electronics

Article

Evaluation of the Improved Extreme Learning Machine for Machine Failure Multiclass Classification

Nico Surantha [1,2,*] and Isabella D. Gozali [1]

[1] Computer Science Department, BINUS Graduate Program—Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia; isabella.gozali@binus.ac.id
[2] Department of Electrical, Electronic and Communication Engineering, Faculty of Engineering, Tokyo City University, Setagaya-ku, Tokyo 158-8557, Japan
* Correspondence: nico.surantha@binus.ac.id

Abstract: The recent advancements in sensor, big data, and artificial intelligence (AI) have introduced digital transformation in the manufacturing industry. Machine maintenance has been one of the central subjects in digital transformation in the manufacturing industry. Predictive maintenance is the latest maintenance strategy that relies on data and artificial intelligence techniques to predict machine failure and remaining life assessment. However, the imbalanced nature of machine data can result in inaccurate machine failure predictions. This research will use techniques and algorithms centered on Extreme Learning Machine (ELM) and their development to find a suitable algorithm to overcome imbalanced machine datasets. The dataset used in this research is Microsoft Azure for Predictive Maintenance, which has significantly imbalanced failure classes. Four improved ELM methods are evaluated in this paper, i.e., extreme machine learning with under-sampling/over-sampling, weighted-ELM, and weighted-ELM with radial basis function (RBF) kernel and particle swarm optimization (PSO). Our simulation results show that the combination of ELM with under-sampling gained the highest performance result, in which the average F1-score reached 0.9541 for binary classification and 0.9555 for multiclass classification.

Keywords: extreme learning machine; multiclass classification; class imbalance; fault diagnosis; predictive maintenance

Citation: Surantha, N.; Gozali, I.D. Evaluation of the Improved Extreme Learning Machine for Machine Failure Multiclass Classification. *Electronics* **2023**, *12*, 3501. https://doi.org/10.3390/electronics12163501

Academic Editors: Giovanni Pau and Xiangjie Kong

Received: 15 June 2023
Revised: 2 August 2023
Accepted: 14 August 2023
Published: 18 August 2023

1. Introduction

The use of machines to assist human work needs to be supported by choosing the right machine maintenance strategy to prevent machine downtime. According to Senseye [1], sudden machine downtime can cause a company to lose as much as 323 production hours a year and average losses of up to $172 million per plant annually.

The most familiar and widely used machine maintenance strategies are reactive and preventive maintenance. Reactive maintenance is executed after a machine fails, so it risks hampering company productivity [2]. This risk prompted as many as 33% of companies to change their maintenance strategy to more proactive machine maintenance, such as preventive maintenance [3]. The preventive maintenance concept allows the machine to be in good condition but still replaced to follow the predetermined maintenance schedule [4]. Thus, it can increase unnecessary maintenance costs.

The next type of maintenance is predictive maintenance, which uses sensors to monitor and store machine condition data and then utilizes this historical data to predict trends, behavior patterns, or correlations using statistical models or machine learning to predict imminent machine damage or failure [5,6]. The results of these estimates will later be used as a reference in taking the necessary actions to ensure the condition and smooth operation of the machine. Predictive maintenance can reduce maintenance costs by 25–30%, eliminate machine breakdowns by 70–75% and reduce downtime by 35–45% [7].

The use of technology in predictive maintenance is inseparable from various studies that continue to try to find innovations and improve the prediction accuracy of learning models. Several studies have used Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Decision Tree algorithms to produce good performance [8]. However, these studies have not considered that in real situations, failures are rare during the life cycle of a machine when compared to the normal operating cycle of the machine [9]. As a result, the imbalance between classes in the historical data may allow misclassification.

Imbalances between classes in the dataset are usually resolved using under-sampling or over-sampling techniques. Some algorithms will then use processed data, e.g., Extreme Learning Machine (ELM), which produces good performance [10]. ELM became popular since it produces comparatively high accuracy and cross-domain adaptation with low time consumption [11]. Moreover, ELM has superior generalization ability and less training time than deep network models, since it does not require an iterative process for tuning parameters [12,13]. Another technique is using a weight element in the dataset, where the class with a more significant number (the majority) will be given a small-valued weight element. In contrast, the minority class will be given a large-valued weighted element to increase the influence of the data on model learning. An example of using the weight element is the Weighted-Extreme Learning Machine (WELM) algorithm created by Zong et al. [14] to handle the imbalance in the class distribution of the datasets.

In addition to data imbalance, selecting the right features is crucial in building a classifier model that produces satisfactory results [15,16]. Particle Swarm Optimization (PSO) can provide promising results in feature selection, where the accuracy results using PSO increase by up to 21.46% compared to traditional methods without PSO [17].

The main contributions of this research are:

1. Build a classification model suitable for predicting machine failure with unbalanced data in predictive maintenance based on the ELM method.
2. Explore the configuration for each ELM-based model to achieve the optimal performance.

The rest of the paper is structured as follows: The Section 2 discusses previous research on predictive maintenance. The Section 3 describes the dataset and the proposed models. The Section 4 discusses the steps and results, while the Section 5 consists of the analysis of the results of the proposed model. The Section 6 provides conclusions from the research conducted.

2. Related Works

Many researchers have used ELM and its derivatives in handling unbalanced data. Mao et al. [18] used online-sequential ELM combined with SMOTE for the under-sampling and over-sampling of unbalanced data, which can effectively reduce the possibility of information loss thanks to the use of granulation division. Apart from under-sampling and over-sampling, various combinations of WELM were also carried out to overcome balanced learning, which gave satisfactory results [19,20]. Although the studies carried out have obtained good performance, there are still things that need to be considered in using ELM and its derivative algorithms in classifying: its characteristics that determine the number of hidden nodes randomly, which allows overfitting [21], so that the use of the kernel in ELM is also used as an alternative option [19,22,23]. The parameters were initially determined randomly, and researchers used PSO to improve performance [22,24,25]. In fields other than fault detection, PSO has proven its excellent capability in selecting features and determining optimal parameters when combined with ELM and WELM [21,26,27].

Machine failure detection or predictive maintenance research usually uses machine historical data collected individually and not shared with the public. There are also some public datasets that are commonly used, e.g., a public dataset prepared by Microsoft under the name Microsoft Azure for Predictive Maintenance [28]. This dataset consists of machine historical information and can be used both for binary and multiclass classification.

Teoh et al. [29] used a two-class logistics algorithm, where class 0 represents equipment in good condition and class 1 represents equipment that has failed. The Microsoft Azure for Predictive Maintenance data was processed to obtain lag features within 24 h and processed by the under-sampling technique to overcome the problem of imbalance in the data. The dataset is divided into training and testing sets with a ratio of 70:30. The designed model produces training and testing accuracy of 95.1% and 94.5%, respectively.

Cardoso and Ferreira [9] used the same dataset to see the effectiveness of random forest and ANN algorithms in predicting and classifying machine faults into five classes. Before training the models, the dataset was pre-processed to obtain lag features within 3 h and 24 h. Both algorithms provide satisfactory predictions based on precision, recall, and F1-score. The summary of related works using the dataset from Microsoft is presented in Table 1.

Table 1. Summary of related works on machine failure classification for predictive maintenance using Microsoft Azure for Predictive Maintenance dataset.

Models (Ref.)	Classes	Evaluation Metrics		
		Precision	Recall	F1-Score
Two-Class Logistic Regression [29]	2	0.946	0.933	0.939
Random Forest [9]	None	0.9988	0.9998	0.9993
	Comp1	0.9718	0.815	0.8865
	Comp2	0.9711	0.9882	0.9796
	Comp3	0.9855	0.9189	0.951
	Comp4	0.983	0.9611	0.9719
Artificial Neural Network [9]	None	0.999	0.9995	0.9993
	Comp1	0.903	0.8425	0.8717
	Comp2	0.9941	0.9853	0.9897
	Comp3	0.9858	0.9392	0.9619
	Comp4	0.9725	0.9833	0.9779

3. Methodology

The research methodology includes data collection, feature extraction, model development, and evaluation. The model built in the experiment is the ELM algorithm and its varieties and combination step by step to obtain optimal performance. The first model is ELM, which is trained using under-sampled and over-sampled data. The second model is the ELM with a weight element called WELM. After that, a third model was created using WELM combined with the kernel to eliminate the possibility of overfitting due to the random determination of the number of hidden nodes. The final model combined WELM with the kernel and PSO, which helped determine the required parameter values and feature selection. In short, the algorithms used in this study are ELM, WELM, WELM with the RBF kernel, and WELM with the RBF kernel and PSO.

The data that will be used and collected is Microsoft Azure for Predictive Maintenance [28], which is a public dataset that has been mostly used for predictive maintenance model training. Then, the raw data will be extracted to obtain features for model training. For ELM, WELM, and WELM with RBF, kernel models will be trained and tested using all the extracted features, but WELM with RBF kernel and PSO will be trained and tested using selected features by PSO. After the training and testing process, the performance of each model will be evaluated using precision, recall, and F1-score.

3.1. Data Collection

Microsoft Azure for Predictive Maintenance contains 876100 real-time telemetry data (or 8761 data per machine), 3919 error log data, 3286 maintenance history data, and 761 fault history data (approximately eight failure data per machine) from a total of 100 machines with four different types of models throughout 2015, except for maintenance history data collected from 2014. Each machine has four components and four sensors measuring

tension, pressure, vibration, and rotation. A controller also monitors the system and can detect five types of errors. In detail, this dataset consists of five data files with the .csv extension:

1. Real-time telemetry data measure the stress, rotation, pressure, and vibration of each engine from four sensors per hour in real time.
2. Error logs contain errors that occur on the machine, but the machine can still operate.
3. Maintenance history contains information on replacing engine components for several reasons.
4. Fault history is a record of component replacements due to machine failures, and
5. Collection of information regarding the type of model and the number of years of work for each of the 100 machines.

From the five data files above, there are five conditions to be classified into:

1. Normal means the machine is in good condition and operates normally.
2. Fail1 means the machine is broken caused by component one of the machine fails.
3. Fail2 means the machine is broken caused by component two of the machine fails.
4. Fail3 means the machine is broken caused by component three of the machine fails.
5. Fail4 means the machine is broken caused by component four of the machine fails.

The dataset can be classified into binary and multiclass classifications. Teoh et al. [29] previously used the binary classification by predicting machine conditions into 0, which means the machine operates normally, or 1, which means the machine fails. Meanwhile, the multiclass classification was previously performed by Cardoso and Ferreira [9] by predicting the machine conditions into five categories, as discussed before.

3.2. Feature Extraction

Feature extraction in this research is carried out by creating lag features, using the same technique as Cardoso and Ferreira [9]. The trick is creating a temporal window to divide the data into short-term and long-term historical data. Short-term historical data are data with an interval of three hours, while long-term historical data are data with an interval of 24 h. In addition, from the error log data, lag features are also made as information on the number of errors that occur every 24 h on the machine. Apart from the lag features, another feature extracted is when the machine is used before being replaced. This period is calculated for each component on each machine. Details of the features that can be extracted from the data can be seen in Table 2.

Table 2. The list of features extracted from dataset.

No.	Feature Name	Feature Description
1	Voltmean_3h	Lag features for the moving average of machine [1] with period time = 3 h
2	Rotatemean_3h	
3	Pressuremean_3h	
4	Vibrationmean_3h	
5	Voltsd_3h	Lag features for the standard deviation of machine [1] with period time = 3 h
6	Rotatesd_3h	
7	Pressuresd_3h	
8	Vibrationsd_3h	
9	Voltmean_24h	Lag features for the moving average of machine [1] with period time = 24 h
10	Rotatemean_24h	
11	Pressuremean_24h	
12	Vibrationmean_24h	
13	Voltsd_24h	Lag features for the standard deviation of machine [1] with period time = 24 h
14	Rotatesd_24h	
15	Pressuresd_24h	
16	Vibrationsd_24h	

Table 2. *Cont.*

No.	Feature Name	Feature Description
17	Error1count	The sum of error [2] in the 24 h prior to each datetime
18	Error2count	
19	Error3count	
20	Error4count	
21	Error5count	
22	Comp1_lastreplacement	The duration of component [3] used since last replaced
23	Comp2_lastreplacement	
24	Comp3_lastreplacement	
25	Comp4_lastreplacement	

[1] Voltage/Rotation/Pressure/Vibration. [2] Error1/Error2/Error3/Error4. [3] Component 1/Component 2/Component 3/Component 4.

All features that have been extracted will be combined and synchronized with the type of failure based on machineID and the date and time of data recording. Data lines that do not have a failure type are considered features for the machine under normal conditions. Next, the data are cleaned by deleting rows with incomplete or empty features; then, the data are normalized using the min–max normalization.

3.3. Model Development

3.3.1. First Model: Extreme Learning Machine with Under-Sampling and Over-Sampling

Extreme Learning Machine (ELM) is a neural network-based learning algorithm that only consists of three layers: one input layer, one hidden layer, and one output layer. Using random numbers of hidden nodes, ELM can provide good generalization results with faster processing time [30]. The output of ELM is calculated using Equation (1).

$$\beta = H^{+}T = H^{T}\left(\frac{1}{C} + HH^{T}\right)^{-1}T, \tag{1}$$

H is the output hidden layer matrix, while H^{+} is the generalized inverse Moore–Penrose matrix of H and H^{T} is the transpose matrix of H. C is the regularization parameter representing the trade-off between minimizing the training error and maximizing the marginal distance.

Under-sampling and over-sampling techniques are applied to the dataset before training the ELM model to handle the data imbalance. The under-sampling works by reducing the majority class sample, either randomly or using some statistical knowledge; then, it can also continue with data cleaning to further filter the majority of class samples if necessary. Meanwhile, the over-sampling adds new samples that have been created or replicated to the minority class to balance the dataset. The new sample can include important information to minority class data to prevent misclassification [31].

3.3.2. Second Model: Weighted-Extreme Learning Machine

Weighted-Extreme Learning Machine (WELM) is a development algorithm from ELM introduced by Zong et al. [14] to handle datasets with an unbalanced class distribution by multiplying the weight elements W into Equation (1) to strengthen the influence of the minority class and weaken the influence of the majority class. There are two types of W value determination schemes. W_1 can be seen in Equation (2) and W_2 can be seen in Equation (3):

$$W_1 : W_{ii} = \frac{1}{\#(t_i)}, \tag{2}$$

$$W_2 \begin{cases} W_{ii} = \frac{0.618}{\#(t_i)} \; if \; t_i > AVG(t_i) \\ W_{ii} = \frac{1}{\#(t_i)} \; if \; t_i \leq AVG(t_i) \end{cases}, \tag{3}$$

W_{ii} is the weight element on the matrix diagonal W, $\#(t_i)$ is the number of samples in class t_i, and $AVG(t_i)$ is the average sample class t_i. Using the schemes, the minority class will have a more significant W value, and the majority class will have a smaller W value, which is calculated using Equation (4).

$$\beta = H^T \left(\frac{1}{C} + WHH^T \right)^{-1} WT, \tag{4}$$

3.3.3. Third Model: Weighted-Extreme Learning Machine with RBF Kernel

Since WELM is a development of ELM, one of the properties passed down to WELM is the number of hidden nodes in the hidden layer, which is determined randomly. This random determination can result in overfitting if the number of hidden nodes selected is too large [21]. One solution to overcome this random selection of hidden nodes is to use a matrix kernel so that WELM output calculations no longer depend on the number of hidden nodes [26]. When using the kernel on WELM, the WELM output function equation can be described in Equation (5):

$$\beta = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{1}{C} + WK(x_i, x_j) \right)^{-1} WT, \tag{5}$$

3.3.4. Fourth Model: Weighted-Extreme Learning Machine with RBF kernel and Particle Swarm Optimization

This model uses PSO to perform feature selection and determine the C and gamma hyperparameters that affect the RBF kernel. The steps of how the model works can be seen in Figure 1.

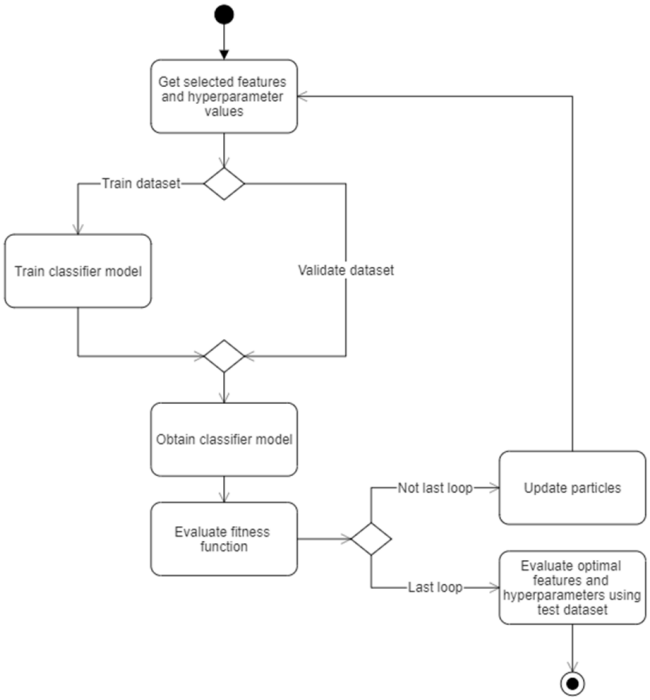


Figure 1. The steps of WELM with RBF kernel and PSO combination.

PSO works by involving a group of particles with their respective positions and velocities to calculate the fitness value using Equation (6) in each iteration.

$$fitness_i = w_s \times fscore_i + w_F \times \left[1 - \frac{\left(\sum_{j=1}^{n_F} f_i \right)}{n_F} \right], \tag{6}$$

w_s is the level of importance of the F1-score result and w_F is the level of importance of the number of features on the fitness of each particle [27]. The position and velocity of the particles will change according to the position and velocity with the best fitness value in each iteration for each particle. The particle position is calculated using Equation (7), while the particle velocity is calculated using Equation (8).

$$x_k^{n+1} = x_k^n + v_k^{n+1}, \tag{7}$$

$$v_k^{n+1} = wv_k^n + c_1 r_1 (pBest_k - x_k^n) + c_2 r_2 (gBest_k - x_k^n), \tag{8}$$

In Equation (6), x_n^k is the position of the n particle in the k iteration. In Equation (7), v_n^k is the velocity of the k particle in the n iteration. The value of c_1 is the value of the tendency of the particle to follow the best position from itself. At the same time, c_2 is the value of the tendency of the particle to follow the best position of the best particle. The values of r_1 and r_2 are random factors with values between 0 and 1, $pBest_k$ is the best position that the k particle has produced, and $gBest_k$ is the best position that all particles have produced during all iterations. PSO particle representation for feature selection is completed, utilizing the particle being initialized with as many features. It will be randomized using a binary number, where one means the feature is selected, and 0 means vice versa. For example, a particle with a binary number 11001 means that the features selected are features 1, 2, and 5 out of a total of 5 features. Table 3 shows the PSO parameter values used in this research.

Table 3. The list of used PSO parameter values.

Parameter Symbol	Parameter Value
w	0.6
c_1	1.2
c_2	1.2
w_s	0.95
w_F	0.05

3.4. Evaluation Metrics

Accuracy is commonly utilized to evaluate the performance of a classifier. However, due to the characteristics of unbalanced data, which favor the majority class more, it is inapplicable to imbalanced learning [32]. Rather than that, the precision, recall, and F1-score are used as evaluation indicators of this work to evaluate the performance of the models [33]. The value range of F1-scores is [0, 1], where the classification's performance improves as the values of the three evaluation indicators become closer to 1. Precision represents the ratio of the true positives to true positives (TP) and false positives (FP), and recall is the ratio of true positives (TP) to the total of the true positives and false negatives (FN). Meanwhile, the F1-score represents the harmonic mean of the precision and recall. The evaluation metrics are defined in Equations (9)–(11).

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

4. Results

4.1. Feature Extraction

The features extracted from the Microsoft Azure for Predictive Maintenance dataset are 25 features consisting of lag features with 3 h and 24 h intervals for the average and

standard deviation of engine voltage, rotation, vibration, and pressure, the number of errors on an engine in the previous 24 h for each type of error, and the length of time the machine was used from the last time it was replaced for each engine component. After the feature extraction results are synchronized with the target, the data are cleaned and normalized. The total data after cleaning and normalization comprise 291,628 rows of data.

It can be seen in Table 4 that the dataset used has a considerable imbalance between classes, where the normal condition of the machine has a percentage of 97.43% of the total data. With a large amount of data, the data are divided into twenty batches.

Table 4. The imbalance between each class in the total dataset.

Target Class	None	Fail	None	Com1	Com2	Com3	Com4
Total	284,141	5873	284,141	1448	2032	999	1394
Percentage (%)	97.43	2.03	97.43	0.5	0.7	0.34	0.48

To divide the data into training data, validation and testing are carried out following the method carried out by Cardoso and Ferreira [9] with the following details:

1. Training data will be taken from a sample of data with a time range from the start of the sample to 31 August 2015, 01:00 (1 January 2015 6:00–31 August 2015 1:00).
2. Validating data will be taken from sample data with times ranging from 1 September 2015, at 01:00 to 31 October 2015, at 1:00 (1 September 2015 01:00–31 October 2015 01:00).
3. Testing data will be taken from data samples with times ranging from 1 November 2015, 01:00 to the end of the sample (1 November 2015 1:00–1 January 2016 6:00).

The data division resulted in 9625 training data, 2478 validation data, and 2476 data lines for testing data.

4.2. First Model: Extreme Learning Machine with Under-Sampling and Over-Sampling

The ELM model experiment was carried out five times, with each experiment having ten iterations. The data were sampled randomly using the random under-sampling and over-sampling library from Python. In this model, all the features that have been extracted are used without selecting features, and the number of hidden nodes used is determined randomly. The best experiment result of ELM with under-sampling is shown in Figure 2, while the best result of ELM with over-sampling is shown in Figure 3.

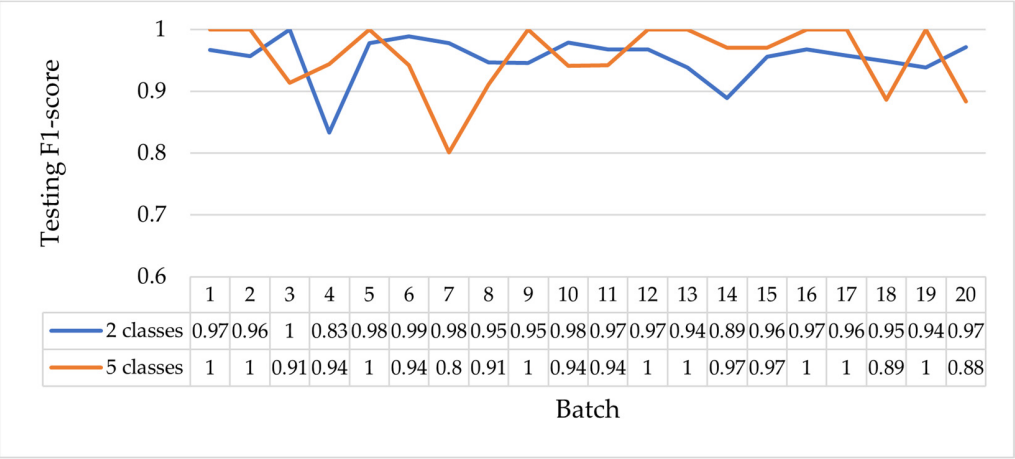
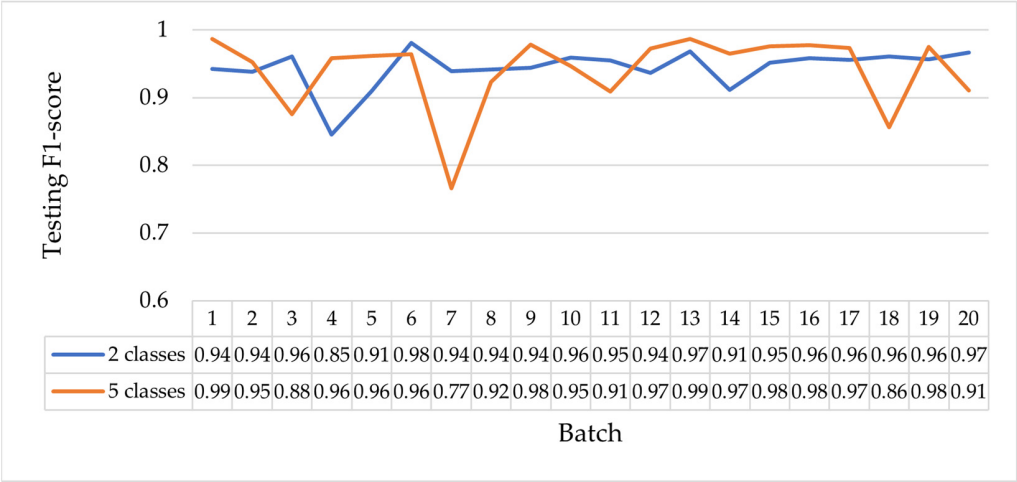


Figure 2. The testing F1-score of ELM algorithm with under-sampling in two sets of classes.

The experiment showed that the performance of ELM with the under-sampling technique is better than that of ELM with the over-sampling technique. ELM with under-sampling produces an average F1-score value in the training set of 0.9201 for binary clas-

sification and 0.9511 for multiclass classification. On the testing set, the average F1-score reaches 0.9541 for binary classification and 0.9555 for multiclass classification. In contrast, ELM with over-sampling produces an average F1-score value in the training set of 0.9315 for binary classification and 0.9458 for multiclass classification and in the testing set of 0.9442 for binary and 0.9409 for multiclass classification.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2 classes	0.94	0.94	0.96	0.85	0.91	0.98	0.94	0.94	0.94	0.96	0.95	0.94	0.97	0.91	0.95	0.96	0.96	0.96	0.96	0.97
5 classes	0.99	0.95	0.88	0.96	0.96	0.96	0.77	0.92	0.98	0.95	0.91	0.97	0.99	0.97	0.98	0.98	0.97	0.86	0.98	0.91

Figure 3. The testing F1-score of ELM algorithm with over-sampling in two sets of classes.

4.3. Second Model: Weighted-Extreme Learning Machine

The WELM model experiment was carried out five times, with each experiment having ten iterations. In this model, all the features that have been extracted are used without selecting features, and the number of hidden nodes used is determined randomly.

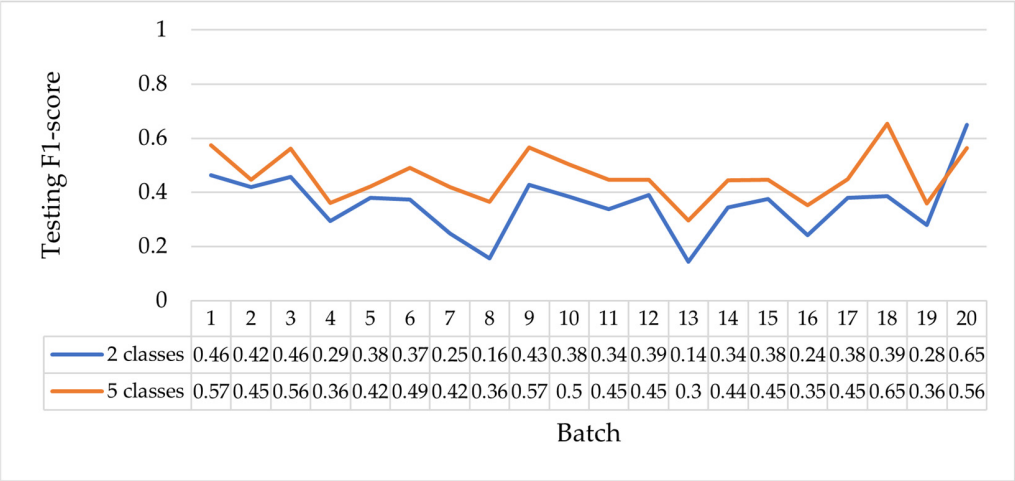
As previously discussed, WELM overcomes data imbalances by using W-weighted elements. There are two types of W-weighted element schemes, so experiments were also carried out using both weight element schemes to determine which weight elements are more suitable for use in this dataset. The best experiment results of WELM are shown in Figures 4 and 5.

In the W_1 weighting scheme, the average F1-score values for binary class and multiclass are 0.8211 and 0.85 for the training set, respectively. Then, the average for the testing set is 0.3563 in the binary class and 0.4579 in the multiclass. There is a significant difference between training and testing scores, reaching 25.81% for binary class and 39.21% for multiclass.

The average F1-scores for the binary class and multiclass of the W_2 weighted-scheme are 0.8697 and 0.8252 for the training set, while the average for the testing set is 0.3486 for the binary class and 0.4621 for the multiclass. The difference between training and testing scores is more significant than that of the W_1 weighting scheme: up to 52.11% for binary class and 36.31% for multiclass.

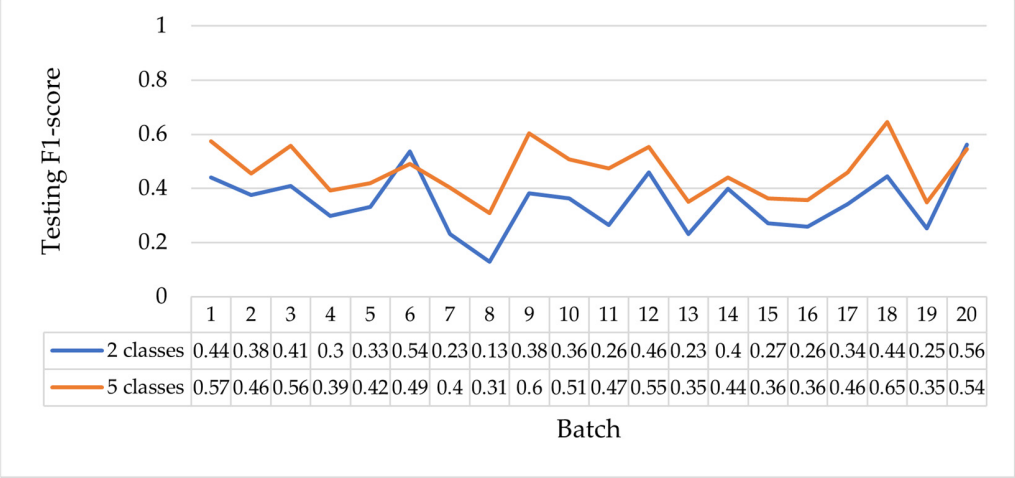
Looking at the results of developing the WELM model further, the reason the results of the WELM evaluation were unsatisfactory could be due to the random selection of hidden nodes. As previously discussed, the number of hidden nodes that are too large can cause overfitting, so the difference in the F1-score values of the training set and the testing set is huge.

Figure 6 shows that the two WELM weighting schemes produce similar results, but the W_1 scheme is slightly superior when used in binary classification. In contrast, the W_2 scheme is superior when performing multiclass classification.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2 classes	0.46	0.42	0.46	0.29	0.38	0.37	0.25	0.16	0.43	0.38	0.34	0.39	0.14	0.34	0.38	0.24	0.38	0.39	0.28	0.65
5 classes	0.57	0.45	0.56	0.36	0.42	0.49	0.42	0.36	0.57	0.5	0.45	0.45	0.3	0.44	0.45	0.35	0.45	0.65	0.36	0.56

Figure 4. The testing F1-score of WELM algorithm using W_1 weighted scheme in two sets of classes.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2 classes	0.44	0.38	0.41	0.3	0.33	0.54	0.23	0.13	0.38	0.36	0.26	0.46	0.23	0.4	0.27	0.26	0.34	0.44	0.25	0.56
5 classes	0.57	0.46	0.56	0.39	0.42	0.49	0.4	0.31	0.6	0.51	0.47	0.55	0.35	0.44	0.36	0.36	0.46	0.65	0.35	0.54

Figure 5. The testing F1-score of WELM algorithm using W_2 weighted scheme in two sets of classes.

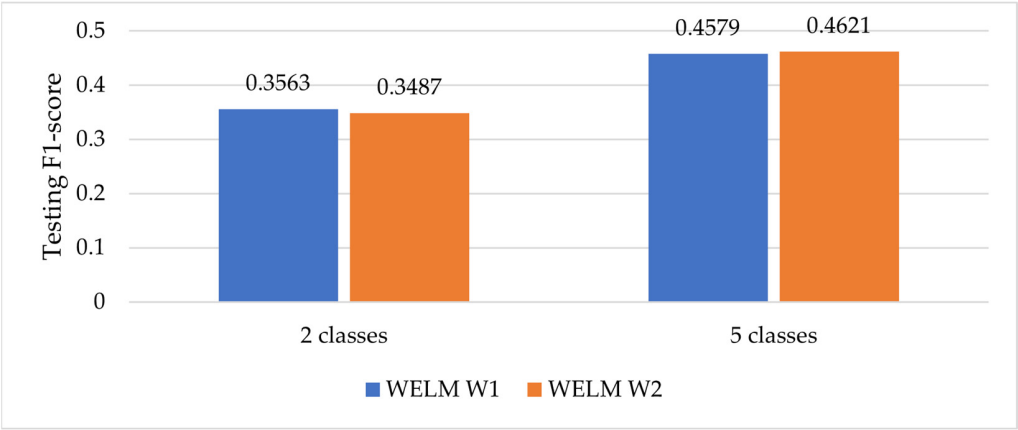
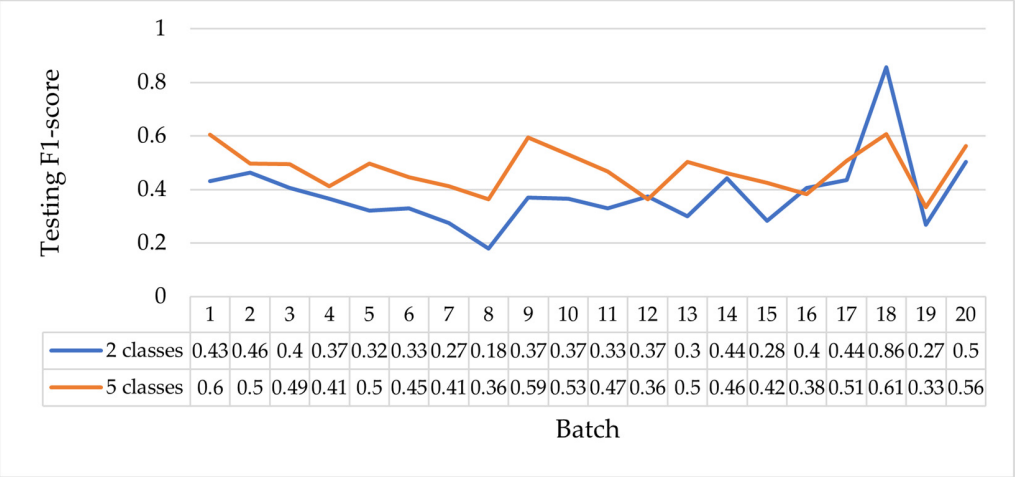


Figure 6. Testing F1-score comparison of W_1 and W_2 in WELM.

4.4. Third Model: Weighted-Extreme Learning Machine with RBF Kernel

The WELM model combined with the RBF kernel model experiment was carried out five times, each with ten iterations. The RBF kernel was used to solve hidden node problems. In this model, all the extracted features are used without selecting features. In addition, two hyperparameters are used by the combination of WELM and this kernel, namely the

C regularization parameter and the kernel bandwidth γ, which are determined randomly. Binary classification uses the W_1 weight element scheme, and multiclass classification uses the W_2 element weight. Figure 7 shows the experiment result of the WELM and RBF kernel combination.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2 classes	0.43	0.46	0.4	0.37	0.32	0.33	0.27	0.18	0.37	0.37	0.33	0.37	0.3	0.44	0.28	0.4	0.44	0.86	0.27	0.5
5 classes	0.6	0.5	0.49	0.41	0.5	0.45	0.41	0.36	0.59	0.53	0.47	0.36	0.5	0.46	0.42	0.38	0.51	0.61	0.33	0.56

Figure 7. The testing F1-score of the integration of WELM with RBF kernel in two sets of classes.

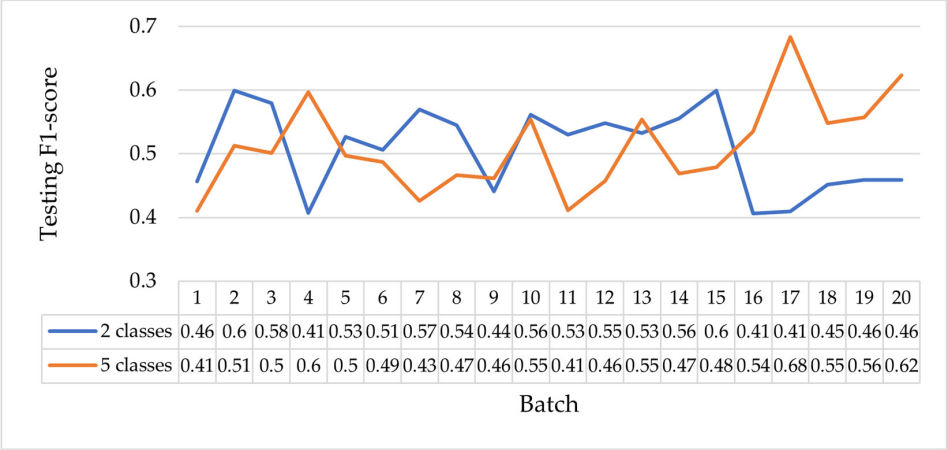
The performance of the WELM model with the RBF kernel is similar to WELM, which uses hidden nodes. The average F1-score results from the training set of this method are 0.8549 for the binary class and 0.9327 for the multiclass. At the same time, the results of the average F1-score from the testing set are 0.3849 and 0.473 for each binary and multiclass. There is a significant difference between the results of training and testing the F1-score: 38.13% for the binary class and 54.78% for the multiclass.

4.5. Fourth Model: Weighted-Extreme Learning Machine with RBF Kernel and Particle Swarm Optimization

Some of the possible reasons that can be factors causing WELM's poor performance with the kernel are the selection of hyperparameter values C and γ in the kernel, which can reduce classification accuracy if not tuned. Another reason could also be that not all the features used significantly influence data classification. Therefore, the development will further use PSO optimization to determine influential features and C and γ hyperparameter values in WELM with the RBF kernel.

The combination method of WELM with the RBF and PSO kernels for feature selection and hyperparameter determination was carried out five times for each data group (where one experiment consisted of 10 PSO iterations). After feature selection and hyperparameter tuning using PSO, the results of the F1-score model slightly increased with the achievement of 0.5072 in binary classification and 0.5116 in multiclass classification. This increase was obtained after 10 PSO iterations. The best experiment result can be seen in Figure 8.

The best experiment result of binary classification was carried out by using feature numbers 1, 2, 4, 9, 10, 11, 15, 17, 19, 20, 21, 22, and 25 as well as the parameter values of 9.15243×10^{14} for C and 604,728.939 for gamma. Meanwhile, the best experiment result of multiclass classification was carried out by using feature numbers 2, 5, 7, 9, 10, 11, 12, 14, 15, 16, 17, 19, 22, and 23 as well as parameter values of 8.99652×10^{14} for C and 649,796.189 for gamma. The details about the feature numbers can be seen in Table 2.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2 classes	0.46	0.6	0.58	0.41	0.53	0.51	0.57	0.54	0.44	0.56	0.53	0.55	0.53	0.56	0.6	0.41	0.41	0.45	0.46	0.46
5 classes	0.41	0.51	0.5	0.6	0.5	0.49	0.43	0.47	0.46	0.55	0.41	0.46	0.55	0.47	0.48	0.54	0.68	0.55	0.56	0.62

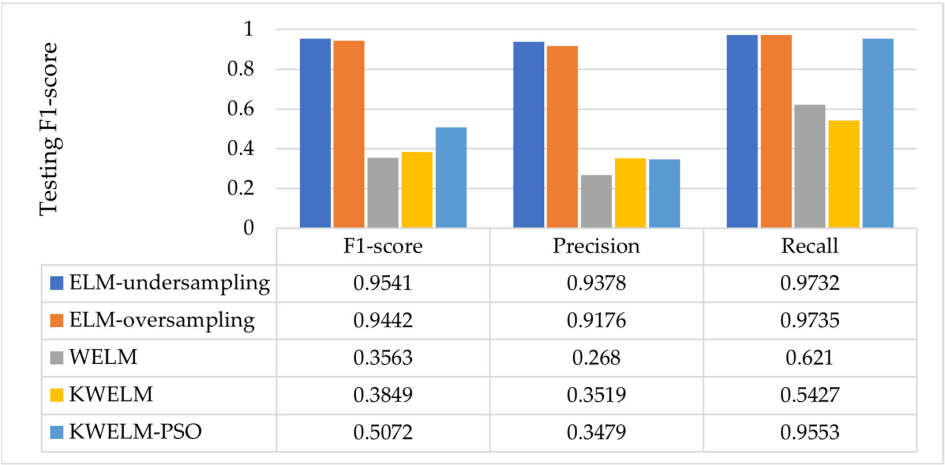
Figure 8. The testing F1-score of the integration of WELM with RBF kernel and PSO in two sets of classes.

5. Discussion

5.1. Comparison of Four Models

This study utilized the ELM model and its varieties to predict machine conditions with the unbalanced dataset. The models were built by combining the ELM algorithm with various methods or algorithms to achieve the optimal model. The first model combines the ELM algorithm with under-sampling and over-sampling methods for handling unbalanced data. Then, the weight element was used in ELM as another way to handle the imbalance, which is known as the WELM algorithm. Another experiment utilized the combination of WELM and RBF kernels, since there is the possibility of overfitting in WELM caused by random hidden node determination. The last model combined WELM with the RBF kernel function and PSO, where PSO was utilized for feature selection and parameter value determination.

In Figures 9 and 10, the WELM method still has lower F1-score, precision, and recall results than the ELM method, which uses under-sampling and over-sampling techniques for binary classification. WELM's weight elements must be optimized with the kernel and PSO to produce a better classification model, which is proved by the increase in F1-score, precision, and recall values of WELM after being combined with the kernel and PSO.

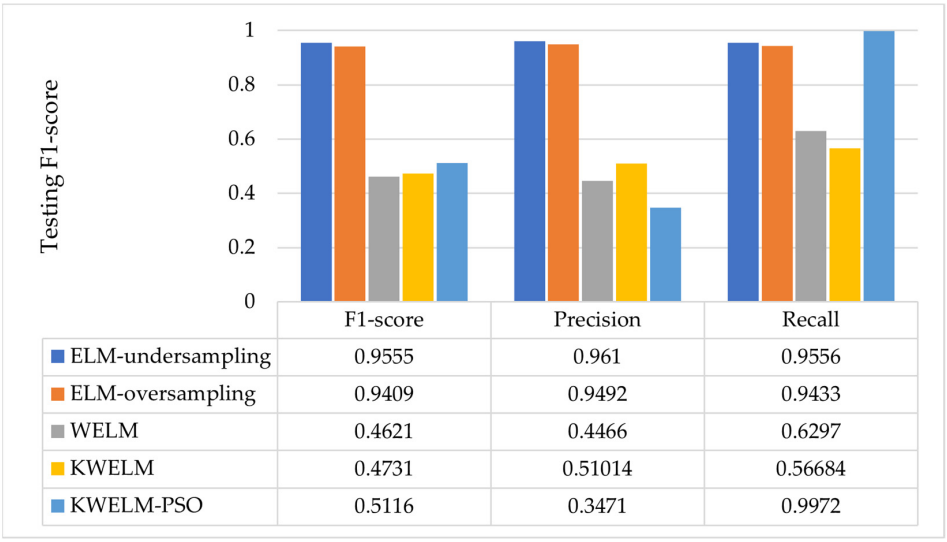


	F1-score	Precision	Recall
ELM-undersampling	0.9541	0.9378	0.9732
ELM-oversampling	0.9442	0.9176	0.9735
WELM	0.3563	0.268	0.621
KWELM	0.3849	0.3519	0.5427
KWELM-PSO	0.5072	0.3479	0.9553

Figure 9. Evaluation result comparison of ELM, WELM, WELM with RBF kernel, and WELM with RBF kernel and PSO combination in binary classification.

Furthermore, for multi-class classification, it is the same as the results of binary classification that ELM using under-sampling and over-sampling techniques has better F1-score, precision, and recall values. The state of the data is more balanced because it uses under-sampling and over-sampling techniques; thus, the classification results improve. Using

WELM with the RBF and PSO kernels can give good results by increasing the number of PSO iterations so that the model can obtain the most optimal features and hyperparameters.



	F1-score	Precision	Recall
ELM-undersampling	0.9555	0.961	0.9556
ELM-oversampling	0.9409	0.9492	0.9433
WELM	0.4621	0.4466	0.6297
KWELM	0.4731	0.51014	0.56684
KWELM-PSO	0.5116	0.3471	0.9972

Figure 10. Evaluation result comparison of ELM, WELM, WELM with RBF kernel, and WELM with RBF kernel and PSO combination in multiclass classification.

5.2. Comparison with Previous Works

The algorithm with the best performance, which is ELM with under-sampling, is compared to previous research in Tables 5 and 6. For binary and multiclass classification, the performance results of ELM with under-sampling and previous research are similar and reached a very good F1-score.

Table 5. Comparison of binary classification with previous research.

Works	Models	Evaluation Metrics		
		Precision	Recall	F1-Score
Teoh et al. [29]	Two-class Logistic Regression	0.982	0.998	0.990
Our work	ELM + Under-Sampling	0.9378	0.9732	0.9541

Table 6. Comparison of multiclassification with previous research.

Works [Ref.]	Models	Failure Type	Evaluation Metrics		
			Precision	Recall	F1-Score
Cardoso and Ferreira [9]	Random Forest	None	0.9988	0.9998	0.9993
		Comp1	0.9718	0.815	0.8865
		Comp2	0.9711	0.9882	0.9796
		Comp3	0.9855	0.9189	0.951
		Comp4	0.983	0.9611	0.9719
	Artificial Neural Network	None	0.999	0.9995	0.9993
		Comp1	0.903	0.8425	0.8717
		Comp2	0.9941	0.9853	0.9897
		Comp3	0.9858	0.9392	0.9619
		Comp4	0.9725	0.9833	0.9779
Our work	ELM + Under-Sampling	None	0.9889	0.9304	0.9546
		Comp1	0.9345	0.9429	0.9352
		Comp2	0.9758	0.9702	0.972
		Comp3	0.941	0.9786	0.9574
		Comp4	0.9649	0.956	0.9585

6. Conclusions

This study uses ELM and its development to achieve the maximum performance of engine failure classification models with imbalanced datasets. To overcome dataset imbalances, the model used is ELM with under-sampling and over-sampling techniques, WELM, WELM combined with the RBF kernel, and WELM combined with the RBF kernel and PSO.

From the experiments and analyses that have been carried out, it is proved that the use of weight elements in WELM alone is still not able to overcome the problem of unbalanced datasets, as evidenced by the results of metric measurements such as F1-score, precision, and recall, which are sensitive to data which have not yet achieved satisfactory results. Meanwhile, under-sampling can balance the dataset, resulting in a better F1-score for each class. For future work, optimization methods other than the RBF kernel and PSO can be sought to improve WELM performance so that there is no need to use under-sampling or over-sampling techniques on data to achieve better performance.

Author Contributions: Conceptualization, N.S. and I.D.G.; methodology, N.S and I.D.G.; software, I.D.G.; validation, I.D.G.; investigation, I.D.G.; resources, N.S.; data curation, I.D.G.; writing—original draft preparation, N.S and I.D.G.; writing—review and editing, N.S.; visualization, I.D.G.; supervision, N.S.; project administration, I.D.G.; funding acquisition, N.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research publication is fully supported by Bina Nusantara University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available online.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Senseye. The True Cost of Downtime: How Much Do Leading Manufacturers Lose through Inefficient Maintenance? 2020. Available online: https://www.senseye.io/downtime-report-download (accessed on 9 April 2022).
2. Ran, Y.; Zhou, X.; Lin, P.; Wen, Y.; Deng, R. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *IEEE Commun. Surv. Tutor.* **2019**, *20*, 1–36. [CrossRef]
3. Plant Engineering. Facilities Maintenance. 2019. Available online: https://www.plantengineering.com/wp-content/uploads/sites/4/2019/02/Plant-Engineering-2019-Maintenance-Report.pdf (accessed on 9 April 2022).
4. Jimenez, J.J.M.; Schwartz, S.; Vingerhoeds, R.; Grabot, B.; Salaün, M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *J. Manuf. Syst.* **2020**, *56*, 539–557. [CrossRef]
5. Satta, R.; Cavallari, S.; Pomponi, E.; Grasselli, D.; Picheo, D.; Annis, C. A dissimilarity-based approach to predictive maintenance with application to HVAC systems. *arXiv* **2017**, arXiv:1701.03633.
6. Zonta, T.; da Costa, C.A.; da Rosa Righi, R.; de Lima, M.J.; da Trindade, E.S.; Li, G.P. Predictive maintenance in the Industry 4.0: A systematic literature review. *Comput. Ind. Eng.* **2020**, *150*, 106889. [CrossRef]
7. Sullivan, G.P.; Melendez, A.P.; Pugh, R.; Hunt, W.D. Operations & Maintenance Best Practices—A Guide to Achieving Operational Efficiency (Release 3.0). Richland. 2010. Available online: https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-19634.pdf (accessed on 1 May 2022).
8. Çinar, Z.M.; Nuhu, A.A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability* **2020**, *12*, 8211. [CrossRef]
9. Cardoso, D.; Ferreira, L. Application of predictive maintenance concepts using artificial intelligence tools. *Appl. Sci.* **2021**, *11*, 18. [CrossRef]
10. Du, J.; Vong, C.-M.; Chang, Y.; Jiao, Y. Online Sequential Extreme Learning Machine with Under-Sampling and Over-Sampling for Imbalanced Big Data Classification. In Proceedings of the ELM-2016, Singapore, 13–15 December 2016; Proceedings in Adaptation, Learning and Optimization. Cao, J., Cambria, E., Lendasse, A., Miche, Y., Vong, C.M., Eds.; Springer: Cham, Switzerland, 2018; pp. 229–239. [CrossRef]
11. Wang, J.; Lu, S.; Wang, S.H.; Zhang, Y.D. A review on extreme learning machine. *Multimed. Tools Appl.* **2022**, *81*, 41611–41660. [CrossRef]
12. Kasun, L.L.C.; Zhou, H.; Huang, G.; Vong, C.M. Representational Learning with ELMs for Big Data. *IEEE Intell. Syst.* **2013**, *28*, 31–34.

13. Tiffany, S.; Sarwinda, D.; Handari, B.D.; Hertono, G.F. The comparison between extreme learning machine and artificial neural network-back propagation for predicting the dengue incidences number in DKI Jakarta. *J. Phys. Conf. Ser.* **2021**, *1821*, 012025. [CrossRef]
14. Zong, W.; Huang, G.B.; Chen, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing* **2013**, *101*, 229–242. [CrossRef]
15. Nieto-del-Amor, F.; Prats-Boluda, G.; Garcia-Casado, J.; Diaz-Martinez, A.; Diago-Almela, V.J.; Monfort-Ortiz, R.; Hao, D.; Ye-Lin, Y. Combination of Feature Selection and Resampling Methods to Predict Preterm Birth Based on Electrohysterographic Signals from Imbalance Data. *Sensors* **2022**, *22*, 5098. [CrossRef]
16. Zhang, C.; Soda, P.; Bi, J.; Fan, G.; Almpanidis, G.; García, S. An Empirical Study on the Joint Impact of Feature Selection and Data Re-sampling on Imbalance Classification. *Appl. Intell.* **2023**, *53*, 5449–5461. [CrossRef]
17. Vashishtha, J. Particle Swarm Optimization based Feature Selection. *Int. J. Comput. Appl.* **2016**, *146*, 11–17. [CrossRef]
18. Mao, W.; He, L.; Yan, Y.; Wang, J. Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine. *Mech. Syst. Signal Process.* **2017**, *83*, 450–473. [CrossRef]
19. Tong, R.; Li, P.; Lang, X.; Liang, J.; Cao, M. A novel adaptive weighted kernel extreme learning machine algorithm and its application in wind turbine blade icing fault detection. *Measurement* **2021**, *185*, 110009. [CrossRef]
20. Jalayer, M.; Kaboli, A.; Orsenigo, C.; Vercellis, C. Fault Detection and Diagnosis with Imbalanced and Noisy Data: A Hybrid Framework for Rotating Machinery. *Machines* **2022**, *10*, 237. [CrossRef]
21. Surantha, N.; Lesmana, T.F.; Isa, S.M. Sleep stage classification using extreme learning machine and particle swarm optimization for healthcare big data. *J. Big Data* **2021**, *8*, 14. [CrossRef]
22. Ma, J.; Wu, J.; Wang, X. Fault diagnosis method based on wavelet packet-energy entropy and fuzzy kernel extreme learning machine. *Adv. Mech. Eng.* **2018**, *10*. [CrossRef]
23. Liu, J.; Xu, H.; Peng, X.; Wang, J.; He, C. Reliable composite fault diagnosis of hydraulic systems based on linear discriminant analysis and multi-output hybrid kernel extreme learning machine. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109178. [CrossRef]
24. Hu, K.; Zhou, Z.; Weng, L.; Liu, J.; Wang, L.; Su, Y.; Yang, Y. An Optimization Strategy for Weighted Extreme Learning Machine based on PSO. *Int. J. Pattern Recognit. Artif. Intell.* **2017**, *31*, 1751001. [CrossRef]
25. Wang, Z.; Wang, N.; Zhang, H.; Jia, L.; Qin, Y.; Zuo, Y.; Zhang, Y.; Dong, H. Segmentalized mRMR Features and Cost-Sensitive ELM With Fixed Inputs for Fault Diagnosis of High-Speed Railway Turnouts. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4975–4987. [CrossRef]
26. Zhu, H.; Liu, G.; Zhou, M.; Xie, Y.; Abusorrah, A.; Kang, Q. Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection. *Neurocomputing* **2020**, *407*, 50–62. [CrossRef]
27. Ahila, R.; Sadasivam, V.; Manimala, K. An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances. *Appl. Soft Comput.* **2015**, *32*, 23–37. [CrossRef]
28. Uz, F.B. Predictive Maintenance Modelling Guide Data Sets. 2016. Available online: https://gallery.azure.ai/Experiment/Predictive-Maintenance-Modelling-Guide-Data-Sets-1 (accessed on 9 April 2022).
29. Teoh, Y.K.; Gill, S.S.; Parlikad, A.K. IoT and Fog Computing based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 using Machine Learning. *IEEE Internet Things J.* **2021**, *10*, 2087–2094. [CrossRef]
30. Dash, N.; Priyadarshini, R.; Rout, S. Performance Comparison of Back propagation Neural Network and Extreme Learning machine for Multinomial Classification Task. *Int. J. Adv. Comput. Technol.* **2013**, *2*, 365–369.
31. Shelke, M.S.; Deshmukh, P.R.; Shandilya, V.K. A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique. *Int. J. Recent Trends Eng. Res.* **2017**, *3*, 444–449. [CrossRef]
32. Wang, L.; Han, M.; Li, X.; Zhang, N.; Cheng, H. Review of Classification Methods on Unbalanced Data Sets. *IEEE Access* **2021**, *9*, 64606–64628. [CrossRef]
33. Wei, H.; Zhang, Q.; Shang, M.; Gu, Y. Extreme learning Machine-based classifier for fault diagnosis of rotating Machinery using a residual network and continuous wavelet transform. *Measurement* **2021**, *183*, 109864. [CrossRef]