorithm and proposed MLBWTCS, provides the better results. The existing algorithm values start from 25 to 55, 45 to 75 and proposed MLBWTCS values starts from 75 to 95. The proposed method provides greater results.

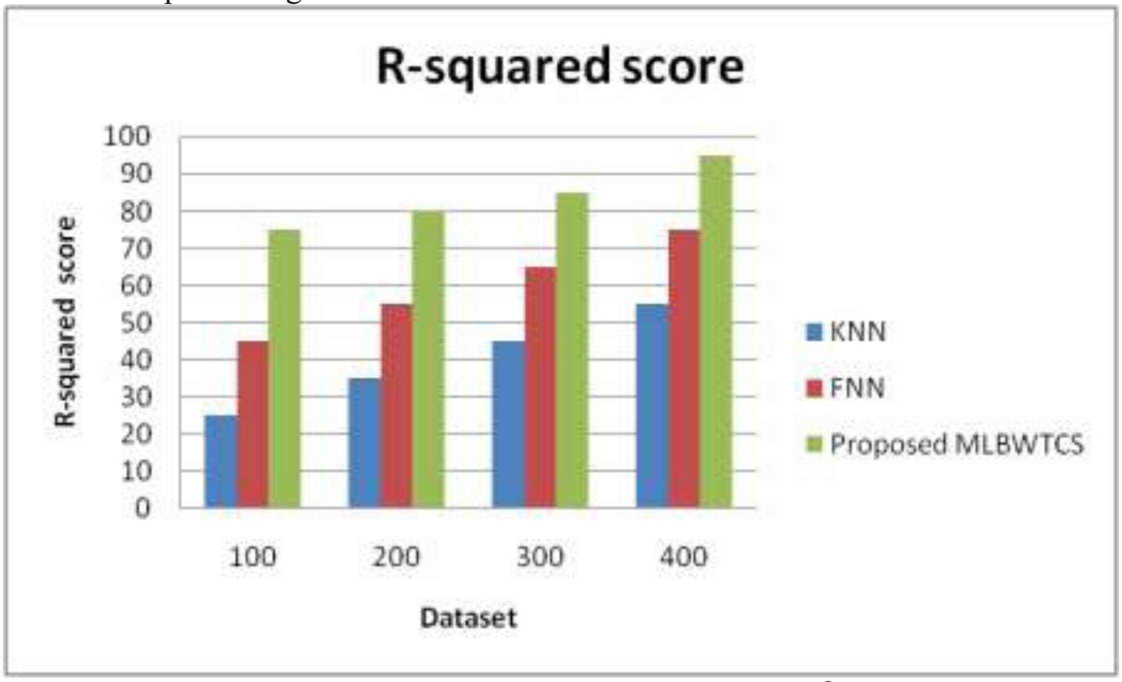

Figure 3.Comparison chart for R-squared (R^2) score

The Figure 3 Shows the comparison chart of R-squared (R^2) scoredemonstrates the existing KNN, FNN and proposed MLBWTCS. X axis denote the Dataset and y axis denotes the Accuracy ratio. The proposed MLBWTCS values are better than the existing algorithm. The existing algorithm values start from 25 to 55, 45 to 75 and proposed MLBWTCS values starts from 75 to 95. The proposed method provides the great results.

4.2 Root Mean Square Error

The RMSE is a measure of the dispersion of the predicted error, or the standard deviation of the predicted error. The RMSE is calculated by using Equation,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

 https://doi.org/10.1051/e3sconf/202338701008

Where $y_i$the observed outcome is$\hat{y}_i$ is the predicted outcome and N is the number of observed outcomes.

No of Datasets	KNN	FNN	Proposed MLBWTCS
100	30	40	79
200	35	55	83
300	40	63	87
400	55	74	97

Table 2.Comparison table for Root Mean Square Error

The Comparison table 2 of Root Mean Square Error demonstrates the different values of existing KNN, FNN and proposed MLBWTCS. While comparing the Existing algorithm and proposed MLBWTCS, provides the better results. The existing algorithm values start from 30 to 55, 40 to 74 and proposed MLBWTCS values starts from 79 to 97. The proposed method provides greater results.

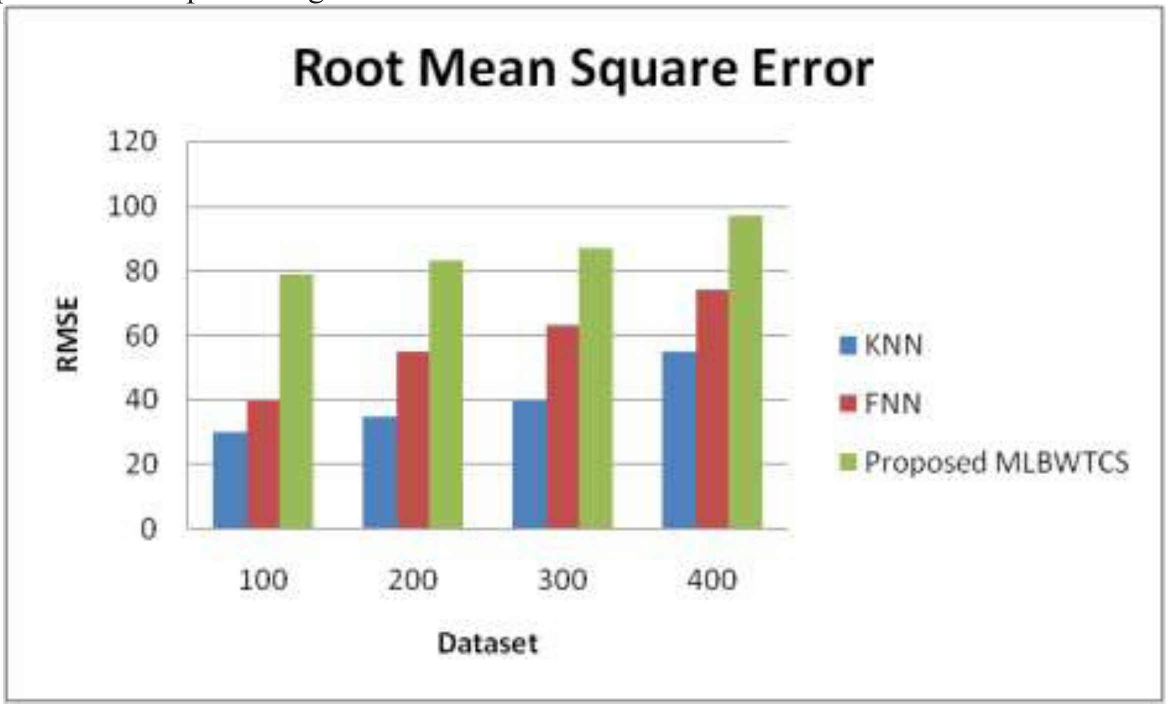

Figure 4.Comparison chart for Root Mean Square Error

The Figure 4 Shows the comparison chart of Root Mean Square Error demonstrates the existing KNN, FNN and proposed MLBWTCS. X axis denote the Dataset and y axis denotes the Accuracy ratio. The proposed MLBWTCS values are better than the existing algorithm. The existing algorithm values start from 30 to 55, 40 to 74 and proposed MLBWTCS values starts from 79 to 97. The proposed method provides the great results.

4.3 Computational Time

No of Datasets	KNN	FNN	Proposed MLBWTCS
100	60	65	88
200	65	70	85
300	70	75	94
400	75	80	98

Table 3.Comparison table for Computational Time

The Comparison table 3 of Computational Time demonstrates the different values of existing KNN, FNN and proposed MLBWTCS. While comparing the Existing algorithm and proposed MLBWTCS, provides the better results. The existing algorithm values start from 60 to75, 65 to 80 and proposed MLBWTCS values starts from88 to 98. The proposed method provides greater results.

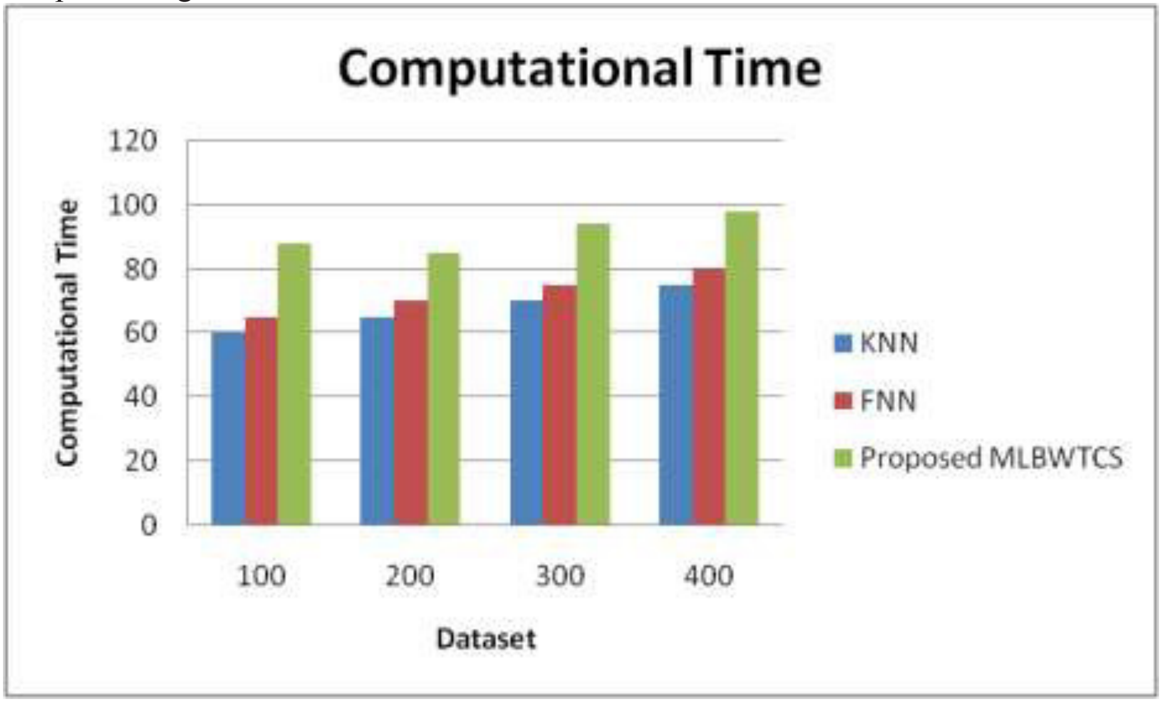

Figure 5.Comparison chart for Computational Time

The Figure 5 Shows the comparison chart of Root Mean Square Error demonstrates the existing KNN, FNN and proposed MLBWTCS. X axis denote the Dataset and y axis denotes the Accuracy ratio. The proposed MLBWTCS values are better than the existing algorithm. The existing algorithm values start from 60 to75, 65 to 80 and proposed MLBWTCS values starts from88 to 98. The proposed method provides the great results.

5. Conclusion

The Machine Learning-Based Wind Turbine Control System (MLBWTCS) is a promising technology that has the potential to revolutionize the wind energy industry. By using machine learning algorithms to optimize the operations of wind turbines, the MLBWTCS can improve their efficiency, reliability, and performance. The system collects data from sensors installed on the wind turbine and uses it to train a machine learning model, which predicts the optimal settings for the turbine operations. The predictions are then used to control the operations of the wind turbine in real-time. The MLBWTCS has Proposed MLBWTCS advantages over traditional control systems, including increased power generation, reduced maintenance costs, and improved safety.

References

1. X. Liu, Z. Zhou, and X. Bai, IEEE Tran on Sustainable Energy, vol. **11**, no. 2, pp. 652-661, (2020). DOI: 10.1109/TSTE.2019.2917423.
2. H. Kazemzadeh, M. Abdi, S. M. Bashi, and H. A. Talebi, Energy Reviews, vol. **113,** pp. 109264, (2019). DOI: 10.1016/j.rser.2019.109264.
3. C. C. Tsai, Y. K. Huang, and C. H. Lin, Expert Systems with App, vol. **36**, no. 2, pp. 3937-3944, (2009). DOI: 10.1016/j.eswa.2008.02.017.

4. J. Cao, J. Zhang, Z. Chen, and Y. Cao, J of Renewable and Sustainable Energy, vol. **2**, no. 6, pp. 063104, (2010). DOI: 10.1063/1.3514513.
5. A. E. Feijoo and J. CidrasIEEE Trans. Power Syst., vol. **15**, no. 1, pp. 110–115, Feb. (2000).
6. A. S. Dobakhshari and M. Fotuhi-Firuzabad, IEEE Trans. Energy Convers., vol. **24**, no. 3, pp. 792–801, Sep. (2009).
7. M. Asmine, J. Brochu, J. FortmFNN, R. Gagnon, Y. Kazachkov, C.-E. Langlois, C. Larose, E. Muljadi, J. MacDowell, P. Pourbeik, S. A. Seman, and K. WienIEEE Trans. Power Syst., vol. **26**, no. 3, pp. 1769–1782, Aug. (2011).
8. Princeton, NJ, Special Report, (2010).
9. M. Singh and S. Santoso, Golden, CO, Subcontract Report, (2011).
10. Hemalatha, S., Sunder Selwyn, T.,(2020),Materials Today: Proceedings,Vol.**46**,no.,pp.3180-3186.doi:10.1016/j.matpr.2020.09.392
11. Abbas, N. J., Zalkind, D. S., Pao, L., Wright, A. (2022). Wind Energy Science, 7(1), 53–73.
12. Sunder Selwyn, T., Hemalatha, S.,(2020),Materials Today: Proceedings,Vol.**46**,no.,pp.3639-3643.doi:10.1016/j.matpr.2021.01.656
13. Johnson, K. E., Fingersh, L. J., Balas, M. J., Pao, L. Y. (2004). Collection of ASME Wind Energy Sym-posium Technical Papers AIAA Aerospace Sciences Meeting and Exhibit, January 2014,103–113.
14. Archetti, Francesco and Candelieri, Antonio. Springer (2019).
15. Pino, F., Schena, L., Rabault, J., Kuhnle, A., Mendez, M. (2022). arXiv preprint arXiv:2202.11664.
16. Rasmussen, Carl Edward. Springer (2003).
17. Rajesh, G., Raajini, X.M., Kritika, N., Kavinkumar, A., Sagayam, K.M., Som, M.M., Wahab, M.H.A.,(2022),International Journal of Integrated Engineering,Vol.**14**,no.3,pp.80-89.doi:10.30880/ijie.2022.14.03.009
18. Dykes, Katherine L and Damiani, Rick R and Graf, Peter A and Scott, George N and King, Ryan N and Guo, Yi and Quick, Julian and Sethuraman, Latha and Veers, Paul Sand Ning, Andrew (2018).
19. Sunder Selwyn, T., Hemalatha, S.,(2020),",Materials Today: Proceedings,Vol.**46**,pp.3292-3296.doi:10.1016/j.matpr.2020.11.461