International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:15, No:7, 2021

Data Analysis Techniques for Predictive Maintenance on Fleet of Heavy-Duty Vehicles

Antonis Sideris, Elias Chlis Kalogeropoulos, Konstantia Moirogiorgou

***Abstract*—**The present study proposes a methodology for the efficient daily management of fleet vehicles and construction machinery. The application covers the area of remote monitoring of heavy-duty vehicles operation parameters, where specific sensor data are stored and examined in order to provide information about the vehicle's health. The vehicle diagnostics allow the user to inspect whether maintenance tasks need to be performed before a fault occurs. A properly designed machine learning model is proposed for the detection of two different types of faults through classification. Cross validation is used and the accuracy of the trained model is checked with the confusion matrix.

***Keywords*—**Fault detection, feature selection, machine learning, predictive maintenance.

I. Introduction

INTERNET of Things (IoT) is about embedding sensors, chips, software etc. into physical objects allowing devices to interconnect and exchange data over the Internet. If we carry over the idea of smart connected devices to the industry area, we can optimize the involved processes through automation, connectivity and analytics. In construction companies, industrial IoT gives the opportunity to scale up their equipment in order to be able to perform remote monitoring and servicing. The common case regarding construction companies is that they need to perform regular maintenance procedures to a big number of machinery, the volume and the diversity of which may vary among different operation locations. Under this consideration, the concept of e-maintenance is a promising tool that may provide remote monitoring and control to the machinery enabling the implementation of efficient decision making schemes to perform. Moreover, the use of IoT technology forwards the maintenance tasks to a predictive approach, leaving behind the preventive one that performs maintenance on the same schedule in regular basis [1].

Predictive maintenance is able to provide solutions for the estimation of machine failure occurrences. The benefits are multiple: reduction of down time due to equipment failures, prolonged equipment lifespan, etc. The application of a predictive maintenance system to the field of fleet management gives the owner company the opportunity to estimate the current condition of its machinery and to be able to foresee a failure long time in advance. In other words, the main benefit is cost saving since the maintenance actions are taken at the time that failure is about to occur. This strategy uses condition monitoring tools to detect abnormalities related to the equipment performance. At the end, the result can be warning signs and alerts activating decisions that have to be taken in time [2].

The proposed methodology uses a set of five engine parameters data (engine RPM, battery voltage, coolant temperature, engine oil temperature and engine oil pressure) representing healthy and faulty operation. Clustering and classification methods are used in order to provide, at the end, a scheme that will be able to identify a machinery fault as it is developing over operation time. The two types of faults that are investigated here are the coolant fluid temperature for the cases that it exceeds the normal limit of 98 °C and a fault at the engine oil filter when its pressure exceeds 3.2 bar and at the same time coolant temperature is more than 70 °C. The accuracy of the trained model is checked with the confusion matrix.

The data ('healthy' and 'faulty') were provided by experts from a global construction company. The data were collected from data loggers (IoT devices) and the data labelling came from experience. This work is part of the IntelligentLogger system, which provides an integrated approach to vehicle and machinery monitoring and maintenance. Sensor data are collected by IoT devices in near real time, validated, transformed, processed and stored in a dedicated Data Warehouse. The output of this methodology is combined with Enterprise Resource Planning (ERP) information and reported to users as alerts for preemptive maintenance.

II. State of the Art

In predictive maintenance, the abnormalities detection steps forward to a recommendation for further maintenance actions (Fig. 1).

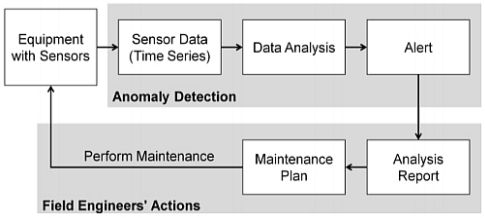

Fig. 1 Example of a timely forecast (possible) failure [3]

Antonis Sideris and Elias Chlis Kalogeropoulos are with Seven Red Lines PC, Vrilissia 15235, Greece (e-mail: asideris@sevenredlines.eu, ekalogeropoulos@sevenredlines.eu).

Konstantia Moirogiorgou is with School of ECE, Technical University of Crete, Chania 73100, Greece (e-mail: dina@display.tuc.gr).

The response is based on the time remaining until the fault occurs.

This field is actually inherently data-driven science where a model is created through system degradation and fault diagnostics. The common practice that is followed involves data classification of the current health condition of the system. The result is examined for prognostics and decision support. Through prognostics an estimation of the time remaining until maintenance is resulting, while the decision support scheme gives a basis for deciding if maintenance is currently necessary.

The field of e-maintenance is not new and various studies have been published proposing solutions with respect to the data that have to examine. For example, specific indications of a monitored parameter through a sensor's operation, as part of a sensor network, may lead to a maintenance task or even to the entire network workflows re-design [4], [5]. These systems can be considered cognitive and involve a set of fields like the one of big data, analytics, machine learning and artificial intelligence. This kind of systems uses either supervised or unsupervised methods for the abnormalities detection. In the supervised learning category, prior knowledge of the type of data (labeled dataset) is required in order each new data point to be categorized as abnormal or normal. On the other hand, in the category of unsupervised learning, the dataset is unlabeled and, so, no prior knowledge or experience is provided in the learning algorithm [6].

In the present study, both clustering and classification techniques are implemented. As a first step, all input data are labelled as 'healthy' or 'faulty' based on expert's opinion and all 'faulty' records are fed to a clustering algorithm in order to distinguish the two different fault types that are under investigation. The clustering labels are then fed to the classification models in order to train them for the detection of future faulty data records (occurrence and type of fault). Two different classification models were trained and tested.

III. Proposed Methodology

The two types of faults that are under consideration are the coolant leakage fault and the engine oil filter fault. The coolant leakage fault is detected when the coolant fluid temperature exceeds the normal limit. Development speed depends on the leakage flow rate and, at the alarming state, a replacement action is required. The threshold coolant temperature values related to this type of fault are the following:

$$\text{Coolant leakage fault} - \text{Fault 1:}$$

$$\text{Coolant leakage flow rate} = \begin{cases} Normal, & if\ coolant\ temperature\ T \leq 85^oC \\ Standard, & if\ coolant\ temperature\ 85^oC < T \leq 98^oC \\ Alarming, & if\ coolant\ temperature\ T > 98^oC \end{cases}$$

The engine oil filter fault occurs when engine oil pressure and coolant temperature meet excess of threshold values at the same time. More specifically, this fault occurs when engine oil pressure exceeds 3.2 bar and at the same time coolant temperature is more than 70 °C. In this case, the oil pressure is high because the oil is not warm enough yet, meaning that the engine is still warming up. Depending on the level of the fault either a cleanup process or filter replacement is required. The details of the engine oil filter fault are the following:

$$\text{Engine oil filter fault} - \text{Fault 2:}$$

$$\text{Engine oil leakage flow rate} = \begin{cases} Normal, & if\ engine\ oil\ pressure\ P \leq 2.4\ Bar \\ Standard, & if\ engine\ oil\ pressure\ 2.4Bar < P \leq 3.2Bar \\ Alarming, & if\ engine\ oil\ pressure\ P > 3.2Bar \end{cases}$$

The main scope of the proposed methodology is to provide a standard methodology that will enable the prediction of each fault and, so will provoke appropriate predictive maintenance tasks. In both fault cases, the frequency of the critical indication occurrence along with the value of the associated critical parameter is examined in order to predict the fault occurrence. The data that are used for present in this study are provided by experts from a global construction company. They used data collected from data loggers that are integrated to the company's fleet of heavy-duty vehicles, while their experience was used towards to the extrapolation and creation of data series in order to be able to implement the proposed methodology and train the proposed model. The experts' knowledge on how to delimit the frequency and the size of the collected data bins that outline the faults prediction was crucial for the current study.

The dataset that we used is a collection of five parameters data values and, more specifically, of the: engine RPM (RPM), battery voltage (V), coolant temperature (Co), engine oil temperature (Co1) and engine oil pressure (Bar), representing healthy and faulty operation. The sampling time is set to 2 mins. The fault occurrences are labelled with the presence of the label 'CO_ALERT' for the Fault 1 instances and the label 'CO1_ALERT' for the Fault 2 instances.

The first thing we calculated was the five data parameters Pearson correlation coefficients (Fig. 2) through which the experts' opinion on the fault events was validated: the coolant and engine oil temperatures (Co and Co1) are highly correlated since they both represent engine's temperature values during engine's operation. The engine's RPM is also highly correlated to the engine's pressure since more engine's rpm results in high pressure.

	RPM	V	Co	Co1	Bar
RPM	1	-0.031	0.23	0.41	0.73
V	-0.031	1	0.13	0.038	0.31
Co	0.23	0.13	1	0.85	-0.057
Co1	0.41	0.038	0.85	1	0.0025
Bar	0.73	0.31	-0.057	0.0025	1

Fig. 2 Pearson correlation coefficients on five sensor data

It is important to notice here that we observed some data records with individually high values at any of the observed parameters and that refer to the engine's start or end of operation. These outliers are not taken into consideration because they do not present during main operation time of the

vehicle.

Using only the data values that represent faulty operation (either Fault 1 case or Fault 2), we implemented clustering in order to distinguish the two types of faults and isolate the outliers. The frequency of fault events occurrence during engine's operation time was the key feature for the data clustering because the records that represent Fault 1 cases are successive in time until replacement occurs while the values that represent Fault 2 cases include parameters that present a successively increase to their values. So, the time difference between successive faulty values is the main key parameter for the clustering process for which the method of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is used. It is an unsupervised learning method that divides the data points into clusters and has the advantage of separating clusters of high density versus clusters of low density within a given dataset while it is great with handling outliers within the dataset.

DBSCAN is a density-based clustering technique which is a more efficient technique when it comes to arbitrary shaped clusters, on the contrary to partition-based clustering techniques, like k-means, that are highly efficient with normal shaped clusters. The parameters that are necessary to be defined are the neighborhood threshold value (Eps) and the point threshold value (MinPts). The clustering is performed with respect to the density of each point, i.e., the number of points in its Eps - neighborhood. The main idea is that the algorithm starts at any point p and retrieves all points in a region with Eps radius around the point [7], [8]. Every neighborhood has to contain at least MinPts other points (Fig. 3).

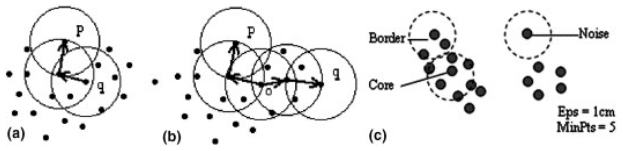

Fig. 3 Basic concepts and terms: (a) p density-reachable point from q point, (b) p and q density-connected points to each other by o point and (c) border object, core object and noise [9]

In our case study, we create a cluster with label '0' for all the detected outliers and the rest of clusters ('1', '2', etc.). represents the different fault states. More specifically, in the case of Fault 2 there are data record labels for '0-NO_MSG', '1 – WARNING MESSAGE' and '2 – ALERT MESSAGE'. in the case of Fault 1 there are data record labels for '0-NO_MSG' and '2 – ALERT MESSAGE'. The Eps parameter is set to 0.4, a value that comes from the dataset observation on the time distance of fault occurrences along with the fact that, in general, small Eps values are preferable. The MinPts parameter is set to 4 points because the initial dataset to which we performed clustering includes only 26 values for Fault 1 and 9 values for Fault 2. Finally, two points are considered neighbors if the distance between the two points is below the threshold Eps and the metric that is used to calculate the distance between the points is the Euclidean distance.

The clustering results regarding the Fault 1 cases showed that they are in accordance to the experts' opinion, i.e., in the cases that we had Fault 1 values for three successive days of operation, recorded data were enough in order to be grouped as Fault 1 occurrence presence. As for the clustering results regarding the Fault 2 cases, that the MinPts parameter equal to 4 was not a good choice for such a small count of faulty records. By changing the MinPts parameter to 3, we observed a clustering output that meets the experts (Fig. 4).

	A	B	C	D	E	F	G	H
1		INPUT				Business Thresholds		Fault Recognition
2	1	26/4/2020 9:44	Data		1	26/4/2020 9:44	Data	Fault
3	2	26/4/2020 9:46	Data		2	26/4/2020 9:46	Data	
4	3	26/4/2020 9:48	Data		3	26/4/2020 9:48	Data	
5	4	26/4/2020 9:50	Data		4	26/4/2020 9:50	Data	Fault
6	5	26/4/2020 9:52	Data		5	26/4/2020 9:52	Data	Fault
7	6	26/4/2020 9:54	Data		6	26/4/2020 9:54	Data	
8	7	26/4/2020 9:56	Data		7	26/4/2020 9:56	Data	
9	8	26/4/2020 9:58	Data		8	26/4/2020 9:58	Data	Fault
10	9	26/4/2020 10:00	Data		9	26/4/2020 10:00	Data	
11	10	26/4/2020 10:02	Data		10	26/4/2020 10:02	Data	
12	11	26/4/2020 10:04	Data		11	26/4/2020 10:04	Data	
13	12	26/4/2020 10:06	Data		12	26/4/2020 10:06	Data	
14	13	26/4/2020 10:08	Data		13	26/4/2020 10:08	Data	
15	14	26/4/2020 10:10	Data		14	26/4/2020 10:10	Data	Fault
16	15	26/4/2020 10:12	Data		15	26/4/2020 10:12	Data	Fault
17	16	26/4/2020 10:14	Data		16	26/4/2020 10:14	Data	Fault
18	17	26/4/2020 10:16	Data		17	26/4/2020 10:16	Data	
19	18	26/4/2020 10:18	Data		18	26/4/2020 10:18	Data	
20	19	26/4/2020 10:20	Data		19	26/4/2020 10:20	Data	Fault
21	20	26/4/2020 10:22	Data		20	26/4/2020 10:22	Data	

Fig. 4 (a) Instances of Fault 1 records: Timestamp and experts 'fault' labelling

	I	J	K	L	M	N
1						
2			Time		Calculate Fault time difference	DBSCAN
3		1	26/4/2020 9:44	Fault	0	cluster 0
4		4	26/4/2020 9:50	Fault	0,004236227	cluster 1
5		5	26/4/2020 9:52	Fault	0,001412095	cluster 2
6		8	26/4/2020 9:58	Fault	0,004236285	cluster 1
7		14	26/4/2020 10:10	Fault	0,008472569	cluster 0
8		15	26/4/2020 10:12	Fault	0,001412095	cluster 2
9		16	26/4/2020 10:14	Fault	0,001412095	cluster 2
10		19	26/4/2020 10:20	Fault	0,004236285	cluster 1

Fig. 4 (b) DBSCAN clustering results on the input data represented in Fig. 4 (a)

	O	P	Q	R
1				
2				
3		Group By Clusters	No_of_Faults	
4		cluster 0	2	avg_data
5		cluster 1	3	avg_data
6		cluster 2	3	avg_data

Fig. 4 (c) Count of points included in every output cluster

In Fig. 4 (a) some of the records with 'Fault 2' labelling are presented and in Fig. 4 (b) the results of the DBSCAN clustering are presented. On 26/04/2020, the combination of values for the coolant temperature greater than 70 ºC with values of the engine oil pressure greater than 3.2 bar should produce a series of alarms for warning or repair alert. As it is shown in Figs. 4 (b) and (c), the cluster 1 and 2 appear successively in time which fits to the expected output, with

some messaging for warning to proceed messages for repair alert. This can be explained due to the lack of a larger dataset with fault events. In any case the fault states are present and a classification scheme where supervised learning is performed may be promising for better results.

The next process after labelling the data is classification. These labels are used as a target variable. The classifiers that were tested are Random Forest and SVM. The Fault 1 prediction case depicts as a binary classification problem, while the Fault 2 case moreover as a multiclassification problem.

Support vector machines (SVMs) separate a set of training vectors for two different classes by mapping the input vectors onto a new higher dimensional feature space [10]. For example, in the case of a linear kernel based SVM, the nonlinear input space is mapped into a new linearly separable space, where all vectors lying on one side of the hyperplane are labelled as -1, and all vectors lying on another side are labelled as +1 (Fig. 5 (a)). Random Forests (RFs) are based on decision trees, where random binary trees implement a subset of the observations over bootstrapping approach [11] (Fig. 5 (b)).

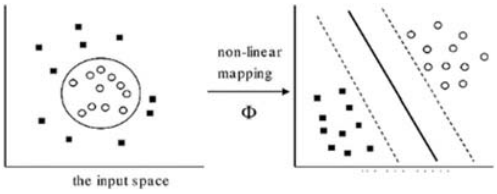

Fig. 5 (a) SVM model generation [12]

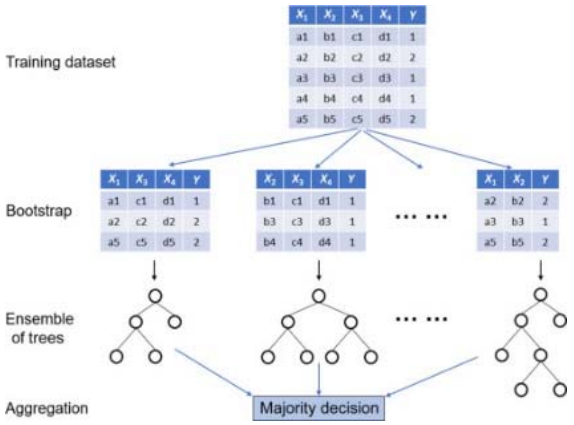

Fig. 5 (b) Implementation of RF classifier [13]

RFs can handle categorical features very well and are non-parametric, so can also handle outliers or non-linearly separable data well. SVMs are also non-parametric models and are capable of doing both classification and regression.

In our study, we performed both models to our labelled fault data because we have a binary classification task (Fault 1) along with a separate multi-classification one (Fault 2) and we needed to compare same classifiers to both cases. DBSCAN clustering provided us with 13 CO-ALERT (alarm) labels (Fault 1) and 8 CO1-ALERT (warning και alarm) labels (Fault 2). The rule we used to divide the dataset to training/test dataset is 70/30 and the cross-validation scheme we implemented is the 10-fold.

The classification results are presented in Tables I-IV.

TABLE I
CO-ALERT (FAULT 1) – RF CONFUSION MATRIX

		Predicted class	
		'0'	'2'
Actual class	'0'	7	0
	'2'	1	5
Accuracy		92%	

TABLE II
CO-ALERT (FAULT 1) – SVM CONFUSION MATRIX

		Predicted class	
		'0'	'2'
Actual class	'0'	6	0
	'2'	0	7
Accuracy		100%	

TABLE III
CO1-ALERT (FAULT 2) – RF CONFUSION MATRIX

		Predicted class		
		'0'	'1'	'2'
Actual class	'0'	4	0	0
	'1'	0	3	0
	'2'	0	0	1
Accuracy		100%		

TABLE IV
CO1-ALERT (FAULT 2) – SVM CONFUSION MATRIX

		Predicted class		
		'0'	'1'	'2'
Actual class	'0'	1	0	1
	'1'	0	2	0
	'2'	0	0	4
Accuracy		91%		

Tables I-IV show the confusion matrices of the classification schemes we implemented. The results present efficiency on fault detection along with the right indices of the level of the detected fault. Fig. 6 presents the classification results for all the clustering labels.

	Target Class	RF_PRED Class	SVM_PRED Class
1	0	0	0
2	2	2	2
3	2	2	2
4	0	0	0
5	0	0	0
6	2	2	2
7	2	2	2
8	0	0	0
9	0	0	0
10	0	0	0
11	2	2	2
12	2	2	2
13	0	0	0

Fig. 6 (a) Class prediction on DBSCAN clustering Fault 1 results using RF and SVM

	Target Class	RF_PRED Class	SVM_PRED Class
1	1	1	1
2	0	0	0
3	1	1	1
4	0	2	2
5	1	1	1
6	2	2	2
7	2	2	2
8	1	1	1

Fig. 6 (b) Class prediction on DBSCAN clustering Fault 2 results using RF and SVM

It is a fact that the input dataset includes small count of faulty records and, so, it is difficult to generalize patterns in training data. The results seem to adjust excessively to the training data, but in any case, the dataset is not complex, so the difficulty in separating the data points into their expected classes is not expected to be high.

The present study specifies a roadmap for further exploration when more faulty data will be available. Nevertheless, the study shows a promising methodology for fault detection allowing predictive maintenance tasks to be supported.

IV. Conclusion

The key idea of the current study was to provide an efficient tool for the detection and classification of two different fault events that may occur to a fleet of heavy-duty vehicles; the coolant leakage fault and the engine oil filter fault. The ground-truth faulty data values were labelled by experts. The clustering method that was used is appropriate for separating clusters of high density and the classification models appropriate for both binary and multi-class problems due to the different action required for every fault we examine. The results are promising for ending up with generalization of abnormal patterns in training data when datasets with more faulty records will be available.

As for the future work, it is important to collect machinery sensor data under varying operating conditions. Capturing all these data will help us develop a robust algorithm that can better detect faults and predict the transition from healthy state and failure. The impact of estimating the degradation path of the selected machinery parameters along with the scheduling of an effective maintenance guide is high for the manufacturing companies.

Acknowledgment

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project code: T1EDK- 01359, IntelligentLogger).

References

[1] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, Guann Pyng Li, Predictive maintenance in the Industry 4.0: A systematic literature review, Computers & Industrial Engineering, Volume 150, 2020, 106889, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2020.106889.

[2] Bousdekis A, Mentzas G., Condition-based predictive maintenance in the frame of industry 4.0., InIFIP International Conference on Advances in Production Management Systems, 2017 Sep 3, (pp. 399-406), Springer, Cham.

[3] P. Zhao et al. "Advanced correlation-based anomaly detection method for predictive maintenance." 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, 2017, pp. 78-83. doi: 10.1109/ICPHM.2017.7998309.

[4] A. Goyal et al. "Asset health management using predictive and prescriptive analytics for the electric power grid." IBM Journal of Research and Development, vol. 60, no. 1, pp. 4:1-4:14, Jan.-Feb. 2016. doi: 10.1147/JRD.2015.2475935.

[5] Loo, A. Van De. “A decision making framework to achieve prescriptive maintenance.” FMCG production industry 2018. TEL. 8282, 2019.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. «Anomaly detection: A survey.» ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. DOI:https://doi.org/10.1145/1541880.1541882.

[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 1996, pp. 226–231.

[8] J. Song, Y. Guo and B. Wang, "The Parameter Configuration Method of DBSCAN Clustering Algorithm," 2018 5th International Conference on Systems and Informatics (ICSAI), 2018, pp. 1062-1070, doi: 10.1109/ICSAI.2018.8599429.

[9] Derya Birant, Alp Kut, ST-DBSCAN: An algorithm for clustering spatial–temporal data, Data & Knowledge Engineering, Volume 60, Issue 1, 2007, Pages 208-221, ISSN 0169-023X, https://doi.org/10.1016/j.datak.2006.01.013.

[10] Vapnik V. Statistical Learning Theory. John Wiley; 1998.

[11] Breiman L (2001) Random forests. Mach Learn 45:5–32.

[12] Huang, M.W., Chen, C.W., Lin, W.C., Ke, S.W. and Tsai, C.F., 2017. SVM and SVM ensembles in breast cancer prediction. PloS one, 12(1), p.e0161501.

[13] Siddharth Misra, Hao Li, Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel times, Editor(s): Siddharth Misra, Hao Li, Jiabo He, Machine Learning for Subsurface Characterization, Gulf Professional Publishing, 2020, Pages 243-287, ISBN 9780128177365, https://doi.org/10.1016/B978-0-12-817736-5.00009-0.