Energy Science & Engineering
Open Access

Wind turbine fault detection and isolation robust against data imbalance using k-Nearest Neighbors (kNN)

Journal:	*Energy Science & Engineering*
Manuscript ID	ESE-2023-06-0358
Wiley - Manuscript type:	Original Article
Search Terms:	Wind energy, Power generation, Electric power systems

Abstract:	Due to the difficulties of system modeling, nonlinearity effects, uncertainties, and the availability of wind turbines SCADA system data, data-driven Fault Detection and Isolation (FDI) methods for Wind Turbines (WTs) have received increasing attention. In this paper, using the wind turbine SCADA data, an effective FDI scheme is proposed using the KNN classifier. The operational dataset is labeled by the status and warning datasets, and the labelled operational dataset, after eliminating invalid data, feature selection, and standardization, is used for training and validation of the FDI model. Data imbalance, which is common in real data sets, almost does not affect the accuracy of the proposed method, hence there is no need for data balancing methods in this algorithm and the accuracy is not affected by false alarms. Therefore, the proposed method has provided impressive accuracy in fault detection and isolation compared to previous research on this dataset. Also, many of the fault classes addressed in this paper were not considered in previous works on this dataset.

Wind turbine fault detection and isolation robust against data imbalance using KNN

Ali Fazli[1], Javad Poshtan[1,*]

[1]Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran
aliifazli1372@gmail.com
jposhtan@iust.ac.ir

Corresponding Author:
Javad Poshtan
Iran University of Science and Technology
Narmak 16846-13114, Tehran, Iran
Tel: +98(21) 73225679
Fax: +98(21) 73225777
Email: jposhtan@iust.ac.ir

Running Title: Wind turbine fault detection using KNN

Wind turbine fault detection and isolation robust against data imbalance using KNN

Abstract. Due to the difficulties of system modeling, nonlinearity effects, uncertainties, and the availability of wind turbines SCADA system data, data-driven Fault Detection and Isolation (FDI) methods for Wind Turbines (WTs) have received increasing attention. In this paper, using the wind turbine SCADA data, an effective FDI scheme is proposed using the KNN classifier. The operational dataset is labeled by the status and warning datasets, and the labeled operational dataset, after eliminating invalid data, feature selection, and standardization, is used for training and validation of the FDI model. Data imbalance, which is common in real data sets, almost does not affect the accuracy of the proposed method, hence there is no need for data balancing methods in this algorithm and the accuracy is not affected by false alarms. Therefore, the proposed method has provided impressive accuracy in fault detection and isolation compared to previous research on this dataset. Also, many of the fault classes addressed in this paper were not considered in previous works on this dataset.

Keywords: Fault Detection and Isolation, KNN, SCADA data, Wind Turbine, Data imbalance

1. Introduction

Wind energy which is renewable and compatible with the environment is developing rapidly and in this regard, large-scale WTs are used in various countries. Therefore, in recent years, the monitoring and maintenance of WTs have attracted much attention. In a harsh environment, fault detection and condition monitoring are necessary for WTs safety and reliability [1]. Fault scenarios in WTs include sensor faults, actuator faults, and system faults [2]. Common faults and failures in WTs and the various methods that researchers use to diagnose them have been investigated and classified in [1], [3], [4]. Various wind turbine fault prognosis approaches including model-based, data-driven, knowledge-based, Artificial Intelligence (AI)-based, stochastic-based, and hybrid methods are reviewed in [4], [5]. From one perspective, fault detection and isolation methods in WTs are divided into three main categories: model-based, data-driven, and hybrid approaches. In model-based methods, a mathematical model of the WT or its subsystem produces a simulated output based on the measured input signal. By comparing the actual output of WT with the estimated output, a residual is produced to be used for fault detection and isolation [6]. Due to the system modeling difficulties and the availability of sensors data, data-driven methods in fault detection of WTs have gained increasing interest compared to the model-based approaches [7]. Data-driven FDI methods in wind turbines include acoustic emission analysis, noise analysis, oil analysis, vibration analysis, machine learning (data mining) methods, and hybrid methods [1]. Major disadvantages of acoustic emission analysis and oil analysis methods are the need for background noise to be shielded and limitation to bearings with closed-loop oil supply system, respectively [8]. Vibration analysis [9], [10] which is an effective tool for the condition monitoring and fault diagnosis of WT drivetrains [11], includes synchronous resampling, Hilbert transform, wavelet transform, statistical analysis, envelope analysis, fast Fourier transform (FFT), and short-term fast Fourier transform (STFT) [6]. Among these methods, FFT is the most important tool, and wavelet analysis has become increasingly popular [12]. Despite the reliability and standardization of vibration analysis methods, they are expensive, intrusive, subject to sensor failures, and have limited performance for low-speed rotation [8].

Machine learning algorithms are divided into two main categories: supervised learning which consists of regression-based and classification-based approaches, and unsupervised learning algorithms such as clustering [13]. If the dataset is labeled, supervised learning algorithms are used, otherwise, unsupervised learning could be a good solution. If labels are categorical, classifiers are used, otherwise, if labels are numerical, regression must be used [13]. In regression methods, which are divided into parametric and non-parametric modeling techniques, an artificial intelligence (AI) model learns the dynamics of a WT or a WT subsystem from the patterns of input and output collected from a dataset [6]. By comparing the predicted behavior of this model with the WT measured actual behavior, possible faults can be detected from the changes in the WT behavior [14]. On the other hand, in the classification methods, an AI model is trained to recognize the patterns (e.g., mode, location, and severity) of different WT faults from the input signals containing the faults information [6]. The main steps of model training in machine learning are data acquisition, preprocessing, feature selection and extraction, model selection, and validation; specifically, these steps for classification are data acquisition, preprocessing, equalization of the classes, feature selection and extraction, model fitting, cross-validation, and using the best model based on validation [13]. A deep learning model is used for feature extraction in [15]. The amount of normal data is much more than that of abnormal data in SCADA datasets, which makes FDI models tend to be biased toward the majority class, i.e., normal data, leading to poor accuracy in diagnosing faults [16]. Several methods such as setting misclassification cost, undersampling majority class, oversampling minority class (e.g. synthetic minority oversampling technique (SMOTE) and adaptive synthetic sampling method (ADASYN) [17]), Tomek links, and cluster centers are used to overcome data imbalance challenge [18]. However, the main focus is to use an appropriate method from the existing methods or develop new algorithms according to the need, because if one technique works well on an imbalanced dataset, it may not work for another imbalanced data set [19].

The main machine learning models already used in the wind turbine FDI problem are Artificial Neural Networks (ANNs), Support Vector Machines (SVM), classifier fusion, and ensemble classifiers. The most common types of neural networks in this area are Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) [20]. Wind turbine SCADA data includes multivariate time series that have temporal correlations within each sensor variable and spatial correlations between different sensor variables. To effectively capture these correlations, in [21] two parallel modules are used for feature extraction; first a Multi-Scale Deep Echo State Network (MSDeepESN) for extracting temporal multi-scale features, and second, a Multi-Scale Residual Network (MSResNet) that can extract spatial multi-scale features. Due to the extreme data imbalance, the focal loss function is used in [21] to reduce its effect on model training. In [22], first, a Multi-Kernel Fusion Convolution Neural Network (MKFCNN) is designed to extract multi-scale spatial correlations of different features, then the Long Short-term Memory (LSTM) is used for learning the temporal correlations among the learned spatial features. After applying a down-sampling method to the training data in [22] to balance normal and faulty samples, only 210 observations from 24108 normal samples are used for the training process. A novel fault diagnosis framework is presented in [23] based on an end-to-end LSTM model, to learn features directly from multivariate time-series data, and capture long-term dependencies through the recurrent behavior and gates mechanism of LSTM. Although using temporal correlations for the training and validation of the FDI system can provide useful information, it leads to the problem of the algorithm's dependence on historical data for fault detection and isolation. Taking into account two new fault scenarios, in [24], time-domain signals recorded from wind turbine simulations are portrayed as images and fed to a multi-channel CNN, which is used for the first time in wind turbine fault detection and isolation. Using a wind turbine real SCADA data based on the support vector machine learning method, a fault diagnosis and prediction model is presented in [18], and despite the use of various data balancing methods, the desired accuracy is not reached because of the lack of attention to the fact that transient observations before and after fault occurrence must be removed. As an alternative to the SVM technique, the GPC, as a Bayesian nonparametric classification method, which gets away from any assumptions about the structural relationship between inputs and the output, provides probabilistic information about the fault types, which is valuable for making maintenance plans [25]. Using an under-sampling method, just 460 samples are representative of the 24108 normal observations of the training data. Therefore, a lot of information is lost by ignoring a large number of samples in the model training process in [18], [22], [25]. A data-driven FDI scheme is proposed in [26] based on the fusion of the decision tree, SVM, KNN, Radial Basis Function (RBF), and MLP classifiers. Although this algorithm is robust against different operational conditions and measurement errors, according to the confusion matrix of [26], it is biased for some minority classes toward the majority class. On the other hand, using data balancing methods may prevent faulty observations from being detected as normal samples, but it can result in false alarms according to confusion matrices of [18], [21], [22], [25]. Figure 1 shows various methods of detection and isolation of WT faults.

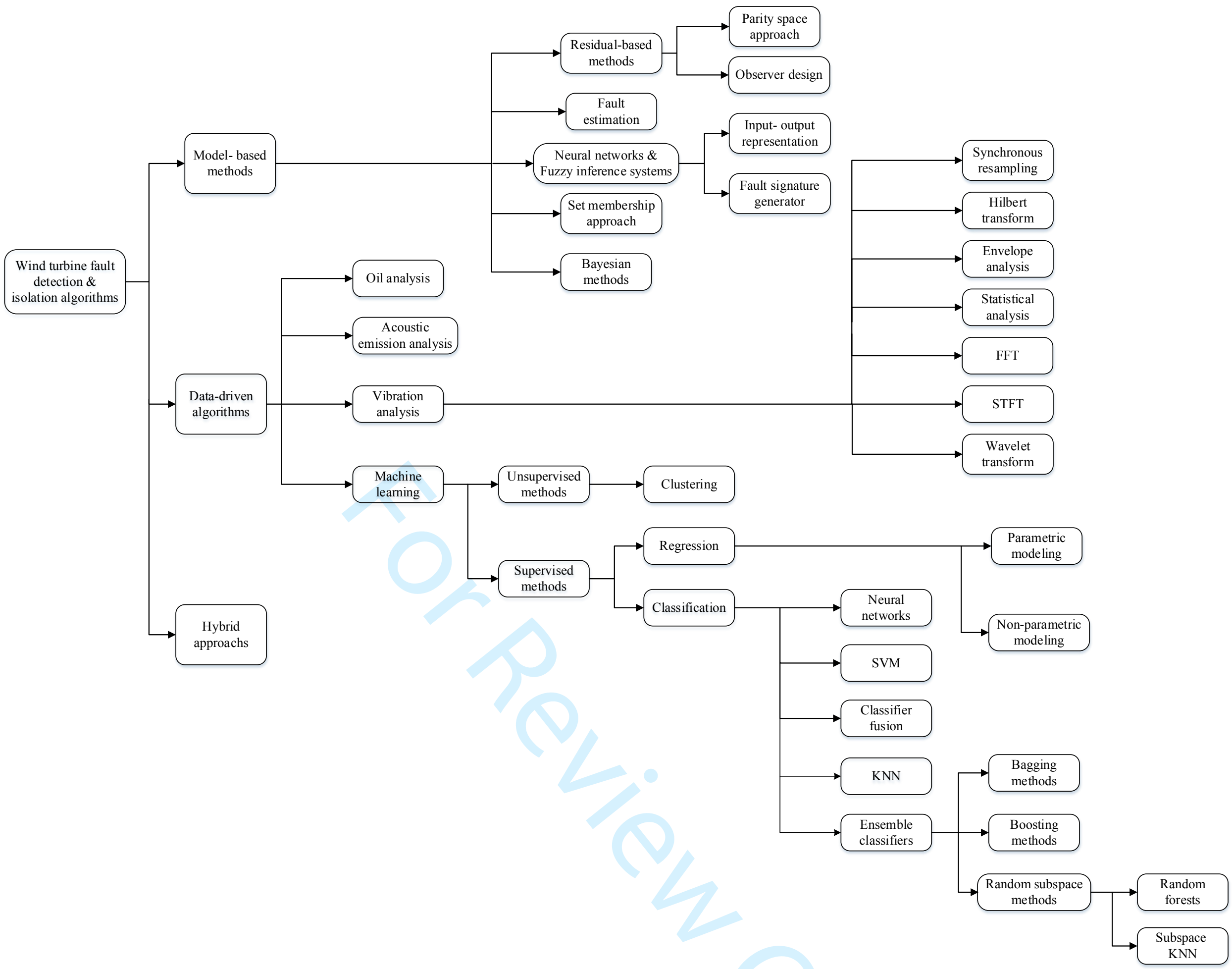

Figure 1. Various methods used for WT fault detection and isolation

In this paper, an FDI method is presented using wind turbine real SCADA data. First, operational SCADA data is labeled using messages of the status and warning datasets. Note that each status and warning dataset includes various faults, while in previous researches only some of the status dataset faults have been considered. Second, informative features are selected from the original features set to form an appropriate feature set. Third, the features selected are standardized in such a way that each feature has zero mean and unit standard deviation, and then the preprocessed operational dataset acts as the KNN input. Finally, WT fault detection and isolation is performed and the holdout validation method verifies remarkable accuracy of the KNN method. The motivation of this paper is to present a data-driven method for wind turbine fault detection and isolation so that the problem of data imbalance as an important challenge of the literature is solved using the potential of machine learning methods without using data balancing techniques. As an example, in the under-sampling method only some sample observations of the majority class, as a representative of this class, are used for the training process, and it is obvious that these samples can not fully represent the behavior of the no-fault (majority) class. Using data balancing methods for handling imbalanced datasets can lead to false alarms, as it can be seen in the confusion matrices of [18], [21], [22], [25]. This is while the proposed algorithm can perform the training and validation processes without the requirement for these interface methods and with the benefit of the capacity of all dataset observations. The contributions of this paper are:

1) The Model is almost robust against imbalanced data, which prevents the classifier from tending to the majority class. Imbalanced classification is one of the most important challenges of the wind turbine FDI methods because faults occur infrequently compared to the healthy operation.
2) Due to the algorithm's robustness against imbalanced data, there is no need for data balancing methods. Therefore, the classifier's accuracy is not affected by false alarms.
3) These two advantages make the algorithm have excellent accuracy to detect and isolate wind turbine faults.
4) Although the collected dataset which is used for training the FDI system is offline, the algorithm determines the labels of the new data observations online.
5) The proposed fault detection and isolation system determines the label of the new data observation only with its current feature values, without any requirement for historical data.
6) Existing SCADA data is used to train and validate the algorithm, so there is no extra cost to collect and record the data.
7) A large number of status and warning faults are considered (in comparison with previous works on this SCADA data [18], [21], [22], [25]).
8) The dependence between different faults based on physical facts and the overlap of these faults' occurrence is analyzed.

The article structure is as follows: Section 2 describes the wind turbine SCADA dataset. In section 3 data preprocessing, including labeling, eliminating invalid data, feature selection, and standardization is explained. Section 4 presents the parameter selection of the WT fault detection and isolation model, and Section 5 includes KNN training and validation results. Finally, in section 6 conclusion and future work are discussed. Figure 2 shows the flowchart of the proposed scheme for WT fault detection and isolation.

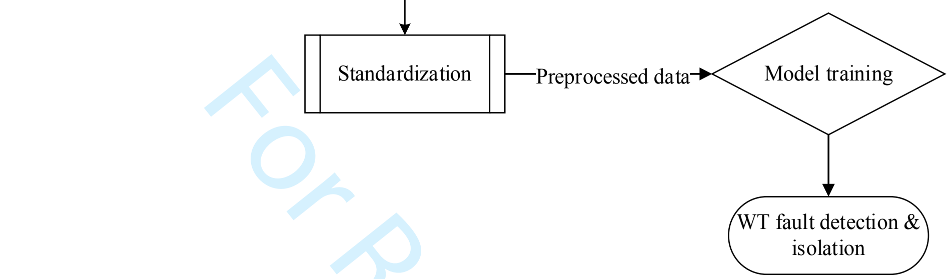

Figure 2. Flowchart of the proposed scheme for WT fault detection and isolation.

2. Data description

The real SCADA data in this study comes from a 3-MW direct-drive wind turbine that supplies the required power of a major biomedical manufacturing plant located off the southern coast of Ireland [18], [21], [22], [25], [27]. This 3-blade wind turbine comprises common major components, including main shaft, generator, rotor, turbine blades, yaw system, pitch system, control and power electronics system, hydraulic and cooling system, etc. The data include observations over an 11-month period from May 2014 to April 2015 which contains three separate datasets:

1) Operational data including 49027 samples with a 10-minute time interval and 66 features such as ambient parameters (e.g. wind speed and ambient temperature), power measurements (e.g. active power and reactive power), WT components recorded temperature (e.g. tower temperature, nacelle cabinet temperature, etc.), and operating conditions (e.g. nacelle position including cable twisting, operating hours).
2) Status data that represents the operating conditions of the wind turbine and is divided into two sets of data: wind energy converter (WEC) and remote terminal unit (RTU). The WEC data includes status messages that are directly related to the WT itself, while the RTU data corresponds to the power control data at the point of connection to the grid, such as active and reactive power. In the status dataset, a status message with a new timestamp is generated each time the status changes. Therefore, it is assumed that the turbine operates

in that mode until the next status message is generated. Each turbine state has a "main status" code and a "sub-status" code, where each main status code greater than zero indicates an abnormal condition and not necessarily a fault; For example, status code 2 ("lack of wind") indicates the lack of sufficient wind for normal operation of a wind turbine.

3) Warning data that corresponds to the general information about the turbine and is not directly related to the turbine operation or safety. Warning messages are about potentially developing faults in the turbine. These messages have timestamps in the same way as status messages and also have a "main warning" code and a "sub-warning" code. This data is also divided into two sets of WEC data and RTU data. The WEC data includes warning messages that are directly related to the turbine itself, while the RTU data is related to the power control data at the point of connection to the grid.

3. Preprocessing

Before applying the FDI model to the operational data, as its input, some steps must be taken. First, considering the messages of the status and warning datasets, the operational dataset is labeled. Then some observations must be removed to prevent feeding the model with invalid data samples. In the next step, non-informative features are removed and z-score normalization is applied to the selected set of features. Note that one of the most important preprocessing steps in classification-based FDI problems, called data balancing, is eliminated by taking advantage of using the KNN model.

3.1. Data labeling

The wind turbine operational dataset is labeled based on the information in the status and warning datasets so that the observations during faults occurrence are distinguished from other observations. Table 1 shows status data faults along with status and sub-status codes, and the symbols assigned to each fault class so that fault classification and evaluation are performed easily. Faults that have not been addressed in previous works on this dataset are highlighted in green. Feeding faults refer to faults in the power feeder cables of the WT, excitation errors are mainly due to problems with the generator excitation system, malfunction air-cooling indicates problems in the air circulation and internal temperature circulation in the WT, mains failure is related to problems with mains electricity supply to the WT, and generator heating faults refer to the generator overheating. Other faults are WT cable twist, inverter over-temperature, WT tower transversal oscillation, fan-inverter malfunction, and yaw control fault. Among these, Feeding fault and Excitation error have overlaps in 95 samples. Also, 12 samples have the labels of Malfunction air cooling and Mains failure at the same time. As these overlaps cannot be ignored and include a significant proportion of the number of samples of each fault, either multi-label classification methods should be used or the overlap section should be considered a separate class. The second solution is adopted, and for this purpose, the modified classes are symbolized as EFF and AMF, respectively. In Table 2 several faults extracted from the WT warning data and used to label the operational data are shown. None of these faults have been investigated in previous researches on this dataset and hence all are highlighted in green.

According to the symbols defined for each fault, Table 3 presents the number of overlapping samples between the faults defined in Tables 1 and 2. Overlaps with less than 10% of both faults samples are negligible and not used in the training and testing of the proposed method. For example, without considering overlap samples, the classes BF and FA with 144 and 271 samples, respectively, have 49 common samples, which exceeds 10% of each class sample size; hence this overlap cannot be ignored. On the other hand, some faults in nature are so close together that have the same main code. Accordingly, faults that are conceptually very close to each other can also be considered as an integrated class. Therefore, two types of class integration have been considered in this paper: first, class integration based on the number of common (overlap) samples in the operational data, and second, class integration based on the nature of faults. Table 4 lists the faults that are merged accordingly. Therefore, the final fault labels used for data labeling are given in Table 5 and the relative frequency of these classes is shown in Figure 3 using a pie chart. Figure 4 shows the scatter plot of the wind turbine's faulty samples on the power curve according to labels presented in Table 5.

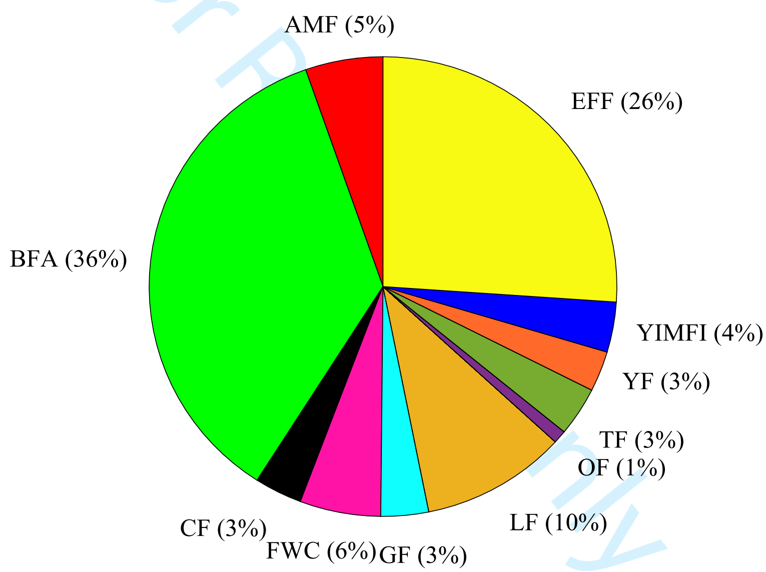

Figure 3. Relative frequency of the WT considered faults

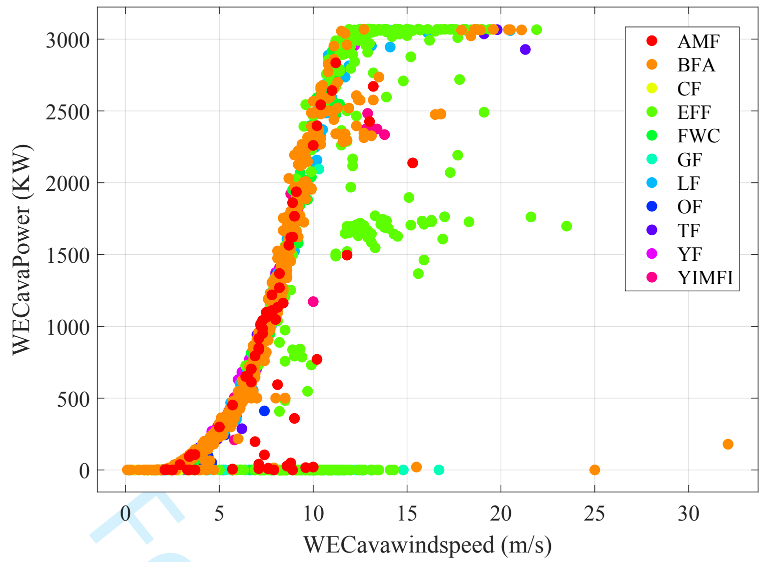

Figure 4. Scatter plot of the wind turbine faulty operational data on the power curve

3.2. Invalid data elimination

Some observations which indicate abnormal behavior of the wind turbine are not labeled with fault labels, because many of these are not associated with a fault, e.g. status code 2 –"lack of wind". Some of these abnormal conditions such as warning code 230 –"Power limitation (10h)" are not considered as the no-fault class in data labeling. According to [22], at least 120 minutes before the change from the normal state and 30 minutes after the change to the normal state should be considered as transient state data. This is due to the fact that, unlike in simulation, in a real SCADA dataset, the change from normal to abnormal performance (and vice versa) occurs with soft behavior, so the data in this transition state should not be considered as normal WT performance data. In this paper with less conservative data cleaning, at least 60 minutes before the change from the normal state and 20 minutes after the change to the normal state are considered as transient observations. Also, sometimes a fault or a group of faults occur intermittently with a small time interval, which indicates that the problem of the turbine is not solved between consecutive faults. Therefore after filtering invalid normal data, we have 34472 observations as the no-fault class which is labeled with the symbol NF. Note that in [21], [22], [25] only 32056 observations are considered as normal class. This means that in this paper, the normal class is more in line with the conditions provided by the SCADA wind turbine dataset. Figure 5 shows the scatter plot of the wind turbine's normal operational data on the power curve.

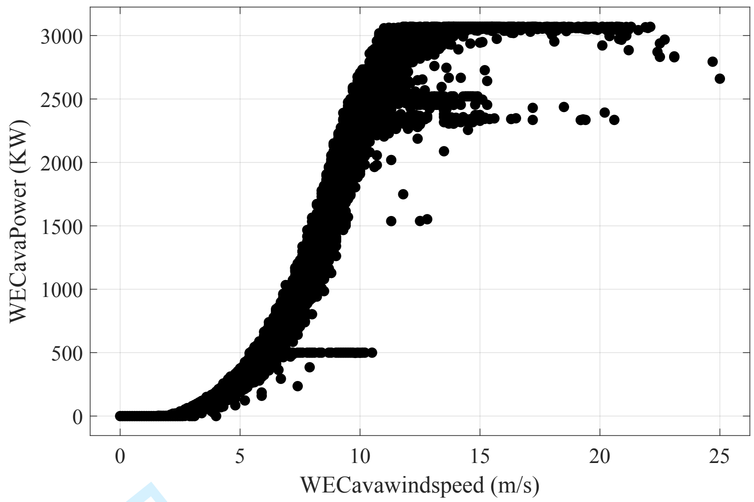

Figure 5. Scatter plot of the wind turbine normal operational data on the power curve

3.3. Feature selection and standardization

Some of the 66 variables of operational data must be used in the fault detection and isolation process. In short, the following steps are taken to select appropriate features, resulting in the final set of features displayed in Table 6.

- ✓ Date Time is not a numeric feature and is eliminated.
- ✓ Variables such as Time, WEC: Operating Hours, WEC: Production kWh, and WEC: Production minutes that are accumulative features result in time-based fault detection and isolation if considered in the training and validation process. Therefore, even if the accuracy obtained is high, it will not be reliable. For example, if a specific fault occurs in March in the training set, the test sample can only be correctly predicted if it occurs in the same month. Hence, for preventing the algorithm from time-based bias these features are not considered in the FDI process.
- ✓ For variables recorded as minimum, maximum, and average, it is better to use only the average because the minimum and maximum values may be affected by outliers and deteriorate the accuracy of the algorithm. Wind speed, Rotation, Power, and Reactive power are of this type.
- ✓ Some parameters including Sys 1 inverter cabinet temp, Bearing temp, Pitch cabinet blade temp, Blade temp, Rotor temp, Stator temp, Nacelle ambient temp, and Inverter cabinet temp are recorded by more than one sensor (seven, two, three, three, two, two, two, and two respectively). Therefore, to prevent the curse of dimensionality, which is an important challenge in the KNN algorithm, these variables are merged into one variable by averaging. For example, Front bearing temp and Rear bearing temp are variables that result in one variable (Bearing temp) by averaging.
- ✓ Some non-informative features that reduce the accuracy of the algorithm are ignored.

Numerical features must be Standardized before feeding to the KNN algorithm to contribute equally to the similarity measures. Standardization (z-score normalization) is a scaling method where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero, and the resultant distribution has a unit standard deviation, that is:

$$X_i^{'} = \frac{X_i - \mu_i}{\sigma_i} \tag{2}$$

where X_i represents the i-th original feature, μ_i and σ_i are its mean and standard deviation respectively, and $X_i^{'}$ is the standardized i-th feature for i=1,…,25.

4. KNN parameter selection

Considering the final fault classes, presented in Table 5, the preprocessed operational data is used to determine the FDI model parameter based on the classification error. A KNN classifier is trained for different values of the parameter K. Figure 6 shows the effect of changes in this parameter on the classification error by the 10-fold validation method. According to this figure, increasing the value of parameter K results in increasing the classification error, so $K=1$ is the best choice for this classifier. Choosing this value usually can yield overfitting, hence to avoid this, the holdout validation is adopted in the procedure of the wind turbine FDI model design. As shown in Figure 6, the KNN classifier can well classify different classes of wind turbine's imbalanced SCADA dataset.

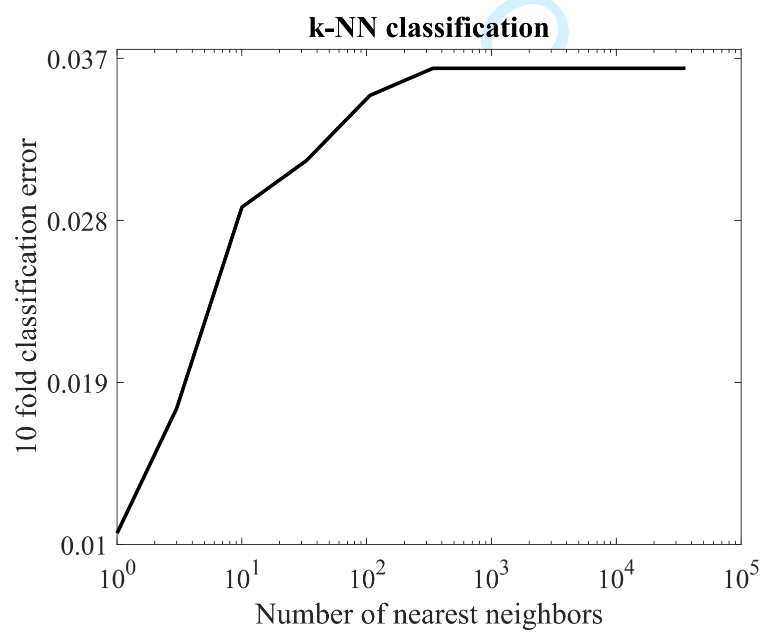

Figure 6. Classification error for different values of parameter K

Many references confirm that the performance of KNN is affected by data imbalance [28]–[30]. Conceptually the KNN algorithm calculates the (Euclidian) distances of the training dataset observations from a validation sample (that is to be labeled) and assigns the majority label of K nearest neighbors of the validation sample to it. Therefore, if the observations of each class (in the feature space of the dataset) have little distance from each other and a small parameter K is selected, the performance of this algorithm will not be affected by data imbalance. In other words, if the distance of samples within a class is less than the distance between samples of different classes, and a small parameter K is selected, this algorithm will be robust against data imbalance. For example, consider that one class of the dataset has 1000 samples and another class has only 2 observations, but these 2 samples are well close to each other in the feature space. In this case, considering $K = 1$, the KNN algorithm will assign a true label to those two samples and will not be affected by the data imbalance. Also, according to Figure 6, by increasing the parameter K, the error reaches 0.03645, which means that the algorithm tends to the majority class: Note that 0.03645 * 35776=1304, exactly equivalent to the false prediction of the minority classes samples (1304 fault observations).

5. Training and validation

After choosing suitable parameters according to the previous section's explanations, the KNN model is applied to the train data, which is 80% of the data. Therefore, 20% of the data is used as validation data, employing the holdout validation method, thus none of the validation data samples are involved in the training process. Considering the faults given in Table 5, the KNN model results in an impressive performance as shown in Figure 7. This confusion chart illustrates that almost all samples of the validation data are assigned properly to the true classes. The following parameters are used to clarify this performance:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3}$$

$$TPR_{ave} = \frac{\sum_{m=1}^{C} TPR_m}{C} \tag{4}$$

$PPV =$

$$PPV_{ave} = \frac{\sum_{m=1}^{C} PPV_m}{C} \tag{6}$$

where C is the number of classes including no-fault and faulty categories, TP is the number of true predictions of the class, FN represents the number of false predictions of the class, P indicates the number of all samples of the class, and FP is the representative of the false alarms of the class. Note that average TPR in (4) and average PPV in (6), which are proposed as validation parameters in this paper, can be useful for the validation of the imbalance classification. The values of TPR and PPV obtained for all classes are greater than 77.8% and 75% respectively according to Figure 7, confirming the remarkable performance of the proposed model in the WT fault detection and isolation is confirmed. Note that the training and validation data are chosen randomly based on the 80-20 proportions. So, each time the training and validation process takes place, the results may be slightly different. In multiple runs, however, the number of false predictions does not exceed 75, which is a good achievement compared to the previously reported methods. Figure 7 also confirms this statement because almost 0.01062 10-fold classification error is equal to 0.01062*0.2*35776≈76 false predictions in the holdout validation. Table 7 compares the proposed FDI model with related previous works on this dataset, in terms of accuracy, the number of the fault classes considered, the number of the normal class samples, overall accuracy (the ratio of the number of the whole true predictions to the number of the whole false predictions), average *TPR*, average *PPV*, and requirement of the method to data balancing. According to the comparison made in [21] and [22], different algorithms including several deep learning methods such as CNN, LSTM, MKFCNN, CNN-LSTM, CNN-GRU, MSDeepESN, and MSResNet neural networks, as well as shallow machine learning techniques such as random forests (RF), SVM, decision tree (DT), and GPC are adopted for comparison on the present SCADA dataset. Note that overall accuracy is not a good validation criterion for imbalance classification, because a case such as "good *TPR* for no-fault class and bad prediction for other classes" also can result in high overall accuracy.

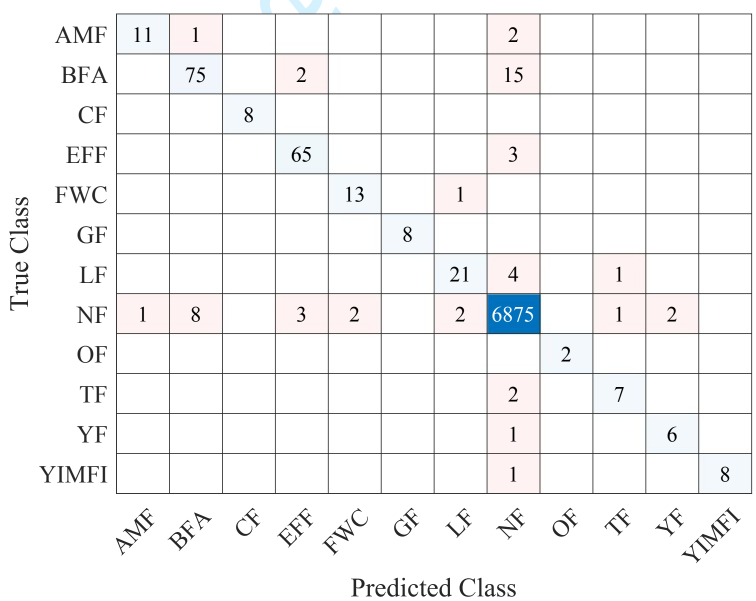

Figure 7. The confusion chart of the proposed model validation

6. Conclusion

In this paper, the KNN model is used for WT fault detection and isolation. For the purpose of preprocessing, four steps of data labeling, invalid data elimination, feature selection, and standardization are applied to the raw SCADA dataset of a wind turbine, respectively. It was shown that the proposed method is robust against the data imbalance challenge, which prevents the classifier from tending to the majority class. Due to the algorithm's robustness against imbalanced data, there is no need for data balancing methods. Therefore, the classifier's accuracy is not affected by false alarms. As a result of the above, the proposed model shows higher accuracy in the WT fault detection and isolation, compared to previous researches on this dataset. The proposed FDI system can be used for online monitoring, unlike the training step which is an offline process. This system determines the label of the new data observation only with its current feature values, without any requirement for historical data. Although using temporal correlations for the training and validation of the FDI system can provide useful information, it leads to the problem of the algorithm's dependence on historical data for fault detection and isolation. Since the existing SCADA data was used to train and validate the algorithm, there is no extra cost to collect and record the data. It is worth mentioning that many faults are considered, especially from the warning data, which are not indicated in the previous related works. Finally, the dependence of the faults is analyzed based on the nature of the faults and the overlap of their occurrence. As a future work, predictive maintenance can be pursued by developing the existing model to predict wind turbine faults.

References

[1] A. Purarjomandlangrudi, G. Nourbakhsh, M. Esmalifalak, and A. Tan, "Fault detection in wind Turbine: A systematic literature review," *Wind Eng.*, vol. 37, no. 5, pp. 535–546, 2013, doi: 10.1260/0309-524X.37.5.535.

[2] D. Yu, Z. M. Chen, K. S. Xiahou, M. S. Li, T. Y. Ji, and Q. H. Wu, "A radically data-driven method for fault detection and diagnosis in wind turbines," *Int. J. Electr. Power Energy Syst.*, vol. 99, pp. 577–584, 2018, doi: 10.1016/j.ijepes.2018.01.009.

[3] Y. Amirat, M. E. H. Benbouzid, E. Al-Ahmar, B. Bensaker, and S. Turri, "A brief status on condition monitoring and fault diagnosis in wind energy conversion systems," *Renew. Sustain. Energy Rev.*, vol. 13, no. 9, pp. 2629–2636, 2009, doi: 10.1016/j.rser.2009.06.031.

[4] M. Rezamand, M. Kordestani, R. Carriveau, D. S. K. Ting, M. E. Orchard, and M. Saif, "Critical Wind Turbine Components Prognostics: A Comprehensive Review," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9306–9328, 2020, doi: 10.1109/TIM.2020.3030165.

[5] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure Prognosis and Applications - A Survey of Recent Literature," *IEEE Trans. Reliab.*, vol. 70, no. 2, pp. 728–748, 2021, doi: 10.1109/TR.2019.2930195.

[6] W. Qiao and D. Lu, "A Survey on Wind Turbine Condition Monitoring and Fault Diagnosis - Part II: Signals and Signal Processing Methods," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6546–6557, 2015, doi: 10.1109/TIE.2015.2422394.

[7] G. Jiang, P. Xie, H. He, and J. Yan, "Wind Turbine Fault Detection Using a Denoising Autoencoder with Temporal Information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, 2018, doi: 10.1109/TMECH.2017.2759301.

[8] S. M. M. Aval and A. Ahadi, "Wind turbine fault diagnosis techniques and related algorithms," *Int. J. Renew. Energy Res.*, vol. 6, no. 1, pp. 80–89, 2016.

[9] M. He and D. He, "A new hybrid deep signal processing approach for bearing fault diagnosis using

vibration signals," *Neurocomputing*, vol. 396, pp. 542–555, 2020, doi: 10.1016/j.neucom.2018.12.088.

[10] C. Liu, G. Cheng, B. Liu, and X. Chen, "Bearing fault diagnosis method with unknown variable speed based on multi-curve extraction and selection," *Meas. J. Int. Meas. Confed.*, vol. 153, 2020, doi: 10.1016/j.measurement.2019.107437.

[11] W. Teng, X. Ding, S. Tang, J. Xu, B. Shi, and Y. Liu, "Vibration analysis for fault detection of wind turbine drivetrains—a comprehensive investigation," *Sensors*, vol. 21, no. 5, pp. 1–33, 2021, doi: 10.3390/s21051686.

[12] D. Strömbergsson, P. Marklund, K. Berglund, and P. E. Larsson, "Bearing monitoring in the wind turbine drivetrain: A comparative study of the FFT and wavelet transforms," *Wind Energy*, vol. 23, no. 6, pp. 1381–1393, 2020, doi: 10.1002/we.2491.

[13] A. Stetco *et al.*, "Machine learning methods for wind turbine condition monitoring: A review," *Renew. Energy*, vol. 133, pp. 620–635, 2019, doi: 10.1016/j.renene.2018.10.047.

[14] A. Zaher, S. D. J. McArthur, D. G. Infield, and Y. Patel, "Online Wind Turbine Fault Detection through Automated SCADA Data Analysis," *Wind ENERGY*, vol. 12, no. 6, pp. 574–593, 2009, doi: 10.1002/we.319.

[15] Y. Liu, H. Cheng, X. Kong, Q. Wang, and H. Cui, "Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning," *Energy Sci. Eng.*, vol. 7, no. 6, pp. 2633–2645, 2019, doi: 10.1002/ese3.449.

[16] L. Chen, G. Xu, Q. Zhang, and X. Zhang, "Learning deep representation of imbalanced SCADA data for fault detection of wind turbines," *Meas. J. Int. Meas. Confed.*, vol. 139, pp. 370–379, 2019, doi: 10.1016/j.measurement.2019.03.029.

[17] A. Burkov, *The Hundred-Page Machine Learning Book*. 2019. doi: 10.1080/15228053.2020.1766224.

[18] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos, A. M. Agogino, and D. T. J. O'Sullivan, "Diagnosing and Predicting Wind Turbine Faults from SCADA Data Using Support Vector Machines," *Int. J. Progn. Heal. Manag.*, vol. 9, no. 1, pp. 1–11, 2018.

[19] N. Rout, D. Mishra, and M. K. Mallick, "Handling Imbalanced Data : A Survey," in *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, 2018, pp. 431–443.

[20] G. Helbing and M. Ritter, "Deep Learning for fault detection in wind turbines," *Renew. Sustain. Energy Rev.*, vol. 98, pp. 189–198, 2018, doi: 10.1016/j.rser.2018.09.012.

[21] Q. He, Y. Pang, G. Jiang, and P. Xie, "A spatio-temporal multiscale neural network approach for wind turbine fault diagnosis with imbalanced SCADA data," *IEEE Trans. Ind. Informatics*, pp. 1–9, 2020, doi: 10.1109/TII.2020.3041114.

[22] Y. Pang, Q. He, G. Jiang, and P. Xie, "Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on SCADA data," *Renew. Energy*, vol. 161, pp. 510–524, 2020, doi: 10.1016/j.renene.2020.06.154.

[23] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renew. Energy*, vol. 133, pp. 422–432, 2019, doi: 10.1016/j.renene.2018.10.031.

[24] S. Zare and M. Ayati, "Simultaneous fault diagnosis of wind turbine using multichannel

convolutional neural networks," *ISA Trans.*, vol. 108, pp. 230–239, 2021, doi: 10.1016/j.isatra.2020.08.021.

[25] Y. Li, S. Liu, and L. Shu, "Wind turbine fault diagnosis based on Gaussian process classifiers applied to operational data," *Renew. Energy*, vol. 134, pp. 357–366, 2019, doi: 10.1016/j.renene.2018.10.088.

[26] V. Pashazadeh, F. R. Salmasi, and B. N. Araabi, "Data driven sensor and actuator fault detection and isolation in wind turbine using classifier fusion," *Renew. Energy*, vol. 116, pp. 99–106, 2018, doi: 10.1016/j.renene.2017.03.051.

[27] R. L. Hu, K. Leahy, I. C. Konstantakopoulos, D. M. Auslander, C. J. Spanos, and A. M. Agogino, "Using domain knowledge features for wind turbine diagnostics," in *Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*, 2017, pp. 300–305. doi: 10.1109/ICMLA.2016.172.

[28] M. E. Kadir, P. S. Akash, S. Sharmin, A. A. Ali, and M. Shoyaib, "A Proximity Weighted Evidential k Nearest Neighbor Classifier for Imbalanced Data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2020, pp. 71–83. doi: 10.1007/978-3-030-47436-2_6.

[29] K. Kim, "Normalized class coherence change-based kNN for classification of imbalanced data," *Pattern Recognit.*, vol. 120, 2021, doi: 10.1016/j.patcog.2021.108126.

[30] S. Zhang, "Cost-sensitive KNN classification," *Neurocomputing*, vol. 391, pp. 234–242, 2020, doi: 10.1016/j.neucom.2018.11.101.

Table 1. Status data faults

Fault	Status code	Sub-status code	Number of samples	Symbol
Generator heating: Hygrostat inverter	9	3	43	GF
Feeding fault	62	various	260	FF
Excitation error : Overvoltage DC-link	80	21	175	EF
Timeout warn message : Malfunction air cooling	228	100	62	AF
Mains failure : Under voltage L1/ Start delay	60	11/2	20	MF
Cable twisted : Left/Right (2-3 turns)	21	1/2	43	CF
Over-temperature : Power choke inverter 4	67	604	12	OF
Tower oscillation : Transversal oscillation	31	1	3	TF
Malfunction fan-inverter: Other control board errors system 3	26	373	3	IF
Yaw control fault: TWK NOCN: fault in cam switch mechanism	22	226	5	YF

Table 2. Warning data faults

Fault	Warning code	Sub-warning code	Number of samples	Symbol
Faulty yaw inverter: Rotational speed error motor x/DC-link voltage instabil system y/Feedback mains contactor faulty(-)	25	various	28	YIF
Yaw control fault : TWK NOCN installed	22	200	32	YCF
Fault blade load control: Blade load curve A/B/C too low (28)	49	193/293/393	195	BF
Fault air cooling: Generator pressure to low (7)	100	11	327	FA
Fault water cooling: Pressure too low (7)/ No flow rate (7)	101	1/5	85	FWC
Fault lubrication system: Grease reservoir empty (rotor) (90)	58	1	140	LF
Tower oscillation: Longitudinal oscill. (low wind speed)	31	32	42	TLF
Malfunction fan-inverter: DC-link voltage instable system 3/ Circuit breaker nacelle fan tripped	26	316, 331, 335	26	MFI

Table 3. Number of the faults overlapping samples

Overlap fault	Number of each fault samples without considering the overlap	Number of overlap samples	Symbol	Can be ignored
BF+FA	144+271	49	BFA	No
YIF+MFI	19+19	5	YIMFI	No
FA+FWC	271+73	5	FAWC	Yes
LF+FWC	131+73	7	LFWC	Yes
LF+FA	131+271	2	LFA	Yes
BF+YIF	144+19	2	BYIF	Yes

AF+MF	50+8	12	AMF	No
EF+FF	80+165	95	EFF	No

Table 4. Integrated classes based on the fault nature

Fault 1	Number of samples	Fault 2	Number of samples	Integrated class symbol	Number of samples
YF	5	YCF	32	YF	37
TF	3	TLF	42	TF	45
IF	3	MFI	19	MFI	22

Table 5. Final fault labels

Fault	Decomposition	Number of samples
BFA	BF+FA	464
YIMFI	YIF+MFI+IF	46
EFF	EF+FF	340
AMF	AF+MF	70
TF	TF+TLF	45
YF	YF+YCF	37
GF	GF	43
CF	CF	43
OF	OF	12
FWC	FWC	73
LF	LF	131

Table 6. Final SCADA variables

No.	Variable Name	No.	Variable Name
1	Ava wind speed	14	Nacelle ambient temp (Averaged)
2	Ava Rotation	15	Nacelle temp
3	Ava Power	16	Nacelle cabinet temp
4	Ava Nacelle's position including cable twisting	17	Main carrier temp
5	Ava Reactive Power	18	Rectifier cabinet temp
6	Ava blade angle A	19	Inverter cabinet temp (Averaged)
7	Sys1 inverter cabinet temp (Averaged)	20	Ambient temp
8	Spinner temp	21	Tower temp
9	Bearing temp (Averaged)	22	Control cabinet temp
10	Pitch cabinet blade temp (Averaged)	23	Transformer temp
11	Blade temp (Averaged)	24	Inverter averages
12	Rotor temp (Averaged)	25	Inverter std dev

13	Stator temp (Averaged)	26	Status (Label)

Table 7. Comparison of the proposed FDI model with related previous works on this dataset

FDI model	Number of fault classes	Number of normal class samples	Overall accurac y	Average TPR	Average PPV	The requirement for data balancing
Proposed model	11	35870	99.27%	90.13%	91.71%	No
STMNN [21]	5	32056	94.93%	90.64%	82.68%	Yes
STFNN [22]	5	32056	93.84%	84.28%	63.74%	Yes
[25]	5	32056	91.79%	78.87%	44.40%	Yes
[18]	4	48503	88.67%	74.99%	40.98%	Yes
RF [22]	5	32056	84.75%	60.95%	46.09%	Yes
SVM [22]	5	32056	72.05%	64.19%	37.24%	Yes
DT [22]	5	32056	80.54%	62.24%	36.57%	Yes
GPC [22]	5	32056	91.80%	79.42%	41.86%	Yes
CNN [22]	5	32056	86.90%	62.86%	32.94%	Yes
LSTM [22]	5	32056	87.59%	53.61%	30.03%	Yes
MKFCNN [22]	5	32056	91.11%	57.42%	40.80%	Yes
CNN-LSTM [22]	5	32056	89.32%	60.84%	39.94%	Yes
CNN-GRU [22]	5	32056	84.08%	59.15%	34.54%	Yes
MSDeepESN [21]	5	32056	87.77%	64.11%	37.45%	Yes
LSTM [21]	5	32056	78.22%	61.87%	33.03%	Yes
MSResNet [21]	5	32056	91.49%	65.51%	50.69%	Yes
CNN [21]	5	32056	82.04%	56.98%	28.50%	Yes