

Multidisciplinary  
SCIENTIFIC JOURNAL  
OF MARITIME RESEARCH



University of Rijeka  
FACULTY OF MARITIME STUDIES

Multidisciplinarni  
znanstveni časopis  
POMORSTVO

<https://doi.org/10.31217/p.37.1.4>

# Condition Monitoring and Fault Diagnosis of a Marine Diesel Engine with Machine Learning Techniques

Gazi Kocak<sup>1</sup>, Veysel Gokcek<sup>1</sup>, Yakup Genc<sup>2</sup>

<sup>1</sup> Istanbul Technical University, Maritime Faculty, Marine Engineering Department, Tuzla, Istanbul, Turkiye, e-mail: kocakga@itu.edu.tr (Corresponding author)

<sup>2</sup> Gebze Technical University, Department of Computer Engineering, Gebze, Kocaeli, Turkiye

## ABSTRACT

A marine engine room is a complex system in which many different subsystems are interacting with each other. At the center of this system is the main diesel engine which produces the propulsion force. Many other components such as compressed air, cooling, heating, lubricating oil, fuel, and pumping systems act as auxiliary machines to the main engine. Automation of many functions in the engine room is starting to play an important role in new generation ships to provide better control using sensors monitoring the engine and its environment. Sensors exist in the current generation ships, but engineers evaluate the sensor data for the presence of any problems. Maintenance actions are taken based on these manual analyses or regular maintenance is carried out at times determined by manufacturers, whether such actions are needed or not. With machine learning, it is possible to develop an algorithm using past evaluations made by engineers. Recent studies show that highly accurate results can be obtained using machine learning methods when there is sufficient data. In this study, we develop new learning-based algorithms and evaluate them on data obtained from a realistic ship engine room simulator. Data for a predetermined set of parameters of a high-power diesel engine were collected and analyzed for their role in a set of fault situations. These fault conditions and the associated sensor data are used to train a set of classifiers achieving fault detection up to 99% accuracy. These are promising results in preventing future damage to the engine or its supporting components by predicting failures before they occur.

## ARTICLE INFO

### Preliminary communication

Received 19 December 2022

Accepted 31 March 2023

### Key words:

Ship engine room  
Condition monitoring  
Fault diagnosis  
Machine learning

## 1 Introduction

Nowadays, computers can carry out many tasks that used to be done by human experts. This is achieved using machine learning provided there is sufficient data. Since computers can acquire and store very large amounts of data and process them quickly, for some tasks they can achieve better results than humans. There are various machine learning methods, and good performance can be obtained by choosing the appropriate method with their best parameterization. Of course, better than human performance depends on the type of problem and data.

The well-known machine learning algorithms offer solutions for regression [1] and classification problems including Support Vector Machines (SVM) [2], Decision Trees [3], Decision Forests [4], Boosting [5], and Arti-

cial Neural Networks (ANN) [6]. Selection of the best method or algorithm depends on the nature of the algorithm as well as the given modeling problem and its data. One important factor in this decision is the dimensionality of the data. Most problems are usually very high dimensional (e.g., a typical image VGA-resolution image has over 300,000 pixels, hence dimensions), and some machine learning methods suffer what is called curse of dimensionality where increasing the number of sensors degrades the performance of the trained model. In practice, the selection is done via trial and error supported with experience of the expert building or training the algorithm. One of the recent popular methods is deep learning [7]. Highly accurate results can be obtained using this approach, yet they rely on even more data for training.

Ship engine room is a very complex machinery system consisting of many subsystems. Since automation is getting more and more important in new generation ships, data from hundreds or even thousands of sensors can be processed and recorded in real time. These data are usually observed and evaluated by the engineer officer on the monitors in the control room. The engineer officers use their experience and engineering knowledge to evaluate the operation and performance of the system. In addition, the critical values for the systems are decided and set by the engineers and the system gives an alarm when these levels are reached. However, there is no system that can understand and evaluate a faulty condition while the values are slowly changing through a series of malfunctions. In other words, there is no intelligent system that can do the work of the engineer officer (in terms of evaluating the data). In this study, we are targeting an intelligent system that can evaluate the status of machinery systems. This system is to predict a malfunction before happening when the sensor indications are at significant levels but unobservable by a human expert. This therefore allows cost reductions by switching from the planned maintenance to the predictive maintenance.

There are studies developing machine learning models to predict engine room failures using multitude of data such as vibration, speed, tribology, etc. Li et al. in [8] uses the simple dynamic model of a 4-stroke small-power (40 kW) diesel engine and the data received from a real engine with an encoder that measures the instantaneous angular velocity. They define the ratio of the instantaneous angular velocity to the average angular velocity and use this ratio mainly for fault detection. Situations such as incomplete lubricating oil in a cylinder and no oil in are examined. Input properties are arranged with kernel independent component analysis (KICA) and Wigner bispectrum analysis. A total of 300 inputs were used in the study [8]. This work is expanded in [9] to learn error detection with fuzzy neural networks (FNN) after fusing the vibration data they received from 4 acceleration sensors placed on the engine in different directions with fast independent component analysis (FastICA). Xi et al. [10] obtained vibration data from a small-powered 4-cylinder diesel engine (boat or yacht engine) by running the engine at constant speed with the accelerometers placed on the cylinders, and error detection and visualization were performed using the ICA method. Antonic et al. [11] performed fault detection in a fully equipped engine room simulator. They proposed a method to correct the expert opinion using fuzzy logic methods without a need to train a machine learning model. Khelil et al. [12] modeled the lubrication system of a ship diesel engine in terms of hydraulics and thermals and made fault detection on this model using artificial neural networks. As input data to an artificial neural network, they first entered the machine speed and the fuel lever position, but then they reduced these to a single parameter because they observed that more accurate results can be obtained with only the machine speed. Kowalski et al. [13] measured 15

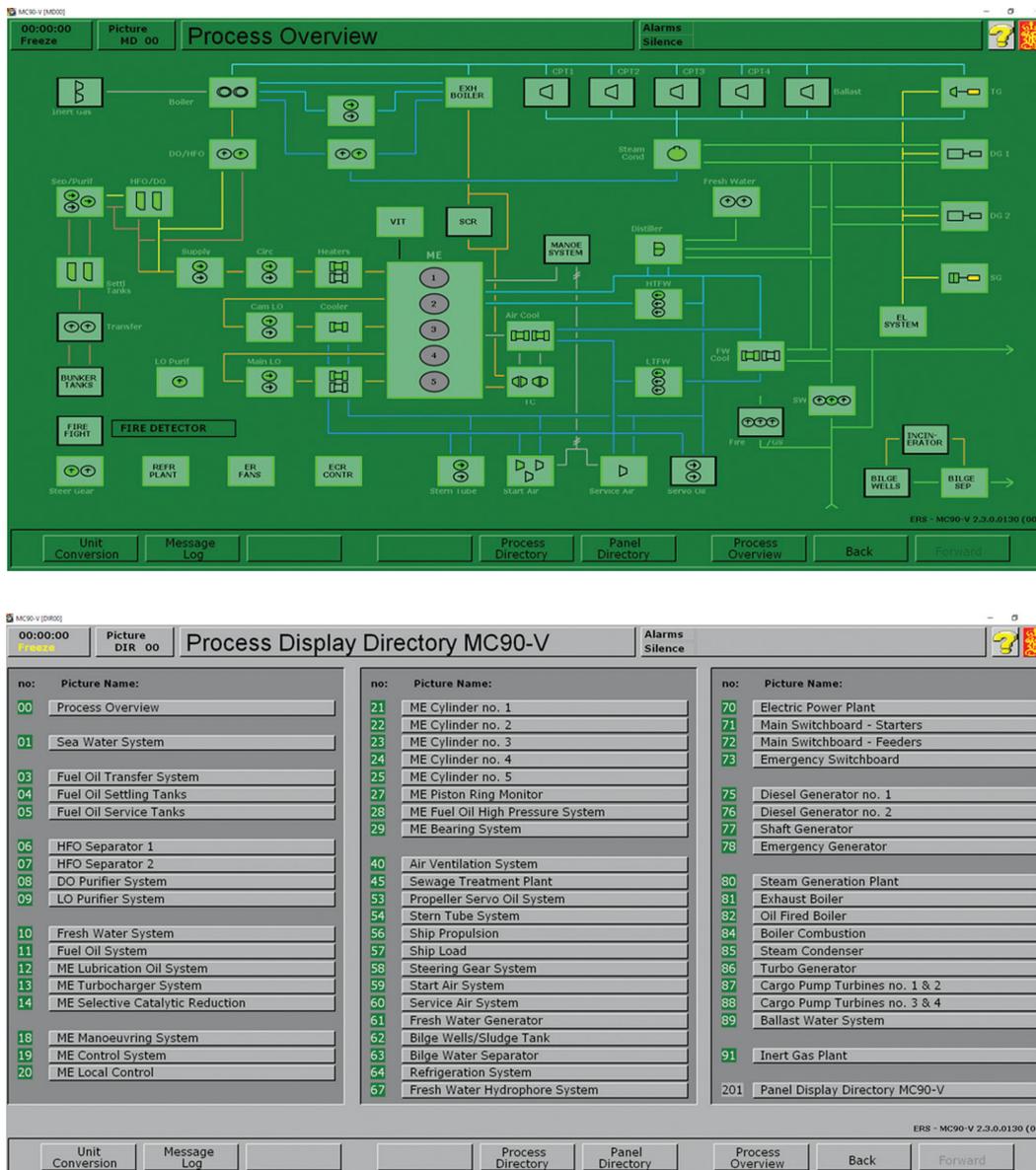
parameters such as ambient air temperature, air humidity, exhaust gas temperature, fuel injection pressure and combustion pressure from a small power (250 kW, 1000 rpm) 4-stroke ship diesel engine operating at constant speed (750 rpm). 10 different failure conditions such as air inlet valve leak, exhaust valve leak, injector blockage were examined. A total of 798 observation data were created and learning was carried out with the ensemble method. Nixon et al. [14] used sensor data and unscheduled maintenance records to monitor the condition of a diesel engine. The fuel pump failure was studied, but no information was given about which sensors were monitored and how much data was used. They experimented with LDA, SVM and Random Forest methods. Lazakis et al. [15] used the data of the noon-report (a set of important parameters from the ship is reported to the operating company on land at noon every day) to evaluate the machine performance. From the noon-report, the power, lubricating oil inlet temperature, lubricating oil pressure, cooling water inlet temperature, cylinder maximum exhaust temperature, cylinder minimum exhaust temperature data of the diesel generator were extracted and used in modeling. SVM method is utilized with 804 data obtained in a 317-day trip. Zhong et al. [16] modeled an AVL BOOST based fault simulation model. They took data for six different fault conditions at 100% load of the machine in their simulations. The conditions studied were variable oil inlet to the cylinder, insufficient air cooler, injector timing error (late injection and early injection), and normal condition. The data (with 21 thermodynamic parameters and 1320 samples) was used to build a model using deep belief networks (DBN). The paper does not give any details on the 21 parameters used. Tan et al. [17] conducted a study to detect faults in the fuel injection system of a ship diesel engine. SVM algorithm was applied to detect failure conditions such as fuel circulation pump wear, circulation pump motor failure, and fuel filter pollution. Hou et al. [18] reported some failure conditions such as cylinder liner crack, burst fuel pump high pressure pipes, scavenger fire, fuel pump wear in a ship diesel engine, using the cylinder exhaust temperature, cylinder coolant temperature, piston cooling oil temperature, cylinder combustion pressure, and cylinder compression pressure as parameters. 7000 samples were used to train a multilayer ANN along with a genetic algorithm to detect failures. The long run-time of the proposed method is stated as a disadvantage. Qi et al. [19] proposed a simple regression model in a ship diesel engine using machine power, engine speed, scavange volume, scavange pressure, exhaust temperature, to detect malfunctions (by thresholding the regressor output). A total of 15800 samples from a MAN B&W 6S42 type machine were used. Ellefsen et al. [20] used error-type independent spectral anomaly detection method to determine the degradation in a diesel engine of an autonomous ferry. Samples were taken from a diesel engine in a laboratory environment for two different load cases. The data was observed for two different load states for normal operation and for each degradation state. Air filter, turbo and cooling deterioration conditions were inves-

tigated. 47 parameters (including machine power, coolant temperature, exhaust gas temperature, and engine speed) were used to train a model with a variational autoencoder (VAE) to predict deterioration conditions.

As one of the most important subsystems in a ship, the main engine is frequently addressed in many automated prediction studies. However, most of these studies concentrate on a part of the main engine. In this study, we address some fault conditions in a ship's main engine. We collect a large amount of data from a high-fidelity simulator representing the fault conditions and use this data to train machine learning models with high accuracy and run-time performance. We evaluate the effectiveness of various machine learning methods. The data and results of the models are presented in detail.

## 2 Methodology

We address the problem of failure detection in a diesel engine used as the main engine of a tanker ship. As mentioned above, learning-based methods to solve this problem requires enough data representing all the possible failure and non-failure cases in a balanced way. A realistic engine room simulator shown in Figure 1 was used to obtain such data. The utilized simulator is the engine room simulator of the Kongsberg Maritime (Norway) and model is K-Sim Engine MAN B&W 5L90MC VLCC (Very Large Crudeoil Carrier) tanker. The specification of the modelled tanker is 187997 tons deadweight, and the navigation speed is 14 knots while the main engine is 18 MW (CSR: Continuous Service Rating) at 74 rpm.



**Figure 1** Two screenshots from the simulator used in the experimental studies: Engine room system (top), process directory (bottom)  
 Source: Author using KS Model MC90V

**Table 1** The parameters and the related malfunctions

	Label	Unit	Information	M2508	M2506	M2507
1	T01010	degC	HTFW temp inlet ME			x
2	T01603	degC	ME exh receiver temp		x	
3	T01613	degC	ME TBCH 1 exh outlet temp		x	
4	T02040	degC	ME cyl 1 exh outlet temp	x	x	x
5	T02041	degC	ME cyl 1 exh outlet temp deviation	x	x	x
6	T02043	degC	ME cyl 1 wtr outlet temp (liner)	x	x	x
7	T02044	degC	ME cyl 1 oil outlet temp (piston)		x	x
8	T02045	degC	ME cyl 1 cover temp (mean)		x	
9	T02046	degC	ME cyl 1 liner temp (mean)		x	x
10	T04600	degC	TG inlet steam temp	x		
11	G01154	ton/h	ME FW exp tank overflow			x
12	G02011	ton/h	ME fuel oil consumption	x		
13	G02012	g/kWh	ME specific fuel oil consumption	x		
14	G02050	kg/h	ME cyl 1 FO flow	x	x	x
15	G02052	ton/h	ME cyl 1 oil flow			x
16	G02053	ton/h	ME cyl 1 wtr flow			x
17	G02057	g/kWh	ME Cyl 1 specific fuel consumption	x		x
18	P01005	bar	HTFW press inlet ME			x
19	P01602	bar	ME exh receiver press		x	
20	L01150	m	ME FW exp tank level			x
21	Z01164	%	ME FW system gas detector			x
22	E02005	MW	ME shaft power (to propeller )	x	x	
23	Q02004	kNm	ME shaft torque	x	x	
24	N01610	rpm	ME TBCH 1 speed		x	

Source: Author

The biggest advantage of using a realistic simulator is that we can create a fault and observe how the observations (sensors) change before, during and after these faults. We examined a few fault conditions in the main engine system. The sensors or parameters are also selected. These fault conditions were created in the simulator and sensor readings were recorded. In addition, data were obtained in the simulator for the normal or non-defective situation. The obtained data and their relations with each other were examined and various machine learnings models were trained, and their performance are analyzed.

We studied the following three faults:

1. Injection valve nozzle clogged (Fault code: M2508)
2. Exhaust valve leakage (Fault code: M2506)
3. Cylinder liner crack (Fault code: M2507)

There are many sensors in the ship engine room. When modeling each fault condition, the parameters related to this fault need to be examined. We decide on the parameters to be used for each fault situation separately. There are different parameters for different faults as well as some common features. For example, 'G02050', 'T02040',

'T02041', 'T02043' parameters are common to all faults. Table.1 lists all these parameters and which parameter is used in which fault condition. After determining the necessary set of parameters, scenarios of failure situations were generated in the simulator, and the changes in the parameters were observed. Faults were inspected for the first cylinder of the main engine. The image of the simulator for this cylinder is shown in Figure 2.

Scenarios for all fault cases were created as follows. After the simulation starts, it works normally for 5 seconds and then the fault is triggered. Faults start from 0% as a percentage and reach their maximum value within 30 minutes. The maximum value differs according to the fault type. The reason for this difference is that the main engine automatically slows down depending on the type of fault. More data after this will not be meaningful to us as the whole system behavior changes when the main engine is at slow down condition. The maximum fault percentage for fuel injection valve clogged is 59%, because when the fault value is around 59-60%, the main engine slows down. The maximum fault percentage for the exhaust valve leakage is determined as 62%. The maximum value is taken as 100% in case of cylinder liner leakage, because

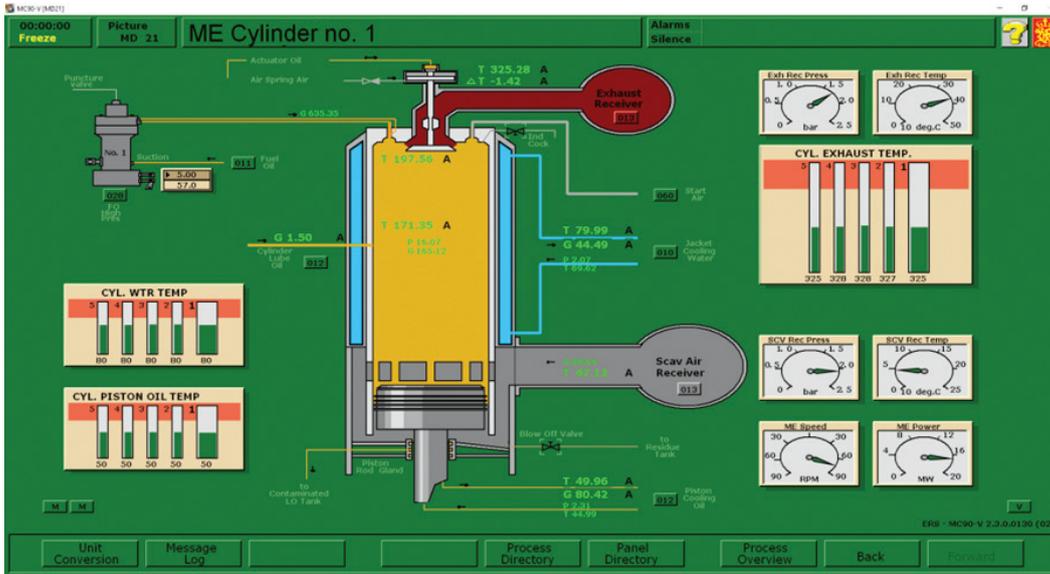


Figure 2 Main engine cylinder no. 1

Source: Author using KS Model MC90V

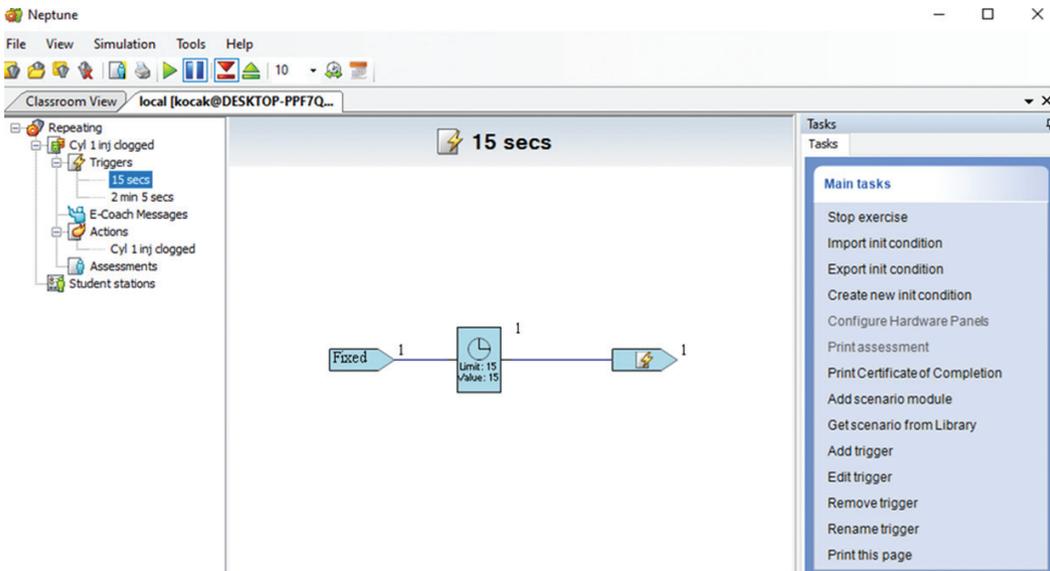


Figure 3 A sample scenario

Source: Author

the main engine does not slow down. Creation of scenarios in the simulation is explained in Figure 3 showing the general view of a scenario. It is seen how “trigger” and “action” operations are modelled in the left menu of the scenario screen. Here trigger specifies the state we want to trigger. “action” can be any failure condition or another process. It is possible to create time dependent scenarios by using these processes and operations in the timer.

Figure 4 shows the “action” activated by this trigger. The action created here is the clogging of the fuel injection

valve of the main engine cylinder no 1. In this example, we can see that there are multiple parameters to model the fault condition such as ramp up duration, on value, off value, on duration, etc.

Figure 5 shows how the values of two selected parameters change during a fault scenario. Automating the data collection was not possible because the simulator allows resetting only the fault state, while the other data continued from the last value in the previous fault state. Therefore, at each iteration the parameters are reset by

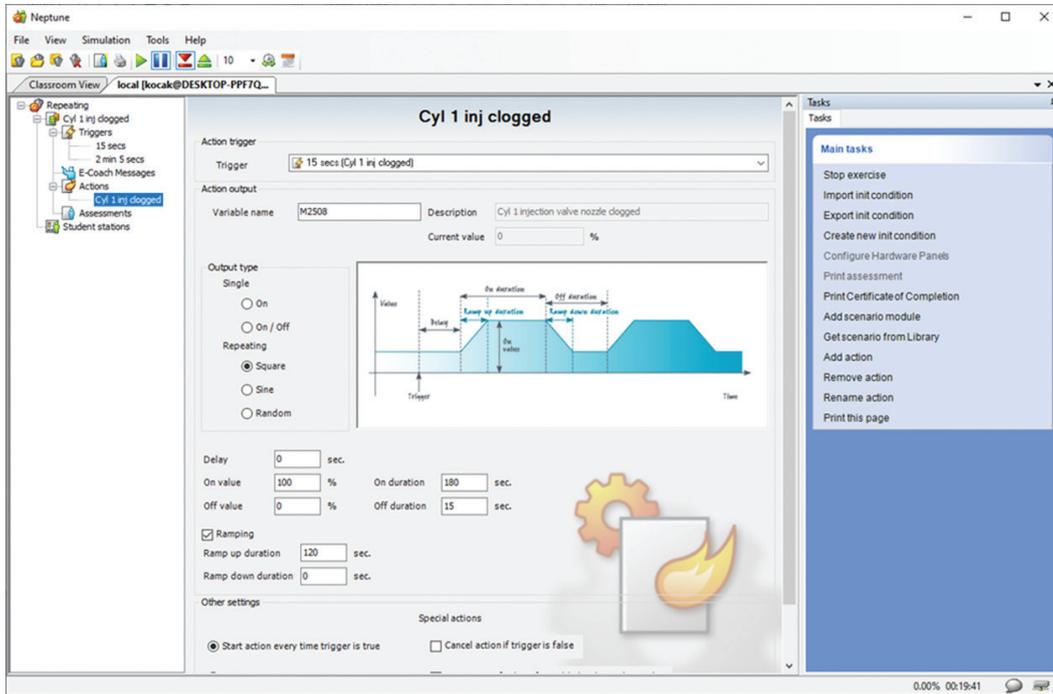


Figure 4 The action settings of a scenario

Source: Author



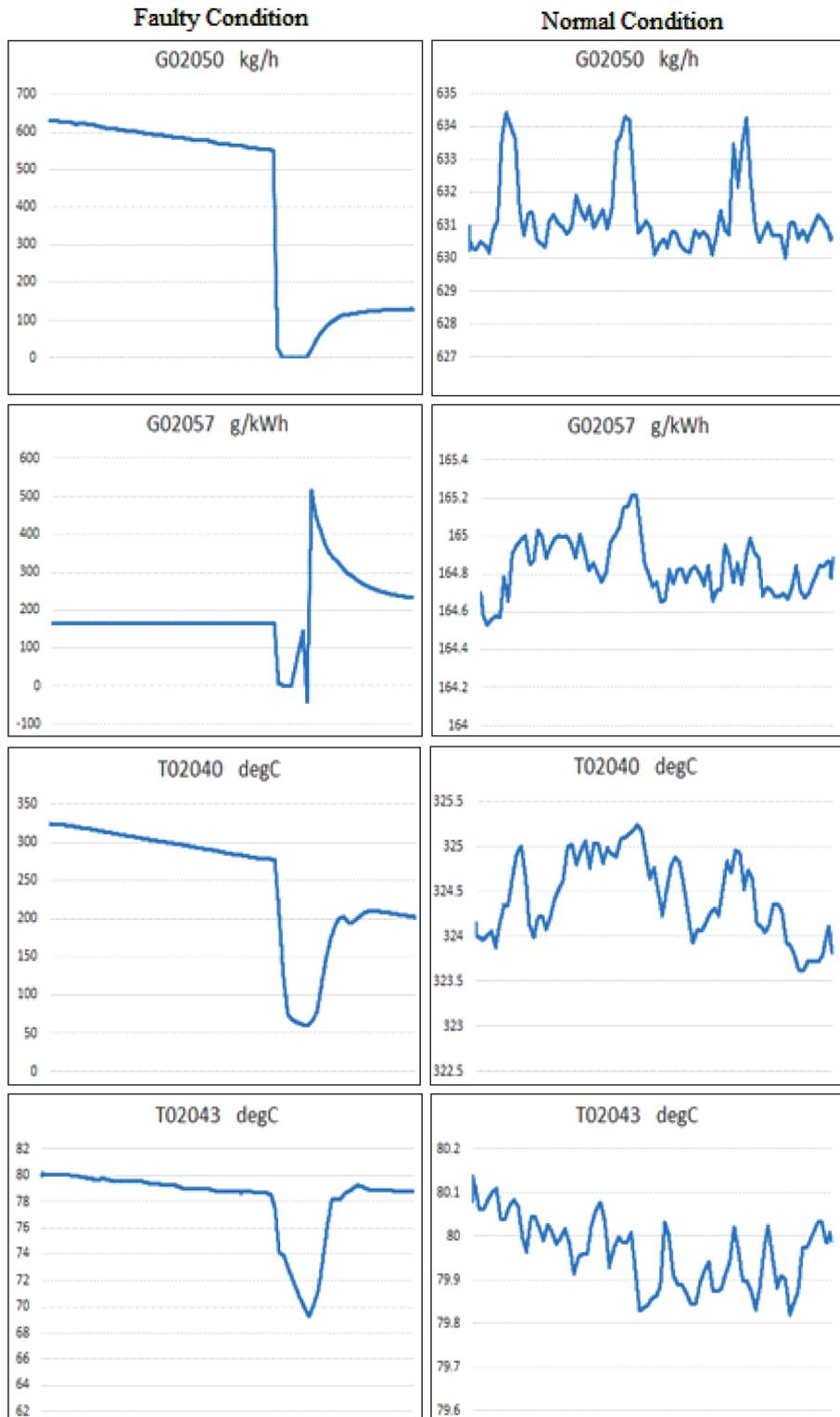
Figure 5 The values of two parameters during a fault condition

Source: Author

restarting the simulation for each fault situation. This of course makes data collection a more laborious.

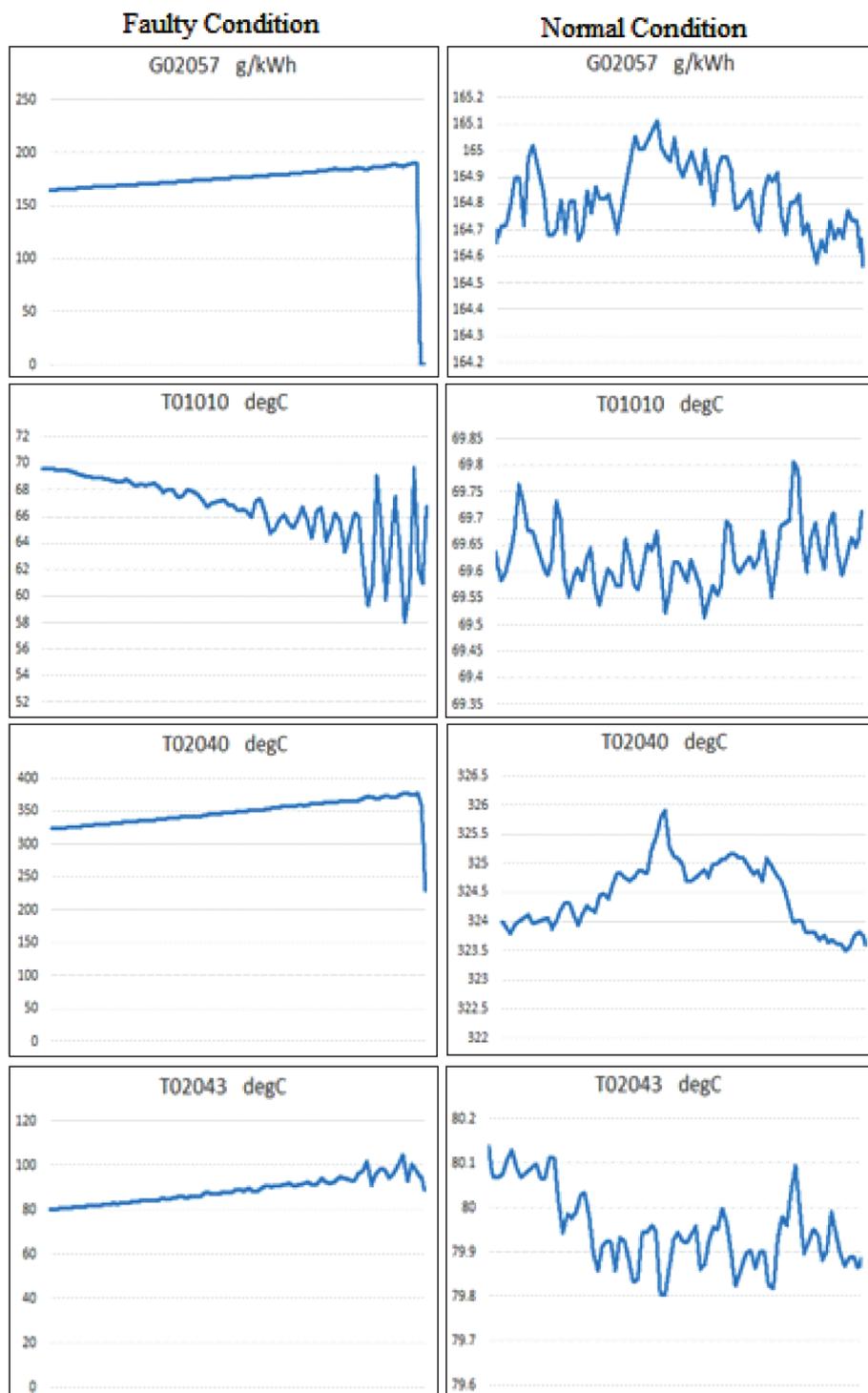
Figure 6 shows the normal and fault state values of some of the parameters used. Temporal variation of four param-

eters during a cylinder liner crack and a normal operation is shown in Figure 7. As seen in the examples given in Figures 6 and 7, there are distinct patterns in normal and fault situations of the temporal variations of the observed sensor data.



**Figure 6** The temporal behavior of some parameters for faulty (left column) and normal (right column) situations for Fault M2508

Source: Author



**Figure 7** The temporal behavior of some parameters for faulty (left column) and normal (right column) situations for Fault M2507

Source: Author

Once the data is obtained (and after the analysis of the data), the machine learning stage starts. We are aiming for classifiers to detect failure cases. This can also be posed as a regression problem with fault values (with percentages, 0% indicating no fault, 100% indicating full fault). The number of features (or the dimensionality) is selected per fault problem (three situations discussed above). For ex-

ample, while the input data is 10 dimensional in fuel injection valve clogged failure, it is 13 dimensional in exhaust valve leakage failure. If we think of it as a 13-dimensional vector, we add the fault percentage, which is the output parameter, to be the 14th dimension. The machine learning algorithm actually uses the test data and estimates the fault value which is the output. Here, the normal and faulty

states of the parameters are collected with corresponding output value and the machine learning algorithm carries out the model building process using the given data. At the normal or fault-free condition the fault status is taken zero. About 60% of the data was used for learning and 40% for testing. While the system is running, it is possible to obtain the percentage of the modeled fault condition continuously. Thus, it will be possible to take precautions when the fault condition is at a certain level. The flow chart of the algorithm used is shown in Figure 8.

Prediction models can be explained as a process of obtaining an output using specified system parameters or sensor inputs. The resulting output may differ depending on the model. In this study, the percentage status of the failure is predicted. Regression models explain the relationship between the multidimensional input parameters and the continuous output parameters. By utilizing the supervised learning method, it is possible to predict the value of the output parameter when building these models. The purpose of the teaching process is to minimize the

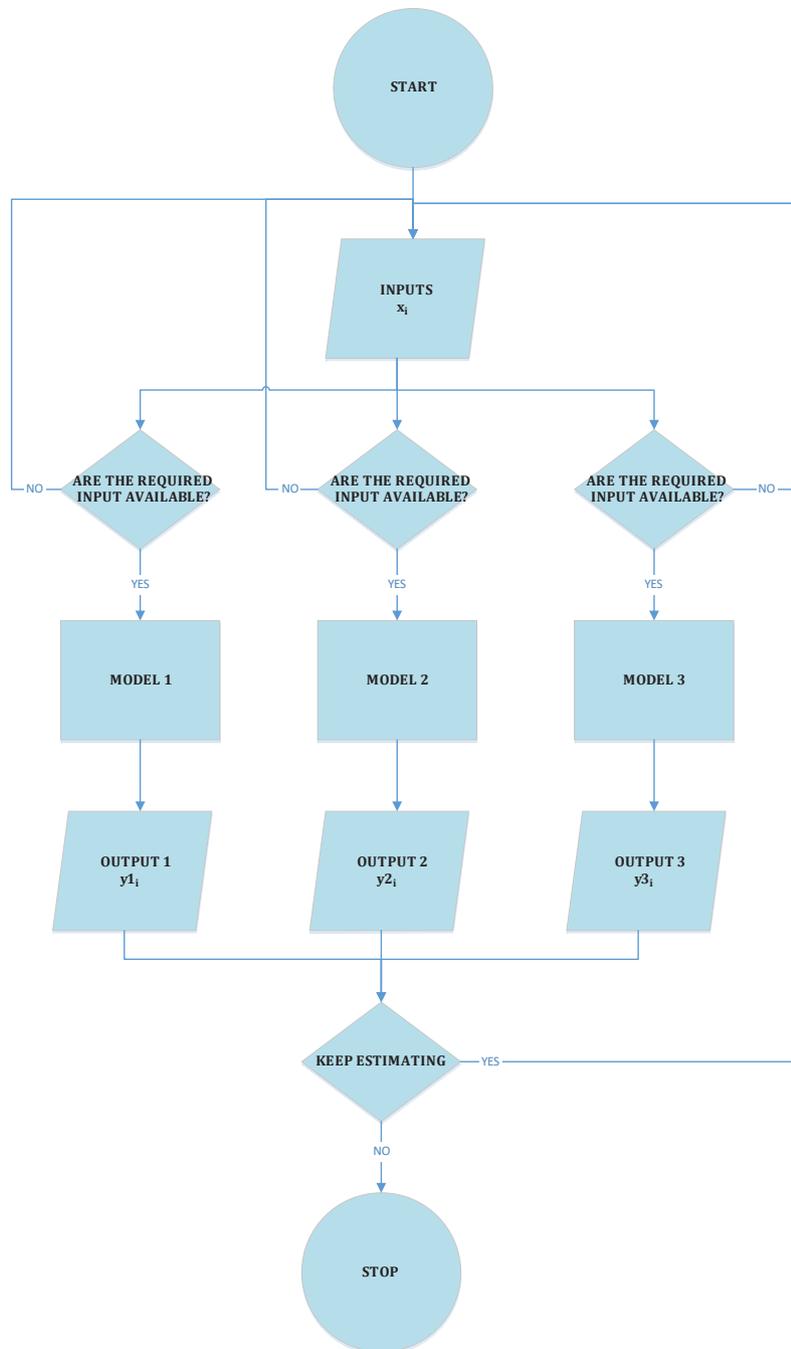


Figure 8 Flow chart for the training process

cost function generated to find the best function representing the data. Thus, the cost function also allows us to measure the error. The most commonly used cost functions are the mean squared error (MSE) and root mean squared error (RMSE). In this study, RMSE was used as a cost function in the learning phase.

Linear regression predicts a dependent variable value based on a particular independent variables. Hypothesis function for linear regression is:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (1)$$

where  $\hat{y}$  is the predicted value of a dependent variable,  $x_1, x_2, \dots, x_n$  are independent variables, and  $\theta_0, \theta_1, \dots, \theta_n$  are the regression coefficients.

If the effect of one variable depends on the value of another variable, we must consider interactions between the variables in linear regression. For two input data, interaction produces another input field for their multiplication:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 \quad (2)$$

Here  $\theta_3$  is a regression coefficient and  $x_1 x_2$  is the interaction. The interaction between  $x_1$  and  $x_2$  is called a two-way interaction as it is the interaction between two independent variables. Two-way interaction was also used in this study.

Another linear regression method is stepwise regression. This regression model performs multiple regressions several times, removing the weakest correlated variable each time. In the end, the variables that best explain the distribution remain. Robust linear regression, on the other hand, provides a more accurate model by clearing outliers. In this study, linear regression, interactive linear regression, robust linear regression and stepwise linear regression models are used.

The support vector machine (SVM) is a popular machine learning tool for classification and regression. There are both linear and nonlinear SVM methods. SVM regression can be extended to a nonlinear domain using a kernel. A nonlinear function moves the original dataset to a higher dimensional space, which makes the data hopefully separable. The polynomial kernel function equation used in this study is as follows:

$$K(x_1, x_2) = (x_1 x_2 + c)^d \quad (3)$$

where  $d$  is the degree of the polynomial function. Value of  $d$  is parameter defining the increase in dimensionality where  $d = 1, 2, 3$  is yield linear, quadratic, and cubic kernels respectively. Apart from these, the Gaussian kernel provides good discrimination in high dimensions for nonlinear problems. Linear SVM, cubic SVM, quadratic SVM, coarse Gauss SVM, medium Gauss SVM and fine Gauss SVM methods were used in this study.

Another family of non-linear algorithms are regression trees. Tree-based models split data multiple times based on certain cutoff values. Different subsets of the dataset

are created, with each sample belonging to a subset. Final subsets are called end or leaf nodes, and intermediate subsets are called internal nodes or split nodes. The average result of the training data at that node is used to predict the outcome at each leaf node. The depth of the tree defines may affect the fitness of the tree (overfit if a fully grown tree is built). Ensemble models of regression trees are also possible. Ensemble models combine many weak (and potentially overfit) learners into a high-quality ensemble model. In this study, various levels of fully grown and pruned (fine, medium, and coarse) trees, as well as boosted ensemble trees and bagged ensemble trees are built.

Performance of a model's prediction (regression) can be measured using mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), or square error (R2). While RMSE is used as the objective function in finding the best regression model, these additional metrics provide different aspects of the error made during testing. Assuming the expected output is  $y_j$  and the estimated output is  $\hat{y}_j$  (and baseline algorithm's performance is  $\bar{y}$ ), the following equations can be used calculate the error metrics (assuming we have  $j = 1, \dots, n$  samples in our test set):

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (4)$$

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} = \sqrt{MSE} \quad (6)$$

$$\hat{R}^2 = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (y_j - \bar{y})^2} = 1 - \frac{MSE(model)}{MSE(baseline)} \quad (7)$$

### 3 Results

The methods described in the previous section were implemented and run using the data obtained for three different fault conditions. An i7-10750H 2.60GHz, 16GB Ram capacity computer was used to train and test the failure data.

For fuel injection valve clogging fault (M2508), a 10-dimensional input data is generated using parameters 'E02005', 'G02011', 'G02012', 'G02050', 'G02057', 'Q02004', 'T02040', 'T02041', 'T02043', and 'T04600'. Since the values of these data are in a wide range, normalization has been applied. 120,000 samples are available for this fault condition. 80,000 samples were used as test data to measure the performance. The regression output is expected to be withing 0-59% range (no observations above 59%).

In addition, principal component analysis (PCA) method is applied to observe how each parameter represents the data. The parameters obtained by determining the

**Table 2** 95% PCA results for M2508 fault

Input parameter	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>
	51.40%	23.90%	7.70%	6.50%	5.60%
'E02005'	0.65	0.10	-0.03	0.11	0.076
'G02011'	-0.10	0.46	-0.16	-0.47	0.705
'G02012'	0.68	0.07	-0.02	0.13	0.095
'G02050'	-0.08	-0.03	-0.19	0.70	0.393
'G02057'	0.30	0.04	0.11	-0.42	-0.226
'Q02004'	-0.01	0.35	0.61	0.19	0.158
'T02040'	-0.04	0.03	0.51	0.04	0.03
'T02041'	-0.04	0.68	-0.20	0.18	-0.287
'T02043'	-0.11	0.42	-0.09	0.10	-0.417
'T04600'	-0.05	0.04	0.49	0.04	0.026

Source: Author

95% variance threshold after applying PCA are shown in Table 2. The most distinguishing parameter is 'G02012', while the least distinguishing parameter is 'Q02004'. The parameters 'G02012' and 'E02050' were observed as the main separators.

The learning and test results for this fault condition are shown in Table 3. The best performance is obtained with the "Fine Tree" method. During learning, a high  $R^2$  value of 0.9996 was achieved with this method. Even though the fully grown tree is expected to overfit the data, this model generalizes well during the tests with a high  $R^2$  value of 0.9992.

**Table 3** Training and test results of M2508 fault

Machine Learning Method	Learning (60%)				Test (40%)			
	$R^2$	MAE	RMSE	MSE	$R^2$	MAE	RMSE	MSE
Linear Regression	0.9987	0.70	0.88	0.78	0.9984	0.80	1.00	0.99
Linear Reg. with Interactions	0.9993	0.52	0.66	0.44	0.9991	0.59	0.75	0.56
Linear Regression Robust	0.9987	0.70	0.88	0.78	0.9984	0.80	1.00	0.99
Stepwise Linear Regression	0.9993	0.52	0.66	0.44	0.9991	0.59	0.75	0.56
Linear SVM	0.9987	0.75	0.94	0.88	0.9984	0.84	1.03	1.07
Cubic SVM	0.9993	0.74	0.88	0.78	0.9991	0.80	0.96	0.91
Quadratic SVM	0.9993	0.75	0.89	0.78	0.9993	0.81	0.96	0.93
Coarse Gaussian SVM	0.9992	0.68	0.84	0.70	0.9990	0.76	0.93	0.86
Medium Gaussian SVM	0.9991	0.69	0.84	0.71	0.9989	0.77	0.95	0.90
Fine Gaussian SVM	0.9986	0.83	1.00	1.01	0.9970	0.96	1.43	2.04
Fine Tree	<b>0.9996</b>	0.36	0.49	0.24	<b>0.9992</b>	0.51	0.71	0.50
Medium Tree	0.9996	0.37	0.50	0.25	0.9992	0.51	0.70	0.49
Coarse Tree	0.9996	0.37	0.51	0.26	0.9992	0.52	0.71	0.51
Ensemble Boosted Trees	0.9991	1.36	1.65	2.74	0.9989	1.36	1.64	2.70
Ensemble Bagged Trees	0.9991	1.36	1.65	2.74	0.9989	1.36	1.64	2.70

Source: Author

**Table 4** 95% PCA results for M2507 fault

Input parameter	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
	70.00%	14.30%	7.40%	3.40%
'G01154'	0.30	-0.04	0.02	0.02
'G02050'	0.38	-0.06	0.02	-0.05
'G02052'	-0.17	0.03	-0.01	0.04
'G02053'	0.38	-0.06	0.02	-0.06
'G02057'	0.39	-0.06	0.02	-0.07
'L01150'	0.27	-0.05	0.02	-0.06
'P01005'	0.37	-0.07	0.02	-0.11
'T01010'	0.11	0.95	-0.03	-0.29
'T02040'	-0.05	0.04	1.00	0.00
'T02041'	-0.18	0.01	0.00	0.05
'T02043'	0.16	0.27	0.00	0.93
'T02044'	-0.01	0.00	0.00	0.11
'T02046'	0.35	-0.04	0.04	0.02
'Z01164'	0.21	-0.01	0.01	0.05

Source: Author

For the fault condition M2507, the 14 relevant parameters are 'G01154', 'G02050', 'G02052', 'G02053', 'G02057', 'L01150', 'P01005', 'T01010', 'T02040', 'T02041', 'T02043', 'T02044', 'T02046', 'Z01164'. For this fault, 270,000 samples were used for learning, and 180,000 samples were used for testing. The output range is 0-100% since the main engine does not slow down (as opposed to the first failure case analyzed above).

Parameter selection was done by determining the PCA 95% variance threshold as shown in Table 4. It is observed that the most distinguishing parameter was the 'G2057' parameter, while the least distinguishing parameter was 'T02044'. The main distinguishing parameters are 'G02057', 'G02053', 'G02050', 'P01005' and 'T02046'.

In the case of this failure, 100% learning performance was achieved with the 'Ensemble Bagged Trees' method during the learning phase. This performance can be expected due to the use of expansion tank level and main engine cooling water gas detector parameters which are known to be related to this fault. During the test phase, the highest performance value is obtained for 'Fine Gaussian SVM'.

**Table 5** Learning and test results of M2507 fault

Machine Learning Method	Learning (60%)				Test (40%)			
	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE
Linear Regression	0.98	1.70	3.72	13.86	0.9678	2.57	7.45	55.51
Liner Regr. with Interactions	0.98	1.78	3.61	13.03	0.9696	2.61	7.23	52.31
Linear Regression Robust	0.98	1.67	3.75	14.03	0.9679	2.56	7.51	56.46
Stepwise Linear Regression	0.98	1.78	3.61	13.03	0.9696	2.61	7.23	52.31
Linear SVM	0.98	1.76	3.74	14.01	0.9677	2.60	7.45	55.51
Cubic SVM	0.99	1.67	3.53	12.48	0.9736	2.42	6.76	45.66
Quadratic SVM	0.99	1.67	3.53	12.48	0.9701	2.48	7.21	51.92
Coarse Gaussian SVM	0.99	1.68	3.54	12.52	0.9701	2.46	7.20	51.82
Medium Gaussian SVM	0.99	1.64	3.09	9.58	0.9758	2.36	6.52	42.45
Fine Gaussian SVM	0.99	1.58	2.47	6.12	<b>0.9816</b>	2.23	5.61	31.43
Fine Tree	0.99	1.13	2.18	4.76	<b>0.9809</b>	1.99	5.72	32.76
Medium Tree	0.99	1.16	2.26	5.11	0.9806	2.01	5.76	33.23
Coarse Tree	0.99	1.21	2.42	5.87	0.9798	2.03	5.88	34.58
Ensemble Boosted Trees	0.98	2.56	3.90	15.20	0.9796	3.09	6.48	42.01
Ensemble Bagged Trees	1.00	1.09	1.96	3.83	<b>0.9796</b>	3.09	6.48	42.01

Source: Author

**Table 6** 95% PCA results for M2506 fault for only the first component

Input parameter	98.80%
'E02005'	0.00
'G02050'	0.32
'N01610'	0.32
'P01602'	0.00
'Q02004'	0.28
'T01603'	0.28
'T01613'	0.31
'T02040'	0.31
'T02041'	0.31
'T02043'	0.29
'T02044'	0.29
'T02045'	0.32
'T02046'	0.29

Source: Author

The third fault is the exhaust valve leakage labelled as M2506. For this fault condition parameters 'E02005', 'G02050', 'N01610', 'P01602', 'Q02004', 'T01603', 'T01613', 'T02040', 'T02041', 'T02043', 'T02044', 'T02045', and 'T02046' were analyzed.

150,000 samples were used for learning and 100,000 samples for testing. The fault value ranges between 0% and 62%. When PCA is applied, only the primary components provide 98.8% representation of the failure. It has been observed that the 'E02005' and 'P01602' parameters have no distinguishing characters. Since these parameters have no effect on the model, they do not need to be monitored for this fault condition.

An  $R^2$  value of 0.9994 was obtained with the 'Fine Tree' and 'Coarse Tree' methods. In testing, it is seen that 0.9993  $R^2$  value is achieved with the 'Fine Tree' and 'Coarse Tree' methods, while the other errors are lower in the 'Coarse Tree' method. These metrics are very close but 'Coarse Tree' may be preferable considering that it uses a smaller tree.

**Table 7** Training and test results of M2506 fault

Machine Learning Method	Learning (60%)				Test (40%)			
	$R^2$	MAE	RMSE	MSE	$R^2$	MAE	RMSE	MSE
Linear Regression	0.9971	1.15	1.40	1.95	0.9972	1.14	1.37	1.88
Linear Reg. with Interactions	0.9971	1.15	1.40	1.95	0.9972	1.14	1.37	1.88
Linear Regression Robust	0.9971	1.15	1.40	1.95	0.9972	1.14	1.37	1.88
Stepwise Linear Regression	0.9971	1.15	1.40	1.95	0.9972	1.14	1.37	1.88
Linear SVM	0.9971	1.38	1.63	2.65	0.9972	1.37	1.61	2.60
Cubic SVM	0.9971	2.03	2.61	6.82	0.9971	2.03	2.61	6.81
Quadratic SVM	0.9983	0.89	1.11	1.23	0.9984	0.88	1.09	1.19
Coarse Gaussian SVM	0.9985	0.91	1.09	1.19	0.9986	0.91	1.08	1.16
Medium Gaussian SVM	0.9987	0.82	0.99	0.99	0.9987	0.81	0.97	0.94
Fine Gaussian SVM	0.9988	0.74	0.91	0.82	0.9989	0.71	0.88	0.77
Fine Tree	<b>0.9994</b>	0.48	0.64	0.41	<b>0.9993</b>	0.52	0.70	0.49
Medium Tree	<b>0.9994</b>	0.49	0.65	0.43	<b>0.9993</b>	0.51	0.68	0.47
Coarse Tree	0.9993	0.50	0.67	0.45	0.9993	0.51	0.67	0.45
Ensemble Boosted Trees	0.9993	1.40	1.68	2.83	0.9993	1.42	1.69	2.87
Ensemble Bagged Trees	0.9993	1.40	1.68	2.83	0.9993	1.42	1.69	2.87

Source: Author

**Table 8** Run-time performance of algorithms for 180,000 samples in testing

	Machine Learning Method	Total Runtime (ms)	Per Test ( $\mu$ s)
1	Tree Fine Tree	320	1,83
2	Tree Medium Tree	92	0,51
3	Tree Coarse Tree	72	0,40
4	Ensemble Boosted Trees	786	4,37
5	Ensemble Bagged Trees	359	1,99
6	Linear Regression	182	1,01
7	Linear Regression with Interactions	109	0,61
8	Linear Regression Robust	63	0,35
9	Stepwise Linear Regression	59	0,33
10	SVM Linear SVM	3335	18,53
11	SVM Cubic SVM	3299	18,33
12	SVM Quadratic SVM	3110	17,28
13	SVM Coarse Gaussian SVM	8213	45,63
14	SVM Medium Gaussian SVM	111649	620,27
15	SVM Fine Gaussian SVM	6431	35,73

Source: Author

While the accuracy of the trained models is very important, it is critical to look at the run-time performances as well. Table 8 shows how long it took to process 180,000 samples for each learning method. It is possible to increase the speed even more with a higher capacity computer and using parallel processing. With the current configuration, the fastest methods, as expected, are linear regression methods.

#### 4 Conclusion

We developed automated fault detection algorithms and evaluated them on a realistic ship engine room simulator. When all error cases are evaluated, we see that decision trees have high accuracy rates in both learning and testing phases. High accuracy rates are also obtained for other algorithms. The lowest value was seen in the linear regression model in the test phase for the cylinder liner crack. Even this is good enough to be used in a practical system. Linear regression gave the best run-time performances among all the algorithms, followed by decision trees. When the fault detection algorithm needs to run on an embedded system, trained linear regression models can be a good choice since they also provide comparable accuracy.

Using a simulator allows us to obtain useful information about the malfunction status of the machines and systems in the engine room. As a result, a step has been taken to develop a system that can understand the status of machine systems. Considering the dozens of systems and hundreds of malfunctions in the engine room, the need for multiple models emerges. Starting with the most important systems and malfunctions, such a system can be realized in stages. While engineers working on the ship evaluate the data, their knowledge can be captured in the data. Using these types of learning-based systems, expert knowledge can be captured and augmented with simulated data for building better predictive maintenance systems.

**Funding:** This study is supported by Istanbul Technical University, Scientific Research Projects (BAP), Project ID: MDK-2019-41955.

**Author Contributions:** Research, Gazi Kocak; Methodology, Veysel Gokcek, Yakup Genc; Data Collection, Gazi Kocak; Conceptualization, Gazi Kocak, Veysel Gokcek; Writing, Gazi Kocak, Yakup Genc.

## References

- [1] Yıldız, B., Bilbao, J.I., Sproul, A.B. (2017). A Review and Analysis of Regression and Machine Learning Models on Commercial Building Electricity Load Forecasting. *Renewable and Sustainable Energy Reviews*, Vol. 73, pp. 1104-1122.
- [2] Brereton, R. G., Lloyd, G. R. (2010). Support Vector Machines for Classification and Regression. *The Analyst*, 135(2), pp. 230-267.
- [3] Rokach, L., Maimon, O. (2005). *Data Mining and Knowledge Discovery Handbook*, Springer, Boston, 2005.
- [4] Rokach, L. (2016). Decision Forest: Twenty Years of Research. *Information Fusion*, Vol. 27, pp. 111-125.
- [5] Freund, Y., Schapire, R. E. (1996). Experiments with a new Boosting Algorithm. In *ICML Vol. 96*, pp. 148-156.
- [6] Jain, A. K., Mao, J., Mohiuddin, K. M. (1996). Artificial Neural Networks: A Tutorial. *Computer*, 29(3), pp. 31-44.
- [7] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), pp. 436-444.
- [8] Li, Z., Yan, X., Yuan, C., Peng, Z. (2010). Intelligent Fault Diagnosis Method for Marine Diesel Engines using Instantaneous Angular Speed. *Journal of Mechanical Science and Technology* 26 (8), pp. 2413-2423.
- [9] Li, Z., Yan, X., Guo, Z., Zhang, Y., Yuan, C. (2012). Condition Monitoring and Fault Diagnosis for Marine Diesel Engines using Information Fusion Techniques. *Electronics and Electrical Engineering*, 7(123), pp. 109-112.
- [10] Xi, W., Li, Z., Tian, Z., Duan, Z. (2018). A Feature Extraction and Visualization Method for Fault Detection of Marine Diesel Engines. *Measurement*, 116, pp. 429-437.
- [11] Antonić, R., Cibilić, A., Medvešek, I.G., Kulenović, Z. (2012). Fault Detection and Identification of Marine Diesel Engine Using Expert Knowledge and Fuzzy Logic Approach. In *16th International Research/Expert Conference TMT 2012*, Dubai, UAE, September 10-12, 2012.
- [12] Khelil, Y., Graton, G., Djeziri, M., Ouladsine, M., Outbib R. (2012). Fault Detection and Isolation in Marine Diesel Engines: A Generic Methodology. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)* August 29-31, 2012. Mexico City, Mexico.
- [13] Kowalski J., Krawczyk B., Woźniak M. (2017). Fault Diagnosis of Marine 4-Stroke Diesel Engines Using a One-vs-one Extreme Learning Ensemble. *Engineering Applications of Artificial Intelligence*, Vol. 57, pp. 134-141.
- [14] Nixon, S., Weichel, R., Reichard, K., Kozłowski, J. (2018). A Machine Learning Approach to Diesel Engine Health Prognostics using Engine Controller Data. In *Annual Conference of the Prognostics and Health Management Society*, 2018.
- [15] Lazakis, I., Gkerekos, C., Theotokatos, G. (2019). Investigating an SVM-driven, One-class Approach to Estimating Ship Systems Condition. *Ships and Offshore Structures*, 14:5, pp. 432-441.
- [16] Zhong, G., Wang, H., Zhang, K., Jia, B. (2019). Fault Diagnosis of Marine Diesel Engine Based on Deep Belief Network. In *Chinese Automation Congress (CAC)*, 2019.
- [17] Tan, Y., Niu, C., Tian, H., Hou, L., Zhang, J. (2019). A One-class SVM Based Approach for Condition-based Maintenance of a Naval Propulsion Plant with Limited Labeled Data. *Ocean Engineering*, 193, 106592.
- [18] Hou, L., Zou J., Du, C., Zhang, J. (2020). A Fault Diagnosis Model of Marine Diesel Engine Cylinder Based on Modified Genetic Algorithm and Multilayer Perceptron. *Soft Computing*, 24, pp. 7603-7613.
- [19] Qi, Z., Qi, Y., Hu, G. (2020). A Practical Approach to Detect Faults of Marine Diesel Engine. *Journal of Computer and Communications*, 8, pp. 12-21.
- [20] Ellefsen, A.L., Han, P., Cheng, X., Holmeset, F.T., Aesoy, V., Zhang, H. (2020). Online Fault Detection in Autonomous Ferries: Using Fault-type Independent Spectral Anomaly Detection. *IEEE Transactions on Instrumentation and Measurement*, 69(10), pp. 8216-8225.