Joint Autoencoder-Classifier Model for Malfunction Identification and Classification on Marine Diesel Engine Diagnostics Data

Kürşat İnce[1,3], Gazi Koçak[2], and Yakup Genc[3]

[1] *Naval Combat Management Technologies Center, HAVELSAN Inc., Pendik/Istanbul, Turkey*
kince@havelsan.com.tr

[2] *Department of Marine Engineering, Istanbul Technical University, Tuzla/Istanbul, Turkey*
kocakga@itu.edu.tr

[3] *Computer Engineering Department, Gebze Technical University, Gebze/Kocaeli, Turkey*
kince,yakup.genc@gtu.edu.tr

ABSTRACT

There has been an increasing demand on marine transportation and traveling, since the voyage of the ships are more economical and efficient than air or land based alternatives. The propulsion of a ship is provided by a main engine system which includes the shaft, the propellers, and other auxiliary equipment. Marine diesel engine is a complex structure that the faults within these machines can cause malfunction of the whole system, which in turn inhibits the ship's mission. It is crucial to monitor the engine and other auxiliary systems during the operation and infer their condition from their diagnostic data. In this study, we analyze monitoring data of a crude oil tanker for different ship loads and conditions. Our primary analysis include main engine fault detection and classification for which we propose an end-to-end joint autoencoder-classifier model that contains a convolutional autoencoder, and a long-short term memory regressor connected to the the latent space. Genetic algorithms optimized models gave us 93.61% accuracy for fault classification task. Further investigation on feature's contributions to the model, we increased the accuracy upto 96%. One concern about marine transportation is the pollution of the air with green house effect gases. In this study, we have developed NOx and SOx emission estimators for different faults and working conditions. Leveraging ship load, working conditions and engine faults in the models helped us to achieve 50% better estimation performance. Although there are other studies regarding gases emissions in the literature, this is the first study that took engine faults into account. We believe that the joint autoencoder-classifier model will be useful for other time series estimation task on other domains, especially where the operating condition plays a role in the process.

1. INTRODUCTION

The propulsion of a ship is provided by a main engine system which includes the shaft, the propellers, and other auxiliary equipment. Almost all the merchant ships utilize marine diesel engines with cylinders, pistons, valves, nozzles, and a turbo charger system, which provides power for the navigation of the ship. The diesel engine is a very complex system that the faults within these machines can cause malfunction of the whole system, which in turn inhibits the ship's mission. So, it is crucial to monitor the engine and other auxiliary systems during the operation and infer their condition from this diagnostic data. The main objective of fault diagnostics are fault detection, fault identification, and fault analysis, which is an essential part of modern industries to ensure safety and product quality. Fault diagnosis has been been active area of research for the last few decades (Heo & Lee, 2018).

In this study, we analyze monitoring data of a crude oil tanker at the full ahead loaded and full ahead unloaded situations for main engine fault detection and identification. The data is obtained from a realistic full-mission engine room simulator from Kongsberg. More than 60 sensor data have been recorded for three cases: normal working conditions plus two cases for malfunction scenarios on diesel engine, namely injection valve nozzle wear and injection valve nozzle clogged. Each run of the failure scenarios ends with one of the malfunctions, if any. As part of our initial research on this dataset, we define a classification problem in which we detect existence of any defined malfunctions and identify what the malfunction could be. We have built a joint autoencoder-classifier model, which contains a convolutional autoencoder, and a long-short term memory regressor connected to the the latent space. The joint architecture helps us to train end-to-end

multi-target deep neural network from the multivariate time series data as in the dataset, and to integrate operating condition of the process into the model added to the latent parameters. Using the best performing models, we have investigated the sensors which might alleviate the performance of the classification. The amounts of released NOx and SOx emissions were also recorded during data collection process. In this initial study, we also built several gradient boosting based regressors to estimate NOx and SOx emissions for different ship loads and for different fault types. Our study shows that a) end-to-end joint autoencoder-classifier model provides up to 95.94% accuracy for classification problems, b) as it is in emission estimation, operating conditions are valuable resource of information for processes, c) leveraging operating conditions and faults in the model helps us to achieve 50% better estimations.

This analysis paper is organized as follows: Section 2 gives information about simulation dataset, which we named MC90-V as the simulator. Section 3 summarizes fault classification problem and gas emissions of the engine. Section 4 describes the workflow and methods we used for analysis. Section 5 describes our implementation details and results.

2. KONGSBERG MC90-V ENGINE ROOM SIMULATOR DATASET

Kongsberg K-Sim is a well-known ship engine room simulator with high fidelity, among maritime departments of the universities. One configuration of the simulator, ERS MAN B&W 5L90MC VLCC L11-V (MC90-V for short), simulates a very large crude carrier with a MAN B&W slow speed turbo charged diesel engine as propulsion unit modeled with fixed and controllable propeller. The model is based on real engine data that make the dynamic behavior of the simulator close to real engine response. The simulator includes control room operator station and panels and bridge and steering panels. K-Sim provides other applications such as Neptune for classroom training and TLDS for engine room monitoring.

We have used Neptune for defining *exercises* that run the core simulator for several times. We have used TLDS to record simulator variables as engine room monitoring data. An exercise in Neptune is defined by an Initial Condition, which is the initial state of the simulator. For the MC90-V dataset, we have created 18 different scenario initializations by defining the conditions on ship's load and speed, sea water temperature, and sea conditions as given in Table 1.

The simulator provides about 1500 malfunctions to be injected into the exercises. For our research, we restricted ourselves to two of the malfunctions on the 1st cylinder of the engine, namely Cyl 1 injection valve nozzle wear and Cyl 1 injection valve nozzle clogged, which are refered as M2503 and M2508 in this paper. For each initial condition (18 in total) and for each malfunction state (3 in total) we run the exercises for 53 times, 2862 runs in total. For each initial condition, we separated 35 of the runs for training and the remaining 18 runs for testing. Each run contains 1000 to 1400 data samples recorded at 1 Hz until the failure occurs. The monitoring data contains more than 60 variables, which can be grouped as *real sensors* and *simulated sensors*. For this initial study we have used real sensors.

Table 1. Initial Conditions for MC90-V Dataset

Condition	Possible Values
Ship Speed/Load	FAL: Full Ahead Loaded FAU: Full Ahead Unloaded
Sea Water Temperature	20^oC 25^oC 28^oC
Sea Condition (Beauf)	0 4 6

We present in the paper is the first of many analysis we plan on the MC90-V dataset. For this study we restrict ourselves to ship load *Full Ahead Loaded* and *Full Ahead Unloaded*, sea water temperature 20^oC and Sea Condition 0.

3. PROBLEM DEFINITION

Detection of faults in a monitored system is the first step in root cause identification, which is crucial before proceeding to the next stages of the diagnosis process. The main aim of fault detection and diagnosis are to identify key indicators which can be used for health prediction of a system and then to take a proper action against a future failures. This key indicators can be used to predict the fault class before the system losses its operation ability.

MC90-V Dataset provides unique challenges, one of which is *fault identification* and *classification*, i.e. identify the state of the engine, whether it is operating normal, and predicting the fault type if it is not. Fault-free exercise runs, which we labeled as M0000, represent the behavior of the engine during its normal operating regime. In this case, the engine does not present any problem and runs smoothly. Faults regarding the Cyl. 1 are injected by the simulator after a random delay. These faults are labeled as M2503 and M2508. Two of the objectives of this study are to identify and classify the faults in test data and to identify the signals having highest contribution to the prediction.

The main engine of the ships run on diesel fuel that produce several gases while burning. As more and more ships travel in each day, their emissions becomes a global concern. The two main pollutants from the ship's emission are Nitrogen oxides (NOx) and Sulphur oxides (SOx) gases, which effect on the ozone layer in the troposphere area of the earth's atmosphere and cause the green house effect and global warming. One other objective in this study is to estimate NOx and SOx

emissions from the burning process.

4. METHODS AND TECHNIQUES

In this study, our main motivation was to build a fault classifier, which would differentiate between M0000, M2503 and M2508. We have used the workflow as given in Figure 1, which contains data preprocessing, model training, and evaluation phases. Data preprocessing step prepares the data for training and evaluation steps.

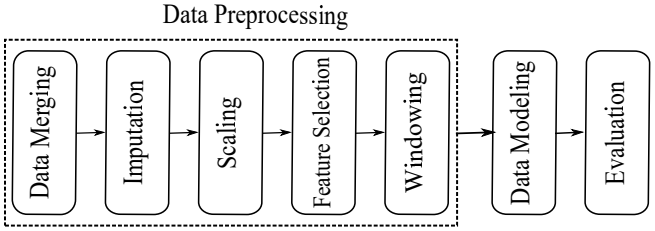


Figure 1. General data processing workflow in this study

Data Preprocessing: The actions we took in this step are data merging, since the simulation data is distributed among 23000 or so files, missing value imputation, data normalization and scaling, feature selection, and windowing. A few of the featured had missing values, which were imputed with *backfill*.

Data Modeling and Optimization: In this study, we propose a joint autoencoder-classifier (JAEC) architecture, which integrates a CNN based autoencoder and an LSTM-based classifier end-to-end for fault classification. The model incorporates CNN based autoencoder. The general architecture is given in Figure 2. Input to autoencoder is $X(t)$, the sensor values at time t. The decoder produces $\widehat{X}(t)$, the signals at time t. Encoded sensor values (latent space) for time t is concatenated to operational conditions $OC(t)$ at time t. This data takes the classification path, which contains two LSTM layers and dense layers that generate $\widehat{Class}(t)$. In JAEC-CNN, convolutional layers in the autoencoder are wrapped in time distributed layers, which applies convolution operation to every temporal slice of the input. Number of filters in CNN layers decreases in the encoder to support latent space generation. Batch normalization is performed between convolutional layers.

For NOx and SOx emission estimations, we propose gradient boosting (GB) based models. GB is one of the powerful techniques for performing classification and regression tasks that builds the model in a stage-wise fashion. GB is an ensemble learner: a complex model based on a collection of individual models. These individual models may have poor predictive power and are prone to overfitting, but combining many such weak models in an ensemble will lead to a much better outcome overall.

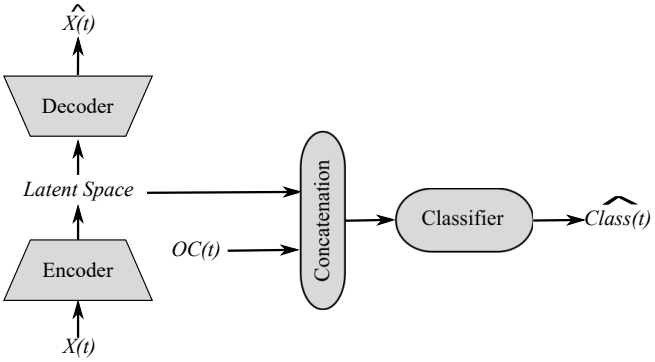


Figure 2. Proposed joint autoencoder classifier architecture

Hyper-parameters for the model architecture are optimized using genetic algorithms (GA), which is based on the biological concept of evolution (Back, Fogel, & Michalewicz, 2000). In genetic algorithms hyper-parameters to optimize are called as *genes*, and a set of gene sequences that construct the unique model is called an *individual*. The genetic algorithms process starts with a set of individuals, ie. population, which are actually an initial set of models. The initial population is *trained* and *evaluated* for their *fitness* to the problem solution. Usually, n best fitted individuals are selected to form the next generation through gene crossover (mating) and gene mutation. The individuals in the new generation goes into training and evaluation stages. The process continues until maximum number of generations is reached or predetermined fitness score is achieved. Finally, best fitted individuals are selected to create the optimal architectural models.

Evaluation Metrics: Commonly used evaluation metrics for classification problems are accuracy (ACC), and F1 score (F1), which are defined by true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) as in Equation 1 and Equation 2, respectively.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$F1 = 2 \times \frac{TP}{TP + \frac{FP+FN}{2}} \tag{2}$$

Evaluation metrics for regression problems are Mean Absolute Error and Root Mean Square Error are defined as in Equation 3 and Equation 4, respectively. In these equations y is the expected result (i.e. ground truth), $\hat{y}$ is the model estimation, and i is the sample index.

$$MAE(y, \hat{y}) = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3}$$

$$RMSE(y, \hat{y}) = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - \hat{y_i})^2} \quad (4)$$

5. Experiments and Results

We have used Scikit-Learn (Pedregosa et al., 2011) and Keras (Chollet et al., 2015) libraries to implement proposed framework for fault classification and emission estimation on MC90-V dataset. This section gives implementation details for each task, and the evaluation results. [1]

5.1. Fault Classification

Data preprocessing step prepared the data for training and evaluation steps. Initially, the dataset contains individual data files for each exercises runs. We have merged individual data files to have two separate subsets, training and test. The dataset contained missing data for each run. Because missing data can create problems for analyzing data, interpolation is used to fill in missing data and avoid pitfalls involving cases that have missing values. After that, data normalization is performed using Z-score normalization, with mean zero and standard deviation one. Initially training data is scaled, and then the scaling parameters are applied to test data. Our exploratory data analysis showed that some of the features has zero variance. We removed these features from the dataset before training. Finally we have used window sizes of 5 up to 100 to create a context for the time series data. The rearranged data have passed to model training phase.

Model training and optimization of JAEC model is performed using DEAP Library (Fortin, De Rainville, Gardner, Parizeau, & Gagné, 2012), a Genetic Algorithms (GA) based optimization framework. Hyper-parameters for GA evolutions are given in Table 2. GA optimization parameters are selected manually after a few experimental runs. The hyper-parameters for the best model is retrieved from these optimizations. We have used different percentages of the training and test data starting from the beginning of each run. This was to divide the runs into regions, which in turn helped us to understand how the fault was developing up in each run.

Table 2. Genetic algorithms search parameters for hyper-parameter optimization

GA Parameter	Value
Initial population size	30
Number of generations	7
Population size per generation	10
Mate probability	0.5
Mutation probability	0.5
Number of selected individuals per generation	5

[1] The source code for this study will be available at `https://github.com/zakkum42/phme22-public` after the publication of this paper.

5.2. Emission Estimation

We have used the same steps as the fault classification task with the exception that we did not perform windowing. Estimation of NOx and SOx emissions were performed with XGBoost library (Chen & Guestrin, 2016). Hyper-parameter optimization was also performed. For this task, we were able to use of the fault label for further investigating the gas emissions.

5.3. Results and Discussion

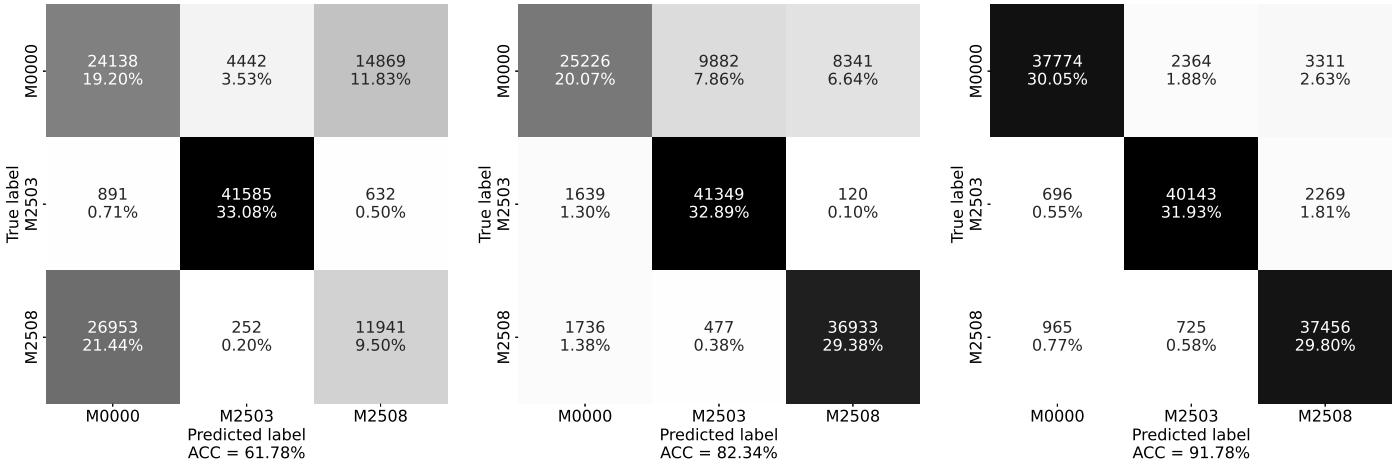
Fault classification results for optimized model are given in Table 3 for different percentages of the training data and for full test data. ACC and F1 metrics are about the same, which is expected as the dataset is well-balanced for each fault. Confusion matrix in Figure 3 shows how ACC is increasing with the addition of new training data. The drop in ACC from 60% to 80% of training data is negligible, and can be attributed to the random initializations of the models. With lesser data M0000 and M2508 were confused the most, but with increasing percentage of the training data the confusion dropped from 21.44% to 2.7%. The other confusions have dropped as well.

Table 3. Classification results for different training percentages

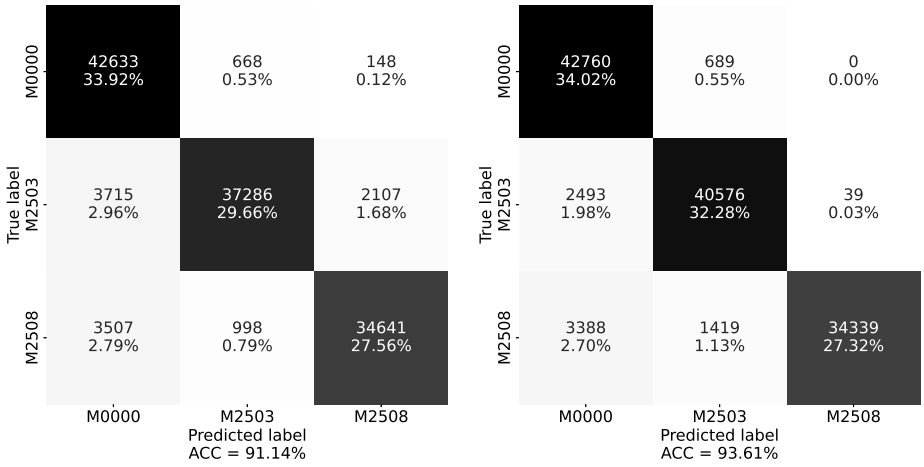
	Test (100%)	
Train (%)	ACC	F1
20	61.78	60.56
40	82.34	81.32
60	91.78	91.78
80	91.14	91.12
100	93.61	93.63

Experimenting with different percentages of the train and test data gave us the scores in Table 4. Highest scores for these experiments are achieved when the same percentage of train and test data was used, i.e. the diagonal of the table. In the left of the diagonal when we added new test samples the scores increased. This was because new samples were coming from known regions as the training data. On the other hand, in the right of the diagonal when we added new test samples the scores decreased. This was because new samples were coming from unknown regions as the training data. One exception to this observation is when the training data was 20% and we increased test samples from 20% to 40% – the first row of the data in Table 4. We suspect that the adjustments to designated initial conditions in the very beginning of the exercise caused fluctuations in the sensor data, whose effect has been dropped with increasing number of samples from a similar region in which the fault has not been fully developed yet.

Using the best classification model, we have analyzed the effect of each feature to the accuracy. Initially we used full train and test data to calculate a base score for the model. Then we iterated over the features: adding noise to a given feature in-



(a) 20% of training data (b) 40% of training data (c) 60% of training data



(d) 80% of training data (e) Full training data

Figure 3. Confusion matrices for varying amount of training data and full test data

Table 4. Classification results for varying amount of train and test data

Test (%)	20		40		60		80		100	
Train (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
20	79.77	79.83	85.39	85.39	73.22	72.88	65.35	64.78	61.78	60.56
40	78.87	79.13	90.55	90.65	90.42	90.45	85.67	85.25	82.34	81.32
60	79.15	79.12	90.68	90.68	94.00	94.00	93.70	93.69	91.78	91.78
80	73.98	74.01	88.37	88.58	92.51	92.63	94.48	94.55	91.14	91.12
100	73.17	73.23	88.01	88.25	92.28	92.41	94.31	94.39	93.61	93.63

validated the features one by one, and a new score can be calculated. The difference between the base score and the new score gave feature's contribution to the model. Sorted list of contributions gave us the feature rank. The contribution of the features are given in Figure 4. As expected, the top features were directly related to the combustion in the cylinder Cyl 1, such as various temperatures and emissions. Removing features with negative contribution increased ACC from 93.61% to 95.94%, which was 2.5% increase in the overall classification performance.

Emission estimations for NOx and SOx gases are given in Table 5 and Table 6, respectively. When the ship engine did not have any fault, i.e. M0000, the emissions were the lowest as we expected. Smaller FAU and FAL scores suggested that when we knew the ship load, we could have better estimation

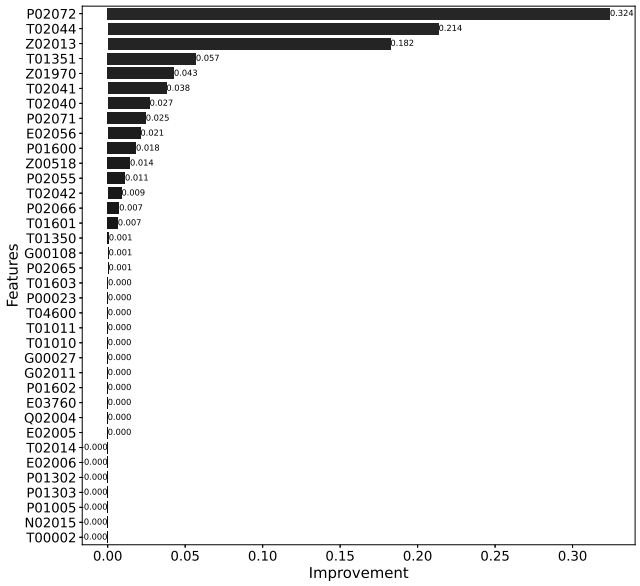


Figure 4. Contribution of each feature to the classification accuracy

of the emissions. When we have used different estimators for ship loads, MAE score decreased upto 50% for full dataset. On the other hand, MAE and RMSE scores for M2508 gas emissions were higher than M2503, which might be an indication of a complications in the burning process that results higher emission variance during M2508 fault development. For comparison, NOx and SOx emissions means and standard deviations of the dataset are given in Table 7 and Table 8, respectively. We observe that our MAE and RMSE scores are comparable to their respective standard deviations.

All the model development and evaluation is done using the simulated dataset. Unfortunately, we could not have any real-world data that we would validate or invalidate our models up-to now.

6. Conclusion

In this paper, we presented our initial analysis on MC90-V dataset, which was constructed via Kongsberg K-Sim, a well-known ship engine room simulator. For this study we aimed for fault identification and classification, namely Cyl 1 injection valve nozzle wear and Cyl 1 injection valve nozzle clogged. For classification, we used a joint autoencoder classifier model trained end-to-end. The optimized models have reached an accuracy score of 93.61%. With further investigation on feature contributions to the score, and removing negative effects, we have reached upto 95.94% model accuracy. We also investigated for NOx and SOx emission estimations for different faults and ship loads. Our findings suggested that if we knew the ship load, working conditions and engine health state we could have upto 50% better estimations with full dataset. We also validated our assumption that M0000 produces less emissions. We believe that the joint autoencoder-classifier model will be useful for other time series estimation task on other domains, especially where the operating condition plays a role in the process. The MC90-V dataset has much more initial conditions than we have used in this study. We will be inspecting other scenarios in the future studies. We also plan developing a remaining useful life estimator, which will predict when the failure will occur in the diesel engine. Also we will analyze other types of gas emissions from the engine.

Acknowledgment

This study is funded by the ITU BAP Unit for the MGA-2018-41459 numbered ITU BAP General Research Project (GAP) with the title ”Condition Monitoring of Marine Machinery by Machine Learning Methods.” The authors also thank to HAVELSAN Naval Combat Management Technologies Center for their support in conducting this research.

References

Back, T., Fogel, D. B., & Michalewicz, Z. (2000). *Evolutionary computation 1: Basic algorithms and operators* (1st ed.). IOP Publishing Ltd.

Chen, T., & Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*. doi: https://dx.doi.org/10.1145/2939672.2939785

Chollet, F., et al. (2015). *Keras.* `https://keras.io`.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012, jul). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, *13*, 2171–2175.

Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, *51*(18), 470-475. (10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018) doi: https://doi.org/10.1016/j.ifacol.2018.09.380

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Biographies

Kürşat İnce received his BSc and MSc degrees from Bilkent University Computer Engineering Department in 1996 and in 1999, respectively. As a software developer, he joined HAVELSAN Inc. in 1996. Currently, he is employed as R&D coordinator in HAVELSAN’s Istanbul R&D Center.

Table 5. NOx estimations for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MAE	RMSE	MAE	RMSE	MAE	RMSE
FULL Data	0.0213	0.0478	0.0094	0.0337	0.0092	0.0276
M0000	0.0021	0.0030	0.0017	0.0023	0.0018	0.0028
M2503	0.0079	0.0115	0.0066	0.0093	0.0056	0.0076
M2508	0.0164	0.0554	0.0151	0.0575	0.0208	0.0562

Table 6. SOx estimations for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MAE	RMSE	MAE	RMSE	MAE	RMSE
FULL Data	0.0053	0.0161	0.0031	0.0084	0.0027	0.0111
M0000	0.0005	0.0009	0.0004	0.0006	0.0005	0.0007
M2503	0.0068	0.0098	0.0053	0.0075	0.0028	0.0038
M2508	0.0039	0.0188	0.0030	0.0114	0.0041	0.0201

Table 7. Mean and standard deviations of NOx emissions for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MEAN	STD	MEAN	STD	MEAN	STD
FULL Data	14.1854	1.0699	15.0664	0.5027	13.2622	0.6433
M0000	13.9247	1.0893	15.0275	0.0207	12.8496	0.0390
M2503	14.6247	1.0358	15.4461	0.5228	13.7989	0.7186
M2508	13.9818	0.9141	14.7040	0.4593	13.1176	0.4605

Table 8. Mean and standard deviations of SOx emissions for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MEAN	STD	MEAN	STD	MEAN	STD
FULL Data	13.0589	0.4050	13.1646	0.4764	12.9478	0.2718
M0000	12.8619	0.0584	12.9208	0.0041	12.8043	0.0039
M2503	13.4252	0.5029	13.6387	0.5666	13.2101	0.3037
M2508	12.8650	0.1027	12.9179	0.0637	12.8013	0.1046

Mr. İnce started PhD in Gebze Technical University Computer Engineering Department in 2015. Still working on his doctoral thesis, his research interests include machine learning, especially applications of deep learning methods in Industry 4.0. He is also one of the coordinators in Data Istanbul Meetup Group, a community for democratizing machine learning, since 2016.

Gazi Koçak was born in Ankara, Turkey. Graduated from İTÜ Marine Engineering Department by 2003. After working on merchant ships about 2 years he started to work at ITU Maritime Faculty as research assistant.He got scholarship for masters and doctoral studies at Kobe University, Japan. He graduated from doctoral course by 2013. He is still working at ITUMF as an Assistant Professor.

Yakup Genc received his PhD in Computer Science from the University of Illinois at Urbana-Champaign. Right after graduation, Dr. Genc joined Siemens Corporate Research (SCR) in September 1999. As research scientist, project manager, program manager and group manager, he developed technology and research strategy in the areas of computer vision, augmented reality and machine learning. His tenure at SCR produced numerous publications and patents. Since September 2012, as a member of the faculty of the Computer Engineering department at the Gebze Technical University, he continuous to conduct research in fields of computer vision, augmented reality, autonomous vehicles, machine learning and deep learning while maintaining close ties with the industry for practical applications of his research.

APPENDIX

The sensors that we used in the study is given in Table 9.

Table 9. Sensor list

Feature	Description	Unit/ Range	Feature	Description	Unit/ Range
E02005	ME shaft power (to propeller)	MW	P02072	ME cyl 1 injection max press (pinjm)	bar
E02006	ME PTO power (to shaftgenerator)	kW	Q02004	ME shaft torque	kNm
E02056	ME cyl 1 indicated power (IKW)	kW	T00002	FO temp inlet ME	degC
E03760	Shaft power	MW	T01010	HTFW temp inlet ME	degC
G00027	FO flow inlet ME (net flow)	ton/h	T01011	HTFW temp outlet ME	degC
G00108	FO meter volume flow - FO supply	m3/h	T01350	ME LO temp inlet ME	degC
G02011	ME fuel oil consumption	ton/h	T01351	Main LO temp outlet ME	degC
N02015	ME Speed	rpm	T01601	ME air receiver temp	degC
P00023	FO pressure at ME	bar	T01603	ME exh receiver temp	degC
P01005	HTFW press inlet ME	bar	T02014	ME mean cylinder exhaust temp	degC
P01302	Main LO press inlet ME	bar	T02040	ME cyl 1 exh outlet temp	degC
P01303	Main LO press inlet ME bearings	bar	T02041	ME cyl 1 exh outlet temp deviation	degC
P01600	ME air receiver press	bar	T02042	ME cyl 1 air inlet temp	degC
P01602	ME exh receiver press	bar	T02044	ME cyl 1 oil outlet temp (piston)	degC
P02055	ME Cyl 1 mean effective pressure (mip)	bar	T04600	TG inlet steam temp (supply line)	degC
P02065	ME cyl 1 combustion press (pmax)	bar	Z00518	ME exh SOx content	g/kWh
P02066	ME cyl 1 compression press (pcompr)	bar	Z01970	ME exh NOx content final	g/kWh
P02071	ME cyl 1 injection open press (pinjo)	bar	Z02013	ME exhaust gas smoke content	%