ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems Part B: Mechanical Engineering
Online journal at:
asmedigitalcollection.asme.org/risk

Júlio Oliveira Schmidt
Electrical Engineering Course,
Universidade Federal de Santa Maria,
Santa Maria, Rio Grande do Sul 97105-900, Brazil
e-mail: julio.oliveira@acad.ufsm.br

Lucas França Aires
Electrical Engineering Course,
Universidade Federal de Santa Maria,
Santa Maria, Rio Grande do Sul 97105-900, Brazil
e-mail: lucas.aires@acad.ufsm.br

Guilherme Ricardo Hubner
Power Electronics and Control Research Group,
Universidade Federal de Santa Maria,
Santa Maria, Rio Grande do Sul 97105-900, Brazil
e-mail: guilherme@gepoc.ufsm.br

Humberto Pinheiro
Power Electronics and Control Research Group,
Universidade Federal de Santa Maria,
Santa Maria, Rio Grande do Sul 97105-900, Brazil
e-mail: humberto.pinheiro@ufsm.br

Daniel Fernando Tello Gamarra[1]
Automation and Applied Robotics Group,
Universidade Federal de Santa Maria,
Santa Maria, Rio Grande do Sul 97105-900, Brazil
e-mail: daniel.gamarra@ufsm.br

Comparison of Convolutional Neural Networks Using Gramian Angular Difference Field, Recurrence Plot, and Markov Transition Fields for Rotor Imbalance Detection in Wind Turbines

This study investigates various data representations derived from wind turbine dynamics, utilizing convolutional neural networks (CNNs) as diagnostic tools to identify rotor mass imbalance. The analysis examines three distinct methods for encoding the dataset into 2D images: Gramian angular difference field (GADF), recurrence plot (RP), and Markov transition fields (MTF). To achieve this, simulations of a 1.5 MW three-bladed wind turbine model were conducted using TURBSIM, FAST, and MATLAB SIMULINK. These simulations generated rotor speed data under diverse conditions, including varying wind speeds and induced mass imbalances achieved by adjusting blade density in the software. Comparative analyses across nine classifiers demonstrated that the Gramian angular difference field provided the most promising results. The study presents detailed results and simulations, offering a thorough evaluation of the proposed architecture's effectiveness in detecting and diagnosing rotor mass imbalances in wind turbines. [DOI: 10.1115/1.4068711]

1 Introduction

Despite significant advancements in recent years, wind energy continues to face considerable challenges related to maintenance. Wind farms often struggle to maintain system efficiency due to adverse weather conditions, such as heavy rainfall, sandstorms, strong winds, thermal fluctuations, and tidal movements. These factors reduce the intervals between required maintenance, leading to increased costs. Studies estimate that approximately 20–35% of the total energy production cost is allocated to maintenance activities [1,2].

This reality has spurred a surge in research efforts aimed at addressing wind turbine failures, refining wind resource assessments, and advancing condition monitoring systems. These initiatives are pivotal in ensuring the continued performance, reliability, and efficiency of wind energy systems [3,4].

Aiming the decrease of maintenance expense, Gao et al. [5] created a method for classifying faults in the gearbox of a wind turbine. Captured vibration signals were processed by the integral empirical load mode decomposition method, classifying then with the least squares support vector machine to see which type of fault was caused [6–9].

To refine the results obtained from a trained convolutional neural network (ConvNet/CNN), the authors employed different approaches. For the qualification, Cho et al. [10] identified distinct kinds of failures and used Kalman filters to recognize the blade pitch angle and an artificial neural network to rank six variant kinds of failures in wind turbines. Chang et al. [11] employed parallel convolutional layers. The convolution is formed by manifold branches, each one of them with manifold layers.

Another approach to the fault detection problem is the use of predictive algorithms, which try to create models to detect failures in turbines or other points of energy generation. Modern research provides interesting results in the area, such as Ref. [12], that presents several methods used in a review of methods for fault diagnosis and detection, and Ref. [13], that presents the use of deep learning and optimization algorithms for fault detection.

This study offers a different idea for the recognition of a mass imbalance in the blades of a wind turbine [14]. A mass imbalance is one of the biggest problems in the blades; it is possible to be provoked by many causes, like icing [15], poor production, exposure to the ambient, and trouble at transportation or at installation. Imbalance typically causes variations in rotor velocity, preventing

[1]Corresponding author.

Manuscript received January 16, 2024; final manuscript received May 13, 2025; published online June 27, 2025. Assoc. Editor: Nikolaos Dervilis.

the wind turbine from operating at rated values. This results in reduced energy generation and decreases the operational lifespan of the rotor [16–18].

The principal contributions of this research are:

(i) The application of Gramian angular difference field (GADF), recurrent plot, and Markov transition fields (MTF) for a signal representation into images of the dataset.
(ii) Evaluation of the performance of the three different image representations.
(iii) Use of convolutional neural networks to diagnose mass imbalance.
(iv) Comparison with others classifiers for consolidation of the results.

The paper is composed of five sections: after an Introduction, the second section of Material and Methods explains the main algorithms and tools used in the article, the third section of the Proposed Approach presents the architecture developed, the fourth section shows the Results obtained, and the last section of Conclusion summarizes the article. The code for the comparison and implementation is available for other researchers.[2]

2 Materials and Methods

This section describes the dataset, the techniques, and algorithms used for the experiments.

2.1 Dataset. The dataset used for detecting mass imbalances in wind turbine rotors was generated using TURBSIM, FAST, and SIMULINK [19,20]. TURBSIM was utilized to produce wind time series, incorporating turbulence characteristics to provide a realistic representation of wind turbine conditions. FAST integrated aerodynamic models, structural components, and control systems, utilizing the wind series generated by TURBSIM. SIMULINK facilitated the development of control loops for maximum power point tracking and nominal power regions, along with modeling a synchronous permanent magnet generator, a generator speed estimator, and performing training and detection based on electrical current measurements. The dataset includes 390 simulations for each rotor unbalance condition (1%, 2%, 3%, 4%, and 5% mass imbalances), with each simulation spanning 120 s and a sampling frequency of 2 kHz. For further details on the dataset's construction, refer to Ref. [21].

In constructing the database, 390 samples were generated for each imbalanced class and 273 for the balanced class, resulting in a total of 2223 samples. The imbalanced classes included random positive and negative variations (e.g., +3% and −3%). After filtering by rotational speed, the dataset was reduced to 1574 samples, while still retaining class imbalance. To address this, the synthetic minority oversampling technique (SMOTE) algorithm was applied to generate synthetic data, equalizing the number of samples across all classes. Figure 1, included below, illustrates this process.

Unbalanced data, characterized by disproportionate class distribution, presents challenges for predictive modeling, particularly affecting algorithms that rely on equal class representation. This study identifies a prevalent imbalance that impedes classifier training, which was addressed using the SMOTE [22–24].

2.2 Convolutional Neural Network. Artificial neural networks are a subset of machine learning and are at the heart of deep learning algorithms; they are composed of layers of nodes, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed to the next layer of the network [25].

For this work, the type of network used was a convolutional neural network. A ConvNet/CNN is a deep learning algorithm that can receive an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiates one from the other. CNN specializes in processing data that has a gridlike topology, such as an image [26–28].

Convolutional neural networks goal is to reduce images so that they are easier to process without losing valuable resources for accurate prediction. The ConvNet architecture has three types of layers: convolutional layer, pooling layer, and fully connected layer.

2.3 Gramian Angular Difference Field. Gramian angular difference field is a very popular method mainly used to transform time series into images. It is well known for maintaining the temporal luminosity between the entries of the one-dimensional series. To carry out the transformation, first all the values of the series undergo a scale change, as they vary only between 0 and 1. After that, the data are represented in polar coordinates and not in Cartesian coordinates [29–33]. For this, it is necessary to calculate the value of r and ϕ (phi) for all points of each sample. The "r" value represents the absolute distance of the sample to the origin (0, 0) of the Cartesian coordinates, and the ϕ value represents the angle formed between the X axis of the Cartesian plane and the imaginary line that connects the sample to the origin (0, 0). Mathematically, r and ϕ can be defined by

$$r = \sqrt{x^2 + y^2} \tag{1}$$

$$\phi = \arccos(x/y) \tag{2}$$

where x and y represent the values of x and y of each point of a sample in the Cartesian plane. With the calculated data, the Gramian angular field (GAF) performs the trigonometric subtraction between all points of the sample

$$\mathrm{GADF} = [\sin(\phi_a) - \sin(\phi_b)] \tag{3}$$

This equation is applied to all points and subtracts against all points other than the one being applied. After that, each value is sorted from left to right and top to bottom. A tone is assigned to each of these values according to a color palette that varies tones according to values from 0 to 1. Finally, with all the points aligned and their respective colors calculated, the image is formed. Figure 2 shows an example of an image generated from the dataset using the Gramian Angular Field method.

The GAF technique converts data vectors into matrices, capturing spatial relationships between features after a power spectral density (PSD) transformation. These matrices are transformed into image-like representations, which serve as inputs to CNNs. By leveraging

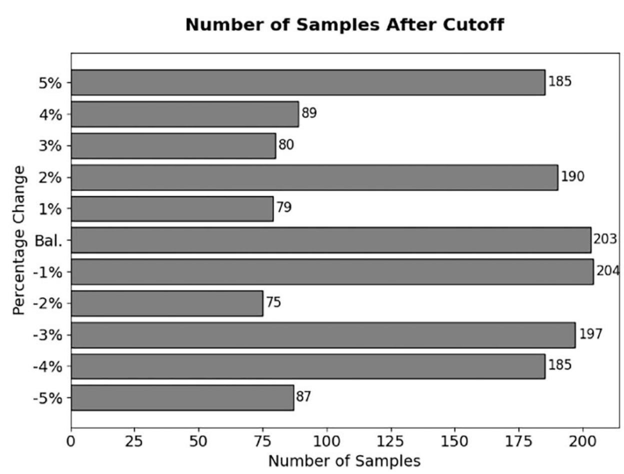

Fig. 1 Gramian angular field algorithm applied to a time signal

[2]https:// github.com/zzzzzzzcz/CNN-RP-MTF-GADF-

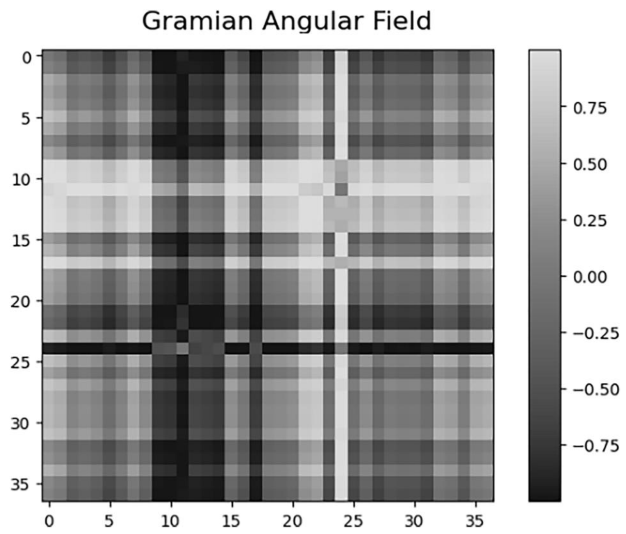

Fig. 2 Gramian angular field algorithm applied to a time signal

their powerful feature extraction capabilities, CNNs maximize performance when processing these image-based inputs. While not inherently superior to time-domain signals, GAF provides an alternative perspective that complements traditional approaches and enhances data analysis for classification tasks.

2.4 Recurrence Plot. Convolutional neural networks are widely used as image recognition classifiers. To train a CNN with a dataset, it is necessary to encode the vector information into an image format [34,35]. Recurrence plot (RP) is a graph or image of a square matrix, in which the elements of the matrix correspond to those times in which a state of a dynamical system is repeated (columns and rows then correspond to a certain pair of times). Technically, the RP reveals all the moments when the phase space trajectory of the dynamical system visits approximately the same area in phase space. This can be expressed mathematically as

$$R(i,j) = \Theta\left(\epsilon - \left|\left|\mathbf{x}(\boldsymbol{i}) - \mathbf{x}(\boldsymbol{j})\right|\right|\right) \tag{4}$$

For $\mathbf{i}, \mathbf{j} = 1, \ldots, \mathbf{N}$, where ϵ is a limit distance, $||.||$ is a norm, Θ is a Headviside function, and $\mathbf{N}$ is the numbers of states.

This equation creates an image where the values are ordered from left to right and top to bottom, that is, the first value is located in the upper left corner and the last value is in the lower right corner. For each of these values, a tone is stipulated according to a color palette that varies the tones according to values from 0 to 1. This technique was applied to the signals from the frequency domain dataset with the inclusion of some statistical data [36,37], and the image generated by the equation can be seen in Fig. 3.

2.5 Markov Transition Field. Markov transition fields are another visualization technique to highlight the behavior of time series, such as the recurrent plot. Each value in the time series is first assigned to a specific quantile, where each quantile is treated as a bin. For instance, when using quartiles (four bins), the lowest 25% of the values define the limits of the first quartile, the next 25% define the limits of the second quartile, and so on. Each bin can be considered a distinct "state." To apply the Markov model, a state transition matrix must be created, and the values within this matrix can be calculated using the following equation:

$$A_{ij} = P(st = j|s(t-1) = i) \tag{5}$$

where A_{ij} is the probability of transition from state i to state j. So the defined MTF array would look like this

$$M_{kl} = A_{qkql} \tag{6}$$

The M_{kl} is the probability of a one-step transition from the bin of x_k to the bin of x_l, where x_k and x_l are two points in the time series in the time-step of k and l, respectively. Then, q_k and q_l, once the transition probability matrix is defined, it is possible to generate an image based on this matrix, as illustrated in the example below in Fig. 4.

3 Proposed Approach

3.1 General Architecture. In Fig. 5, a flowchart is illustrated with all the steps of the methodology to better elucidate the methodology used. The dataset exhibits an imbalance, prompting the utilization of the SMOTE algorithm for data augmentation and rebalancing. Furthermore, statistical features in both frequency and time domains are extracted to compose a comprehensive dataset. Once this dataset is prepared, three distinct data representations—Gramian angular, Markov transition, and recurrent plot—are employed to convert information into images.

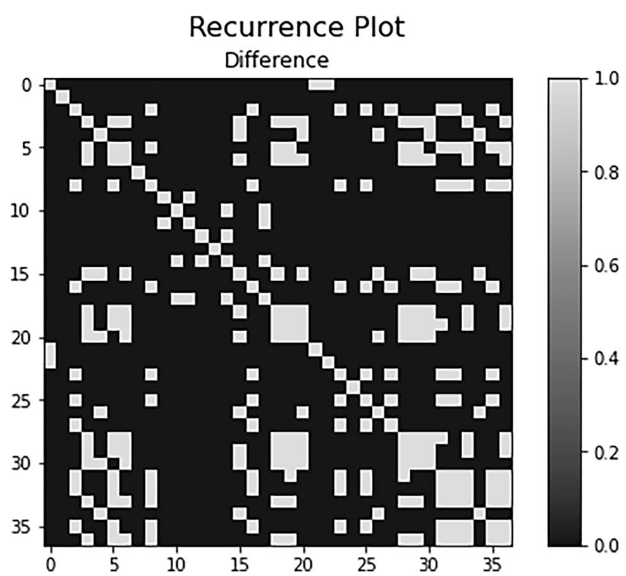

Fig. 3 Recurrence plot representation

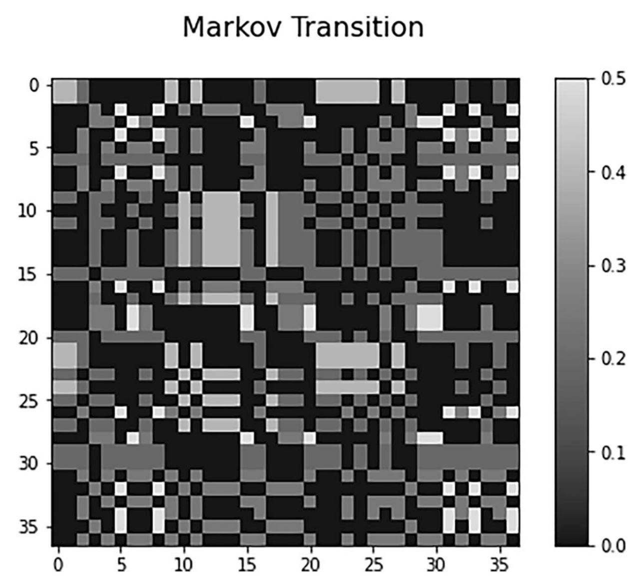

Fig. 4 Image generate using MTF

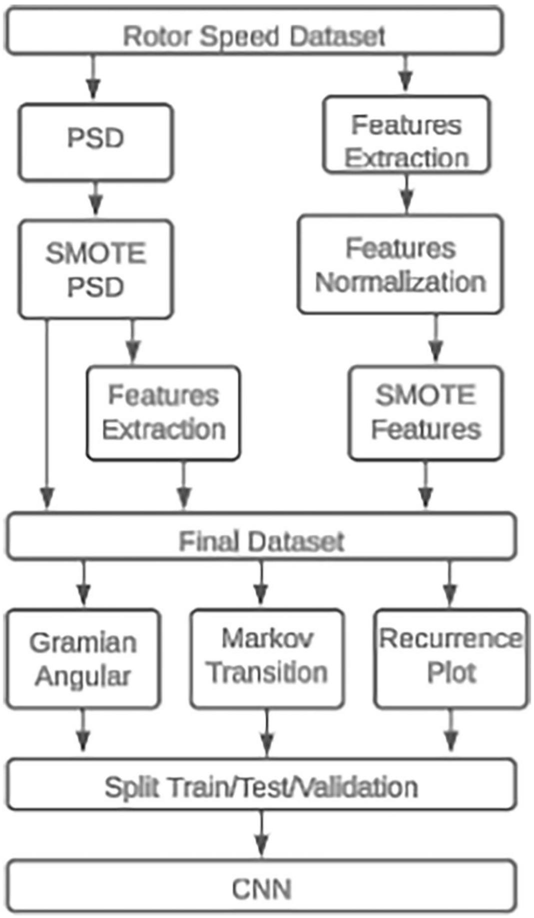

Fig. 5 Flowchart of the methodology employed for this study

Subsequently, the data undergoes division into training, test, and validation sets. The training set is used to optimize the model during the learning phase, the validation set is employed to monitor and evaluate the model's performance throughout the training process, and the test set is dedicated to assessing the model's final performance on unseen data, ensuring its ability to generalize effectively. This segmentation sets the stage for utilizing convolutional neural networks (CNNs) to effectively identify and address the imbalance within the wind turbine data. It is also worth noting that by transforming time-signals into images, we leverage the powerful feature extraction capabilities of CNNs, which are well-suited for handling image-like inputs. This approach is particularly helpful when spatial relationships between features, as captured by GAF, RP, or MTF, play a significant role in classification.

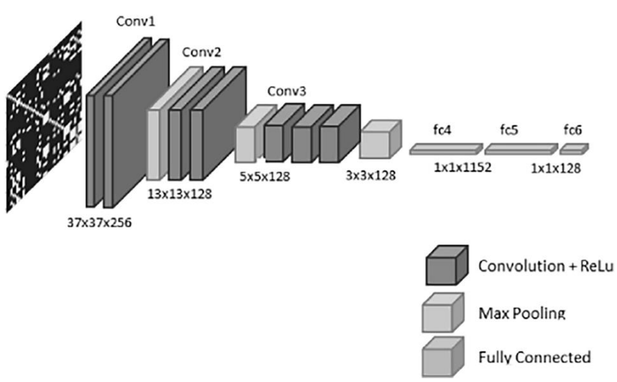

Fig. 6 Type and output shape of each layer in the convolutional neural network

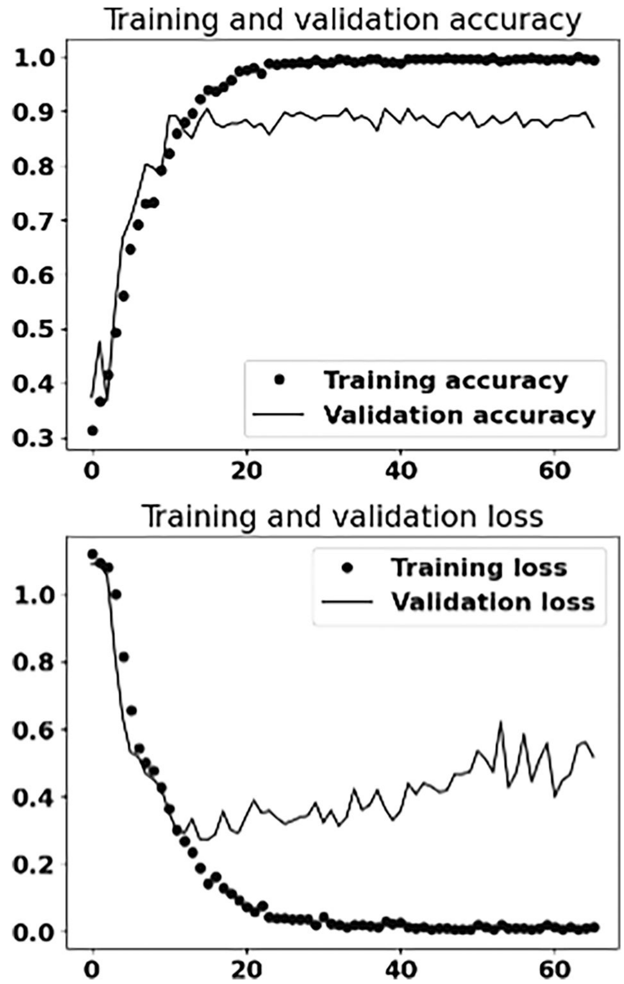

Fig. 7 Accuracy and loss of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using MTF

3.2 Embedding Data Into a Vector. Seeking to improve the accuracy of the algorithm, data of temporal characteristics were introduced to the code, taken from the original dataset, and data of characteristics of the PSD [38], from the normalized samples of the fragment of frequencies of the PSD [39].

The first feature extraction is performed [40] in a dataset with 1574 samples and 240,000 characteristics for each sample, generating a dataset which is called "time's features" [41], with

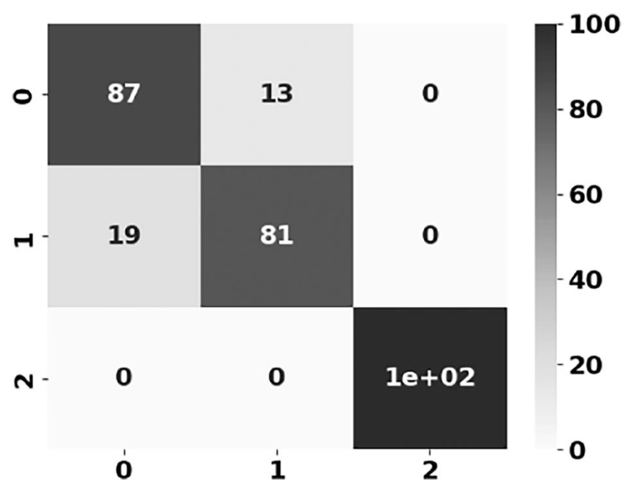

Fig. 8 Confusion matrix of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using MTF

the following features: maximum value, kurtosis, median absolute deviation, mean absolute deviation, standard deviation, variance, total energy, trimmed mean, peak-to-peak value, mean, and minimal value. Since these 1574 samples are divided into imbalanced classes, SMOTE is applied to generate synthetic samples, resulting in a total of 2244 samples across 11 balanced classes. By unifying the negative and positive percentage classes, such as −5% and 5%, the dataset is reduced to 2244 samples distributed across six classes. However, since all classes except for normal functioning (0% imbalance) are doubled, additional synthetic data must be created, leading to a final dataset of 2448 samples divided into six perfectly balanced classes.

Then, it was calculated the power spectral density of the original 1574 time series. Considering that the mass imbalance malfunction mainly affects generation at low frequencies, only the first 15 values of the entire spectrum were used. The process of creating synthetic data is identical to the one used with the time-series features. At this stage, a second feature extraction is performed, resulting in a dataset referred to as "PSD features," which includes the following features: kurtosis, median absolute deviation, mean absolute deviation, standard deviation, variance, total energy, trimmed mean, peak-to-peak value, mean, RMS value, and zero-crossing value. The final dataset is constructed by concatenating the PSD values, the PSD features, and the time series features. The 2448 samples with 37 features each are then encoded as 37×37 images with three dimensions of color; this encoding process is made using the recurrence plot, Gramian angular difference fields, and Markov transition field algorithm. Those images are split into the train, test, and validation sets. The train set represents 68% of the images, the validation set represents 12%, and the test set represents the last 20%.

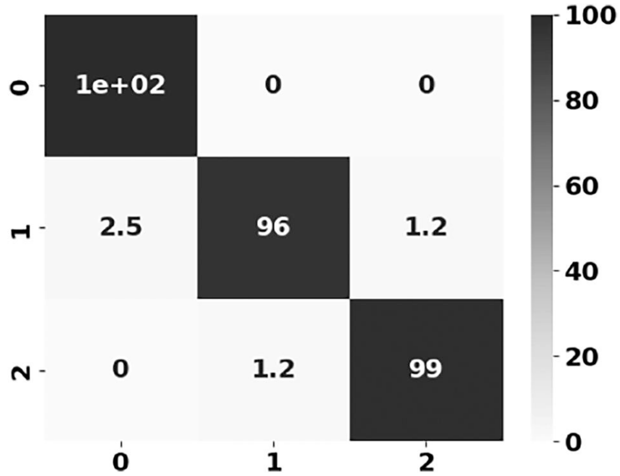

Fig. 10 Confusion matrix of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using RP

3.3 Convolutional Neural Network Architecture. The CNN model is composed of three convolutional layers, the first one has 256 nodes, and the last layer with 128 nodes, three max pooling layers, four dropout layers, four activation layers, one flatten layer, and two fully connected layers. The whole summary model is in Fig. 6.

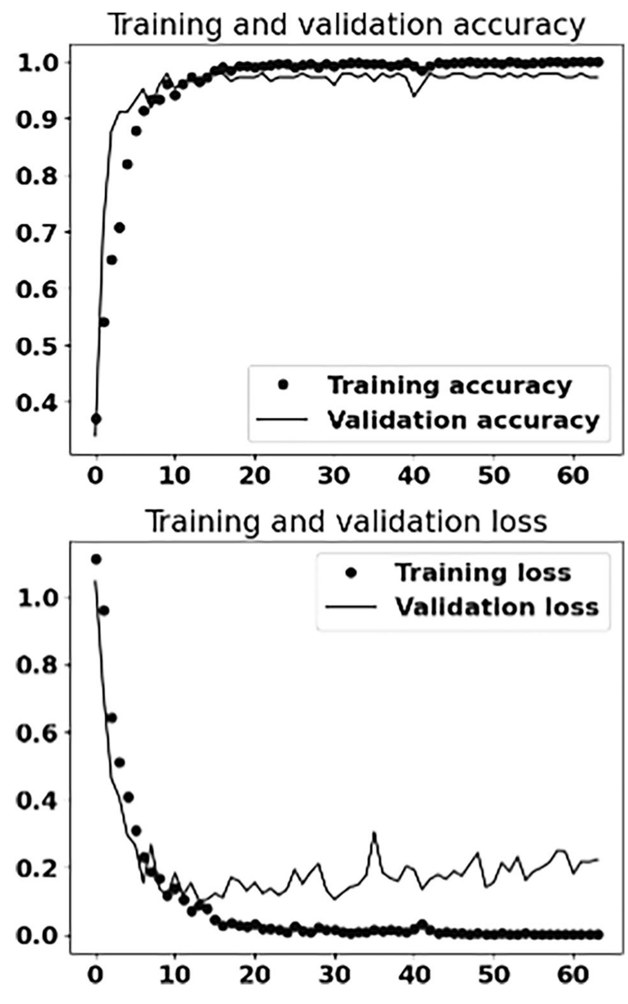

Fig. 9 Accuracy and loss of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using RP

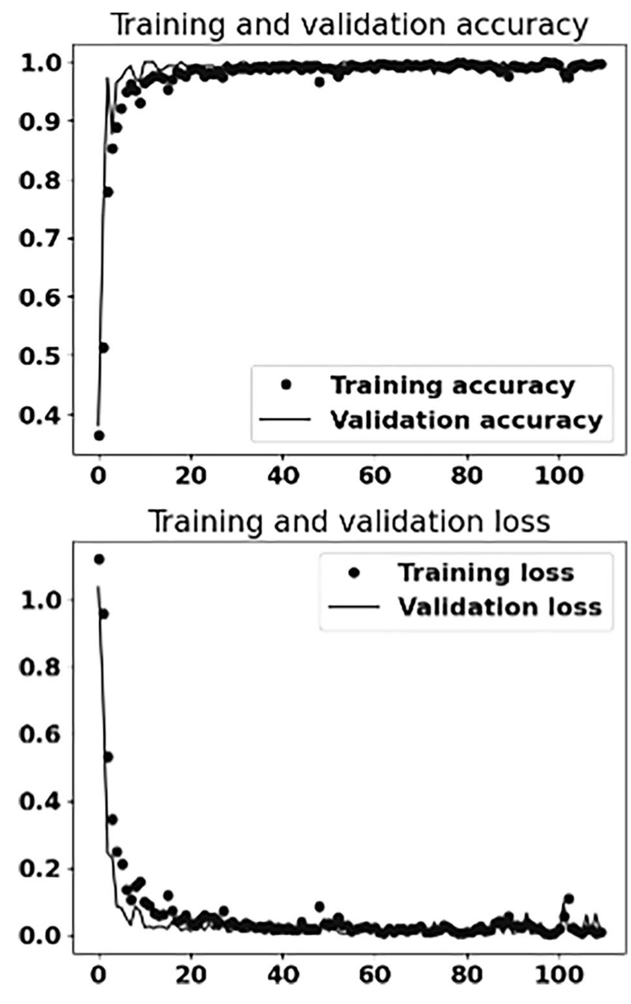

Fig. 11 Accuracy and loss of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using GADF

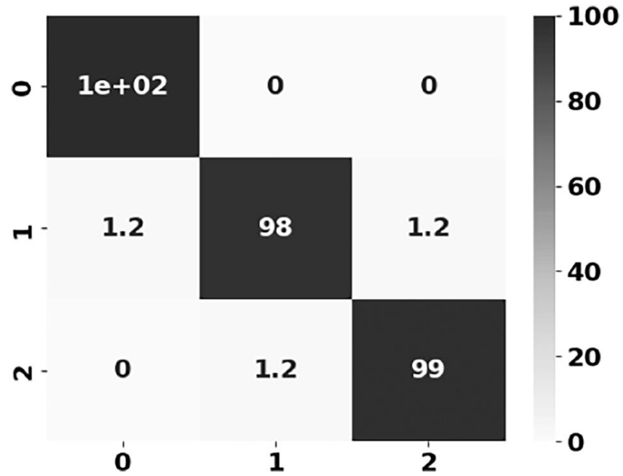

Fig. 12 Confusion matrix of the CNN with three classes (5% of imbalance, 3% of imbalance, and balanced) using GADF

Besides the CNN, other nine classifiers were used for comparative matters. The nine used classifiers are nearest neighbors, linear support vector machine (SVM), radial basis functions (RBF) SVM, decision tree, random forest, multilayer perceptron (MLP) neural network, AdaBoost, Gaussian Naive Bayes, and quadratic discriminant analysis (QDA). All of them were trained with 80% of the data and tested 20%.

4 Results

The results will be broken down by the number of classes used, and each of these subdivisions will have two types of processing for comparison. With the CNN network, data formed by the PSD were used, adding the data of PSD characteristics and time characteristics.

4.1 Experiments With Three Classes: 5% of Imbalance, 3% of Imbalance, and Balanced. With three classes, it was possible to obtain the best performance of the implemented algorithms. There were three different tests, each test using a different algorithm to generate images. The first test used recurrence plot as an image generator, subsequently, Markov transition fields were used, and finally, Gramian angular difference fields were employed.

The loss and accuracy of the CNN with three classes using MTF are presented in Fig. 7, while the confusion matrix is illustrated in Fig. 8, utilizing Markov transition field.

Figure 9 displays the accuracy and loss curves of the CNN with three classes using recurrent plots, and Fig. 10 exhibits the confusion matrix, both using recurrence plot.

In Fig. 11, it is possible to see the accuracy and loss of the CNN with three classes using Gramian angular difference fields, and Fig. 12 depicts the confusion matrix, generated with the use of Gramian angular difference fields.

Table 1 Best result for three classes

Classifier	Accuracy (%)
Linear SVM	99.51
MLP neural net	99.18
QDA	99.59
Decision tree	97.95
RBF SVM	99.02
Nearest neighbors	97.06
Naive Bayes	99.18
Random forest	94.61
AdaBoost	99.02
CNN with RP	97.96
CNN with MTF	90.20
CNN with GADF	99.59

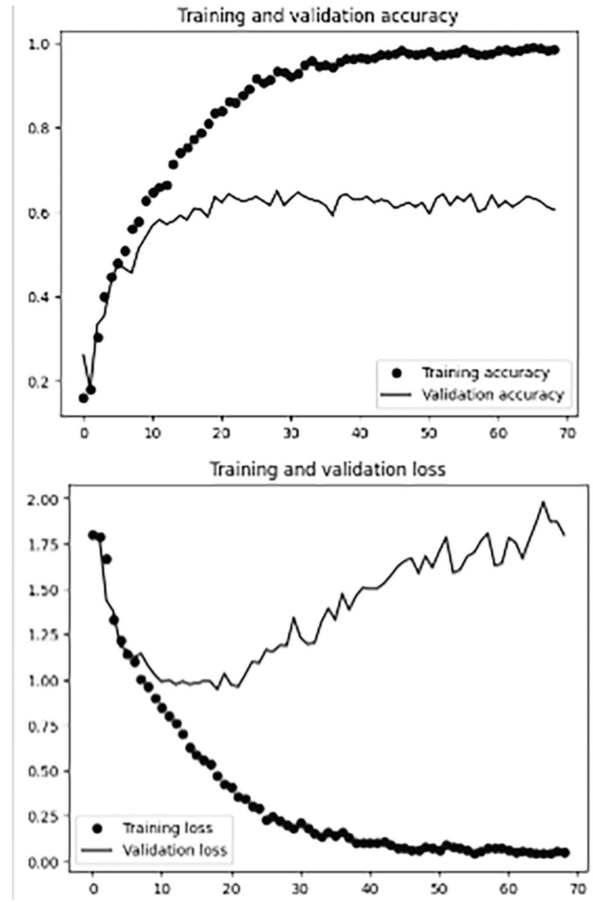

Fig. 13 Accuracy and loss of the CNN with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using MTF

4.2 Experiments With Different Classifiers With Three Classes. Due to the reduced number of only three classes in use, all algorithms performed better than 94% accuracy for PSD data. The highest performance achieved was 99.59% accuracy, which was achieved by the QDA classifier, both using recurrent plot, Gramian angular difference fields, and Markov transition. CNN, on the other

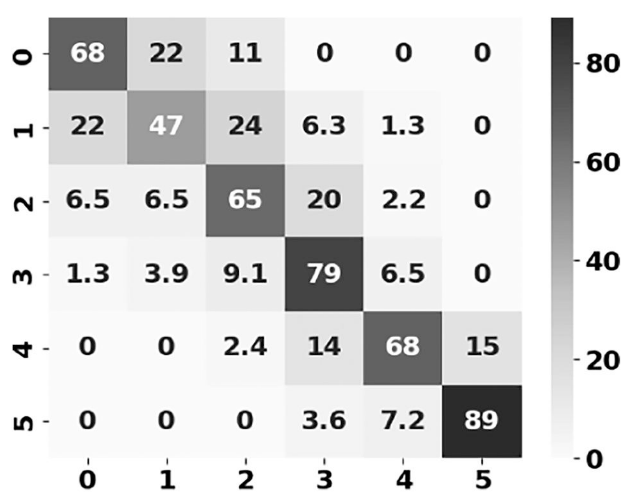

Fig. 14 Confusion matrix of the CNN with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using MTF

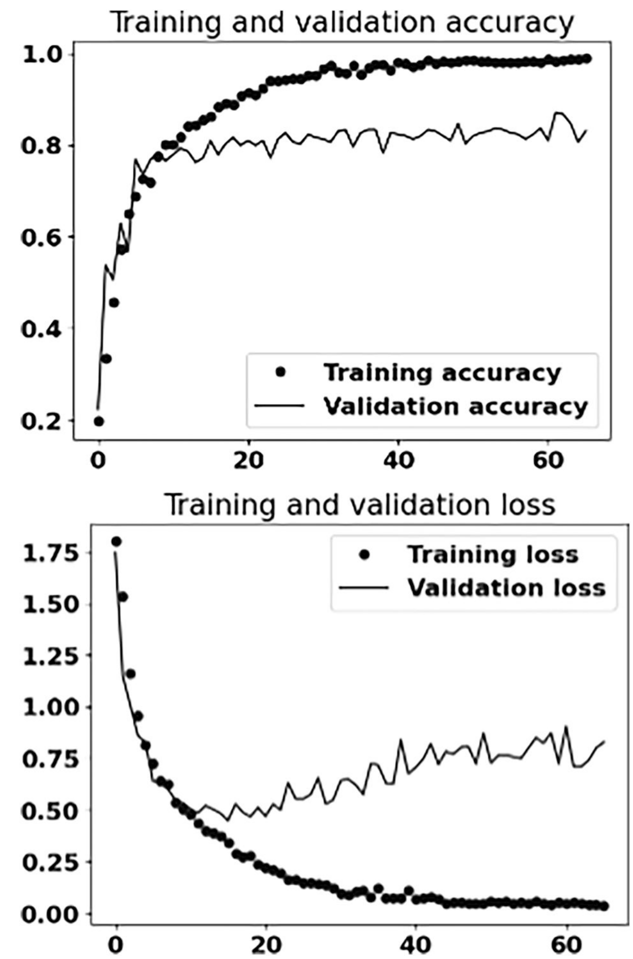

Fig. 15 Accuracy and loss of the CNN with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using RP

hand, showed a great difference in performance between those algorithms; Gramian angular difference fields manage to get the highest score reaching up to 99.59%, but the other two algorithms were unable to achieve the same results. Table 1 shows the best results achieved for the classifiers.

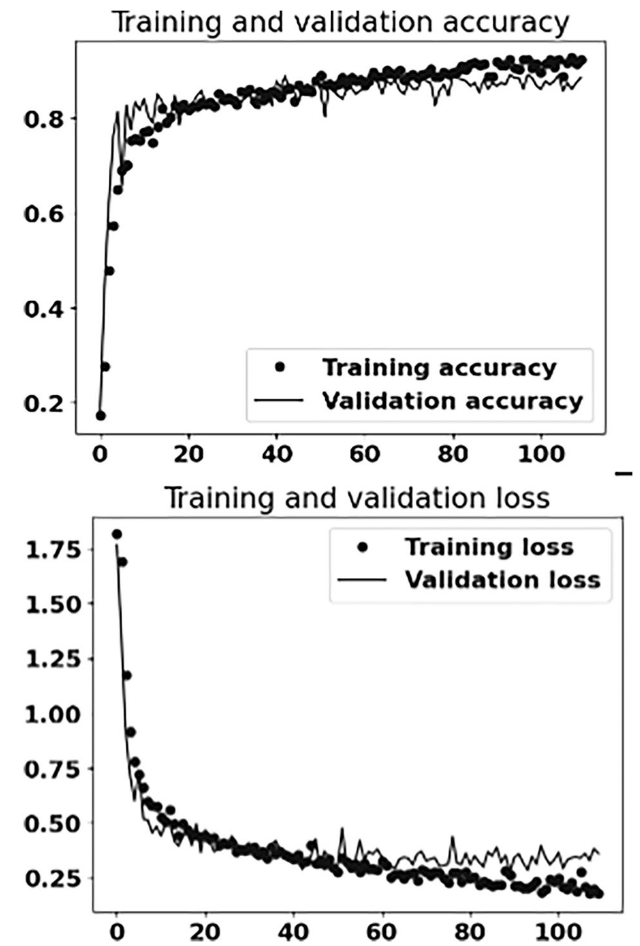

Fig. 17 Accuracy and loss of the CNN with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using GADF

4.3 Experiments With Six Classes: 5% Imbalance, 4% Imbalance, 3% Imbalance, 2% Imbalance, 1% Imbalance, and Balanced. Figure 13 displays the accuracy loss curves of the CNN with the six classes, while Fig. 14 illustrates the confusion matrix, both generated using Markov transition field.

In Fig. 15, the accuracy and loss curves of the CNN with six classification classes are presented. Additionally, in Fig. 16, the

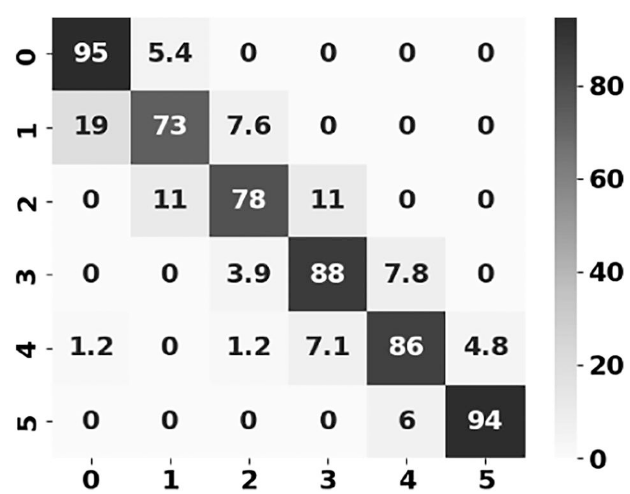

Fig. 16 Confusion matrix obtained after training with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using RP

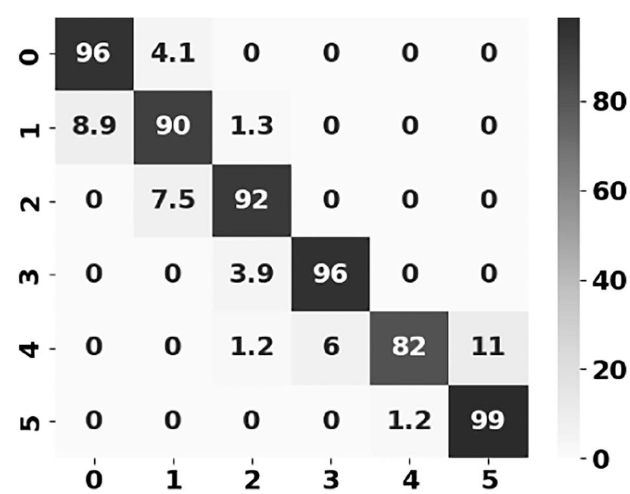

Fig. 18 Confusion matrix obtained after training with six classes (5% imbalance, 4% imbalance, 3% imbalance, 2% imbalance, 1% imbalance, and balanced) using GADF

Table 2 Best result for six classes

Classifier	Accuracy (%)
Linear SVM	91.84
MLP neural net	83.88
QDA	88.20
Decision tree	86.33
RBF SVM	84.44
Nearest neighbors	79.52
Naive Bayes	75.50
Random forest	71.94
AdaBoost	58.57
CNN with RP	84.89
CNN with MTF	69.18
CNN with GADF	92.45

confusion matrix is presented, with all matrices generated using the recurrence plot technique.

Figure 17 presents the accuracy and loss curves of the CNN with six classes, providing a visual representation of their performance. Figure 18, on the other hand, offers an insight into the confusion matrix, and it was created using Gramian angular difference fields.

4.4 Experiments With Different Classifiers With Six Classes. Due to the increase of number of classes to six, all algorithms performed with less accuracy compared to the results using three classes. The highest performance achieved between classifiers was 90.20% accuracy, which was achieved by the linear SVM classifier, using Gramian angular difference fields. CNN, on the other hand, showed a better performance between those classifiers; Gramian angular difference fields manage to get the highest score reaching up to 92.45%. Table 2 shows the best results are achieved.

5 Conclusions

In summary, this study implemented and evaluated three distinct image representations of wind turbine data signals. The Gramian angular difference field emerged as the most effective in detecting mass imbalance. Additionally, a CNN architecture was employed to classify and assess the degree of imbalance in wind turbines. When compared to nine different classifiers, the CNN achieved the highest accuracy in isolation, particularly in the six-class scenarios. It also shared the top position with linear SVM in the three-class scenarios. The code used in this study has been made publicly available to the scientific community.

Future research could explore alternative deep neural network architectures to further improve the effectiveness of imbalance detection in wind turbines.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

References

[1] Wang, J., Liang, Y., Zheng, Y., Gao, R. X., and Zhang, F., 2020, "An Integrated Fault Diagnosis and Prognosis Approach for Predictive Maintenance of Wind Turbine Bearing With Limited Samples," Renewable Energy, **145**, pp. 642–650.

[2] Mishnaevsky, L., Jr., 2019, "Repair of Wind Turbine Blades: Review of Methods and Related Computational Mechanics Problems," Renewable Energy, **140**, pp. 828–839.

[3] Kaasik, M., and Mirme, S., 2020, "Wind: Generating Power and Cooling the Power Lines," Adv. Sci. Res., **17**, pp. 105–108.

[4] Cho, S.-Y., Choi, S.-K., Kim, J.-G., and Cho, C.-H., 2018, "An Experimental Study of the Optimal Design Parameters of a Wind Power Tower Used to Improve the Performance of Vertical Axis Wind Turbines," Adv. Mech. Eng., **10**(9), p. 168781401879954.

[5] Gao, Q., Liu, W., Tang, B., and Li, G., 2018, "A Novel Wind Turbine Fault Diagnosis Method Based on Integral Extension Load Mean Decomposition Multiscale Entropy and Least Squares Support Vector Machine," Renewable Energy, **116**, pp. 169–175.

[6] Cao, Z., Xu, J., Xiao, W., Gao, Y., and Wu, H., 2019, "A Novel Method for Detection of Wind Turbine Blade Imbalance Based on Multi-Variable Spectrum Imaging and Convolutional Neural Network," 2019 Chinese Control Conference (CCC), Guangzhou, China, July 27–30, pp. 4925–4930.

[7] Kumar, R., Ismail, M., Zhao, W., Noori, M., Ya, Dav, A., Chen, S., and Singh, V., et al., 2021, "Damage Detection of Wind Turbine System Based on Signal Processing Approach: A Critical Review," Clean Technol. Environ. Policy, **23**(1), pp. 1–20.

[8] Zhang, J., and Yang, Y., 2024, "Fault Diagnosis of Nonlinear Analog Circuits Using Generalized Frequency Response Function and LSSVM," PLoS One, **19**(12), p. e0316151.

[9] Chen, M., Guo, S., Xing, Z., Folly, K. A., Liu, Y., and Zhang, P., 2024, "Diagnosis of a Rotor Imbalance in a Wind Turbine Based on Support Vector Machine," AIP Adv., **14**(4), p. 045315.

[10] Cho, S., Choi, M., Gao, Z., and Moan, T., 2021, "Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks," Renewable Energy, **169**, pp. 1–13.

[11] Chang, Y., Chen, J., Qu, C., and Pan, T., 2020, "Intelligent Fault Diagnosis of Wind Turbines Via a Deep Learning Network Using Parallel Convolution Layers With Multi-Scale Kernels," Renewable Energy, **153**, pp. 205–213.

[12] Lan, J., Chen, N., Li, H., and Wang, X., 2024, "A Review of Fault Diagnosis and Prediction Methods for Wind Turbine Pitch Systems," Int. J. Green Energy, **21**(7), pp. 1613–1640.

[13] Mahendra Bhatu Gawali, S. S. G., and Patil, M., 2023, "Fault Prediction Model in Wind Turbines Using Deep Learning Structure With Enhanced Optimisation Algorithm," J. Control Decis., **12**(3), pp. 471–488.

[14] Gong, X., and Qiao, W., 2012, "Imbalance Fault Detection of Direct-Drive Wind Turbines Using Generator Current Signals," IEEE Trans. Energy Convers., **27**(2), pp. 468–476.

[15] Zeng, J., and Song, B., 2017, "Research on Experiment and Numerical Simulation of Ultrasonic Deicing for Wind Turbine Blades," Renewable Energy, **113**, pp. 706–712.

[16] Helbing, G., and Ritter, M., 2018, "Deep Learning for Fault Detection in Wind Turbines," Renewable Sustainable Energy Rev., **98**, pp. 189–198.

[17] Fantino, R., Solsona, J., and Busada, C., 2016, "Nonlinear Observer-Based Control for PMSG Wind Turbine," Energy, **113**, pp. 248–257.

[18] García Márquez, F. P., Tobias, A. M., Pinar Pérez, J. M., and Papaelias, M., 2012, "Condition Monitoring of Wind Turbines: Techniques and Methods," Renewable Energy, **46**, pp. 169–178.

[19] Jonkman, J. B., 2006, *TurbSim User's Guide*, National Renewable Energy Laboratory, Golden, CO.

[20] Jonkman, J. M., 2003, "Modeling of the UAE Wind Turbine for Refinement of FAST," National Renewable Energy Laboratory, Golden, CO, Report No. NREL/TP-500-34755.

[21] Hübner, G., Pinheiro, H., de Souza, C., Franchi, C., da Rosa, L., and Dias, J., 2021, "Detection of Mass Imbalance in the Rotor of Wind Turbines Using Support Vector Machine," Renewable Energy, **170**, pp. 49–59.

[22] Zhang, H., Huang, L., Wu, C. Q., and Li, Z., 2020, "An Effective Convolutional Neural Network Based on Smote and Gaussian Mixture Model for Intrusion detection in Imbalanced Dataset," Comput. Networks, **177**, p. 107315.

[23] Li, P., Abdel-Aty, M., and Yuan, J., 2020, "Real-Time Crash Risk Prediction on Arterials Based on LSTM-CNN," Accid. Anal. Prev., **135**, p. 105371.

[24] Xiong, Y., Wang, L., Wang, Q., Liu, S., and Kou, B., 2022, "Improved Convolutional Neural Network With Feature Selection for Imbalanced ECG Multifactor Classification," Measurement, **189**, p. 110471.

[25] Bishop, C. M., 2006, *Pattern Recognition and Machine Learning*, Vol. 2, Springer Google Scholar, Cambridge, UK, pp. 1122–1128.

[26] Indolia, S., Goswami, A. K., Mishra, S. P., and Asopa, P., 2018, "Conceptual Understanding of Convolutional Neural Network—A Deep Learning Approach," Procedia Comput. Sci., **132**, pp. 679–688.

[27] Wessner, R. N., Frozza, R., Duarte da Silva Bagatini, D., and Molz, R. F., 2023, "Recognition of Weeds in Corn Crops: System With Convolutional Neural Networks," J. Agric. Food Res., **14**, p. 100669.

[28] Zhou, T.-Y., and Huo, X., 2024, "Learning Ability of Interpolating Deep Convolutional Neural Networks," Appl. Comput. Harmonic Anal., **68**, p. 101582.

[29] Zhang, Q., Qi, Z., Cui, P., Xie, M., and Din, J., 2023, "Detection of Single-Phase-to-Ground Faults in Distribution Networks Based on Gramian Angular Field and Improved Convolutional Neural Networks," Electr. Power Syst. Res., **221**, p. 109501.

[30] Wang, Q., Pian, F., Wang, M., Song, S., Li, Z., Shan, P., and Ma, Z., 2022, "Quantitative Analysis of Raman Spectra for Glucose Concentration in Human Blood Using Gramian Angular Field and Convolutional Neural Network," Spectrochim. Acta Part A: Mol. Biomol. Spectrosc., **275**, p. 121189.

[31] Liao, Y., Qing, X., Wang, Y., and Zhang, F., 2023, "Damage Localization for Composite Structure Using Guided Wave Signals With Gramian Angular Field Image Coding and Convolutional Neural Networks," Compos. Struct., **312**, p. 116871.

[32] Jiang, H., Deng, J., and Zhu, C., 2023, "Quantitative Analysis of Aflatoxin B1 in Moldy Peanuts Based on Near-Infrared Spectra With Two-Dimensional Convolutional Neural Network," Infrared Phys. Technol., **131**, p. 104672.

[33] Xie, J., Xu, J., Wei, Z., and Li, X., 2023, "Fault Isolating and Grading for Li-Ion Battery Packs Based on Pseudo Images and Convolutional Neural Network," Energy, **263**, p. 125867.

[34] Wang, Z., and Oates, T., 2015, "Imaging Time-Series to Improve Classification and Imputation," Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, July 25–31, pp. 3939–3945.
[35] Bai, R., Meng, Z., Xu, Q., and Fan, F., 2023, "Fractional Fourier and Time Domain Recurrence Plot Fusion Combining Convolutional Neural Network for Bearing Fault Diagnosis Under Variable Working Conditions," Reliab. Eng. Syst. Saf., **232**, p. 109076.
[36] Jiang, G., Jia, C., Nie, S., Wu, X., He, Q., and Xie, P., 2022, "Multiview Enhanced Fault Diagnosis for Wind Turbine Gearbox Bearings With Fusion of Vibration and Current Signals," Measurement, **196**, p. 111159.
[37] Pichika, S. V. V. S. N., Yadav, R., Geetha, Rajasekharan, S., Praveen, H. M., and Inturi, V., 2022, "Optimal Sensor Placement for Identifying Multicomponent Failures in a Wind Turbine Gearbox Using Integrated Condition Monitoring Scheme," Appl. Acoust., **187**, p. 108505.
[38] Rahi, P. K., and Mehra, R., 2014, "Analysis of Power Spectrum Estimation Using Welch Method for Various Window Techniques," Int. J. Emerging Technol. Eng., **2**(6), pp. 106–109.
[39] Sarraf, S., 2017, "EEG-Based Movement Imagery Classification Using Machine Learning Techniques and Welch's Power Spectral Density Estimation," Am. Sci. Res. J. Eng., Technol., Sci. (ASRJETS), **33**(1), pp. 124–145.
[40] Gecgel, O., Ekwaro-Osire, S., Dias, J. P., Serwadda, A., Alemayehu, F. M., and Nispel, A., 2019, "Gearbox Fault Diagnostics Using Deep Learning With Simulated Data," 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, June 17–20, pp. 1–8.
[41] Ali, J. B., Saidi, L., Harrath, S., Bechhoefer, E., and Benbouzid, M., 2018, "Online Automatic Diagnosis of Wind Turbine Bearings Progressive Degradations Under Real Experimental Conditions Based on Unsupervised Machine Learning," Appl. Acoust., **132**, pp. 167–181.