

*Article*

## Real-Time Induction Motor Health Index Prediction in A Petrochemical Plant using Machine Learning

Waritsara Khrakhuean<sup>1,a</sup> and Parames Chutima<sup>1,2,b,\*</sup>

<sup>1</sup> Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Thailand

<sup>2</sup> The Academy of Science, The Royal Society of Thailand, Thailand

E-mail: <sup>a</sup>waritsara.k@gmail.com, <sup>b,\*</sup>parames.c@chula.ac.th (Corresponding author)

**Abstract.** This paper presents real-time health prediction of induction motors (IMs) utilised in a petrochemical plant through the application of intelligent sensors and machine learning (ML) models. At present, maintenance engineers of the company implement time-based and condition-based maintenance techniques in periodically examining and diagnosing the health of IMs which results in sporadic breakdowns of IMs. Such breakdowns sometimes force the entire production process to stop for emergency maintenance resulting in a huge loss in the company's revenue. Hence, top management decides to switch the operational practice to real-time predictive maintenance instead. Intelligent sensors are installed on IMs to collect necessary information related to their working statuses. ML exploits the real-time information received from intelligent sensors to flag abnormalities of mechanical or electrical components of IMs before potential failures are reached. Four ML models are investigated to evaluate which one is the best, i.e. Artificial Neural Network (ANN), Particle Swarm Optimization (PSO), Gradient Boosting Tree (GBT) and Random Forest (RF). Standard performance metrics are used to compare the relative effectiveness among different ML models including Precision, Recall, Accuracy, F1-score, and AUC-ROC curve. The results reveal that PSO not only obtains the highest average weighted Accuracy but also can differentiate the statuses (Class 0 – Class 3) of the IM more correctly than other counterpart models.

**Keywords:** Real-time prediction, machine learning, artificial neural network, particle swarm optimisation, gradient boost tree, random forest.

**ENGINEERING JOURNAL** Volume 26 Issue 5

Received 28 January 2022

Accepted 10 May 2022

Published 31 May 2022

Online at <https://engj.org/>

DOI:10.4186/ej.2022.26.5.91

## 1. Introduction

In the petrochemical industry, the maintenance team usually performs time-based maintenance (i.e. scheduled maintenance and scheduled replacement whereby the maintenance is undertaken on equipment based on a fixed interval or schedule without considering equipment conditions) and condition-based maintenance (i.e. actual condition of the equipment is analysed to determine what kind of maintenance should be done). Condition-based maintenance needs to be performed only when symptoms of deteriorating performance or imminent failure are notified. The most famous techniques used in condition-based maintenance are condition monitoring maintenance and fault diagnosis. Moreover, condition-based maintenance can be further categorised into offline and online assessments. Various critical process parameters are normally monitored during the condition monitoring maintenance, e.g. speed, sound, vibration, electrical signal and temperature. The irregularities of these parameters are used as an early warning signal of potential failures in the key equipment and/or production process. In addition, they could be utilised in gauging equipment's health index which is an indicator to judge whether any proper maintenance operation is needed or not before a critical failure of the equipment will occur.

An asynchronous motor, normally referred to as an AC induction motor (IM), has been widely used as a machine driver in the petrochemical industry for many decades due to its stable speed control. Besides, the IM is relatively inexpensive and convenient to maintain because there is no carbon brush and commutator like in a DC electrical motor. The speed of the IM depends on the frequency of the AC power source. When the IM is operated with a variable speed drive controller, the speed can be easily adjusted from zero to its rated speed. The critical factors that cause IM breakdowns arise from both mechanical and electrical problems, e.g. the changes in vibration level, three-phase current, three-phase voltage, percentage of load, speed, and operating temperature. The allowable levels of changes are crucial inputs for determining the health index (status) of the IM.

One of the problems encountered in the case study petrochemical company is sporadic unplanned downtimes of the IMs. According to the historical data kept in the corrective/emergency maintenance (CM) datalogger (Table 1), it is observed that CM has been constantly performed on the IMs every year. However, at present, manual spot data collection is the only technique used by maintenance officers. As a result, these data are prone to human errors and inaccuracies due to the discrepancy in the qualification, competency and skills that exist among maintenance officers. Moreover, the failure mode and effect analysis focus mostly on mechanical damages, whereas only a few electrical parameter check-points are considered. Unsurprisingly, breakdowns are still something often witnessed in the production process. Such unplanned breakdowns not only could disrupt the entire production system, but also result in production

loss, unacceptable product quality, and incur substantial repair costs.

To improve the aforementioned situation, real-time maintenance in which equipment can communicate its operating condition, particularly abnormal signs, in real-time with the control room through smart sensors is implemented. Since the status of the equipment is continuously monitored, the predictive maintenance plan can be developed more effectively and faster than condition-based maintenance. Consequently, a real-time data collection system through smart sensors is additionally installed on the IMs of the case study company to increase the accuracy of the collected data and to rectify the human error problem. This system enables the data collections of both mechanical and electrical parameters of the IMs to be done much easier resulting in more reliable motor health index levels to be observed. Since the ultimate goal of the maintenance department is to switch from time-based and condition-based maintenances to real-time predictive maintenance, machine learning (ML) is employed to categorise and flag possible abnormalities in the mechanical or electrical components of the IMs before potential equipment or system failures are reached. Note that the real-time maintenance in this study is 1 hour of operation.

To predict the health index levels of the IMs in dynamic operating conditions, several input parameters retrieved from the intelligent sensors attached to the IMs are utilised. For example, the parameters like the swinging amplitude of the vibration, input current and temperature are among the key factors that affect the health degradation of the IMs. If the vibration amplitude of an IM is high, an unbalance or misalignment of the machinery may probably occur. This could subsequently result in a malfunction or breakdown of the IM shortly. Therefore, these parameters need to be identified by maintenance officers since they are necessary for an effective IM health prediction and diagnosis.

In this research, four ML models are employed to predict the health index levels and notify abnormalities of the IM including two computational models, i.e. Artificial Neural Network (ANN) and Particle Swarm Optimization (PSO), and two ensemble models which integrate different models in their prediction processes, i.e., Gradient Boost Tree (GBT) and Random Forest (RF). Several standard performance measurements are used to benchmark the relative effectiveness among different ML models, i.e. Precision, Recall, Accuracy, F1-score and AUC-ROC curve.

The remaining section of the paper is organised as follows. Section 2 provides a comprehensive review of related literature. Section 3 explained the experimental design which illustrates how data are gathered, clean, and prepared before embarking on the training process of the ML. ANN, PSO, GBT and RF used in the prediction of the IM health index levels are presented in Section 4. Section 5 discusses the experimental results and compares the effectiveness of different ML models. Finally, the conclusion is given in Section 6.

Table 1. Corrective maintenance history.

IM Class	Number of Corrective Maintenances											Grand Total
	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	
S	-	-	-	-	3	1	-	-	1	2	1	8
A	8	16	7	10	8	17	4	5	2	2	7	86
B	10	17	19	14	18	11	16	14	12	13	11	155
C	12	13	5	8	2	7	6	7	2	3	5	70
Grand Total	30	46	31	32	31	36	26	26	17	20	24	319

(Note: IM Class is the criticality class assessment when the equipment is malfunction; S = having an impact on people or environment, A = suddenly causing production disruption and/or off-specification product, B = causing production disruption and/or off-specification product but not suddenly, C = not causing a direct effect on product specification)

## 2. Literature Review

This section provides a comprehensive review of related literature on the methodology for predicting the health index levels and abnormalities of the IM using the machine learning model. Seera et al. [1] provided a hybrid intelligent model that can be used to monitor the state of IMs. This model consists of an ensemble of classification and regression trees. A hybrid intelligent model combining the fuzzy min-max neural network and the random forest model. Tao et al. [2] combined three classifiers to diagnose faults in IMs, i.e. naive Bayes classifier, random forest classifier, and the support vector machine classifier. The states of the motor were accurately predicted using a multi-classifier algorithm. Toma et al. [3] adopted a hybrid motor-current data-driven approach for bearing fault diagnosis, which incorporated statistical features, genetic algorithms, and ML models. Three distinct classification algorithms, namely the k-nearest neighbour algorithm, decision tree, and random forest were used to generate experimental results indicating that all three classifiers achieved greater than 97% accuracy.

Ghate et al. [4] developed a novel method for intelligent fault detection and classification in three-phase IMs using radial basis function - multilayer perceptrons cascade neural network. When tested on testing data and cross-validation data, it was discovered that the network could detect faults in an induction motor with an average classification accuracy of 98.41% and 98.11%, respectively. Su et al. [5] provided a system that computed the fault indicator by comparing a vibration spectrum model generated using a multilayer perceptron neural network to the standard technique and the root mean square values. Martins et al. [6] demonstrated the use of an unsupervised Hebbian-based neural network-based algorithm to perform a fully automatic online diagnosis of three-phase induction motor stator faults. The direction of the neural network eigenvectors indicated the phase during which the fault occurred. The relationship between the eigenvector space components was used to determine whether or not the motor was healthy.

Noel [7] introduced the gradient penalties to select the structure method, a hybrid optimisation technique that

combines PSO and gradient-based local search algorithms. Its objective was to accomplish faster convergence and improve the accuracy of the final solution while avoiding becoming caught in local minima. The PSO method converged faster to a much more accurate final solution for a range of benchmark test functions. Qolomany et al. [8] employed the PSO approach which could optimise parameter settings and conserve significant computational resources throughout the deep learning model tuning process. Compared to the grid search method, it was proven to be a more efficient way to set the ideal number of hidden layers and neurons in each layer of the deep learning algorithm. Kim et al. [9] built a deep neural network model and used it in conjunction with two well-known ML techniques, i.e. logistic regression and random forest, to predict the motor outcome six months after stroke. The current study has shown that by employing fourteen input variables clinicians. Notably, the deep neural network could be beneficial for predicting motor outcomes in the upper and lower limbs at six months after stroke.

Tama et al. [10] studied how a gradient boosted machine could be used to increase the detection performance of an anomaly-based intrusion detection system. The gradient boosted machine was then compared against the performance of four well-known classifiers, i.e. random forest, deep neural network, support vector machine, and classification and regression tree. The experimental results demonstrated that the gradient boosted machine significantly outperformed intrusion detection system approaches. Krauss et al. [11] developed and applied a statistical arbitrage technique based on deep neural networks, gradient-boosted trees, random forests from various ensembles methods to deploy with the S&P 500 constituents dataset. Calzavara et al. [12] extended adversarial training to gradient-boosted decision trees and demonstrated its efficacy on a publicly available dataset. Raja [13] deployed the diabetes dataset on the standard UCI Pima Indian and found that the gradient boosting classifier outperformed random forest and neural networks. The standard measurements like AUC, Recall, and Accuracy were used to evaluate the constructed models. Manna et al. [14] examined the effectiveness of

this successful paradigm in predicting air traffic delays. The gradient boosted decision tree modelled sequential data exceptionally accurately. Selvi et al. [15] developed a novel map-reduce-based optimal data classifier technique for efficiently diagnosing diabetic Mellitus. The simulation results demonstrated that the map-reduce-based optimal data classifier model, which incorporated multiple stages of the Hadoop ecosystem, data collecting, and classification using the gradient boosting tree, consistently outperformed the compared methods.

Pal [16] recommended that the random forest classifier outperformed the support vector machine in classification accuracy and training time when applied to Landsat enhanced thematic mapper plus data for an area in the United Kingdom with seven distinct land cover types. Rodriguez-Galiano et al. [17] investigated the random forest classifier's effectiveness in classifying the land cover of a complex area. The study took numerous factors into account, including mapping accuracy, sensitivity to data set size, and noise.

Chevachapimol et al. [18] applied conventional ML, classical feed-forward deep learning, and a novel hybrid deep learning approach to predict the on-time performance of domestic flights departing from the top ten busiest airports in the United States. The purpose of this study was to evaluate the hybrid deep learning model's efficacy to those of a feed-forward ANN and gradient boosted tree machine learning.

From the review of literature, it is obvious that none of the articles has compared the effectiveness of computational and ensemble models of ML on the IM dataset before. Additionally, under the context of the IM health index prediction, PSO has never been applied. The closest article to this research was conducted by Tao et al. [2]. However, their IM parameters were substantially fewer than those presented in this research and also different ML models were used as the prediction tool.

### 3. Experimental Design

The dataset used in this research is collected from the IM which has specifications as follows: 475 kW, 2 Poles, 50 Hz, 3 Phase, 6.6 kV, 2988 rpm, squirrel cage class S. The one-hour interval dataset which combines both mechanical and electrical parameters from Yr 2016 to Yr 2021 include the vibration level, temperature, input current, input power and flow rate.

#### 3.1. Data Cleansing

Data cleansing is the process of identifying and resolving corrupt, inaccurate, or irrelevant data caused by sensor errors or data entry errors by maintenance officers. This critical stage of data processing, also referred to as data scrubbing or data cleaning, is a vital step to improve the consistency and reliability of data before constructing operation and maintenance schedules. Inaccuracies in data usually include missing values and typographical errors. In many cases, that requires certain values to be filled in or

corrected. On the other hand, in some circumstances, the values will need to be removed.

While conducting the data cleansing process, the considered attributes of the IM include the vibration (acceleration and velocity units) in three-dimensional (axial, horizontal and vertical) and surface temperature, input current, input power and flow rate of the system. The directional measurements of the IM are depicted in Fig. 1, and the attributes/parameters (inputs) collected in the IM health index dataset are shown in Table 2.

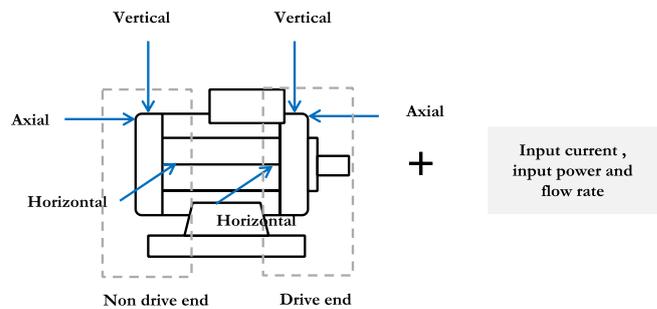


Fig. 1. Dimensional measurement of the IM.

#### 3.2. Feature Extraction

The international standards used to evaluate the IM health index levels in this study are shown in Table 3, including the vibration, temperature, current, power and flow rate. The health index level is classified into four groups (i.e. Class 0, Class 1, Class 2, and Class 3), the details of which are explained in Table 4. Note that Class 2 and Class 3 are the health indexes that are the most important categories for the company since they indicate the prone-to-failure state of the IM.

#### 3.3. Data Preprocessing

##### 3.3.1. Data normalisation

Data normalisation is the process of converting the attributes with a mixture of different unit scales into the same scale. One of the famous data normalisation processes is called feature scaling. The standard data normalisation method is defined as follows.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where  $x$  is the original measured value,  $\mu$  is the mean of the samples, and  $\sigma$  is the standard deviation of the samples. The formulations of  $\mu$  and  $\sigma$  are shown in Eq. (2) and Eq. (3), respectively. Note that  $N$  is the number of members in the dataset.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (2)$$

Note that all data in the IM dataset are converted into normalised data before further processing.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (3)$$

### 3.3.2. Multi-class imbalanced classification

The class imbalance problem has drawn growing interest recently due to the classification difficulty caused by the imbalanced class distributions [19]. To rectify the problem of the oversampling imbalanced classification

dataset, in this study, the synthetic minority oversampling technique (SMOTE) is applied. Although SMOTE could help balance the class distribution, it does not provide any additional information to the ML model [20]. The datasets before and after applying the SMOTE technique are shown in Fig. 2 and Fig. 3, respectively. It is noticeable that the dataset obtained after the SMOTE technique is applied is perfectly balanced and ready to be used in training the ML. The summarised description of the dataset is listed in Table 5.

Table 2. The attributes/parameters used in the dataset for the IM health prediction.

Attribute Name	Parameter	Unit	Direction	Location
Acc_DE_Ax	Acceleration	m/s <sup>2</sup>	Axial	Drive end
Acc_DE_RH	Acceleration	m/s <sup>2</sup>	Horizontal	Drive end
Acc_DE_RV	Acceleration	m/s <sup>2</sup>	Vertical	Drive end
Acc_NDE_Ax	Acceleration	m/s <sup>2</sup>	Axial	Non drive end
Acc_NDE_RH	Acceleration	m/s <sup>2</sup>	Horizontal	Non drive end
Acc_NDE_RV	Acceleration	m/s <sup>2</sup>	Vertical	Non drive end
Temp_DE	Temperature	degrees Celsius	Horizontal	Drive end
Temp_NDE	Temperature	degrees Celsius	Horizontal	Non drive end
Vel_DE_Ax	Velocity	mm/s	Axial	Drive end
Vel_DE_RH	Velocity	mm/s	Horizontal	Drive end
Vel_DE_RV	Velocity	mm/s	Vertical	Drive end
Vel_NDE_Ax	Velocity	mm/s	Axial	Non drive end
Vel_NDE_RH	Velocity	mm/s	Horizontal	Non drive end
Vel_NDE_RV	Velocity	mm/s	Vertical	Non drive end
I_SYS	Current	Ampere	-	System
P_SYS	Power	kW	-	System
Q_SYS	Flow Rate	m <sup>3</sup> /h	-	System

Table 3. Standard and specification of IM health index.

Attribute	Standard and specification
Vibration	ISO-10816 Part 3
Temperature	ANSI/NETA MTS-2019
Current, Voltage and Flow rate	Machine specifications and manufacturing guidelines

Table 4. IM health index level classifications.

Health index levels	Definitions
Class 0	New machine or new installation.
Class 1	Acceptable for long-term operation.
Class 2	Unacceptable for long-term operation. Need to create condition monitoring program or mitigation plan.
Class 3	Need to adjust operating condition or perform corrective action immediately.

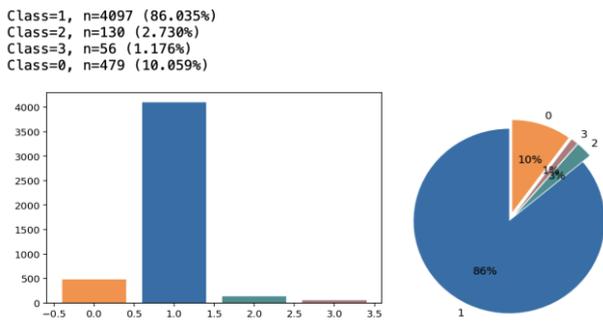


Fig. 2. The IM health index level before applying SMOTE.

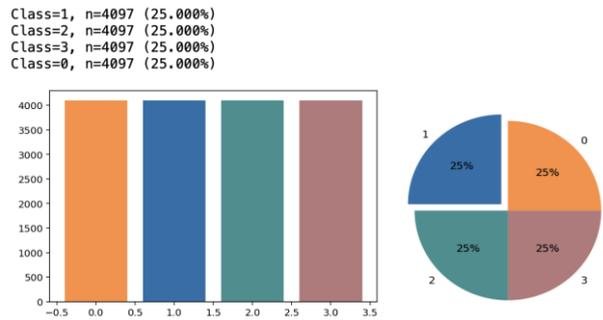


Fig. 3. The IM health index level after applying SMOTE.

Table 5. Summarised Description of the Dataset.

Description	Value
Number of instances	16,388
Number of attributes	17
Number of health index levels	4
% Class 0	25%
% Class 1	25%
% Class 2	25%
% Class 3	25%

### 3.4. Evaluation Metrics

The evaluation of the ML model plays an essential role in identifying how effectively the ML model works. ML models are mainly classified into two types, i.e., classification and regression. Due to the problem of interest, this study focuses on the classification category. As such, the relative effectiveness evaluation of the ML model includes Precision, Recall, Accuracy, F1-Score, and Area Under the Curve (AUC) - Receiver Operating Characteristics (ROC) curve.

The AUC - ROC curve is the performance evaluation metric to demonstrate how well the ML model can classify the output into different categories across a range of threshold values. ROC is a probability curve, whereas AUC represents a measure of separability degree. When AUC approaches 1, it indicates that the ML model has a high degree of separability. In contrast, the ML model has no potential for class separation if its AUC is less than 0.5 [21].

The confusion matrix is another performance measurement that demonstrates how well the predicted labels of the ML model are matched with the true labels. A good ML model will have only the diagonal elements (from left to right) of the confusion matrix, whereas the other elements are 0. The structure and definition of the confusion matrix are shown in Fig. 4 and Table 6, respectively.

		Predicted label	
		Predicted Positive (1)	Predicted Negative (1)
True label	Actually Positive (1)	TP	FP
	Actually Negative (0)	FN	TN

Fig. 4. The structure of the confusing matrix.

Table 6. The definitions of the confusion matrix.

Values	Definitions
TP	Predicted positive and the actual result is true.
TN	Predicted negative and the actual result is true.
FP	Predicted positive and the actual result is false.
FN	Predicted negative and the actual result is false.

The formulations to compute Accuracy, Precision, Recall, and F1-score metrics are shown in Eq. (4) - Eq. (7), respectively [22].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

## 4. Machine Learning Models

At the beginning of the research, abundant IM parameters (attributes) expected to affect the IM health index and the prediction accuracy of the ML model are often available for consideration. However, in reality, several of them may later be proved not significant by considering the result of the experiments. Therefore, the IM parameter selection through the application of statistical analyses is necessary. After the potential

parameters for modelling are identified, the next step is to establish the appropriate settings for each of them.

#### 4.1. Parameter Selection

Parameter selection is an essential process that is used to find the smallest subset of the IM parameters that significantly affects the IM health index. First, in this study, the Pearson Correlation Coefficient (PCC) technique, which measures the strength of a linear association between two or more variables, is applied. Such correlations enable us to see how the behaviour of one parameter is varied in relation to another. If the value of PCC between any two or more variables is more than 0.95 (highly correlated parameters), the strength of association among them is significant. Consequently, the only parameter  $x$  with the highest effect on the response  $y$  (severity) is selected as a representative of the group and the others are dropped from further consideration. Fig. 5

shows the PPC between 17 different parameters ( $x$ ) of the IM health data. Obviously, none of the elements in Fig. 5 is more than 0.95. This means that their correlations are not high enough to group any of them.

The next step is to apply multiple regression to the IM health index dataset to find if there exist any significant parameters and to create a regression equation that expresses the statistical relationship between one or more significant IM parameters and the response variable (severity). The Minitab software is a tool to analyse multiple regression. The analysis of variance (ANOVA) obtained from Minitab is shown in Table 7. It is observed that the P-values of the IM parameters including Acc\_DE\_RV, ACC\_NDE\_Ax, and VelDE\_RH are greater than 0.05 (significance level). This means these parameters are not statistically affected by the IM health index and therefore they are removed from the dataset [23],[24].

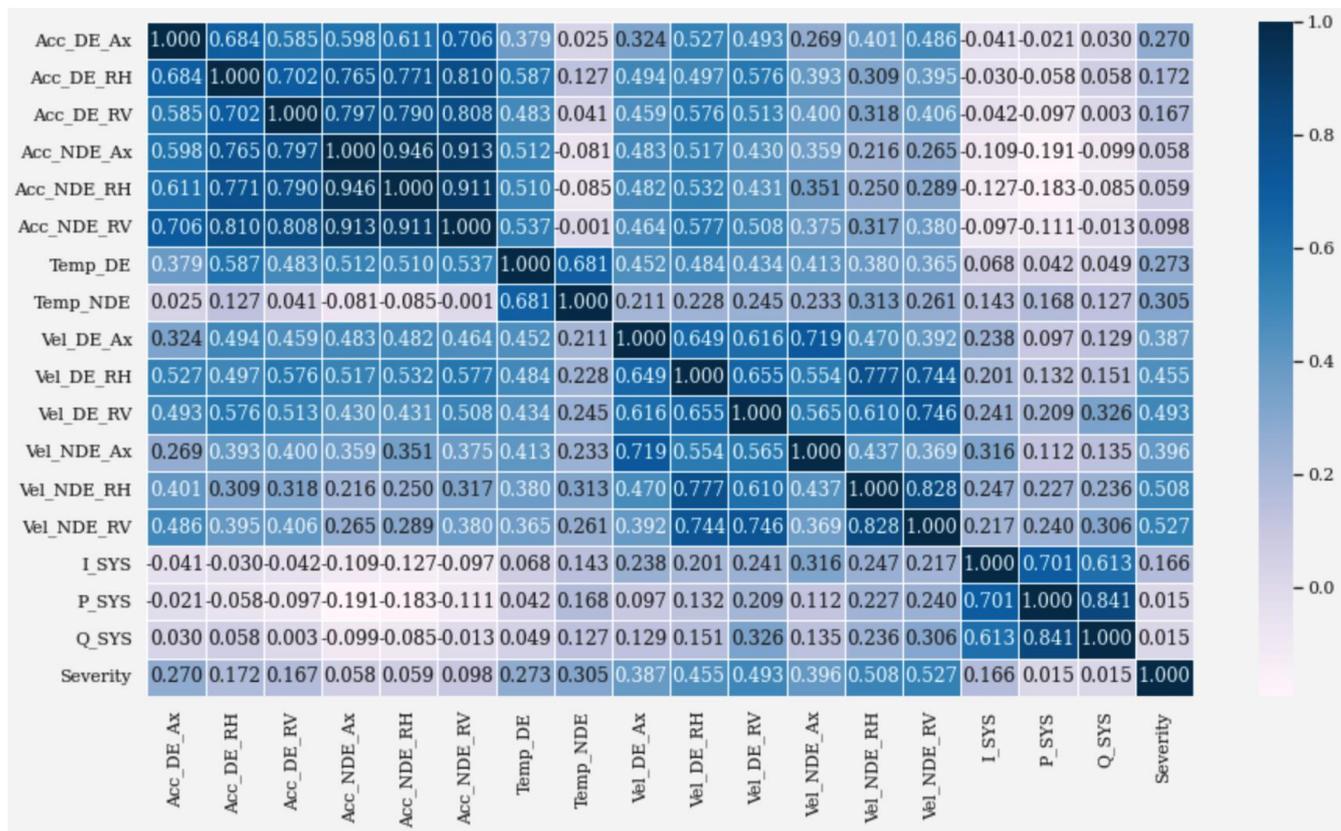


Fig. 5. PPC between different parameters for the IM health index prediction.

Table 7. Analysis of variance (ANOVA).

Analysis of Variance					
Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	17	382.358	22.4916	243.14	0.000
Acc_DE_Ax	1	15.266	15.2665	165.03	0.000
Acc_DE_RH	1	1.454	1.4541	15.72	0.000
Acc_DE_RV	1	0.067	0.0674	0.73	0.393
Acc_NDE_Ax	1	0.117	0.1169	1.26	0.261
Acc_NDE_RH	1	0.486	0.4860	5.25	0.022
Acc_NDE_RV	1	6.347	6.3469	68.61	0.000
Temp_DE	1	0.475	0.4754	5.14	0.023
Temp_NDE	1	1.576	1.5762	17.04	0.000
Vel_DE_Ax	1	4.514	4.5142	48.80	0.000
Vel_DE_RH	1	0.212	0.2117	2.29	0.130
Vel_DE_RV	1	12.135	12.1348	131.18	0.000
Vel_NDE_Ax	1	3.054	3.0541	33.02	0.000
Vel_NDE_RH	1	0.455	0.4548	4.92	0.027
Vel_NDE_RV	1	10.272	10.2724	111.05	0.000
I_SYS	1	6.096	6.0959	65.90	0.000
P_SYS	1	4.441	4.4412	48.01	0.000
Q_SYS	1	6.239	6.2391	67.45	0.000
Error	4744	438.847	0.0925		
Total	4761	821.205			

## 4.2. Artificial Neural Network

Artificial Neural Network (ANN) is an information processing and computing technique inspired by the human brain's information processing, called “biological nervous systems”. ANN comprises several nodes and layers which is analogous to neural in the human brain. Layers in ANN are categorised into input, (at least one) hidden and output layers. Several nodes often exist in each layer. The input nodes are connected to several nodes in the hidden layer which are subsequently coupled with the output nodes. A typical example of the ANN architecture is shown in Fig. 6. A weight and a threshold appear between linking nodes. If the output of an individual node (summing junction as shown in Fig. 7) exceeds a prespecified threshold value, that node is activated and data is allowed to transmit to the next layer. Otherwise, no data is sent to the next network's layer at all. While developing the model, weights and hidden layers are adjusted to maximise the effectiveness of the ML model.

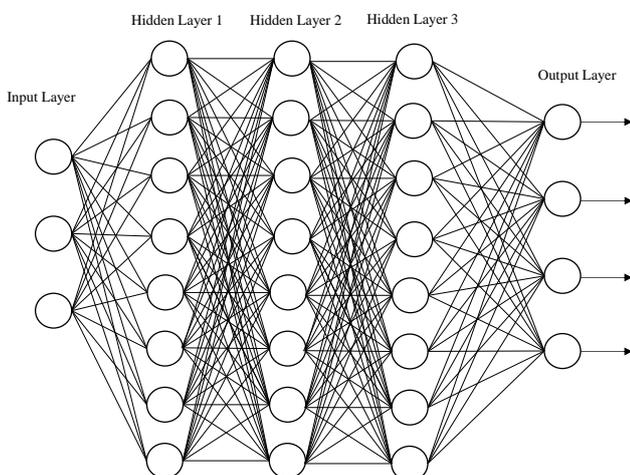


Fig. 6. ANN architecture.

### 4.2.1. Mathematical model

Consider a set of input nodes (signals), their respective weights, a bias, and an output as shown in Fig. 7, the formula to compute the output  $y_k$  of the  $k$  neuron is given in Eq. (8).

$$y_k = \left( \varphi \sum_{j=1}^N w_{kj} x_j + T_k^{\text{hid}} \right) \quad (8)$$

where  $\varphi$  denotes the activation (or transfer) function,  $N$  denotes the number of input neurons,  $w_{kj}$  denotes the weight,  $x_j$  denotes the input to the input neuron, and  $T_k^{\text{hid}}$  denotes the hidden neurons' threshold terms.

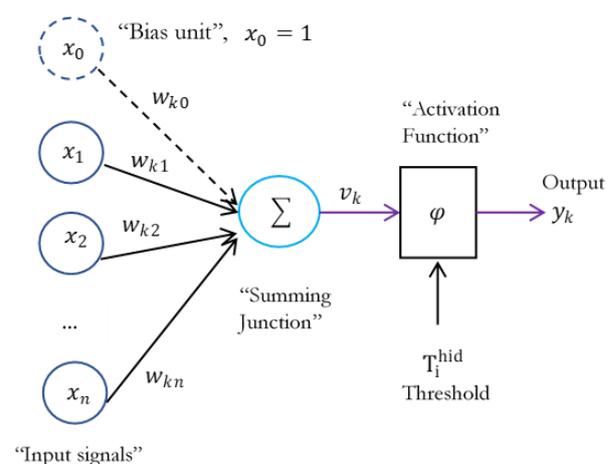


Fig. 7. Mathematical model of ANN.

### 4.2.2. Backpropagation algorithm

Backpropagation modifies the weights of the network connections repeatedly to minimise the difference

between the network's actual and desired output vectors [25]. Backpropagation seeks to reduce the cost function by modifying the network's weights. The gradients of the cost function for all those weights determine the level of adjustment. The goal is to identify the network that is most closely matches the observed data.

#### 4.2.3. ANN model

In this research, the ANN model is developed under the Python programming language installed in the Jupyter notebook. Scikit-learn which is a free Python library is used in the model construction. Optuna is used as a framework for automated hyperparameter tuning to dynamically select a set of hyperparameters suitable for the learning process of the algorithm. Besides, Optuna also allows the user to specify a search space for hyperparameter tuning. The hyperparameters of ANN in this research are shown in Table 8. Note that the default values are used for those that do not appear in Table 8. The ranges of the hyperparameters are adopted from [18].

An activation function in ANN, also known as a transfer function, is used to determine the output of the neural. In this study, Rectified Linear Unit (ReLU), as

shown in Eq. (9), which is the non-linear activation function mostly utilised in convolutional neural networks or deep learning, is used. It has become the default activation function for many types of neural networks because of its simplicity in model training and often achieves a better result. In addition, Stochastic Gradient Descent (SGD), a solver in ANN, is used to optimise the weights and generate updated parameters that minimise the loss function of the model.

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (9)$$

When the Optuna process is completed, the hyperparameter indicates how important the model is built from the IM dataset (Hyperparameter Importance). Where the number of trials ( $n_{\text{trials}}$ ) is set to 30 due to the calculation limitation. The resultant Hyperparameter Importance is shown in Fig. 8. The hyperparameter that mainly affects the model is `learning_rate_int` which controls the step size to update weights. The final hyperparameters used in the ANN model to create the output from the IM dataset are shown in Table 9.

Table 8. Hyperparameter search space of the ANN model.

Parameter	Parameter keys	Range
Hidden layer size	<code>hidden_layer_sizes</code>	[16,4], [32,4], [64,4], [128,4], [256,4]
Activation function for hidden layer	<code>activation</code>	['logistic', 'tanh', 'relu']
The solver for weigh optimisation	<code>solver</code>	['lbfgs', 'sgd']
Learning rate	<code>learning_rate_int</code>	[0.01,1.00, step=0.05]
Maximun number of iterations	<code>max_iter</code>	[50,500, step=50]

Table 9. Final hyperparameter of the ANN model.

Parameter keys	Range
<code>hidden_layer_sizes</code>	(64,4)
<code>activation</code>	'relu'
<code>solver</code>	'sgd'
<code>learning_rate_int</code>	0.35
<code>max_iter</code>	150

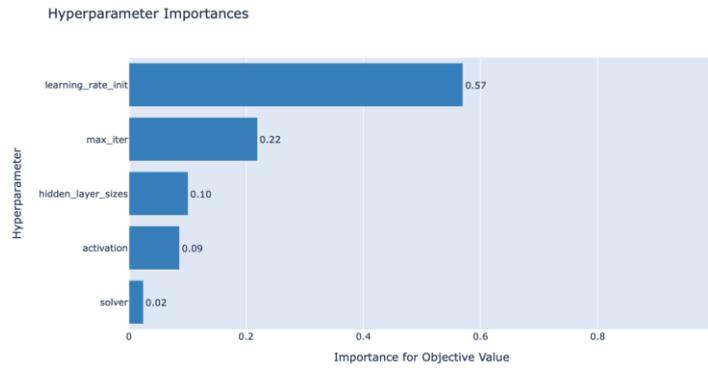


Fig. 8. Hyperparameter Importance of the ANN model.

### 4.3. Particle Swarm Optimisation

Particle swarm optimisation (PSO) is an evolutionary algorithm that is used to find the global maximum or minimum of a function by utilising a model of social interaction between independent agents (particles) that make use of social knowledge (also known as swarm intelligence) [25]. It has established itself as a practical global optimisation algorithm and a formidable competitor to the standard benchmark for function optimisation as a genetic algorithm.

PSO has been extensively used to solve a variety of different types of optimisation problems. In meta-heuristic algorithms, population initialisation is very important [26]. PSO places several simple entities called “particles” in the search space of a problem or function, and each evaluates the objective function at its current location. Each particle then determines its path through the search space by combining some aspect of its current and optimal (best-fit) locations with those of one or more swarm members, along with some random perturbations. After all, the particles have been moved, the next iteration begins. Eventually, the swarm as a whole, similar to a flock of birds foraging for food collectively, is likely to approach an optimal fitness function [27].

The traditional PSO searches through a population of particles corresponding to GA individuals. Each particle  $i$  represents a possible solution and is equipped with a current position vector  $x_i$ , a velocity vector  $v_i$ , and the best personal position  $p_i$  [28]. With the epoch  $t$ , the swarm consists of  $n$  particles travelling into the  $n$ -dimensional search space. Each particle calculates the cost function during each epoch, which is used to determine the particle's local best fitness, and then the least cost is used to determine the swarm's global best fitness. Eq. (10) is used by particles to update their position and velocity vectors.

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(p_g(t) - x_i(t)) \quad (10)$$

where  $w$  denotes the inertia weight,  $c_1$  and  $c_2$  denote the acceleration coefficients,  $r_1$  and  $r_2$  are uniformly distributed random numbers in the range (0,1). The

particle  $i$  can search around its individual best position  $p_i$ , and the global best position  $p_g$ . Based on the updated velocities is defined in Eq. (11). Note that  $x_i(t)$  and  $v_i(t)$  are the position vector and the velocity vector of the  $t^{\text{th}}$  speed and the position components of  $i^{\text{th}}$  particle.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (11)$$

The search space of the hyperparameters for PSO to experiment are adopted from [26],[29] as shown in Table 10. The final hyperparameters after the completion of the tuning process are shown in Table 11.

Table 10. Hyperparameter search space of the PSO model.

Parameter	Parameter keys	Range
Hidden layer size	n_hidden	[16, 32, 64, 128]
Acceleration coefficients	c1	[0.5, 1.49, 2.5]
	c2	[0.5, 1.49, 2.5]
Inertial weight	w	[0.4, 0.9]

Table 11. Final hyperparameter of the PSO model.

Parameter	Parameter keys	Range
Input Layer	input values	(14,)
	output values	(14,)
Hidden Layer	input values	(14,)
	output values	32
	Activation function	relu
Output Layer	Input values	(14,)
	Output values	32
	Activation function	softmax
Acceleration coefficients	c1	0.5
	c2	0.5
Inertial weight	w	0.4
Number of particles	Num of particles	500
Number of iterations	iter	400

#### 4.4. Gradient Boosting Tree

The decision tree is a rule-based model that creates an if-then-else rule from each feature's value without a formal relationship equation between the features and the target. The vital thing about decision tree construction is the selection of splits for each feature that results in the minimised cost function. Gradient boosting tree (GBT) is a technique that can be used for regression and classification. It consists of several decision trees that are connected in sequential order. Each decision tree learns errors from the previous trees and exploits this information to improve its prediction accuracy. When the learning of decision trees is deep enough, the model is stopped because no more error patterns from the previous trees to learn. In this study, GradientBoostingClassifier which is a multi-class classification decision tree is used.

Let  $x$  be the set of input variations.,  $x = \{x_1, \dots, x_n\}$  and the output is defined as  $y, y = \{y_1, \dots, y_n\}$ . So, the input data used for training are  $\{x_i, y_i\}_{i=1}^n$ . The loss function is defined in Eq. (12). Note that  $p$  is referred to the predicted probability, and  $N$  is the number of samples.

$$L(y_i, F(x)) = - \sum_{i=1}^N y_i \cdot \log(p) + (1 - y_i) \cdot \log(1 - p) \quad (12)$$

The initialised model with a constant value  $F_0(x)$  is defined in Eq. (13).

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, \gamma) \quad (13)$$

Similar to the ANN model, the Optuna framework is used to search for the appropriated parameters of the GTB model. The search space of the hyperparameters to experiment are shown in Table 12. The final hyperparameters and the hyperparameter importance for the GBT model are shown in Table 13 and Fig. 9, respectively.

Table 12. Hyperparameter search space of the GBT model.

Parameter	Parameter keys	Range
The rate of learning reduces the contribution of each tree.	learning_rate	[0.01,1.00, step=0.01]
The number of boosting stages to be performed	n_estimators	[0, 500, step=1]
The sample size fraction will be used to fit the individual base learners.	subsample	[0.1,1.0, step=0.1]
The minimum sample size is required to split an internal node.	min_samples_split	[2,5]
The minimum sample size is required to split an internal node.	min_sample_leaf	[1,5]
Individual regression estimators' maximum depth.	max_depth	[2,5]

Table 13. Final hyperparameter of the GBT model.

Parameter keys	Range
learning_rate	1.69
n_estimators	350
subsample	0.7
min_samples_split	5
min_sample_leaf	5
max_depth	5

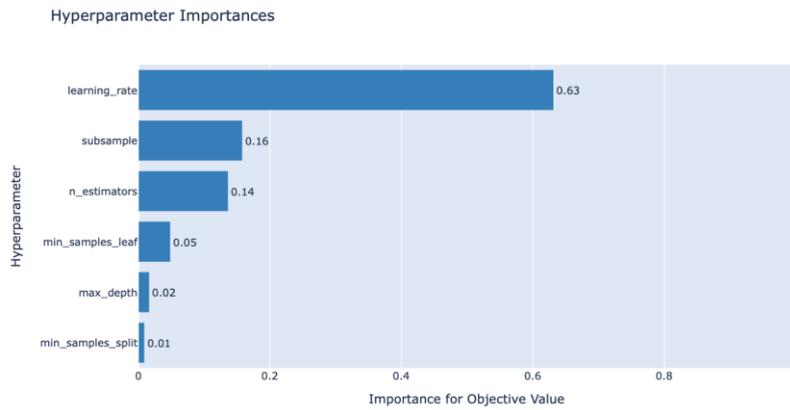


Fig. 9. Hyperparameter Importances of the GBT model.

#### 4.5. Random Forest

As a competitor to GBT, random forest (RF) have been developed. RF is a tree-based combination that relies on a set of random variables for each tree. As implied by its name, the RF algorithm is a supervised classification algorithm that classifies data by constructing multiple classifiers to improve prediction accuracy [30].

The RF classifier is made up of several different tree classifiers. A random vector sampled independently from the input vector is used to create each classifier. To classify an input vector, each tree casts a unit vote for the most popular class [16]. The final prediction, the mode of the classes for classification, is made by combining the predictions from all trees.

In the RF model, each tree in the ensemble is constructed from a sample drawn from the training set with a replacement, called a bootstrap sample. Furthermore, the best split is found either from all input features or a random subset of size max features when splitting each node during tree construction, from which the best parameter of the model can be found. Fig. 10 depicts the RF model. Instead of letting each classifier vote for a single class, the Scikit-learn method combines different classifiers by averaging their probabilistic predictions.

Once again, the Optuna framework is used to find the hyperparameter importance of the RF model. Table 14 shows the search space of the hyperparameters, some of which are adopted from [31]. The final values of the hyperparameters used for the RF model are shown in Table 15, and the hyperparameter importance is shown in Fig. 11.

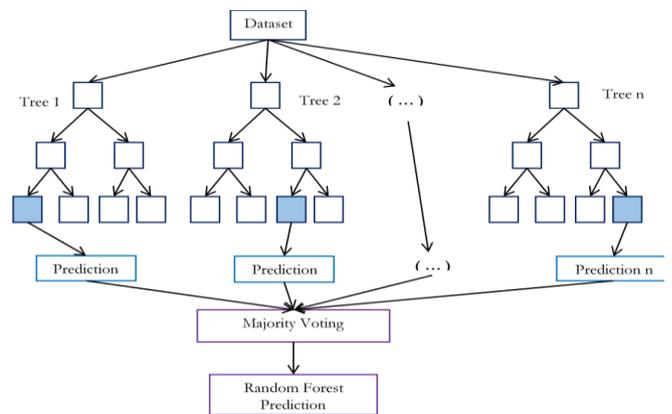


Fig. 10. Architecture of RF.

Table 14. Hyperparameter search space of the RF model.

Parameter	Parameter keys	Range
Number of trees in the forest	n_estimators	[200, 2000, step=10]
Max number of features considered for splitting a node	max_features	["auto", "sqrt"]
Max number of levels in each decision tree	max_depth	[10, 110, step=10]
min number of data points placed in a node before the node is split	min_samples_split	[2,5]
min number of data points allowed in a leaf node	min_samples_leaf	[1,4]
method for sampling data points (with or without replacement)	bootstrap	["True", "False"]

Table 15. Final hyperparameter of the RF model.

Parameter keys	Range
n_estimators	160
max_features	auto
max_depth	100
min_samples_split	5
min_sample_leaf	4
bootstrap	True

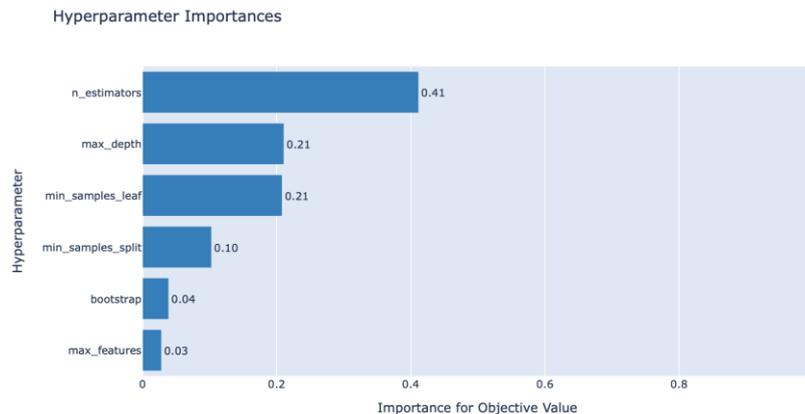


Fig. 11. Hyperparameter Importance of the RF model.

## 5. Result and Discussion

This section presents the relative performances of different ML models that are experimented on the IM dataset with the following specification, i.e. 475 kW, 2 Poles, 50 Hz, 3 Phase, 6.6 kV, 2988 rpm, squirrel cage. The IM operates under various daily load conditions specified by the case study petrochemical company. The ratio between the training dataset and the test dataset is 70:30. The parameters of all contestant algorithms are set at their optimal values to ensure that all of them are executed at their best performances to classify the IM health index into four categories. The metrics used in the evaluation of different methods include the classification report (i.e. Precision, Recall, F1-score and Accuracy) and confusion matrix.

### 5.1. Classification Report

As mentioned earlier, the objective of this study is to use ML to categorise the statuses of the IM into different health index levels ranging from Class 0 (new machine installation) to Class 3 (need immediate corrective action). Moreover, Class 2 and Class 3 must be given special attention since they are related to the near-failure states of the IM. As a result, a better model should have the values of precision, recall and F1-score metrics as close to 1 as possible. Table 16 - Table 18 show the classification reports of ANN, PSO, GBT and RF, respectively.

Considering in terms of the ability to predict the statuses of the IM, PSO is the most effective model since

the distribution of the Precision, Recall and F1-score metrics are quite uniform across four different classes (Class 0 to Class 3). Besides, these values are very close to 1. Although the performance of RF looks slightly inferior to PSO, it is much better than GBT and ANN.

The F1-score metric is a hybrid measurement. It is defined as a harmonic mean between the Precision and Recall metrics. As a result, F1-score is given higher priority than Precision and Recall and is often used to evaluate the overall performance of the ML model. From Table 16 - Table 19, it is obvious that the weighted average accuracy of PSO in terms of F1-score is the highest (0.97), followed by RF (0.96), GBT (0.93) and ANN (0.61), respectively. This means that PSO apart from having an excellent ability to classify data into different categories, such categories are higher accurate than the other competing ML models.

Table 16. Classification Report for the ANN Model.

	precision	recall	F1-score	support
Class 0	0.43	1.00	0.60	1195
Class 1	0.00	0.00	0.00	1253
Class 2	1.00	0.73	0.84	1241
Class 3	1.00	1.00	1.00	1228
accuracy			0.68	4917
Macro avg	0.61	0.68	0.61	4917
Weighted avg	0.61	0.68	0.61	4917

Table 17. Classification Report for the PSO Model.

	precision	recall	F1-score	support
Class 0	0.94	1.00	0.97	1195
Class 1	1.00	0.94	0.97	1253
Class 2	0.98	0.98	0.98	1241
Class 3	0.98	0.98	0.98	1228
accuracy			0.97	4917
Macro avg	0.98	0.97	0.97	4917
Weighted avg	0.98	0.97	0.97	4917

Table 18. Classification Report for the GBT Model.

	precision	recall	F1-score	support
Class 0	1.00	1.00	1.00	1195
Class 1	0.99	1.00	0.99	1253
Class 2	0.82	0.94	0.88	1241
Class 3	0.94	0.79	0.86	1228
accuracy			0.93	4917
Macro avg	0.94	0.93	0.93	4917
Weighted avg	0.94	0.93	0.93	4917

Table 19. Classification Report for the RF Model.

	precision	recall	F1-score	support
Class 0	0.99	1.00	1.00	1195
Class 1	1.00	0.99	0.99	1253
Class 2	0.95	0.92	0.94	1241
Class 3	0.92	0.95	0.94	1228
accuracy			0.96	4917
Macro avg	0.97	0.96	0.96	4917
Weighted avg	0.96	0.96	0.96	4917

result is true or predicted negative and the actual result is false.

Figures 12 - 15 show the confusion matrixes and AUC-ROC curves of ANN, PSO, GBT, and RF, respectively. Again, PSO is superior to the other ML models in distinguishing groups of data since its confusion matrix has dark blue highlights (high values) on all diagonal elements and all of its AUCs are greater than 0.5. This means the predicted labels (Class 0 – Class 3) of PSO is matched very well with the true labels. The second best is RF, followed by GBT and ANN, respectively. Among these four models, ANN is the worst and high predicted errors are often obtained from this model.

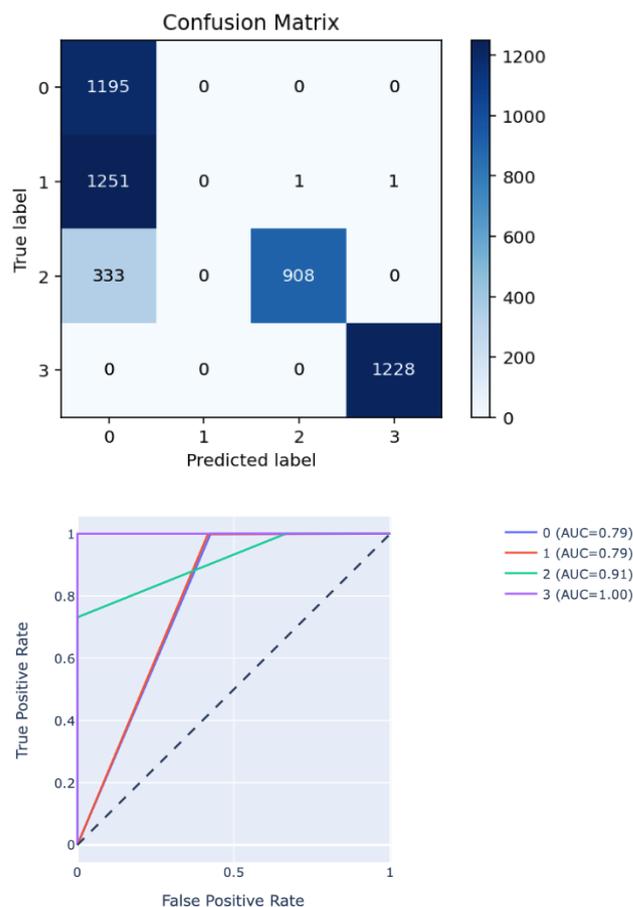


Fig. 12. Confusion Matrix and ROC curve for ANN.

## 5.2. Confusion Matrix and AUC-ROC curve

Another two metrics that can be used in measuring the ability to classify groups of data are the confusion matrix and the AUC-ROC curve. If the model is effective in categorising data into the correct groups, its confusion matrix should have data only on the diagonal elements (from left to right) of the matrix and the values of AUC of the ROC curve should be greater than 0.5. Note that the elements in the diagonal of the confusion matrix represent the matching between predicted positive and the actual

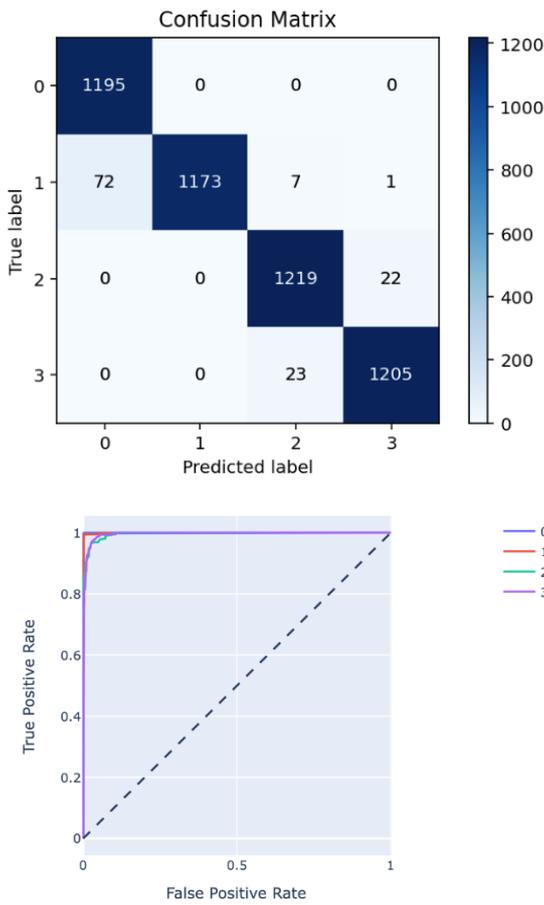


Fig. 13. Confusion Matrix and ROC curve for PSO.

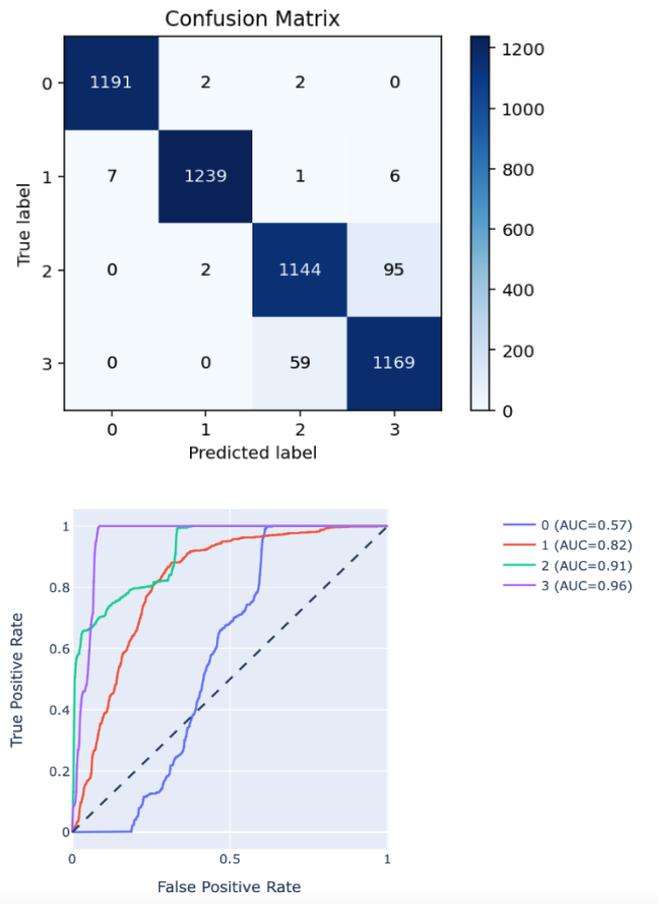


Fig. 15. Confusion Matrix and ROC curve for RF.

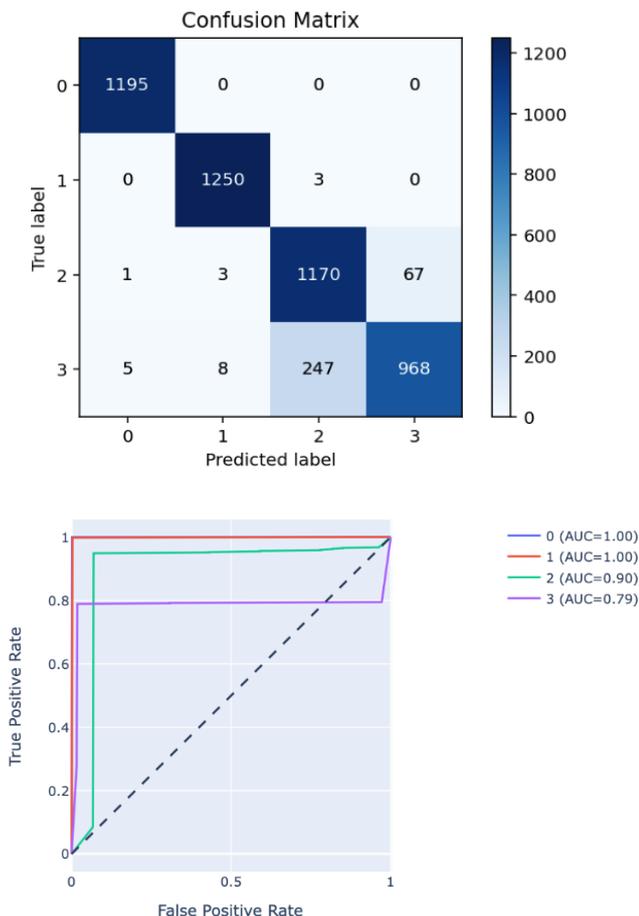


Fig. 14. Confusion Matrix and ROC curve for GBT.

## 6. Conclusions

This study combines data science and machine learning technique to predict the health index levels (Class 0 – Class 3) of the IM to help maintenance officers of the case study company to identify the need for appropriate maintenance actions in real-time. Intelligent sensors are used as a means for big data collection. Currently, such enormous information makes maintenance officers more prone to making wrong and non-timely decisions. Data cleansing is needed before the IM dataset is used in the training and prediction processes of ML to identify and resolve corrupt, inaccurate, or irrelevant data caused by sensor errors. In addition, the parameters (attributes) of the IM have to be checked for independence to determine if these attributes can be combined and reduced. Four predictive ML models are used including two computational models (i.e. ANN and PSO) and two ensemble models (i.e. GBT and RF). The performance measures used in benchmarking different ML models comprise Precision, Recall, F1-Score, Accuracy, confusion matrix and AUC-ROC curve. The experimental results indicate that PSO outperforms the other comparative ML models in terms of its ability to differentiate the statuses of the IM (i.e. highest values of the diagonal elements of the confusion matrix) as well as its discrimination accuracy (i.e. highest weighted average accuracy based on F1-score).

The results of this study are of great benefit to the performance of maintenance officers since the real-time health index status prediction, especially Class 2 (medium-risk severity level) and Class 3 (high-risk severity level), enable them to instantly make corrective or emergency maintenance on time. Besides, this information is very useful for consulting with experts for possible corrective maintenance solutions, rescheduling the maintenance plan, spare part procurement planning, etc. to prevent unplanned shutdowns of the plant.

In terms of financial benefits, the installation cost of the real-time data collection system including intelligent sensors connected to the IMs is around 700,000 baht. This system is utilised to predict and prevent unplanned equipment and/or production system downtime. The cost for equipment breakdown is approximated around 720,000 baht/time. Under the historical data of Yr 2020, seven breakdowns (class A) have occurred, which exclude the entire system disruption (class S). The payback period is less than one year (only 1.7 months) which is a very good figure for such an investment. Due to the successful implementation is achieved in this research, the company has a plan to extend this concept to cover other critical equipment such as rotating machinery. Moreover, a new computer server is also planned to purchase in the fiscal Yr 2022 to facilitate hourly predictive health monitoring of the entire system realisable.

## References

- [1] M. Seera, C. P. Lim, S. Nahavandi, and C. K. Loo, "Condition monitoring of induction motors: A review and an application of an ensemble of hybrid intelligent models," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4891-4903, 2014.
- [2] H. Tao, L. Mo, F. Shen, Z. Du, and R. Yan, "Multi-classifiers ensemble with confidence diversity for fault diagnosis in induction motors," in *2016 10th International Conference on Sensing Technology (ICST)*, IEEE, Nov. 2016, pp. 1-5.
- [3] R. N. Toma, A. E. Prosvirin, and J. M. Kim, "Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers," *Sensors*, vol. 20, no. 7, p. 1884, 2020.
- [4] V. N. Ghate and S. V. Dudul, "Cascade neural-network-based fault classifier for three-phase induction motor," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1555-1563, 2010.
- [5] H. Su and K. T. Chong, "Induction machine condition monitoring using neural network modeling," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 241-249, 2007.
- [6] J. F. Martins, V. F. Pires, and A. J. Pires, "Unsupervised neural-network-based algorithm for an on-line diagnosis of three-phase induction motor stator fault," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 259-264, 2007.
- [7] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Applied Soft Computing*, vol. 12, no. 1, pp. 353-359, 2012.
- [8] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta, and D. Benhaddou, "Parameters optimization of deep learning models using particle swarm optimization," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, June 2017, pp. 1285-1290.
- [9] J. K. Kim, Y. J. Choo, and M. C. Chang, "Prediction of motor function in stroke patients using machine learning algorithm: Development of practical models," *Journal of Stroke and Cerebrovascular Diseases*, vol. 30, no. 8, p. 105856, 2021.
- [10] B. A. Tama and K.-H. Rhee, "An in-depth experimental study of anomaly detection using gradient boosted machine," *Neural Computing and Applications*, vol. 31, no. 4, pp. 955-965, 2019.
- [11] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500," *European Journal of Operational Research*, vol. 259, no. 2, pp. 689-702, 2017.
- [12] S. Calzavara, C. Lucchese, and G. Tolomei, "Adversarial training of gradient-boosted decision trees," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, November 2019, pp. 2429-2432.
- [13] B. Raja, "Diabetics prediction using gradient boosted classifier," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 3181-3183, 2019.
- [14] S. Manna, S. Biswas, R. Kundu, S. Rakshit, P. Gupta, and S. Barman, "A statistical approach to predict flight delay using gradient boosted decision tree," in *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, IEEE, June 2017, pp. 1-5.
- [15] R. T. Selvi and I. Muthulakshmi, "Modelling the map reduce based optimal gradient boosted tree classification algorithm for diabetes mellitus diagnosis system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1717-1730, 2021.
- [16] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217-222, 2005.
- [17] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93-104, 2012.
- [18] W. Cheevachaipimol, B. Teinwan, and P. Chutima, "Flight delay prediction using a hybrid deep learning method," *Engineering Journal*, vol. 25, no. 8, pp. 99-112, 2021.
- [19] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1-6, 2004.

- [20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
- [21] S. Narkhede, "Understanding AUC-ROC curve," *Towards Data Science*, vol. 26, pp. 220-227, 2018.
- [22] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.
- [23] M. S. Thiese, B. Ronna, and U. Ott, "P value interpretations and considerations," *Journal of Thoracic Disease*, vol. 8, no. 9, p. E928, 2016.
- [24] R. A. Fisher, *Statistical Methods for Research Workers. Biological Monographs and Manuals No. V*, 11th ed. Edinburgh and London: Oliver and Boyd, 1950.
- [25] Y. Chauvin and D. E. Rumelhart, Eds., *Backpropagation: Theory, Architectures, and Applications*. Psychology Press, 1995.
- [26] H. T. Rauf, W. H. Bangyal, J. Ahmad, and S. A. Bangyal, "Training of artificial neural network using PSO with novel initialization technique, in *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, IEEE, November 2018, pp. 1-8.
- [27] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview," *Soft Computing*, vol. 22, no. 2, pp. 387-408, 2018.
- [28] J. Yu, L. Xi, and S. Wang, "An improved particle swarm optimization for evolving feedforward artificial neural networks," *Neural Processing Letters*, vol. 26, no. 3, pp. 217-231, 2007.
- [29] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, 2004.
- [30] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in *International Conference on Innovative Computing and Communications*, Springer, Singapore, 2019, pp. 253-260.
- [31] W. Koehrsen. "Hyperparameter tuning the random forest in Python." <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74> (accessed Jan10, 2018).



**Waritsara Khrakhuean** received her bachelor's degree in Electrical Engineering from Kasetsart University Sriracha Campus. She is a master's student in the Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University. She is currently working as an inspection engineer at GC Maintenance and Engineering Company Limited, Thailand.



**Parames Chutima** received his Engineering degree from Chulalongkorn University. He received his Master's degree from Chulalongkorn University and the Asian Institute of Technology. He received his PhD degree in Manufacturing Engineering and Operations Management from the University of Nottingham. Currently, he is a Professor in the Faculty of Engineering at Chulalongkorn University, Bangkok, Thailand. His research interests include multi-objective optimisation in operations management, production planning and control of assembly lines, just-in-time production systems and simulation modelling. He is the author of several national and international publications in conference proceedings and referred journals.