NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
DIVISION OF MARINE ENGINEERING

Fault Diagnosis Of Two-Stroke Marine Diesel Engines Through Torsional Vibrations

Diploma Thesis

Stefanos Mokas

Thesis committee:

Supervisor:	Christos I. Papadopoulos, Associate Professor NTUA
Members:	L. Kaiktsis, Professor NTUA
	G. Papalamprou, Assistant Professor NTUA

Athens, October 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΝΑΥΤΙΚΗΣ ΜΗΧΑΝΟΛΟΓΙΑΣ

Μελέτη Διακύμανσης Στρεπτικής Ροπής Για Διάγνωση Βλαβών Σε Ναυτικούς Κινητήρες Diesel

Διπλωματική Εργασία

Στέφανος Μόκας

Εξεταστική επιτροπή:

Επιβλέπων: Χ. Ι. Παπαδόπουλος, Αναπληρωτής Καθηγητής ΕΜΠ
Μέλη: Λ. Καϊκτσής, Καθηγητής ΕΜΠ
Γ. Παπαλάμπρου, Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2019

Acknowledgments

I would like to dedicate this thesis to my loving mother and brother and express my deepest gratitude to them without whose love, support and understanding I could never have completed this final milestone of my studies at the School of Naval Architecture and Marine Engineering of the National Technical University of Athens (NTUA).

The research included in this thesis could not have been performed without the assistance, patience, and support of many individuals. I would like to extend my gratitude first and foremost to my thesis advisor, associate Professor Christos I. Papadopoulos for mentoring me over the course and I sincerely thank him for his confidence in me.. His technical insights, innovative ideas and support not only made this thesis possible but also shaped my mind as an engineer.

Finally, I would like to express my gratitude to all the professors with whom I shared these last years, the people who supported me wholeheartedly during my student life and my fellow students for the experiences and the skills we shared.

Abstract

The aim of this thesis is to provide a tool for the predictive maintenance of two-stroke marine diesel engines through monitoring of torsional vibrations and performance data.

Costly failures can be avoided and downtime can be reduced with predictive maintenance, which is crucial for the vessel operation, sustainability and profitability. In order to achieve this goal, a reliable modelling of the shafting system is very important. Such a model can be used for generating a dataset of the torsional vibrations during normal or faulty operation of a marine engine. This dataset can later be used for machine learning and training of a classifier model that would be able to discern between two classes of data: intact and faulty condition. Neural Networks (NN), Support Vector Machines (SVM) and Decision Trees are the most common algorithms used for classification based on machine learning. Within this thesis the Modelica language is used as the tool for developing the shafting system model and generating the torsional vibrations dataset. Emphasis is given on the presentation of Modelica language as a powerful tool for simulation and its use for torsional vibration analysis.

Furthermore, within this study a literature review on the subject of predictive maintenance is conducted demonstrating that even though in other industries significant steps have been made on developing a diagnostics system, the maritime industry falls behind on the preventive maintenance of the marine diesel engines and thus there is still ample room for research and developments.

A case of a container ship vessel driven by a two-stroke low-speed Diesel engine is studied. In particular, a 10,000 TEU Container Vessel's propulsion system was modelled, based on the existing torsional vibration analysis which was available within the frame of this study and simulations were conducted. The simulation requires that the model represents accurately the dynamic behaviour of the system for correct transient torsional vibration calculations. The shafting system was modelled using the Modelica language. In addition to that, the torsional vibrations theory was utilized and the steps for developing the case study model are analysed.The natural frequencies and modes of the shafting system being studied are determined, and the forced torsional vibration response is then calculated. The forced torsional vibration stress curves are obtained from the calculated vibrations. The same work is carried out for the case of a cylinder misfire and the results are compared with the available ones for verification.

Finally, data generated from simulations is used for training machine learning algorithms in order to classify between intact and faulty operation. The trained classifier is able to distinguish between the intact condition and the one of cylinder misfiring, based on the dataset features that were extracted from the torsional vibration signals of the

developed model. Moreover, the classifier is able to locate the fault location, indicating the most probable cylinder and percentage of misfiring. The developed method shows promising results for further research on a predictive maintenance tool that can be used for marine diesel engines through monitoring of torsional vibrations. Furthermore, the combination of additional performance data could provide more accurate and precise predictions.

Σύνοψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μίας μεθόδου πρόβλεψης βλαβών στους μεγάλους ναυτικούς κινητήρες Diesel, μέσα από τη μελέτη των ταλαντώσεων στρέψεως και των δεδομένων απόδοσης του κινητήρα.

Δαπανηρές αστοχίες μπορούν να αποφευχθούν, ενώ επιπλέον ο χρόνος εκτός λειτουργίας μπορεί να μειωθεί μέσα από την πρόβλεψη βλαβών, γεγονός το οποίο είναι κρίσιμης σημασίας για τη τη βιώσιμη και αποδοτική λειτουργία του πλοίου. Προκειμένου να επιτευχθεί αυτός ο στόχος, είναι πολύ σημαντικό να γίνει μια αξιόπιστη μοντελοποίηση του αξονικού συστήματος. Ένα τέτοιο μοντέλο μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός συνόλου δεδομένων, μέσα από την προσομοίωση του κατά τη διάρκεια κανονικής ή ελαττωματικής λειτουργίας. Αυτό το σύνολο δεδομένων, μπορεί στη συνέχεια να χρησιμοποιηθεί για τη χρήση ενός αλγόριθμου μηχανικής μάθησης που θα είναι σε θέση να διακρίνει μεταξύ δύο κατηγοριών δεδομένων: άθικτη και ελαττωματική κατάσταση. Τα νευρωνικά δίκτυα (Artificial Neural Networks -ANNs) , οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines-SVMs) και τα δέντρα απόφασης είναι οι πιο συνηθισμένοι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούνται για ταξινόμηση-κατηγοριοποίηση. Στην παρούσα διπλωματική, η γλώσσα προγραμματισμού Modelica χρησιμοποιείται για την ανάπτυξη του μοντέλου του αξονικού συστήματος και την προσομοίωση των ταλαντώσεων στρέψεως προκειμένου να γίνει η δημιουργία ενός συνόλου δεδομένων. Επιπροσθέτως, δίνεται ιδιαίτερη έμφαση στην παρουσίαση της γλώσσας Modelica ως ένα ισχυρό εργαλείο προσομοίωσης και η δυνατότητα της στη μελέτη των στρεπτικών ταλαντώσεων.

Επιπλέον, στο πλαίσιο της διπλωματικής εργασίας διεξάγεται μια βιβλιογραφική ανασκόπηση στο θέμα της προληπτικής συντήρησης, υποδεικνύοντας ότι παρόλο που σε άλλες βιομηχανίες έχουν γίνει σημαντικά βήματα για την ανάπτυξη ενός συστήματος διάγνωσης, ο τομέας της ναυτιλίας υστερεί στην ύπαρξη διαδικασιών προληπτικής συντήρησης των ναυτικών κινητήρων Diesel, δίνοντας έτσι μεγάλο περιθώριο για έρευνα και ανάπτυξη.

Ακόμα, εξετάστηκε το παράδειγμα ενός πλοίου μεταφοράς εμπορευματοκιβωτίων, το οποίο διέθεται αργόστροφο κινητήρα Diesel. Συγκεκριμένα, μοντελοποιήθηκε το σύστημα πρόωσης ενός πλοίου μεταφοράς εμπορευματοκιβωτίων χωρητικότητας 10.000 TEU, λαμβάνοντας όλα τα απαραίτητα στοιχεία για την ανάπτυξη του μοντέλου από προηγούμενη μελέτη των ταλαντώσεων στρέψεως που ήταν διαθέσιμη για αυτό το πλοίο.

Η προσομοίωση ενός συστήματος απαιτεί το μοντέλο να αντιπροσωπεύει με ακρίβεια τη δυναμική συμπεριφορά του συστήματος, προκειμένου να γίνεται ο σωστός υπολογισμός των ταλαντώσεων στρέψεως σε συνθήκες μεταβατικής κατάστασης. Το αξονικό σύστημα μοντελοποιήθηκε χρησιμοποιώντας τη γλώσσα προγραμματισμού Modelica. Επιπλέον, παρουσιάζεται η χρησιμοποιούμενη θεωρία των στρεπτικών ταλαντώσεων και η μεθοδολογία

ανάπτυξης του μοντέλου για το παράδειγμα ενός πλοίου μεταφοράς εμπορευματοκιβωτίων. Αρχικά, προσδιορίζονται οι φυσικές ιδιοσυχνότητες και ιδιομορφές του αξονικού συστήματος και στη συνέχεια εξετάζονται οι αποκρίσεις των στρεπτικών ταλαντώσεων. Μέσα από τις προσομοιώσεις υπολογίζονται οι γωνίες ταλάντωσης και η διακύμανση της στρεπτικής ροπής, καθώς και η τάση λόγω στρέψης. Ίδια διαδικασία ακολουθείται και για την περίπτωση προσομοίωσης της ύπαρξης σφάλματος ατελούς καύσης σε έναν κυλίνδρο και τα τελικά αποτελέσματα συγκρίνονται για επαλήθευση με αυτά της αρχικής διαθέσιμης μελέτης.

Τέλος, τα δεδομένα που παρήγχθηκαν από τις προσομοιώσεις χρησιμοποιήθηκαν για την εκπαίδευση αλγορίθμων μηχανικής μάθησης προκειμένου να γίνει κατηγοριοποίηση μεταξύ άθικτης και ελαττωματικής λειτουργίας. Ο αλγόριθμος που εκπαιδεύτηκε ήταν σε θέση να διακρίνει μεταξύ της άθικτης κατάστασης και εκείνης με έναν κύλινδρο να εμφανίζει σφάλμα, χρησιμοποιώντας χαρακτηριστικά γνωρίσματα που είχαν εξαχθεί από τα σήματα που προσομειώθηκαν μέσω του μοντέλου ταλαντώσεων στρέψεως που αναπτύχθηκε. Επιπλέον, ο αλγόριθμος ήταν ικανός να εντοπίσει την τοποθεσία του σφάλματος, υποδεικνύοντας τον πιθανότερο κύλινδρο και το πιθανό ποσοστό ατελούς καύσης. Η μέθοδος που αναπτύχθηκε προσφέρει πολλά υποσχόμενα αποτελέσματα για την περαιτέρω έρευνα σχετικά με ένα εργαλείο προγνωστικής συντήρησης, που μπορεί να χρησιμοποιηθεί στους ναυτικούς κινητήρες Diesel μέσα από τη μέτρηση των στρεπτικών ταλαντώσεων. Επιπλέον, ο συνδυασμός πρόσθετων δεδομένων απόδοσης του αξονικού συστήματος πρόωσης θα μπορούσε να παρέχει πιο λεπτομερείς και ακριβείς προβλέψεις.

Contents

Nomenclature

$FMEA$	Failure Mode and Effect Analysis
FFT	Fast Fourier Transformation
ANN	Artificial Neural Network
FNN	Fuzzy Neural Network
ACA	Ant Colony Algorithm
TVA	Torsional Vibrations Analysis
θ	Angle amplitude of harmonic $[rad,\ deg]$
θ_0	Maximum angle amplitude of harmonic $[rad,\ deg]$
ω:	Rotational speed $[rad/s]$
ϕ:	Phase of harmonic$[rad]$
$\ddot{\theta}$	Rotational speed $[rad/s]$
$\dot{\theta}$	Rotational acceleration $[rad/s^2]$
I	Inertial mass vector
C	Damping vector
K	Stiffness vector
F	Excitation vector
T	Tangential force $[Nm]$
$T_0,\ T_1,\ T_2...$	Amplitudes of torque harmonics $[Nm]$
e_k	Phase relative to T.D.C. of k-order component.
$T.D.C.$	Top Dead Center
MCR	Maximum Continuous Rating
$OMEdit$	Open Modelica Connection Editor
OMC	Open Modelica Compiler
TEU	Twenty-foot equivalent unit
T_m	Constant harmonic component of torque
$T_{d_{rel}}$	Relative damping torque $[Nm]$
D	Damping constant in Nms/rad
ω_{rel}	Relative speed between two inertias in rad/s
κ	Unidimensional relative damping
$T_{d_{abs}}$	Absolute damping torque $[Nm]$
$I_{propeller}$	Propeller moment of inertia in kgm^2
I_{cranck}	Crank moment of inertia in kgm^2
k	Knearest neighbors algorithm parameter
r	Radius Neighbors Classifier learning parameter
$MLPC$	Multi-Layer Perceptron classifier
$\bigcirc$	MLPC output
$\bigcirc^l$	MLPC output vector
$f,\ f_j{}^l$	MLPC activation function
$I_i,\ (\bigcirc^{l-1})_i$	MLPC inputs

$w_i,\ w_{j,i}{}^l$	MLPC neuron weights
$b,\ b_j{}^l$	MLPC bias term
f_T	Torque correction factor
MIP	Mean Indicated pressure
MIP_{MCR}	Mean Indicated Pressure at maximum continuous rating
MEP_{MCR}	Mean Effective Pressure at maximum continuous rating
n	Engine rotational speed
n_{MCR}	Engine rotational speed at maximum continuous rating
f_D	Damping correction factor
k	order
$f_n^{center},\ f_n^{low},\ f_n^{high}$	Center, lower and upper frequencies of an octave band

List of Figures

List of Tables

1 Introduction

Two-stroke diesel engines are widely used for the propulsion of vessels in the maritime industry. Fault detection for the vessel's main engine is important to ensure reliable and profitable operation, by avoiding costly failures and reducing downtime. There have been continuing advances in the application of vibration based condition monitoring techniques to machines in operation. Such techniques have been applied to rotating machinery, typically operating at constant speed and load in various industries. However, the application of condition monitoring on marine two-stroke diesel engines has developed slowly.

On the other hand, torsional vibrations of marine propulsion shafting systems have been studied for more than a century. During the operation of a marine propulsion system driven by a Diesel engine, the occurrence of torsional vibrations is constant. The torsional vibrations are generated by the various (periodic,dynamic) external loads which induce excitation torques. The main engine and the propeller are the two main sources of excitations, with the bellow being possible causes of effecting the expected torsional vibrations:

- Ageing of the frontal damper
- Misfiring of a cylinder
- Inadequate flexible coupling
- Excess diameter in the propulsion line, among others

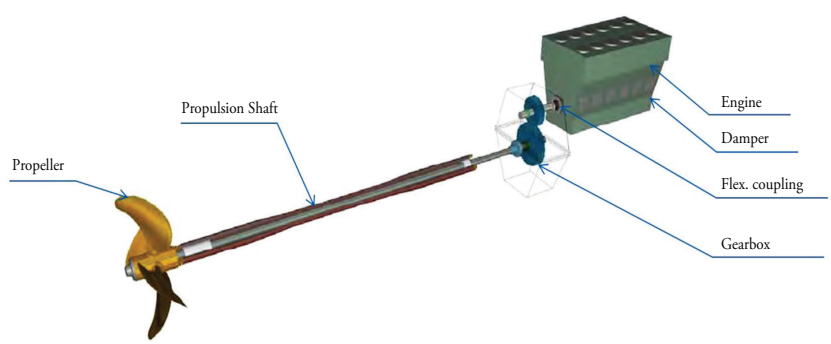

Figure 1: Typical propulsion system

In most cases the reason of studying the torsional vibrations of a marine propulsion shafting system is for the prevention of shaft or other structural failures that could lead to huge economic or even human disasters. Meanwhile, maintenance of machinery components can affect the sustainability and profitability of a vessel and current maintenance state of practice in shipping offers ample room for improvement. Early detection of potential failures can be accomplished by periodic intelligent monitoring and analysis of torsional vibrations. A predictive maintenance technique, that includes real-time equipment monitoring during operation, should be performed in order to drive maintenance decision-making. Predictions are based on vibration signals generated by a healthy shafting system. Vibrations can be measured periodically and any increment in their reference levels indicates the possibility of a failure.

1.1 Literature Review

The need for the continuous monitoring of the machinery room is constantly gaining ground in order to keep perfect operation and maintenance. Marine Diesel engines are well known for their operational robustness and efficient performance, but despite their known reliability there are operational issues that make monitoring of critical engine components and subsystems necessary. Moreover, engines typically constitute a significant fraction (1/10-1/5) [1] of the acquisition cost and a comparable fraction of the life cycle cost, thereby providing the motivation for engine condition monitoring on the basis of reducing life cycle costs.

The work of Liang Xingyu et.al. [16] presents the progress and the recent trends in the torsional vibrations of internal combustion engines. Their work is focusing on presenting the available literature on the modelling of torsional vibrations, the analysis methods, the measures and torsional vibration control as well as the research directions on these topics. The utilization of torsional vibrations of the crankshaft for identifying faults by torsional vibration signals is also depicted in their work. They present the research direction and refer to studies on fault diagnosis by torsional vibration signal of internal combustion engine crankshaft.

.

In their study, Banks et.al(2001) [1] presented an overview of previous research conducted on diesel engine diagnostics and presented diesel engine diagnostics developments. Many research efforts aiming to develop a diesel engine diagnostic system have typically been guided by a thorough knowledge of component failure modes knowns as Failure Mode and Effects Analysis (FMEA). Within their work is presented a comparison of component failure rates obtained from studies of twenty similar marine diesel engines, with failures of the fuel oil valve representing greater than 30% of the recorded failures. Their work indicates that the fuel injection system is the most prevalent source of problems for diesel engines, while engine components subjected to high levels of wear,

such as pistons, liners and bearings rank near the bottom. They also demonstrated that the parts that are most prone to failure or are critical to engine performance and thus given higher maintenance attention also had the higher failure rates.

Condition monitoring systems and fault diagnostics techniques have been developed to reduce maintenance costs and to increase availability for critical mechanical systems. These techniques compare measured operational parameters to normal machine condition baseline levels. The most common parameters that can be measured and monitored on diesel engines are pressure, temperature, flow rates and vibrations. Cylinder pressure analysis has been used extensively to monitor the engine combustion process, by evaluating deviations from pre-established, 'healthy' pressure-time curves of each of the cylinders. In addition, vibration and acoustic analysis of diesel engine has shown a great potential for predictive diagnostic through spectrum analysis of engine vibration data. Even though today's techniques of diesel engine diagnostics have been well developed, there is still no mechanism for predicting the remaining useful life. Current financial situation of the shipping industry, combined with an average global merchant fleet vessel age of almost twenty years [10] makes clear that a high level of operations optimisation is required for the vessel to remain sustainable and profitable. The introduction of novel methods of monitoring the condition of machinery equipment, suggesting suitable maintenance actions, and scheduling those actions in an optimised fashion is considered necessary.

Predictive maintenance can monitor equipments during operation. Predictions are based on a vibration signature generated by a healthy machine. Vibrations are measured periodically and any increment in their reference levels indicates the possibility of a failure. There are several approaches to analyse the vibrations information for machinery diagnosis. Conventional time-domain methods are based on the overall level measurement, which is a simple technique for which reference charts are available to indicate the acceptable levels of vibrations. Processing algorithms have been developed to extract some extra features in the vibrations signature of the machinery. Among these is the Fast Fourier Transforms (FFT) that offers a frequency-domain representation of a signal where the analyst can identify abnormal operation of the machinery through the peaks of the frequency spectra. Since FFT cannot detect transient signals that occur in non-stationary signals, more complex analysis methods have been developed such as the wavelet transform. This approach converts a time-domain signal into a time-frequency representation where frequency components and structured signals can be localized. Also, there are now available fast and efficient computational algorithms to process the information of these new techniques.

Regarding the engine faults, which arouse the need of predictive maintenance, these can be classified into two main categories:

- Combustion faults
- Mechanical faults

Misfire is a very common combustion fault and many works have been put forward to study the vibration-signal-based misfire diagnosis. One of the main approaches is based on the torsional vibration signal of the crankshaft. The other is based on the translational acceleration signals measured on the engine block. Even though some researchers studied the engine fault diagnosis, these diagnostic approaches normally require an expert to interpret the analysis results from measured vibration signals making machine learning a potential solution to the problem of automated diagnostics of faults in marine diesel engines.

In their work, Jian Chen et.al(2013) [3], developed an Artificial Neural Network (ANN) based automated system to diagnose a range of different faults in internal combustion (IC) engines, including combustion faults (misfire) and mechanical faults (piston slap and bearing knock). They developed simulation models, which can simulate combustion faults and mechanical faults in engines, since network training is a critical issue of ANN applications. In addition, in order to evaluate and update the simulation models, they carried out a small number of experiments with combustion faults and mechanical faults and studied the experimental vibration signals. To detect engine misfires they used the torsional vibration of the crankshaft and angular acceleration of the engine block. Moreover, for the diagnosis of mechanical faults they demonstrated that envelope analysis of the vibration signals as the appropriate signal processing approach. Finally, they trained networks on simulated data that can efficiently detect both the combustion faults and mechanical faults in real tests and identify the location and severity of the faults as well.

Williams (1996) [4], used the torsional vibration signal of the crankshaft for the misfire diagnosis. He presented the technical literature for engine misfire detection and faulty cylinder identification using crankshaft angular velocity measurements. Techniques to identify the location of specific faulty cylinders, real time detection of engine misfire using measured speed fluctuations and torsional vibration signal, are categorized into three main approaches: threshold criteria, pattern recognition, and model based deconvolution. In his work a comparison of most commonly used methods of fault detection is presented as well as various assumptions for the location of misfiring cylinders.

In their study, Desbazeille et.al(2010),[5] focused at monitoring large diesel engines by analysing the crankshaft angular speed variations. They modelled both the

crankshaft dynamical behaviour and the excitation torques. The angular speed variations were modelled at the crankshaft free end. The excitation torques depended on the in-cylinder pressure curve which was modelled with a phenomenological model while mechanical and combustion parameters of the model were optimized based on actual data. They proposed an automated diagnosis based on an artificially intelligent system. They used neural networks for pattern recognition of the angular speed waveforms in normal and faulty conditions. The reference patterns required in the training phase were computed with the developed model and were calibrated using a small number of actual measurements. Finally, they managed to successfully diagnose an experimental fuel leakage fault, including detection and localization of the faulty cylinder, as well as the approximation of the fault severity.

For the misfire diagnosis, the vibration based condition monitoring based on the torsional vibration signal of the crankshaft creates the need for modelling of the torque generated by the engine. In the literature, the subject of in-cylinder pressure reconstruction has been widely studied. In the papers of Zhang (2007), [6] and Moro et.al(2002),[7] methodologies for the instantaneous in-cylinder pressure reconstruction have been developed. The methodologies are based on the existence of a linear correlation, characterized by frequency response functions, between in-cylinder pressure and engine speed signals. They also present the signal processing methodology and experimental results, obtained during transient tests. They propose the methodologies for the development of diagnostic tools for the combustion process, as in the presence of a misfiring cylinder the instantaneous engine speed waveform is strongly affected by the absence of combustion, and they confirmed that the reconstructed in-cylinder pressure shows a good agreement with the actual measurements.

Another work aiming to develop a method for machine health monitoring is that of Crupi et.al.(2004) [8]. They analysed the rotating machinery of the Refinery of Milazzo (Italy) and they developed a method using neural networks that allowed to diagnose faults, which were not considered in the training data. The designed net was able to evaluate the vibration signatures and recognize the fault presence. With their analysis they concluded that the net was able to detect the presence of imbalance and bearing wear, even if these typologies of faults were not present in the training data set.

Nahvi and Esfahanian (2005)[9], also designed an artificial neural network system for fault prediction of rotating machinery systems. In their work they used multi-layer feed forward networks, constituted of non-linear neurons and implemented a normalization scheme on the input and output vectors. To train the network they used data consisting of vibration signals of more than 40 rotating common rotating machinery faults. Moreover, they developed a computer software able to detect machinery faults by using the above techniques which was validated for fault detection of different ma-

chinery systems. Lastly, they displayed the effectiveness of the proposed neural network algorithm by several tests and found that the designed network was capable of identifying unknown faults in rotating machinery.

The work of Gkerekos et.al.(2017) [11] presents a methodology for intelligent monitoring of marine machinery using performance data. The proposed methodology includes a suitable pre-processing technique for the acquired dataset and the development of a self- learning model that can estimate whether a given data point corresponds to a reference (nominal) condition considered during training. The proposed self-learning model can be trained without the need of obtaining data corresponding to faulty conditions. For training the models they used measurements that were suitably analysed and processed to retain most of the information (variance) of the original dataset while minimising number of required dimensions. Finally, the proposed methodology was elaborated and showcased through several case studies, simulating faults in different subsystems. A similar advanced work is that of Peng Li et.al.(2010)[13] in which a self-learning algorithm for fault diagnosis in the combustion system of a marine diesel engine was developed. In this paper an advanced method of intelligent fault diagnosis based on fuzzy neural network (FNN) was optimized and trained by ant colony algorithm (ACA).

It must be mentioned that the maintenance and condition monitoring in maritime industry is left behind when compared to defence, aviation, manufacturing, auto mobile, and nuclear power production, where maintenance focus has recently shifted from reactive to preventive/predictive. The work of G Chandroth [12] presented a methodology where vibration data were combined with performance data (cylinder pressures) for the condition monitoring of a main engine to create robust diagnostic systems. Orthogonal wavelet transforms, principal component analysis and time domain information were used to extract features from the data. and several artificial neural net classifiers were developed using these data. In another work of Gkerekos et.al.(2016) [14] a self-learning model for the condition monitoring of ship machinery was developed based on vibration measurements.

Last but not least, the recent work of Hesari et.al. (2018)[15] presents a combustion fault detection technique for a 12-cylinder 588 kW transient diesel engine based on vibration signature analysis using fast Fourier transform, discrete wavelet transform, and artificial neural network. They found that power spectra of vibration signals in the low-frequency range reliably distinguish between normal and faulty conditions, but the fault's location couldn't be identified. For the identification of faults location they used a feature extraction method based on discrete wavelet transform and energy spectrum that were later used as inputs in a neural network for classification according to the location of sensors and faults. Their proposal was that a two-step fault detection method is more reliable than other one-step methods for complex engines.

1.2 Goals of the Present Study - Thesis Outline

Simulation is proving to be a viable way of generating data to train neural networks to diagnose and make prognosis of faults in machines. Advanced software packages are now available, which allow modelling of the kinetics of whole engines, usually with the components such as engine block, crankshaft, connecting rods and pistons treated as rigid. Up until now they have been used primarily in the design and development of new engines, but they could also be used to simulate a wide range of faults, both in combustion, and in the mechanical components.

In the present work, the Modelica language is used as the tool for modelling and simulation of the shafting system. Furthermore, significant emphasis is given on the presentation of the Modelica language and its use for torsional vibration analysis, for a deeper understanding of the modelling problem. One of the main objectives of this study is to develop a model that can be used to simulate various steady state and transient state cases of a propulsion system. The overall aim of the above objectives is to provide reliable data generated from the modelling phase for the later machine learning and training of a classifier model which will be able to discern between two classes of data: healthy and faulty. A critical issue with machine learning applications in machine condition monitoring is the model training, and it is neither likely nor economical to experience a sufficient number of different actual faults, or to generate them in seeded tests, to obtain sufficient experimental results for the training. The proposed methodology can be used for training a classification algorithm using sufficient number samples form the developed model.

Further, investigating the potential of the proposed method a case study was carried out. Based on the available data from a torsional vibration analysis already conducted on a container ship, a Modelica model was developed. After verifying the results produced, several machine learning classifier algorithms (Decision Tree, Random Forest, KNeighbors, Multi-layer Perceptron) were trained using simulation data for normal and faulty operation conditions with one cylinder misfiring at different levels. The classification method is divided into three steps predicting a possible fault condition, the location of the fault as well as the misfire level. The features for the machine learning algorithms are extracted by performing an Octave Band Analysis, which utilizes the power spectra of vibration signals in the low-frequency range.

The present work aims to design an automated system for Main Engine failure diagnostics, through monitoring of torsional vibrations and performance data. The above will provide a simplified but robust framework for the early detection of emerging faults that will lead to minimisation of ship's downtime and operation costs.

2 Torsional Vibrations

Two-stroke diesel engines are the most common propulsion prime mover of ships, transmitting the required power through the main propulsion shafting to the propeller. The subject of torsional vibrations of the propulsion systems have been widely studied over the past years. The main sources of excitations of the aforementioned systems are produced by the propeller and the main engine, with the torsional vibrations being a constant problem. The propeller produces an alternating excitation torque due to its hydrodynamic behaviour with the water flow, while on the other end of the propulsion system, the diesel engine produces excitation torque on each of the crank throws created by the combination of the gas combustion forces inside the cylinders and the tangential forces induced by the reciprocating masses.

2.1 Modelling of Propulsion Systems

The most common methods of modelling a shafting system are that of the lumped parameter system, the continuous system, or by using a finite element analysis model. The lumped parameter model is the most commonly used in the shipbuilding industry as a method for simplifying the model of complex propulsion systems. This method is usually further simplified as a straight branched-chain type (damped or undamped) and calculations can be performed using Holzer method, the system matrix eigenvalue extraction method, the trial and error search method based on the transfer matrix, or with the finite element which allows to calculate the torsional vibration of more complex ship propulsion systems.

The shafting system is usually composed by the following components:

- Engine crankshaft, which is the main component of the internal combustion engine
- Engine damper.
- Flexible coupling
- Gearbox
- Propulsion shaft, which can further consists of the intermediate shafts and the propeller shaft
- Propeller

Bellow are listed the three basic methods that can be used for modelling the shafting system and analysing the torsional vibrations.

- **Lumped Mass Parameter Model**

  Also known as the simple mass-spring model this is the simplest model for the calculation of shaft vibration. With this method a crankshaft is modelled as illustrated in figure 2, with the crankshaft being demonstrated by mass spring components with equivalent stiffness and inertia moments, elastic axis without mass, internal damping and external damping. The rotational inertia of each crank includes the equivalent rotational inertia of connecting rod and piston.

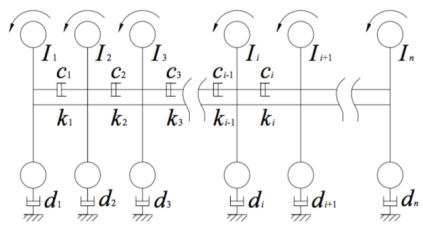

Figure 2: Simple mass-spring model

This modelling method has clear physical content and is more precise for lower frequency of torsional vibration modal. Moreover, this simplified model offers ease of use and simpler calculations, but at the cost of precision. Also this model is established completely for rigid shaft and rotation parts, so it can not simulate the actual shaft.

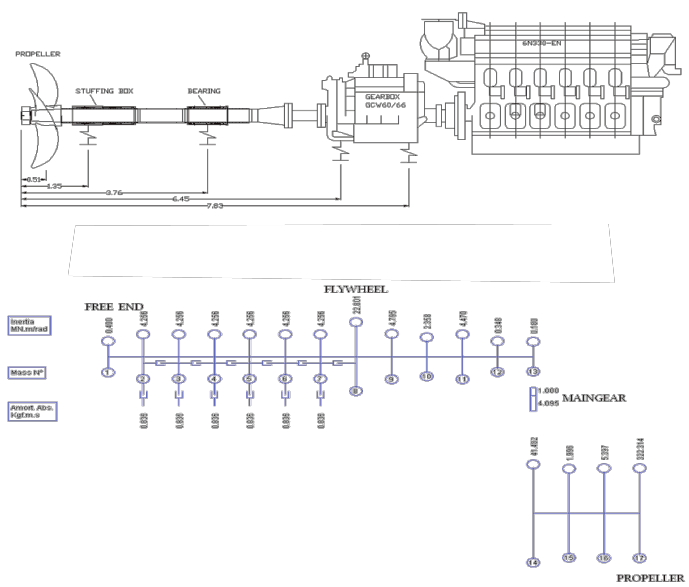

Figure 3: Representation of propulsion system using simple mass-spring model

- **Continuous Mass Model**

  This model is based on continuum theory and the shaft is considered as elastomer. The finite element method is adopted, dissecting the crankshaft entities directly into finite element calculation model of division. Compared to the lumped mass model the mass of the shaft is distributed continuously along the shaft and partial differential equations can be used for more precise calculations of low as well as high frequency and vibration modal of the shaft The model is solved using numerical method, and also can calculate arbitrary section stress conveniently. The cons of this model is the lower speed of calculations and the higher complexity, thus this model is not preferred for system simulation and design and it's mainly used in the calculation of free vibration. Two modelling methods deriving from the continuous mass model, are the framework model and the multi diameter model[16], which can offer higher precision.

- **Multi Segment Concentrated Mass Model**

  Even though, this model is very similar to the lumped mass model, can be separated into many sections according to the structure characteristics upon analysis demand. Compared to the lumped mass model and the continuous model, high order torsional vibration frequencies can be calculated with higher calculation speeds.

New modelling methods continue to develop, such as the continuous beam model [16] and graphical methods which are based on system matrix method. The research direction is mainly focusing on continuous mass model using finite element analysis. But, it's not only the modelling method that will assure that the theoretical calculations are accurate, since the model's parameters also have to be correct and accurate. Considerable amount of precision errors lies at the accuracy of model's rigid parameters, which are calculated through experience formula and approximate calculations.

2.2 Calculation of Torsional Vibrations

2.2.1 Basic Equations

The common approach to the torsional vibration analysis is through analysing steady-state vibrations in a frequency domain. Another approach, is the transient torsional vibration analysis performed in the time domain, which can simulate events such as passing through a barred speed range. However, the later is more demanding and intensive calculations and multiple runs are required.

External and internal vibratory forces are responsible for the vibratory occurring motion around the equilibrium point. These vibrations are time-dependent phenomena, appearing in the form of periodic movements, and could be described by several simple harmonics having different periods and phases according to Fourier harmonic analysis. A harmonic is a sinusoidal movement described by the following equation:

$$\theta = \theta_0 \times sin(\omega \times t + \phi)$$

where, θ: the amplitude at time t
θ_0: the maximum amplitude
ω: the pulsation (in rad/s)
ϕ: the phase (in rad)

The base of the torsional vibration analysis is the equations of motion for the system. The methodology to model a propeller system for torsional analysis is based on the equation:

$$[I]\ddot{\theta} + [C]\dot{\theta} + [K]\theta = [F] \tag{1}$$

Where $[I]$, $[C]$, $[K]$, $[F]$ are the inertial mass, damping, stiffness and excitation vectors respectively. Equation (1) is a non-homogeneous system of second-order linear ordinary differential equations with constant coefficients, with each equation corresponding to the number of nodes (lumped masses) in the torsional vibration model.

The most common solving methods of the torsional vibration models also described in [16] include the following:

- The Holzer method
- The system matrix method
- The transfer matrix method
- The energy method and amplification coefficient method
- The modal analysis method

- The finite element
- The substructure analysis method of the torsional vibration of systems with branch shafts

As a first step the investigation of natural frequencies is very important for the analysis of resonance phenomena. The natural frequencies of the undamped system can be found from solving the differential equation of the free vibrations given by the equilibrium of internal forces (potential energy plus kinetic energy equals zero):

$$[I]\ddot{\theta} + [K]\theta = 0 \tag{2}$$

where $[I]$, $[K]$ are the inertial mass and stiffness matrices.

Then the Campbell's diagram is constructed in order to compare the system natural frequencies with the possible excitation frequencies, referred as the critical speeds of the engine system. In a two stroke-cycle internal combustion engine, the engine crankshaft makes one revolution per cycle. The engine cylinder pressure and the inertial effects are periodic in one engine revolution and by applying a Fourier series expansion a mathematical description of the torque acting at the crankshaft as a summary of harmonic functions(orders) can be generated. Therefore a critical speed for a two stroke-cycle internal combustion engine is defined as:

$$Critical\ Speed = \frac{Natural frequency}{Order No.}$$

If a natural frequency of the system corresponds to any of the forcing frequencies,the engine speed may be critical if there is insufficient damping in the system. It is obvious that there are many engine critical speeds due to the multiple natural frequencies and order numbers, but are only important when they coincide with engine running speed.

2.2.2 Excitation Forces

The torque produced by the engine is the cause of the inertial forces due the reciprocating and rotating masses of the crankshaft and of the gas forces acting on each cylinder by the combustion of fuel in the combustion chamber. The tangential force T is a function of crank angle and is periodic over one crank revolution as it can be seen at figure 4.

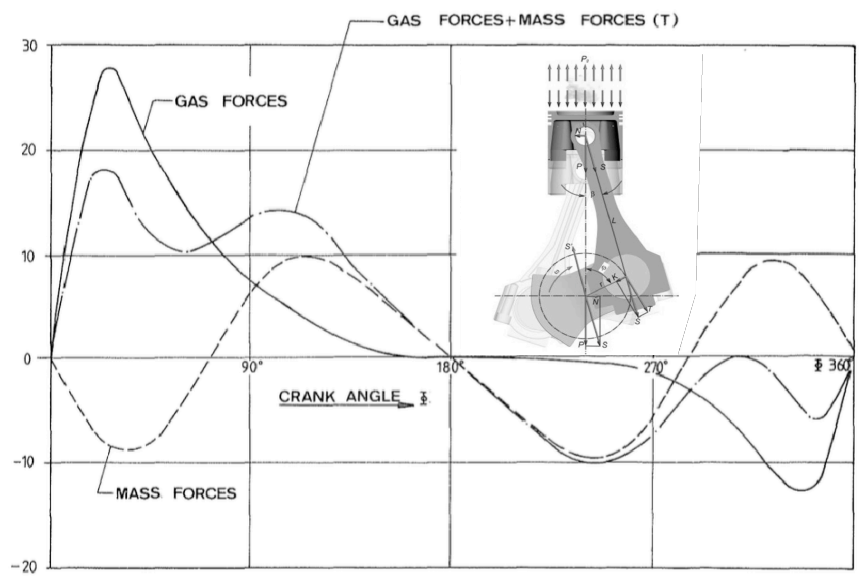

Figure 4: Tangential forces during one revolution

A harmonic analysis is usually performed on the function of the tangential force T to consider the various orders separately. In addition, only a certain number of harmonics can be considered, since higher harmonics of high frequency and small amplitudes do not significantly affect to the torsional vibrations.

$$T = T_0 + T_1 sin(\phi + e_1) + T_2 sin(2\phi + e_2) + ... + T_k sin(k\phi + e_k)$$

where,

e_k is the phase relative to T.D.C. of k-order component.

The propeller is the other main excitation source of torsional vibrations. In order for a ship to voyage at a desired speed, specific power is demanded to be delivered to the propeller. The torque-power demand of the propeller is always equal to the torque-power supplied by the main engine in order for the system to be at an equilibrium state and the ship to voyage at the desired speed under the relevant weather conditions. The

propeller's torque demand curve is known as the propeller law curve and most of the times is a a cubic variation of power demand as a function of speed and is based on the engine's maximum continuous rating (MCR). The cubic curve of the propeller's law is applicable for displacement vessels using a propeller for their propulsion and for cruising at low and medium speeds. The curve can also change for a specific ship based on its loading condition, the fouling og the hull and the weather conditions. In addition there are many types of ships and speed ranges for which the propeller's law is not applicable, thus for better precision ship specific resistance curves should be used based on experimental data.

3 Open Modelica

The OpenModelica language was used within this thesis as a tool for modelling and simulating the torsional vibrations of the shafting system. The definition of simulation is the imitation of a system response over time. The simulation requires that a model correctly represents the behaviour of the simulated system and thus the correct transient torsional vibration calculation needs to be based on adequate simulation of propulsion plant dynamics. Modelica is a relatively new (2000) object-oriented programming language, developed by the countries of north Europe in cooperation with companies such as *SAAB* and *Dassult*. The Modelica language in not only used for academic purposes but also in the industry for model based development. Many automotive companies, such as Audi, BMW, Daimler, Ford, Toyota, VW use Modelica to design energy efficient vehicles and/or improved air conditioning systems and have also been used by power plant providers, such as ABB, EDF, Siemens.

3.1 Introduction To Modelica

Modelica is a modelling language for computer simulation of dynamic systems where behaviour evolves as a function of time. Modelica is an object-oriented equation-based programming language, oriented toward computational applications with high complexity requiring high performance. Specifically, the models of the systems are created by using their equations and the models can be used in different domains while building complex systems. The language uses definitions used also in languages such as *JAVA*, *C++*, *C#* and others. Common used terms of the aforementioned languages such as class, inheritance, methods and functions are also used in Modelica, while new terms such as Connections, Connectors, Packets, models .etc. are introduced. Furthermore, due to the fact Modelica being an object-oriented language the vast availability of developed libraries must be mentioned as one of its main advantages.

The four most important features of Modelica are:

- Modelica is primarily based on equations instead of assignment statements. This allows acasual modelling that gives better reuse of classes since equations do not specify a certain data flow direction. Thus a Modelica class can adapt to more than one data flow context.

- Modelica has multi-domain modelling capability, meaning that model components corresponding to physical objects from several different domains such as, electrical, mechanical, thermodynamic, hydraulic, biological, and control applications can be described and connected.

- Modelica is an object-oriented language with a general class concept that unifies

classes, generics-known as templates in C++ and general sub-typing into a single language construct. This facilitates reuse of components and evolution of models.

- Modelica has a strong software component model, with constructs for creating and connecting components. Thus the language is ideally suited as an architectural description language for complex physical systems, and to some extent for software systems.

3.2 Modelica Environments And OpenModelica

There are several commercial and open source environments for implementing the Modelica language, each one with it's own capabilities developed for specific applications. For the purposes of this thesis the OpenModelica environment was used for developing the models. The OpenModelica environment is an open pen source Modelica based modelling and simulation environment for industrial, research and teaching usage, which has been developed and funded by the Linköping university, the Open Source Modelica Consortium and fast developing companies. The OpenModelica environment supports most of the Modelica language capabilities (algorithms, equations, functions and packages) and supports the ability of connecting with Eclipse and learning of Modelica through DrModelica.

Bellow is a list of the commercial and open source Modelica enviroments, each one with it's own capabilities developed for specific applications.

- Commercial Modelica environments:
  - CATIA Systems
  - Dymola
  - LMS Imagine.Lab AMESim
  - MapleSim
  - SystemModeler
  - SimulationX
- Open source Modelica enviroments:
  - JModelica.org, developed by Lund University and Modelon AB, Sweden
  - OpenModelica, developed by Linköping University, Sweden

In the OpenModelica environment the user has the ability of developing the model either in a graphic editor or by using an interactive command handler.

The OpenModelica environment applications are:

- OMShell – an interactive command handler
- OMNotebook – a literate programming notebook
- MDT – an advance textual editor in Eclipse
- OMEdit – graphic editor

- OMOptim - optimization tool
- ModelicaML - UML Profile
- MetaModelica - extension
- ParModelica – extension

Within this thesis, the OpenModelica Connection Editor (OMEdit) was used for the modelling purposes., which is a user interface of the OpenModelica that provides the ability to build models, edit connections and simulation of the models and plot of the results. OMEdit offers modellers the ability to create models both textually and graphically. When using the graphic environment of OMEdit the user can use already developed models just by drug and drop of the relative icon from the available libraries. The parameters of the available models-components can be changed to suit each application enabling the user to develop a more complex model by connecting the available models-components.

As already mentioned, Modelica being an object-oriented language uses the concept of class, the meaning of which is reduced to a more abstract level, and the concept of object which has a tangible value and is undoubtedly related to the class, with each object having reference to a specific class. Moreover, for reasons related to the maintainability and scalability of the code, Modelica introduces new concepts for specific use, identical to the class concept. These are the concepts of model, connector, record, block and type, which are restricted classes and are used as alternatives to the class under appropriate conditions.

3.3 OMEdit

As already mentioned OMEdit is the OpenModelica Connection Editor, which is a graphical environment for developing of models. OMEdit is developed in C++ using the Qt 4.7 graphical user interface library. OMEdit communicates with OpenModelica Compiler (OMC) and requests the model information and creates models based on the Modelica annotations. After the implementation of the models, OMC performs several steps for the simulation of the object oriented models. Firstly, from Modelica source code, parsing of the codes, type checking, class expansion and generation of connection equations are performed. Later, OMC flattens the object oriented models into a system of differential algebraic equations (DAE) and in order to reduce the size of the equations performs optimizations. Furthermore, it reduces index of the system for numerical solutions and minimal set of equations in state-space form are generated. Lastly, sequential C code is generated with a numerical solver to simulate the models.

When using the OMEdit the user can navigate through the bellow basic views:

- Modeling View

  Can be used by the user for creating models both textually and graphically

- Plotting View

  Is used for plotting the simulated results offering different options for presenting the results appropriately

- Interactive Simulation View

  This view is similar to the plotting view with the difference that the user can change the parameters and variables value during the simulation

In addition, during the model development modeller is using three different views:

- Icon view

  The icon view demonstrates the icon that is used for each model-component in the model space of the diagram view. In this view he user can also create a new icon for representing a new model.

- Diagram view

  When in diagram view the user can drag and drop components-models from the available libraries into the model space and connect the various components in order to create a new model.

- Text view

  The text view enables the user to see the generated code of the developed model for each of the components used in the diagram view. Moreover, when in this the user can edit the generated code of the components used in the diagram view or add lines to the code of the model using Modelica language. Every change made in diagram or text view is simultaneously depicted to all views.

The ability of acasual and multi-domain modelling, show the strong capabilities of OpenModelica when developing a model. Bellow, figure 5 compares a model developed with OpenModelica to a model developed in Simulink, where simulation is made through assignment statements with limited data flow context and the view of the developed model not demonstrating physical model.

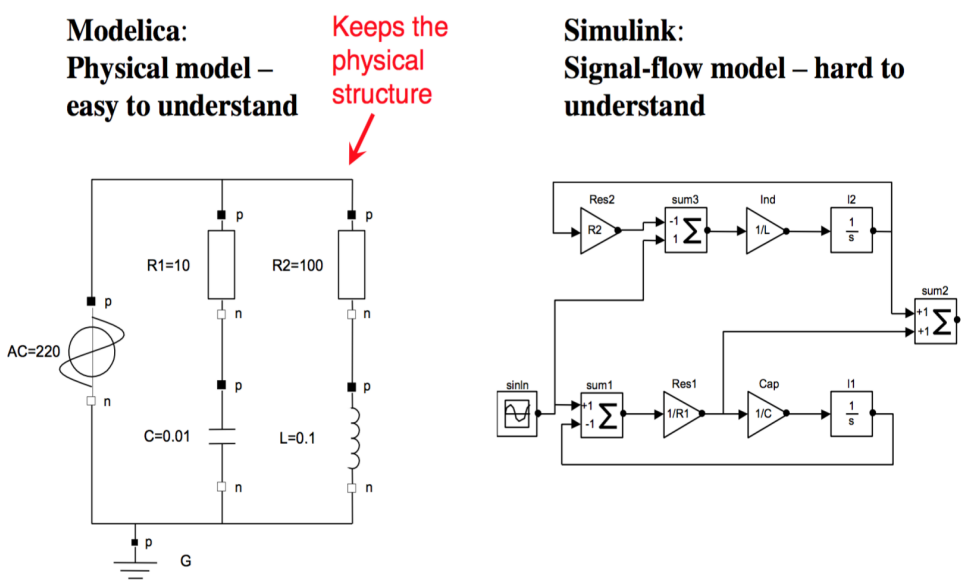

Figure 5: Modelica model vs Simulink model

4 Torsional Vibration Model Development with OpenModelica

In this section, the methodology of developing a model in OpenModelica for a Vessel's propulsion system is presented. The data used for this purpose are obtained from of a torsional vibration analysis of a 10,000 TEU Container Vessel's propulsion system., equipped with a 10 cylinder *MAN B&W 10S90ME-C Mk9.2* engine. Figure 6 demonstrates the developed model which will be described later.

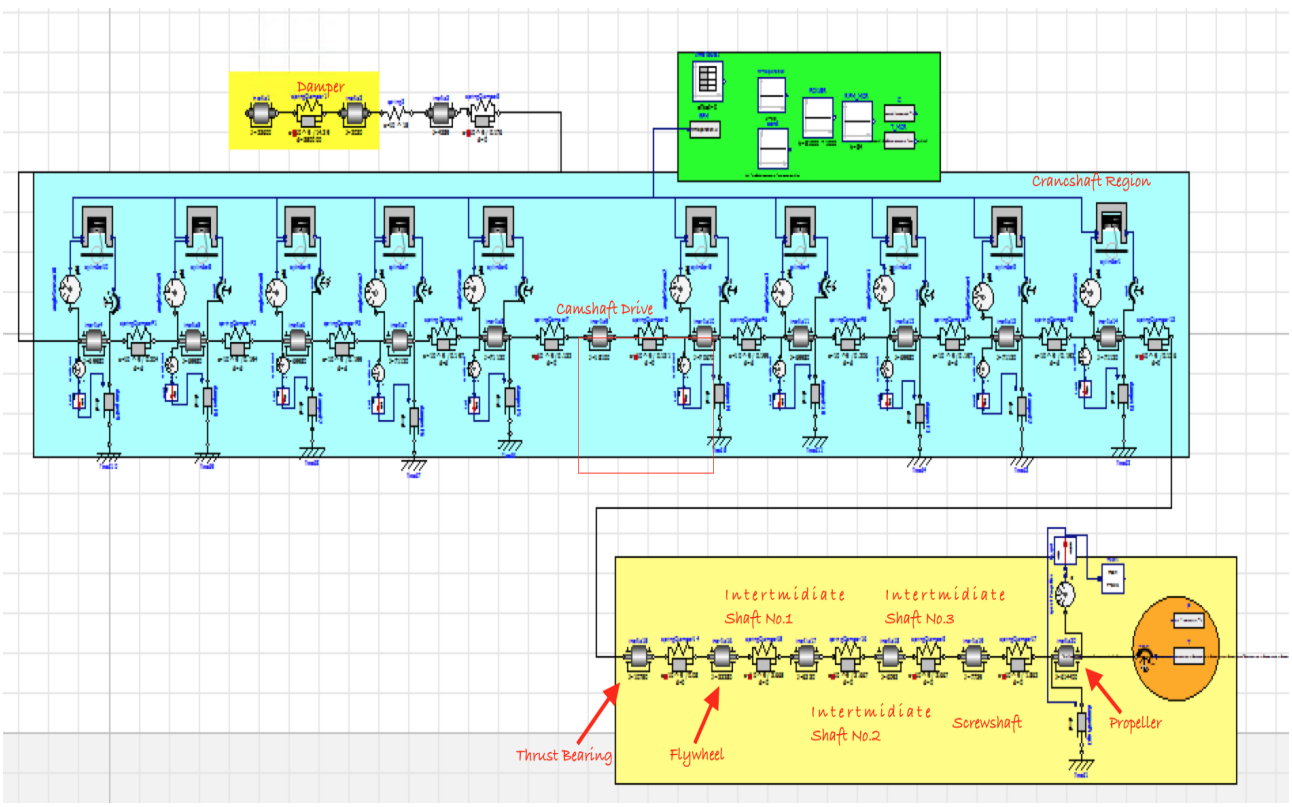

Figure 6: OpenModelica model for torsional vibrations

The presented technique of modelling for the solution of torsional vibration problem uses a Lamped Mass parameter model, composed of a finite number of elements describing the torsional stiffness of the propel and intermediate shafts and crankshaft, the rotating inertia and the internal and external system damping. The first components used to set the model refer to the rigid parameters:

- Torsional stiffness

- Polar mass and moment of inertia
- Relative damping
- Absolute damping

The next step in analytically determining the torsional response is to calculate the torsional natural frequencies of the system. For this purpose the stiffness and mass inertia of the shaft and components being analysed, referred also as the mass-elastic data, are required. In this case the mass moment of inertia and torsional stiffness, as already mentioned, are obtained from the torsional vibration analysis of a 10,000 TEU Containership. If not provided they can be calculated using simple formulas. Table 1, illustrates the data of the torsional system.

Table 1: Torsional System

Engine Data					
Engine Model	10S90ME-C Mk9.2, 2-cycle	**Bore**	900 mm		
Engine MCR	51000 kW	**Stroke**	3260 mm		
Speed at MCR	84 RPM	**Number of Cylinders**	10		
Torque at MCR	5798 kNm	**Connecting Rod Ratio**	0.4528		
MEP	17.75 bar	**Reciprocating Mass**	$17,840$ kg		
MIP	18.69 bar	**Firing sequnce**	$1, 8, 7, 3, 5, 9, 4, 2, 10, 6$		
Torsional System					
No.	**Inertia kgm^2**	**Elasticity nradNm**	**Eq. Diam. mm**	**Component**	**Firing angle**
1	23600	14.29		Damper	
2	2030	zero			
3	4389	0.176	1130	Flange	
4	69980	0.204	1130	Cylinder 1	0°
5	69980	0.194	1130	Cylinder 2	254°
6	69980	0.199	1130	Cylinder 3	115.5°
7	71120	0.197	1130	Cylinder 4	229.2°
8	71120	0.132	1130	Cylinder 5	141.5°
9	18100	0.131	1130	Camshaft Drive	
10	70670	0.199	1130	Cylinder 6	343.8°
11	69980	0.206	1130	Cylinder 7	82.2°
12	69908	0.197	1130	Cylinder 8	36.4°
13	71120	0192	1130	Cylinder 9	185.2°
14	71120	0.126	1130	Cylinder 10	288.4°
15	10750	0.08	1179	Thrust Bearing	
16	33280	3.665	805	Flywheel Intermediate Shaft No.1	
17	6330	3.667	805	Intermediate Shaft No.2	
18	6093	3.667	805	Intermediate Shaft No.3	
19	7739	1.862	988	Screwshaft	
20	614400			Fixed Pitch Propeller	

The natural frequencies of the undamped system can be found from solving the equation of the free vibrations. Table 2 shows the calculated natural frequencies and mode shapes for the 20 nodes(inertias) of the system, which agree with the results of the available study. The results of this study are also include in the appendix A for reference

Table 2: Natural frequencies without damping

Natural Frequency	**Node Relative Angle/Mode Shape**																			
(RPM)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
138.6	1.08	1	1	1	0.996	0.989	0.979	0.966	0.955	0.944	0.925	0.902	0.877	0.851	0.832	0.82	0.241	-0.339	-0.917	-1.21
508	29.5	1.35	1.35	1	0.552	0.105	-0.358	-0.803	-1.079	-1.346	-1.698	-1.993	-2.197	-2.312	-2.328	-2.333	-1.746	-1.045	-0.277	0.124
751.3	-0.904	0.984	0.984	1	0.931	0.786	0.571	0.308	0.114	-0.081	-0.369	-0.634	-0.834	-0.958	-0.986	-0.999	-0.828	-0.537	-0.172	0.028

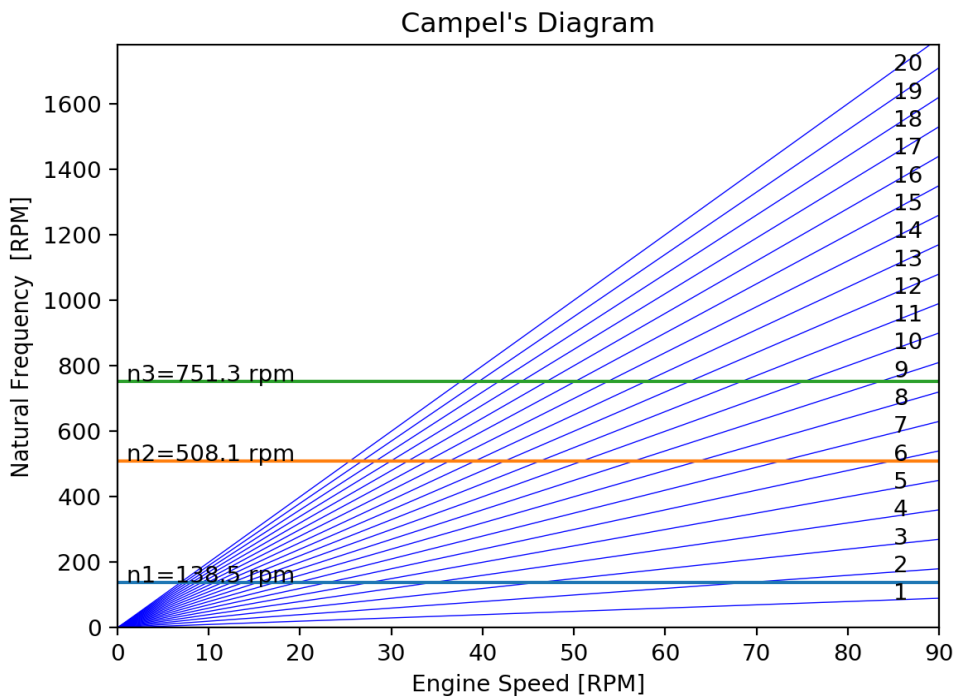

Figure 7: Campbell diagram resonance of natural frequencies due to harmonic excitations

In table 2 only the first three natural frequencies are shown since these are the ones interfering with the excitation harmonics for the engine's speed range. Figure 7 illustrates the Campbell's diagram comparing the system's first three natural frequencies, to the excitation frequencies that may be critical for the specific engine speed, if there is insufficient damping in the system. It is obvious that there are many engine critical speeds due to the multiple natural frequencies and order

numbers, but are only important when they coincide with engine running speed.

The next components that need to be set are those representing the excitation forces. The forced torsional vibrations of the propulsion system's shafts are caused by the torque of the cylinder gas pressure forces and the inertia forces of the reciprocating masses. The method used in this study, performs harmonic analysis on the function of the tangential force T, considering the first 20 harmonics. Furthermore, the total tangential force is assumed being composed of the ideal firing gas, the mass and compression gas components. The values for the amplitudes of each harmonic can be seen in appendix A.

$$T = T_m + T_1 sin(\phi + e_1) + T_2 sin(2\phi + e_2) + ... + T_k sin(k\phi + e_2 0)$$

The torque of one cylinder is the sum of a constant term, T_m and of the harmonics components. The constant component cannot produce torsional vibrations and only creates twisting of the shaft section between the engine and the propeller, on which the alternating twisting due to the harmonic components are superimposed. In this work the amplitude of the constant component produced by all (10) cylinders is assumed to be calculated by a cubic variation of power demand as a function of speed, known as propeller's law curve. The values for this curve showing the power produced for each engine speed can also be found at appendix A.

In contrast to the constant torque T_m, produced by the cylinder to accelerate the system, the absolute damping components of the system create an opposite constant toque which is dependent on the engine's speed.

The other main excitation source, the propeller, is assumed that the torque-power demand of the propeller is the difference between the total torque-power supplied by the engine and the opposing torque, due to the absolute damping of the system. In this way the power demand of the system is equal to the torque-power supplied by the main engine and the system oscillates around the equilibrium point. This assumption was made due the available data from the original torsional study of the vessel. For demonstrating the excitation of the propeller at the OpenModelica environment the combination of the *torque* and *real exression* components is used.

4.1 Creating New Modelica Classes

In figure 8 the components used for demonstrating the rigid parameters are shown. The *inertia* component represents the polar mass and moment of inertia. The

springdamperF is used for demonstrating the torsional stiffness and relative damping of the shaft. In the original *springdamper* component found in the standard Modelica library, the damping torque is calculated as:

$$T_{d_{rel}} = D\omega_{rel}$$

where,

D, is the damping constant in Nms/rad

ω_{rel}, is the relative speed between two inertias in rad/s

The new component calculates the relative damping as proposed by *MAN B&W*, using the unidimensional relative damping $\kappa = \frac{DampingTorque}{ElasticTorque}$.

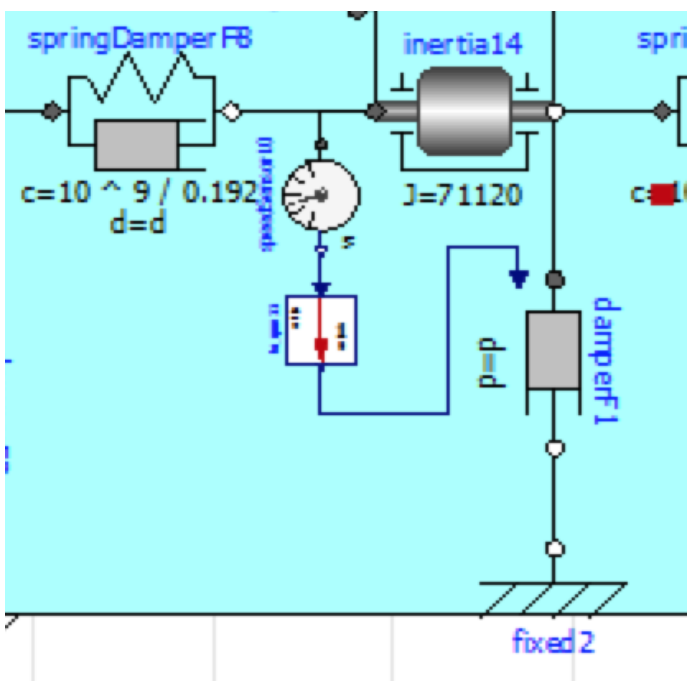

Figure 8: Rigid parameters demonstrate with OpenModelica components

Similarly, new components are created for the absolute damping of the system incorporating the *MAN B&W* proposal. The absolute damping in this case is calculated as:

$$T_{d_{abs}} = D\omega_{rel}$$

where,

D, is the damping constant in Nms/rad, calculated as $2 \cdot 5.5 \cdot I_{propeller} \cdot \omega/100$ for the propeller and as $2 \cdot 0.85 \cdot I_{cranck} \cdot \omega/100$

I, being the moment of inertia in kgm^2

ω_{rel}, being the relative speed of the shaft with the environment

A completely new component is used for modeling the torque generated form each cylinder. This components takes as input parameters the cylinder firing angle and the desired misfire percentage of the cylinder. Then calculates the required torque for the requested rotational speed, by superimposing the calculations for the constant term, T_m and for the harmonic components of the ideal fire, mass and compression. The amplitude of the harmonic components is assumed to change relative to the rotational speed by a factor $f_T = \frac{MIP}{MIP_{MCR}}$. The use of this correction factor will discussed later at section 4.2.

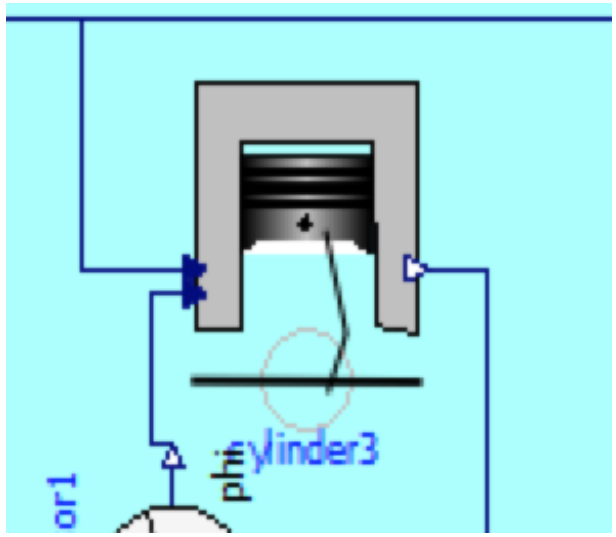

Figure 9: New model class for cylinder

Lastly, it is useful mentioning some of the components used to set the parameters for the system's desired speed. In the case of a steady state simulation a parameter named *RPMoperation* is set at desired speed needed for the simulation. This speed is then used as the system's initial rotational speed and as the input parameter for the *cylinder* components to produced the relative torque. The torques produced by the relative damping, absolute damping and the propeller are then calculated based on the rotational speed and torque of cylinders. In the case of a transient state simulation, an initial rotational speed can be set for the model in addition to a function for the desired speed increase rate.

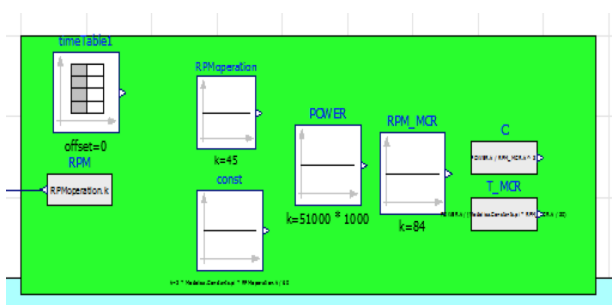

Figure 10: Block components used for system's initialization

4.2 Model Validation

In order to validate the developed torsional vibration model, the results are compared to those of the available torsional vibration study of a 10,000 TEU Container Vessel's propulsion system.

In the original study, a modal summation is used for calculating the vibration amplitudes at each inertia and the stress between them. With this method the amplitude due to the effect of each excitation order is calculated for the engine's speed range. Moreover, the relative and absolute damping are order dependent and the total amplitudes at each inertia are calculated as the synthesis, superposition, of the magnitudes of all orders. The OpenModelica model presented is equation based, thus by setting the right parameters it's as able to represent vibration amplitudes, as those would appear in a real life situation.

For the purposes of validating the model, multiple steady-state simulation where performed for various speeds at the engine's operational range, $20 - 90\ rpm$. Due to the large volume of simulations, the OpenModelica model was simulated using a *.mos* script, run from command line. Using this script one can change the initial parameters, select the variables to be saved, import Modelica model and generally add lines of code as would do at the OMEdit text view. The results are then processed using a *Python* script. For each steady state simulation corresponding to a specific speed,the vibration and torque amplitude is calculated as the peak to peak distance for each obtained signal. Repeating the process for all simulation the total vibration amplitude at each speed is calculated and compared to those of available study.

Bellow, figures 11 and 12, demonstrate the comparison to the available study, for the total stress amplitude at each speed for the intermediate shaft at a point after the flywheel and for the total stress amplitudes for the screwshaft respectively.

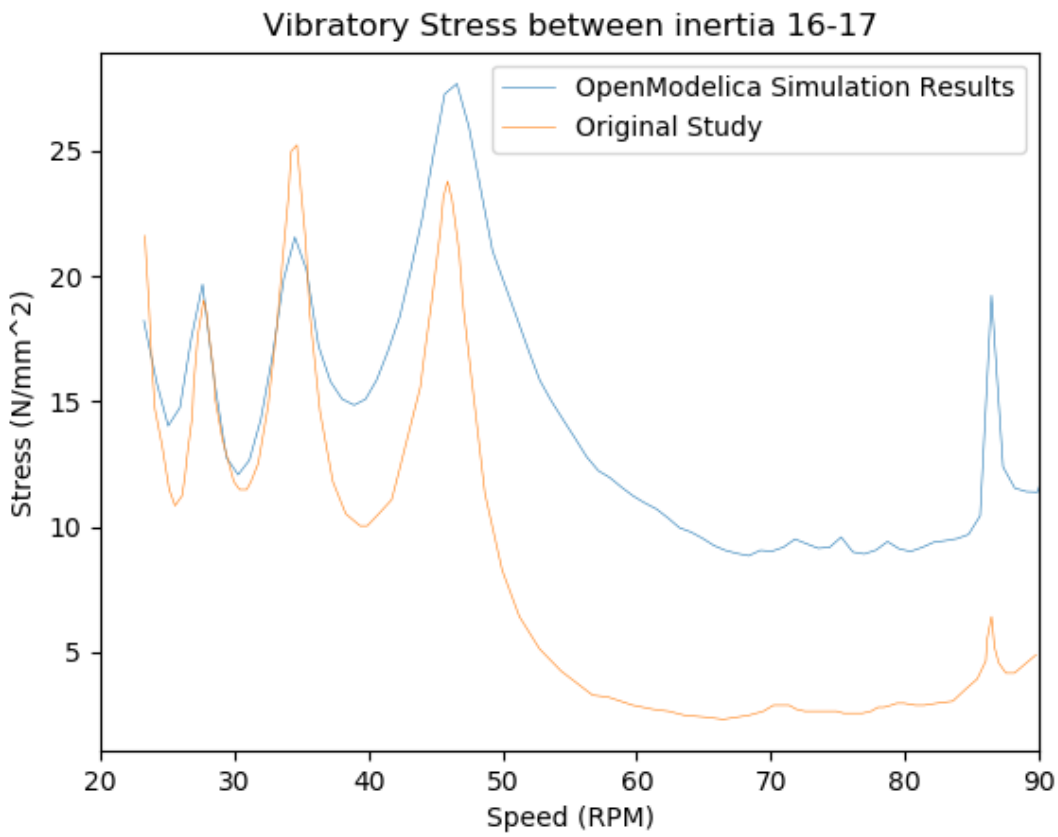

Figure 11: Vibratory stress after the flywheel between inertia No.16-17

Figures 11 and 12 illustrate the results simulated with the assumption of correcting factor f_T for the amplitudes of the harmonics synthesizing the cylinders' torque and correcting factor f_D for the absolute damping constant.The torque correcting factor is calculated as $f_T = \frac{MIP}{MIP_{MCR}}$. This assumption is based on the available data from the study and in order to correct the amplitudes of the simulated results. Specifically, in the original study the mean indicate pressure is calculated as :

$$MIP_{(n)} = MEP_{MCR}(n/n_{MCR})^2 + MIP_{MCR} - MEP_{MCR}$$

Moreover, the amplitudes of the harmonics are given for the MCR speed and no informations about the amplitudes at other speeds is available. For that reason, it was assumed that the amplitudes of the harmonics, consisting the cylinder torque, will be decreased by the factor $f_T = \frac{MIP}{MIP_{MCR}}$. Otherwise, a higher value for the torque would be calculated at speeds lower than MCR, resulting to higher values for the calculated vibratory stress.

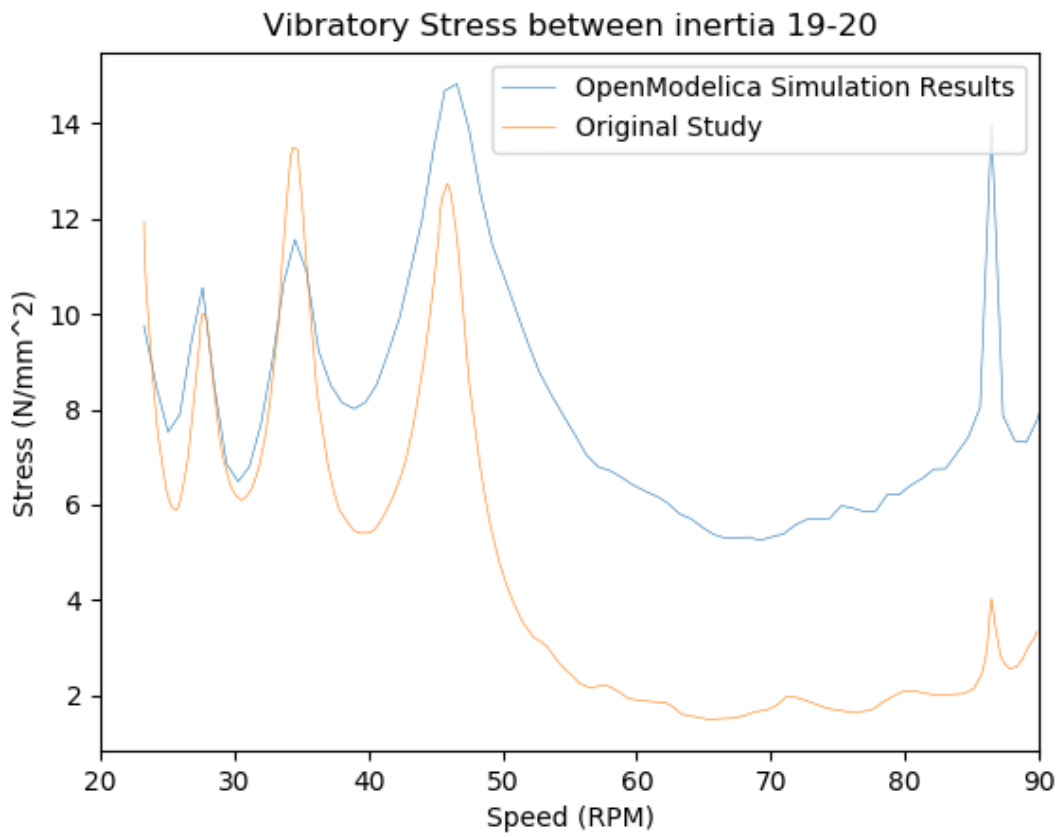

Figure 12: Vibratory stress at screwshaft between inertia No.19-20

The damping correcting factor is calculated as $f_T = \frac{MIP}{MIP_{MCR}}$. This assumption is based on the available data from the study and in order to correct the amplitudes of the simulated results. Specifically, in the original study the absolute damping is calculated as :

$$MIP_{(n)} = MEP_{MCR}(n/n_{MCR})^2 + MIP_{MCR} - MEP_{MCR}$$

The damping torque $T_{d_{abs}} = D\omega_{rel} = 2 \cdot 0.85 \cdot I_{cranck} \cdot \omega/100$ of the crankshaft is multiplied by a factor $f_D = 10.5$ to correct the effect what all harmonics would have to the damping constant. In a classic modal analysis such as that of the original study, the effect of each individual harmonic ($k = 1, 2, ..., 20$) is studied and the absolute damping constant has a linear relationship with the harmonic order. Thus, an assumption is made that when the summation of all harmonics is studied with the Modelica model,

the absolute damping constant at each crank is calculated as:

$$T_{d_{abs}} = f_D D\omega_{rel} = \frac{1}{20}\sum_{k=1}^{k=20} kD\omega_{rel} = 10.5D\omega_{rel}$$

Last, it must be confirmed that any fault simulated with the OpenModelica will indeed produce results that can be later easily distinguished and classified using machine learning algorithms. Bellow figures, show the simulated results of the vibratory stress amplitudes, for the intermediate shaft at a point after the flywheel, at intact condition and for Cylinder No.1 with a 50% misfire respectively.

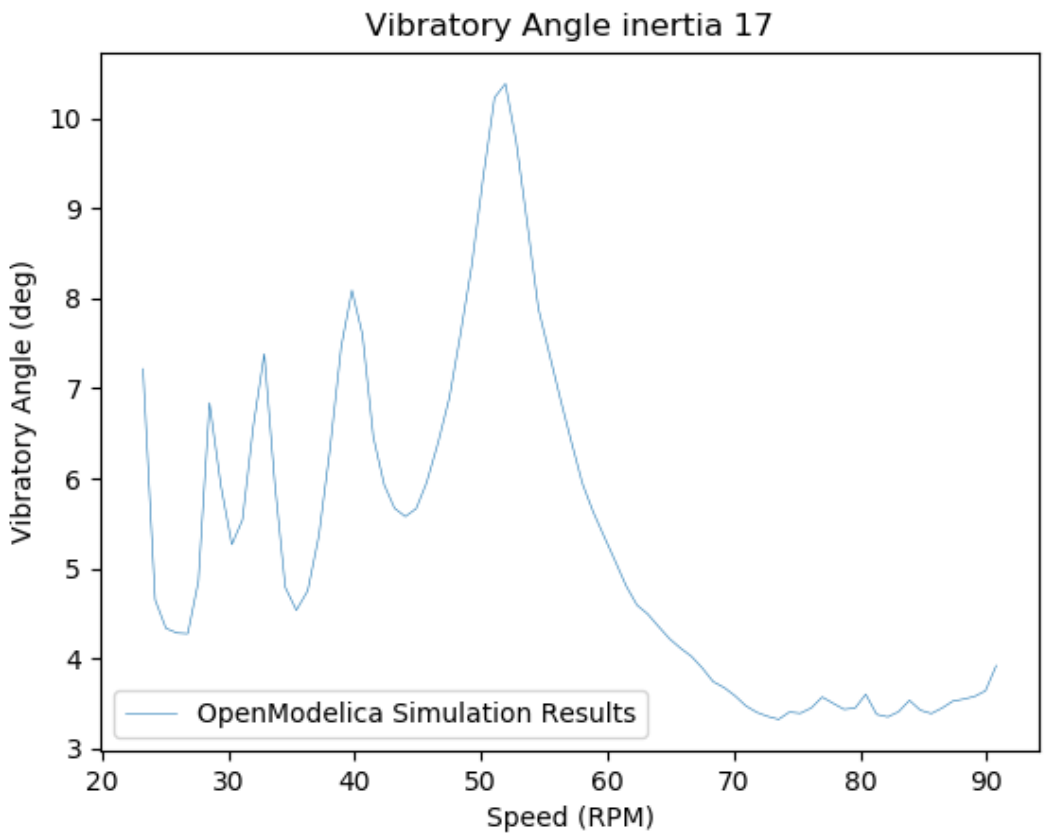

Figure 13: Vibratory angle after the flywheel between inertia No.17, intact condition

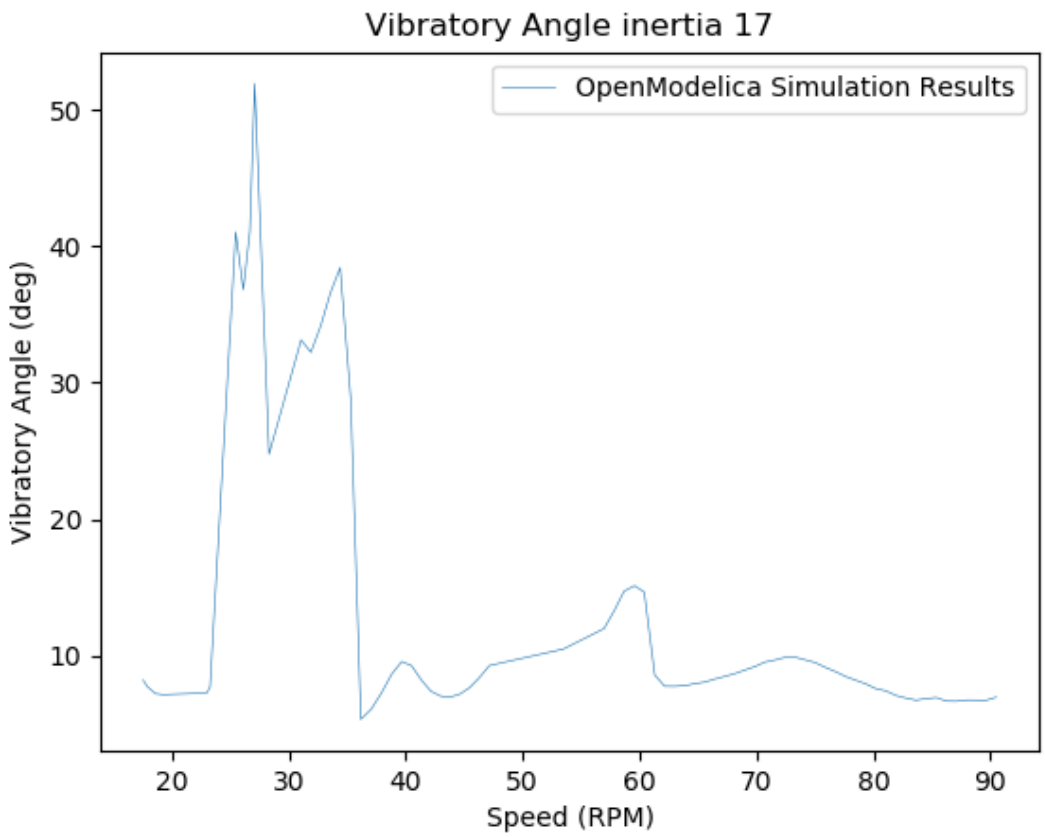

Figure 14: Vibratory angle after the flywheel inertia No.17, Cylinder No.1 50% misfiring

From figures 13 and 14, it is easily observed, that simulations for cylinder No.1 misfiring produce results with significant differences. From these results it can be concluded that machine learning algorithms will perform well at distinguishing between intact and fault conditions.

5 Common Cylinder Failures and Maintenance Techniques

5.1 Common Cylinder Failures

A preventive maintenance tool aims on the diagnosis of common failures. Therefore, it is important that the failures with the highest occurrence rates or/and impact to vessels operational life should be studied. As already mentioned, main engine's faults can classified in two main categories.

- Combustion faults
- Mechanical faults

One of the most common combustion faults is that of cylinder misfire. Other commonly occurring faults are that of the fuel oil valve, of the fuel delivery system, the cylinder head and cooling system.

- Fuel Injection System
- Cylinder Head and Valves
- Charging and Exhaust System
- Cooling System
- Bearings, Pistons, Liners, Timing Gears, etc.

Most of the faults can be detected by changes from expected values or patterns measured by speed, temperature, pressure and vibration sensors. A common fault can be located at piston rings used to keep gas blow-by from the combustion chamber to the crankcase. A possible blow-by will appear as reduction in compression pressure and power loss. Piston rings assist also to the lubrication of the cylinder liner, meaning that a malfunction will result in piston scraping the cylinder liner and increase in heat transferred from the piston to the liner. Apart from machinery faults, a fault can arouse due to fuel oil quality. Marine fuel oil's consistency and quality varies for different suppliers. The impact of a bad quality fuel oil can be seen at the cylinder's pressure diagram as a delayed and prolonged combustion with high pressure peaks or increased rates in the rise of pressure . In addition, the faults due to unsuitable fuel oil, increase scavenging temperatures and high fuel to air ratio can appear at the cylinder's pressure diagram as irregular peaks known as "knocking", due to the ignition of fuel at irregular intervals. Another, of the common problems appearing is due to malfunction of the fuel injection system and specifically fuel being injected earlier or with delay. These may mainly be

due to a fault at the fuel injector or at the fuel pump and their effect can be seen at the cylinder's pressure diagram. Similar effects on the pressure diagram can be due to incorrect timing of valve opening or even due to damaged exhaust valve. Increased scavenging temperatures can be an indication of fault at the cooling system and the scavenging air system. Faults located at the turbocharger are usually translated as a change in its rotational inertia, which then result into delay of the turbocharger during transient states. Thus, during acceleration or deceleration a delay in the air supplied by the turbocharger results into a rich or poor air-fuel mixture respectively.

The origin of the above mentioned faults can easily be detected after an inspection. For example,during an inspection a piston crown damage by erosion, due to mechanical overloads and thermal disintegration can easily be observed They cause usually derives from excessive and/or premature fuel injection and lack of oxygen for the fuel available, translating to incorrect spraying and leaking injector nozzles injectors. Another fault that can be easily detected with inspection is that of burning low sulphur fuels with increased additives. Over-additivation lead to mechanical and chemical bore polish, which then lead to micro-seizures and latent risk of scuffing.

Figure 15: Damage due to low sulphur fuel

From the available literature it is clear that the fuel injection system has been the most prevalent source of problems for diesel engines. Meanwhile, engine components subjected to high levels of wear, such as pistons, liners and bearings rank near the bottom. Furthermore, the amount of attention invested in the maintenance of marine engine components compared to to their failures rates is unequal and it is clear that

this is not happening due to lack of proper maintenance attention. For that reason preventative maintenance should be used for parts that are most prone to failure or are critical to engine performance and avoid any unnecessary component replacements, regularly scheduled maintenance and yearly overhauls. Vibration analysis can be used as a technique for monitoring the performance and failures by measuring vibrations against known failure vibrations, by Interpreting and classifying the information from acquired data. It must be noted, that even though an abnormal condition could be detected, by using vibration analysis for calculation of the vibration spectrum, that condition may be the result of not just one fault of but sometimes multiple causes. The later makes difficult to distinct the fault's location.

5.2 Maintenance Philosophy

Maintenance's role is not to repair damaged equipment, but to prevent its damage, enabling machinery to work efficiently, reliably and safely. The selection of the type of maintenance depends on the application considering the cost and safety. The maintenance philosophy can be divided based on the following approaches:

- Reactive or run to failure maintenance

  This type of maintenance is based on the philosophy of run-to-failure. In this case maintenance is only performed after failure of a component. In most cases no repair is possible and the component is replaced by a new one. This type of maintenance can offer the longest time between shutdowns but failures are catastrophic and can possibly affect multiple components and/or machines and additionally a large spare-parts inventory is required. For these reason, reactive maintenance is mainly applied to relatively not expensive and non-critical machines or where there is back up equipment.

- Preventive or time based maintenance

  This method has been established for more expensive equipment with more costly operation. When preventive maintenance is implemented, maintenance and inspections of equipment is performed at a fixed frequency, following Original Equipment Manufacturers (OEMs) recommendations and past experience. With this method unplanned downtime is minimised since maintenance is performed to prevent failure of the machine.. Significant increase in machine lifespan is also one of its advantages as well as the lower cost due to the reduced number of components that need complete replacement. Preventive maintenance generally aims to provide such maintenance intervals determined by the rate of failures of similar equipment, aiming for a 1-2% of them to experience failures between maintenance intervals. Thus, the clear majority of machines would be able to continue working without maintenance for multiple maintenance intervals. The weakness of this method the difficulty to estimate time between repairs. Not taking into consideration the actual machine condition results to failures that can occur before the planned repair or maintenance been conducted too early.

- Predictive or condition based maintenance

  In this case the maintenance scheduling for each component is decided based on its past and present condition, rather than at predetermined intervals by performing monitoring and diagnostics. Predictive maintenance is extremely cost effective for marine engines that need to run for long periods without any shutdowns. Downtime due to unplanned maintenance can be avoided, while lifetime, and safety can be increased. The main disadvantage is that higher initial cost is

required, but still proves to be more economical during a machine's lifespan and offers better management of spare parts.

- Proactive or prevention maintenance

  This method aims to identify the root cause of a fault or a condition. The objective is to proceed to corrective actions based on the root cause in order to avoid faults or improve a condition.

6 Machine Learning Algorithms

6.1 Introduction To Machine Learning

Over the past years, machine learning algorithms have gained a lot of attention and have been extensively used for classification problems, aiming to recognise or predict different classes within a dataset. Machine learning algorithms are programs (math and logic) that adjust themselves to perform better as they are exposed to more data. The "learning" part of machine learning means that those programs change how they process data over time, much as humans change how they process data by learning. So a machine-learning algorithm is a program with a specific way to adjusting its own parameters, given feedback on its previous performance making predictions about a dataset.

- **Supervised and unsupervised learning**

  There are many different machine learning algorithms and can be distinct in two main categories, supervised and unsupervised learning. For the purposes of this thesis a supervised learning model was used. By this term it is meant that the model is trained to reproduce known outputs for a training set whose output data patterns are known. On the contrary unsupervised learning models are trained with unlabelled data and don't have any outcome to predict data . Thus they may be used to uncover information about the data structure as a preliminary step before applying supervised learning.

- **The Dataset**

  A training dataset is the actual dataset used to train the model during development process. During the preprocess the training dataset is divided into features and labels. Features are the algorithm's inputs and each one is represented as one column of the dataset, whereas labels are the algorithm's output. The data can have one or more labels and the aim of the machine learning model is to be able to predict new unlabelled data. For the training process the model is fed with the training dataset receiving both known input and output data and the selected algorithm creates the appropriate rules between the input and output data. During this stage many algorithms can be tested and compared in order to opt for the one with the highest performance

The supervised learning algorithms can be divided into two learning techniques:

1. **Classification**

   This is a technique that separates the training data into classes and aims to predict the output of unknown data by reproducing class assignments from the

training set. The classification of data can be done according to specific individual requirements.

2. **Regression**

   This is a technique that fits the training data and aims to reproduce the output value.

For the purposes of this thesis the classification technique was used, since the developed OpenModelica model offers the ability to obtain data for various loads and engine conditions. In general the classification during machine learning can be binary, multi-class, multi-labelled and hierarchical.

- **Data Processing And Feature Engineering**

  When creating a machine learning model, one of the first considerations is to decide the features to be used. Features are pieces of information given as input to the algorithm and can be either numeric values or text. Datasets are usually of high dimension including irrelevant or redundant features, which may lower the performance of a machine learning algorithm. For this reason, feature selection methods must be applied in order to increase performance

In machine learning the terms of bias and variance used in statistics, are used for in order to decide whether supervised learning algorithms can generalize beyond training dataset.

- **Underfitting-Bias error**

  The bias error occurs when the model cannot adequately capture the important structure of the data, missing the relevant relations between features and target outputs.

- **Overfitting-Variance error** The variance error appears as sensitivity to small fluctuations in the training set and occurs from "overtraining" or when training samples are rare, adjusting to specific random features of the training data. Thus, the model performs well on the training dataset, but is less accurate in predicting unknown data.

6.1.1 Python For Machine Learning

Python is an object-oriented programming language and an easy-to-use language making it simple to use. In addition is a free software and can also be freely modified and re-distributed because while the language is copyrighted it's available under an open-source license. Python comes with a large standard library that supports many common programming tasks and is easily extended by adding new modules implemented

in a compiled language such as C or C++. One of it's advantages is the ability to run on different Operating Systems and can be embedded into an application to provide a programmable interface.

Python is broadly used in machine learning and scientific applications, having libraries such as Scikit-learn and SciPy for machine learning and data analysis. It must be noted that when selecting a programming language for machine learning and data science, the selection of the best language depends on the project or experience on previous projects. Scikit-learn is a machine Learning library that has a simple interface and it is well documented with many examples and tutorials. In continuation, Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests and gradient boosting, and is cooperates very well with Python's numerical and scientific libraries NumPy and SciPy, as well with the rest of the wealthy open-source available libraries.

6.2 Random Forest Algorithm

The Random Forest algorithm shows great performance in regression and classification problems in various fields and have gained increased attention within the machine learning community. Random decision forests known as Random forests is a machine learning technique useful for prediction problems. It is an ensemble learning method for classification or regression. Classification is achieved with a set of decision trees that grow using randomly selected subspaces of data at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forest handles non-linearity by exploiting correlation between the features of the data.

6.2.1 Decision Trees

A decision tree is a decision support tool based on the concept of the cutting off of possibilities. Can be used to classify data, and they cut off possibilities of what a given instance of data might be by examining a data point's features. They have a flowchart-like structure where the flow starts at the base with a single node and extends to the many leaf nodes that represent the categories that the tree can classify. Each internal node represents a decision, each branch represents their possible consequences, including chance event outcomes, resource costs, and utility outcome and each leaf node represents a decision taken after computing all attributes.

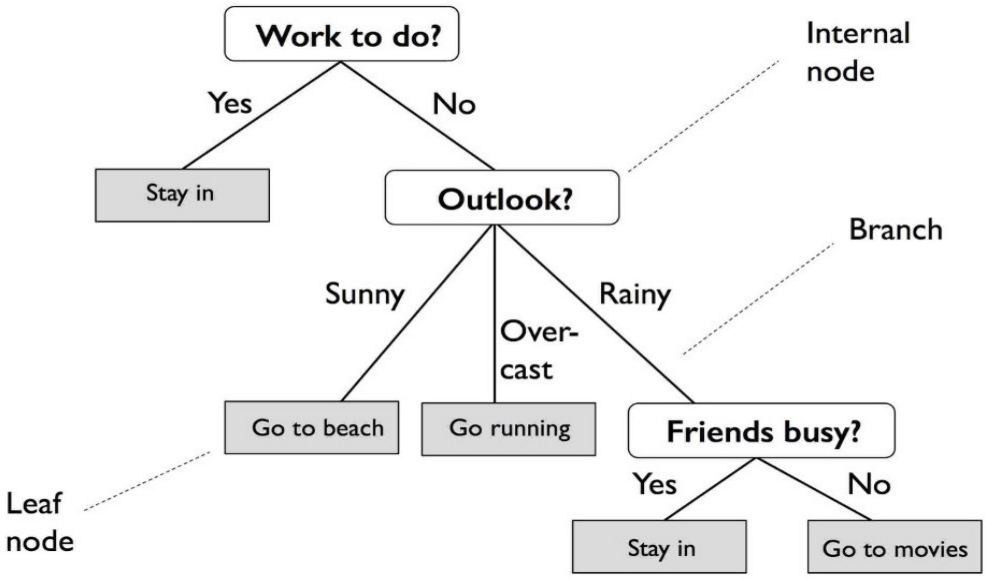

Figure 16: Decision tree example

Decision tree structure:

- Root Node: A root node is at the beginning of a tree. It represents entire population being analysed. From the root node, the population is divided according to various features, and those sub-groups are split in turn at each decision node under the root node.
- Parent Node and Child Node: These are relative terms. Any node that falls under another node is a child node or sub-node, and any node which precedes those child nodes is called a parent node.
- Decision Node: When a sub-node splits into further sub-nodes, it's a decision node.
- Leaf Node or Terminal Node: Nodes that do not split are called leaf or terminal nodes.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Pruning: Removing the sub-nodes of a parent node is called pruning. A tree is grown through splitting and shrunk through pruning.
- Branch or Sub-Tree: A sub-section of decision tree is called branch or a sub-tree, just as a portion of a graph is called a sub-graph.

Generally, are simple to understand and can give important insights and help determine worst, best and expected values for different scenarios, while be combined with other methods for improved results. Some of their disadvantages is the tendency to overfit when trees are grown very deep and tend to learn highly irregular patterns and thus becoming unstable and . Decision trees are able to handle multi-output problems and use continuous numerical input, they are not a practical way to predict such values, since decision-tree predictions must be separated into discrete categories.

6.2.2 Random Forest

The Random Forest algorithm is a a supervised classification algorithm which consists of a group of decision trees, which are the base learner. The decision trees are grown by feeding on training data using the random subspace method. The algorithm applies bootstrap aggregation and random feature selection to individual classification trees for prediction. That means all decision trees, in the Random Forest, do polling during the prediction and majority of the polls is considered the result of prediction. To break this down, decision trees can suffer from high variance and tend to over-fit the model to the training data. The bootstrap aggregation, called bagging, can reduce

this variance, but the trees are highly correlated. Random Forest is an extension of the above using the random subspace method to build trees based on multiple samples of the training data, enabling also to opt for the number of features that can be used to build the trees.

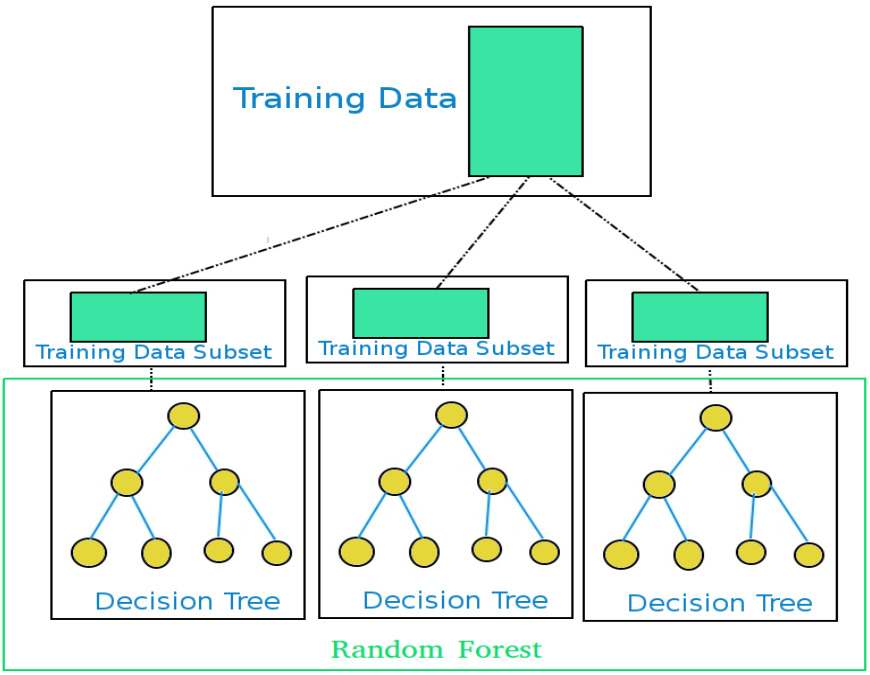

Figure 17: Random forest structure example

Random forest is considered an ensemble method that uses ensembles of decision trees. The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator. The scikit-learn implementation combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class.

6.3 Nearest Neighbors Algorithm

Nearest neighbors algorithm is based on the principle of finding a predefined number of training samples closest in distance to the new point, and predict the label from these, thus the model representation is the entire training dataset. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The algorithm stores the entire dataset, so training data must be carefully selected and some cases requires to be updated. Predictions are made by looking through the entire training dataset and finding the most similar "neighbour" based on a distance measure. The distance measure is most often used is Euclidean distance, calculated as the square root of the sum of the squared differences between a new point and an existing point across all input attributes. Other distance measures include Tanimoto, Jaccard, Mahalanobis and cosine distance. As an example Euclidean distance measure is suggested to be used if the input variables are similar in type, while the Manhattan distance is a good measure to use if the input variables are not similar in type. Selecting a higher value for the parameter k of the knearest neighbors, minimises noise effect of the dataset, but the classification boundaries become less distinct.

Within the Scikit-learn library two different nearest neighbors classifiers can be found:

1. KNeighborsClassifier implements learning based on the knearest neighbors of each query point, where k is an integer value specified by the user.

2. RadiusNeighborsClassifier implements learning based on the number of neighbors within a fixed radius r of each training point, where r is a floating-point value specified by the user. The k-neighbors classification in KNeighborsClassifier is the most commonly used technique.

6.4 Multi-layer Perceptron classifier

The Multi-Layer Perceptron classifier (MLPC) relies on an underlying Neural Network to perform the task of classification. Artificial Neural Networks (ANN) are widely used for classification problems and Scikit-Learn offers a library for an easy implementation. MLPC uses an ANN with three or more hidden layers and because of its many hidden layers, it's considered a deep learning method. Each hidden layer contributes to the overall performance of the network, using a back propagation technique for training the network. MLPC can learn and model complex and non-linear relationships and also generalize well, even when trained with incomplete data. Even though, the error in one node cannot affect the entire network, it is difficult to understand how the output layer was predicted and the algorithm tends to be computationally expensive.

The concept of an ANN is inspired by the human nervous system, which consists of the dendrite that receives information signal, the nucleus which processes the signals and the axon which connects one neuron to another. When a neuron receives an electrochemical signals from many sources for which the excitation is high enough, the neuron fires, passing on the signal. Based on that concept an ANN is a network of artificial neurons made up of an input layer which receive the input data, one or more hidden layers that processes the data, and output layer which makes the predictions.

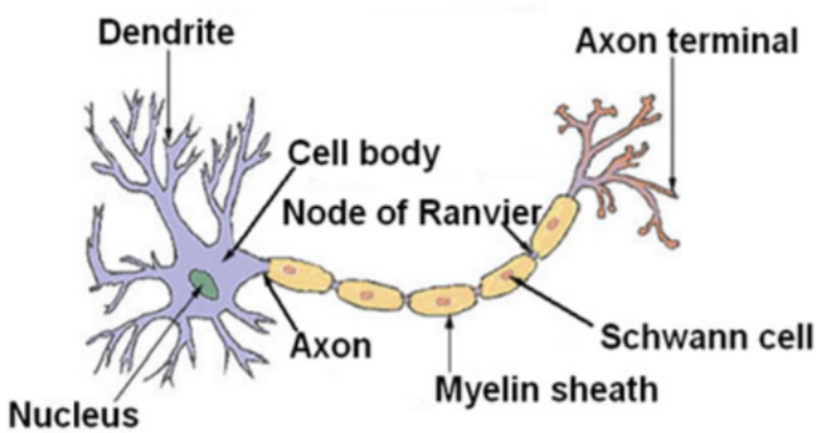

Figure 18: Structure of human neuron

An ANN architecture has the following structure:

- **Input Layer**, which is the visible layer receiving the input feature, with the number of input nodes representing the number of features from the input dataset.

  Hidden Layers, in contrast to the input layer, remain hidden and are the main

contributor of model's performance. MLPC consists of multiple hidden layers which are connected together.

Neuron, is the computational unit of the neural network. MLPC's neurons are connected in a directed graph with the output of one neuron acting as the input of the next neuron. Each neuron has weight, biases and an activation function. Weights are the fundamental elements behind the learning process of the network. Learning happens by adjusting the weights during data processing, while the activation function maps the weighted sum of inputs to the output in the neuron.

Output Layer, the final hidden layer of the network and represents the model's prediction. The output can be a single output neuron, binary or multiple output neurons.

The learning process takes place through the stages of feed-forward and back propagation. During *feed-forward* stage, random weights are assigned for each input, adding a bias to the sum of input and weight. Then that information is sent to each neuron in the hidden layer for further processing. The neuron uses an activation function, for processing the information based on weights and biases, by multiplying the weights and bias added. One complete cycle of the forward pass and backward pass is usually referred to as an epoch. *Back propagation* is needed during the learning process to adjust weights and biases in a backward direction, in order to improve the accuracy. The final predicted output occurs after a feed-forward phase and is compared to the expected output by calculating the error, using a loss function.

A single artificial neuron can be represented mathematically as:

$$\bigcirc = f\left(\sum_{i=1}^{k} w_i(I)_i + b\right)$$

where,

$\bigcirc$, is the output
f, is the activation function
I_i, are k inputs
w_i, are summation weights
b, is a bias term determining how much activation induces firing

A layer of multiple artificial neurons can be represented mathematically as bellow, where indices $l-1$for the input vector and l for the output vector anticipate the next step, where layers are interconnected.

$$(\bigcirc^{l})_{j} = {f_{j}}^{l}\left(\sum_{i=1}^{k_{l-1}} {w_{j,i}}^{l}(\bigcirc^{l-1})_{i} + {b_{j}}^{l}\right)$$

where,

$\bigcirc^{l}$, is the output vector
${f_{j}}^{l}$, is the activation function
$(\bigcirc^{l-1})_{i}$, are k_{l-1} inputs
${w_{j,i}}^{l}$, is the weight of the i_{th} input in the j_{th} neuron
${b_{j}}^{l}$, is the bias term of the j_{th} neuron

By combining layers a MLPC is achieved as shown in figure 19 bellow.

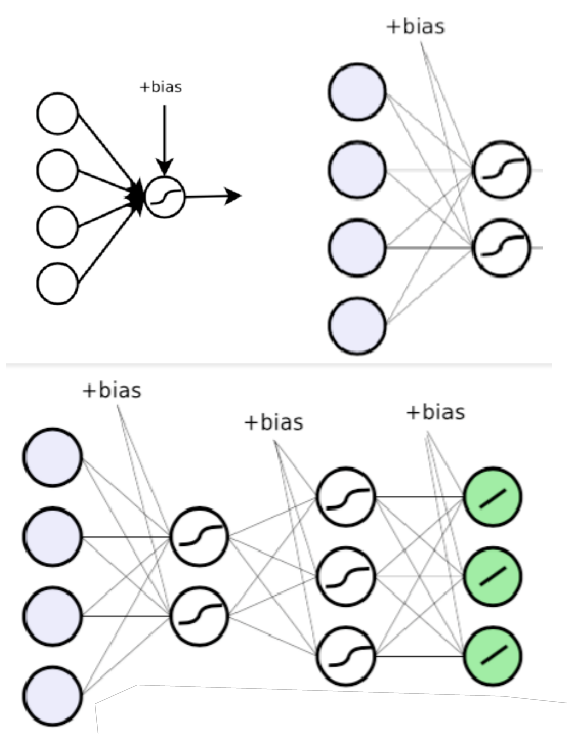

Figure 19: Structure of an MLPC

7 Development of a Fault Diagnosis Method

7.1 Data Processing

7.1.1 Data Extraction From OpenModelica Model

To diagnose a fault the monitoring of a machine is essential in order to obtain and detect data that underlay a fault. The diagnosis occurs after the analysis of the detected data which usually consists of vibration signals measured at regular time intervals at points of interest. Within this thesis fault diagnostics are based on vibration signals, speed and power.

The ISO 17359 standard, *Condition monitoring and diagnostics of machines*, offers guidelines about how various operational parameters associate with various machine faults. Most of the faults for rotating machinery can be identified by changes of amplitudes and spectral content of vibrations. By applying the right analysis tool to the vibration signals information regarding the fault point can be derived.

Machine types: **pumps**	**Symptom or parameter change**									
Examples of faults	Fluid leakage	Length measurement	Power	Pressure or vacuum	Speed	Vibration	Temperature	Coast down time	Oil debris	Oil leakage
Damaged impeller		•	•	•	•	•	•	•	•	
Damaged seals	•	•		•	•	•				
Eccentric impeller			•	•	•	•	•	•		
Bearing damage		•	•		•	•	•	•	•	•
Bearing wear		•				•	•	•	•	
Mounting fault						•				
Unbalance						•				
Misalignment		•				•				
• Indicates symptom may occur or parameter may change if fault occurs.										

Table 3: Typical monitoring parameters

The data extracted from the Modelica model simulation, for the training of machine learning algorithms, is that of the vibration signal at the intermediate shaft after the flywheel and the speed signal. The signals are obtained after multiple simulations at steady state conditions with a $1RPM$ mean speed difference between them for various engine conditions, in the range of $20 - 90RPM$. Simulations' sampling frequency is

100 Hz, since the maximum engine speed is $n = 90\ rpm\ or\ n = 1.5\ Hz$ and thus the highest order of excitation(20) has a frequency of 30 Hz. Moreover, the simulated signals are obtained after the system has reached steady state and have a 25 s duration. Each signal is further divided into five equal time duration signals in order to obtain more samples for the training of machine learning algorithms.

Then each signal is processed further using time domain analysis where statistical features are calculated from the vibration data. By comparing the statistical features, particular faults can be identified. In method proposed within this study, the statistical features used are that of the mean value, peak amplitude, the standard deviation and the variance. In addition, a lot of valuable information can be extracted from a frequency domain analysis, showing important information about the problem source. The frequency domain analysis technique used for this study is presented next at section [7.1.2].

7.1.2 Octave Band Analysis

Time-domain processing generally provides an intuitive representation of the relationships between data and is often used to identify morphology of transient information. In the case of the ship's main engine the information of interest repeats over regular intervals, transformations can be used to convert the time-domain information to the frequency-domain. Specifically, the oscillatory information from the vibration signal can, which may be hidden in the time-domain may be easier to assess in the frequency-domain. The vibration signal can be represented as a sum of sinusoids, with each sinusoid having a specific oscillation frequency, amplitude, and phase shift, by using commonly used Fourier Transform. Using the Fourier Transform, which can be easily implemented in Python using the Fast Fourier Transform (FFT) algorithm , the signal from the time-domain can be represented to the frequency-domain to perform Spectral Analysis

In order to use the results of a Spectral Analysis as features for a machine learning algorithm, one must select based on a frequency by frequency basis. That would result at selecting to many features in the case of deciding to use all the available information or to a number of features that are not important in the case of selectively deciding the frequencies to be used. As a solution to that the Octave Band Analysis is used in this case, which can generates results that can be easily be used as features of an algorithm.

Octave Band Analysis has been developed for analysing audibility performance, by grouping more naturally frequencies of audio signals, so that the distributed signal power scales better for analysis. The frequency range is divided into sets of frequencies called bands, with each band covering a specific range of frequencies. A octave has an upper band frequency which is twice the lower band frequency. Similarly, a one-third octave band is a frequency band whose upper band-edge frequency is the lower band frequency times the cube root of two.

The audio spectrum from 20 Hz to 20 KHz can be divided up into 11 octave bands. If we set/define the 7th octave band's center frequency to be $f_7^{center} = 1000\ Hz$ then all lower center frequencies for octave bands can be defined from each other using the formula $f_{n-1}^{center} = f_n^{center}/2$. All higher center frequencies for octave bands can be defined from each other using the formula $f_{n+1}^{center} = 2f_n^{center}$. For each center frequency,the half- octave low, or high, frequency for each octave band are (respectively) given by the formulae $f_n^{low} = f_n/2^{1/2}$ and $f_n^{high} = 2^{1/2}f_n$. Thus the center frequency of an octave band is $f_n^{center} = f_n^{center}/2$ and the fractional percent bandwidth per octave band is constant.

In the 1/n-octave analysis, the signal to be analysed is split into partial signals by a

digital filter bank before the sound level is determined. The filter bank consists of several filters connected in parallel, each with a bandwidth of 1/n-octave. An octave filter is a filter whose upper cut-off frequency is twice the lower cut-off frequency, whereas 1/3-octave filters further subdivide each octave band into three parts and so on. This means that octave filters or 1/n-octave filters do not have a constant absolute bandwidth, but a constant relative bandwidth, i.e., the frequency bands are equidistant on a logarithmic frequency scale.

Each octave band may be separated into three ranges - referred to as one-third-octave bands. The audio spectrum from 20 Hz to 20 KHz can be divided up into 31 1/3-octave bands. If we set/define the 19th 1/3 octave band's center frequency to be $f_{19}^{center} = 1000\ Hz$ then all lower center frequencies for 1/3octave bands can be defined from each other using the formula $f_{n-1}^{center} = f_n^{center}/2^{1/3}$. All higher center frequencies for 1/3octave bands can be defined from each other using the formula $f_{n+1}^{center} = 2^{1/3} f_n^{center}$. For each center frequency,the1/6 octave low, or high, frequency for each 1/3 octave band are (respectively) given by the formulae $f_n^{low} = f_n/2^{1/6}$ and $f_n^{high} = 2^{1/6} f_n$. The percent fractional bandwidth per octave band is constant.

In this study, the maximum engine speed is $n = 90\ rpm\ or\ 1.5\ Hz$ and the highest expected frequency is $n = 30\ Hz$ corresponding to the $k = 20$ order of excitation. Thus, one third octave band analysis is performed to the frequency range between 0 Hz and 50 Hz. With this analysis the frequencies present in the spectrum are grouped into bands making it easier to extract information about the spectrum amplitudes at different rotational speeds.

7.2 Algorithm Evaluation Methods

In order to decide about the capabilities of the tested algorithms, they will go through a validation process for the available dataset. As already mentioned at the data processing section, the features that will be included in the dataset for training the model are as shown bellow:

- Average speed
- Peak angle
- Standard Deviation
- Variance
- 1/3 Octave Bands

The above features correspond to the speed and angle signal obtained for the part of the shaft after the flywheel.

In order to avoid overfitting the selected dataset is divided to a training set and a test to be used for training and validating the algorithm. Otherwise, training the model and testing it on the same data, would result to a model capable to perfectly predict classes of the samples that has been trained, but will have a very poor score at predicting anything useful unknown data. Usually, a random split of 80% to 20% is used for the training and the test set respectively. In this work, a random split of $80\% - 20\%$ after shuffle of the dataset, is also used with the addition of an initial *Corss validation* procedure called *k-fold CV*. In this approach the training set is split into k smaller sets called k folds. The model is trained on the $k - 1$ folds and is tested on the remaining fold. The final performance is then the average of the values computed in the aforementioned loop. The *k-fold CV* is computationally expensive and is used only as an initial step to check if the random split provides sufficient results, based on the calculation of algorithms' accuracy.

- **Correlation matrix**, is used to obtain more insight in how strongly each feature in the dataset is correlated with others. Due to the fact that the dataset contains a lot of features and the correlation becomes difficult to discern, the correlation of features with the class is visualized it in the form of a bar-graph.
- **Learning curve**, is used to visualize the the validation and training score of a model for varying numbers of training samples. It shows whether the model is over fitted or under fitted and whether an increase in the size of training set would be beneficial.

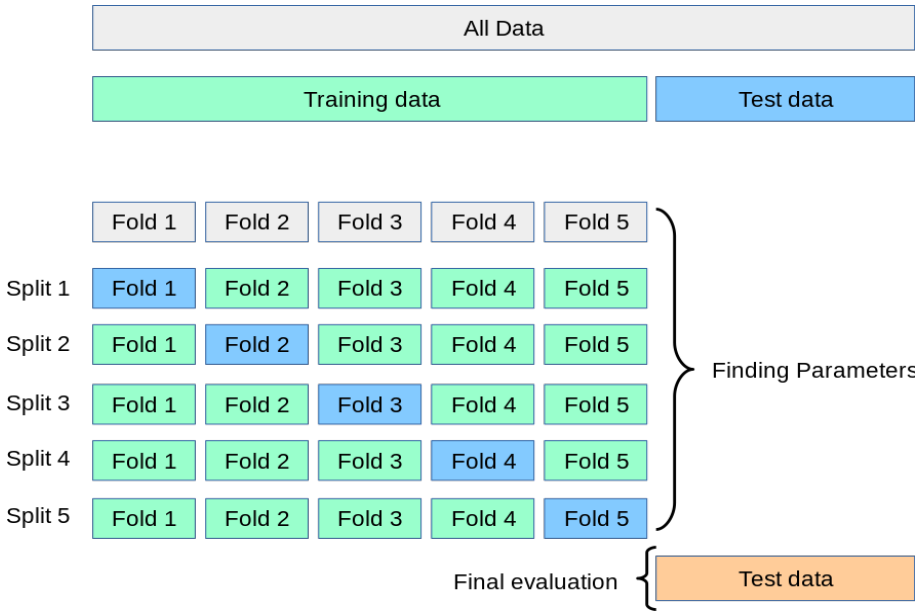

Figure 20: Illustration of data split with *k-fold CV*

- **Confusion Matrix**, is used for assessing the performance of a classification algorithm by comparing observed and predicted results. It is used to provided more information compared to the classification accuracy alone by showing *T*rue Positive and *False Positive* rates (number of correct and incorrect predictions) for each class

- **Classification report** displays the precision, recall, F1, and support scores per class, which are the main classification metrics. Metrics are defined in terms of true and false positives, and true and false negatives. The *Precision* metric shows the ability of classifier not to label an instance positive that is actually negative, calculating the ratio of true positives to the sum of true and false positives (for all instances classified positive, what percent was correct?). The *Recall* metric measures the ability of classifier to find all positive instances calculating the ratio of true positives to the sum of true positives and false negatives (for all instances that were actually positive, what percent was classified correctly). The *F1* score is a weighted harmonic mean of precision and recall and is lower than accuracy. The *Support* is just the number of actual occurrences of the class in the specified dataset, used for diagnosing imbalance problems.

7.3 Case Study

7.3.1 Method

This case study aims on specifying and locating a misfire due to a possible fault based on the vibration signals generated from the OpenModelica model. The algorithms to be tested are the following:

i. Decision Tree Classifier

ii. Random Forest Classifier

iii. KNeighbors Classifier

iv. Multi-layer Perceptron classifier

The proccess is divided in three steps as described bellow:

1. In this step the classifier predicts the possible appearance of a fault. A Classification problem is solved with two classes, *0* and *1*, where all signals corresponding to intact condition are classified as class *0* and those corresponding to a condition with one cylinder misfiring are classified as class *1*.

2. In this step the classifier predicts the possible the possible location of the fault. A Classification problem is solved with 10 classes, *1*,...*10*, where all signals corresponding to misfiring of cylinder no.1 are classified as class *1* and as class *2*,.., ,class *10* respectively for the rest of the cylinders misfiring.

3. In this step the classifier predicts the possible the possible misfire percentage for the predicted fault location of step 2. A classification problem is solved with 3 classes, *20*, *50* and *80* where all signals corresponding to the specific cylinder location misfiring at a 20%, 50% and 80% percentage are classified as class *20*, *50* and *80* respectively.

At each step the algorithms are trained using the features obtained as described at the previous sections. In particular, only the vibration signal at the intermediate shaft after the flywheel is obtained. The signal is obtained through multiple simulations at steady states with a $1RPM$ mean speed difference between them for various engine conditions. Moreover, in order to avoid aliasing errors and satisfy the Nyquist frequency theorem, the simulation's sample rate is to 100 Hz, since the maximum engine speed is $n = 90\ rpm\ or\ n = 1.5\ Hz$ and thus the highest order of excitation(20) has a frequency of 30 Hz. For the various conditions the signals are classified as described in the following scenarios in order to be used as classes.

- Intact condition with 0% misfire for all cylinders

Table 4: Intact condition scenarios

Cylinder No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Scenario 1 (misfire %)	**2**	**1**	**0**	**0**	**3**	**0**	**1**	**0**	**3**	**2**
Scenario 2 (misfire %)	**1**	**0**	**3**	**2**	**0**	**2**	**0**	**1**	**2**	**1**
Scenario 3 (misfire %)	**0**	**2**	**1**	**3**	**0**	**2**	**0**	**2**	**0**	**2**
Scenario 4 (misfire %)	**2**	**2**	**3**	**0**	**1**	**2**	**0**	**1**	**0**	**0**

- Four scenarios of intact condition with a range of $0 - 3\%$ misfire for various cylinders

All the signals obtained from the aforementioned simulations have a recorded duration of 25 seconds and each of them is further divided into five equal time duration signals in order to obtain more samples for the machine learning algorithms. Then for each signal the mean engine speed, the peak amplitude, standard deviation and variance are calculated. In addition to these, the frequency domain features calculated with the one-third octave analysis are included to the dataset.

7.3.2 Algorithm Initial Evaluation Results-Step No.1

At this point the selected algorithms will be evaluated based on the methods mentioned later at this section and for each of the previously described threes steps. At the bellow figures the correlation between all features, as well as the correlation of all features with the class *Cylinder Condition(0-1)* are presented respectively.

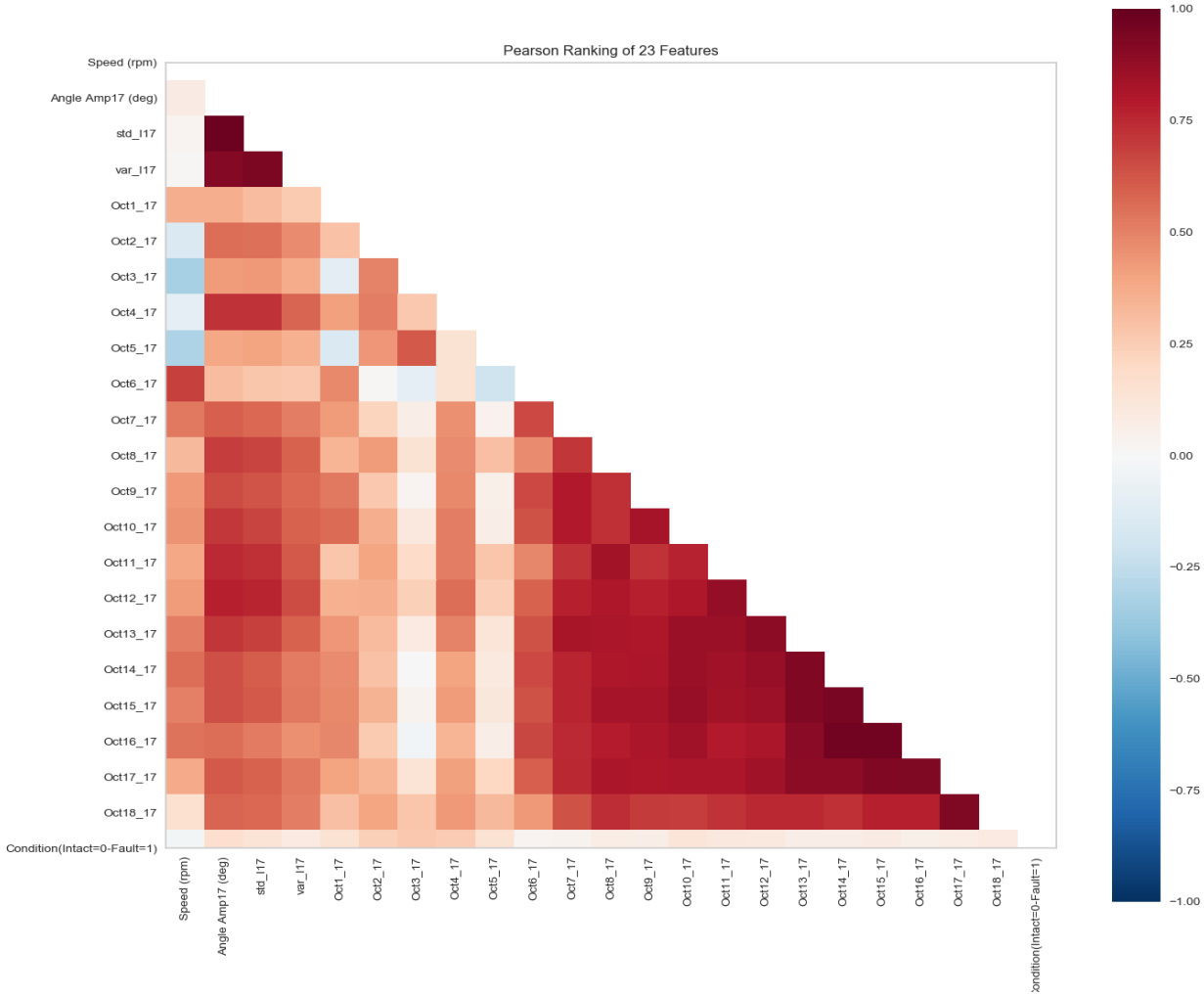

Figure 21: Correlation heatmap between features

Each coloured square illustrates the correlation between the variables on each axis. Dark colour indicates strong correlation with values ranging from -1 to $+1$. A zero correlation value indicates there is no linear trend between two variables. Correlation values close to $+1$ show a positive correlation so that if one feature increases so does the other. Instead correlation values close to -1 show a negative correlation so that if one feature increases the other will decrease. Figure 21 visualizes the strong correlation between all selected features and the correlation of all features with the output class *condition*.

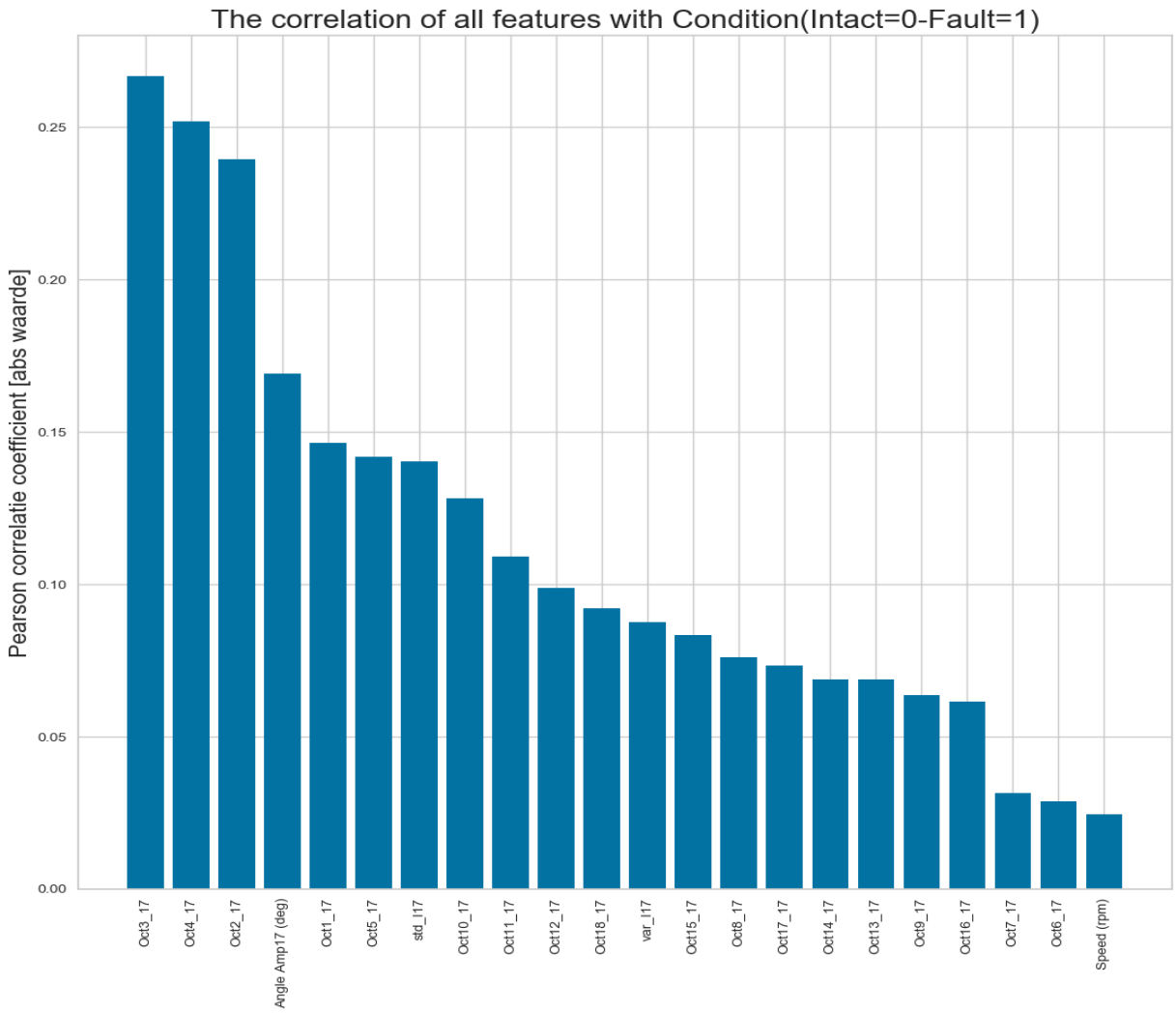

Figure 22: Correlation heatmap of features with class

Figure 22 visualizes the correlation using the Pearson's Correlation Coefficient as before, but ranking the correlation of all features with the output class *condition*. As discussed previously the value of Pearson's Correlation Coefficient can be between -1 and $+1$. In figure 22 the mean speed value has the least correlation and the addition of that feature may interfere with accuracy. In general, when dealing with large datasets with many features, non-correlated features should be ignored, in order to avoid noise and increased complexity of the system. The most important feature in this step is the third one third Octave. Lastly, for a dataset as that of this case study a small coefficient it's not necessarily bad, since the dataset might have a large statistically significant correlation and further investigation is needed.

- **Classification report**

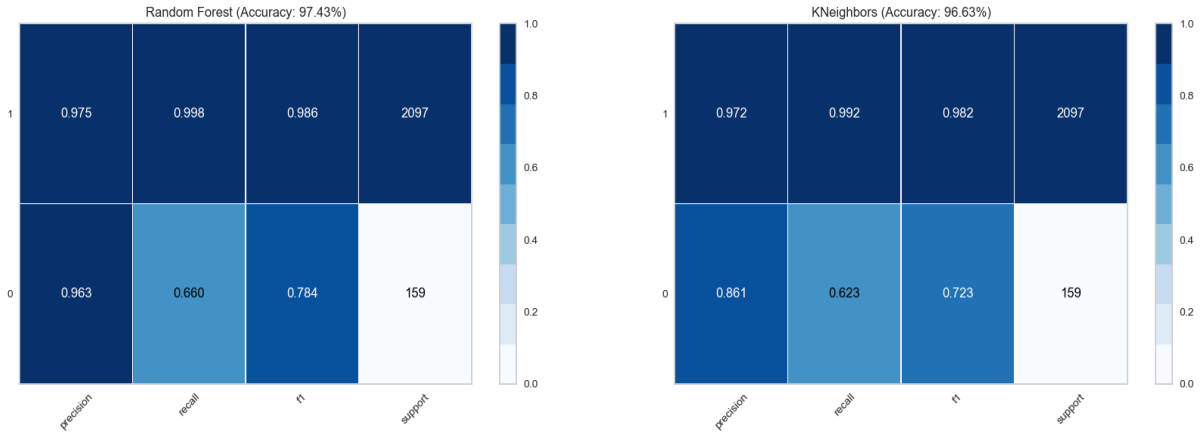

Figure 23: Random forest and KNNeighbors classification report for step No.1

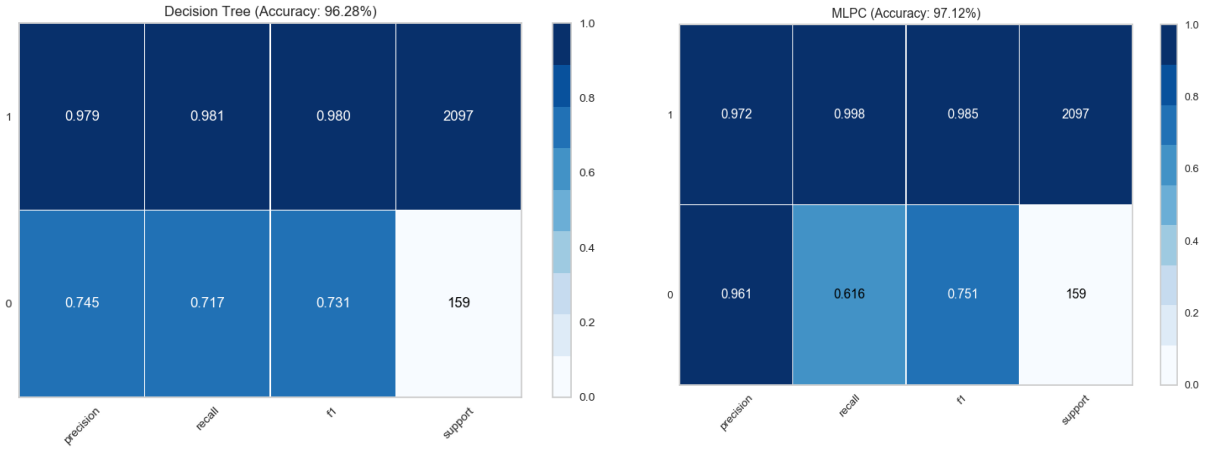

Figure 24: Decision tree and Multi-layer Perceptron classification report for step No.1

A high accuracy score is displayed for all algorithms. In addition, the precision score is high but the recall seems to be the one lowering the F1 score. That means that all algorithms have classified correctly most of their prediction but don't perform as well at finding all positive instances of the class.

- **Confusion Matrix**

Figure 25: Random forest and KNNeighbors confusion matrices for step No.1

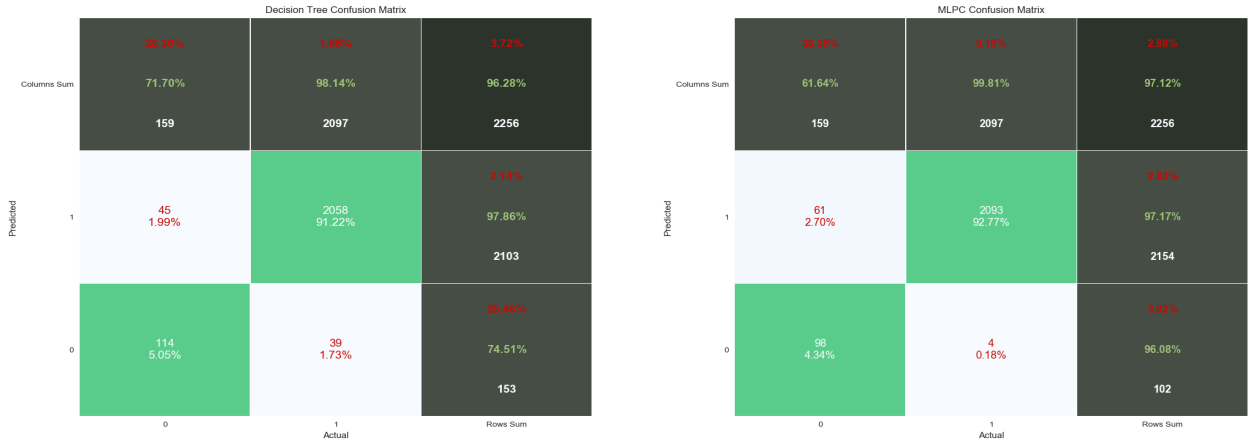

Figure 26: Decision tree and Multi-layer Perceptron confusion matrices for step No.1

The confusion matrix confirms the high precision and lower recall scores. Lower recall score is due to algorithms tendency to classify samples as fault condition, which is due to the class imbalance.

- **Learning curves**

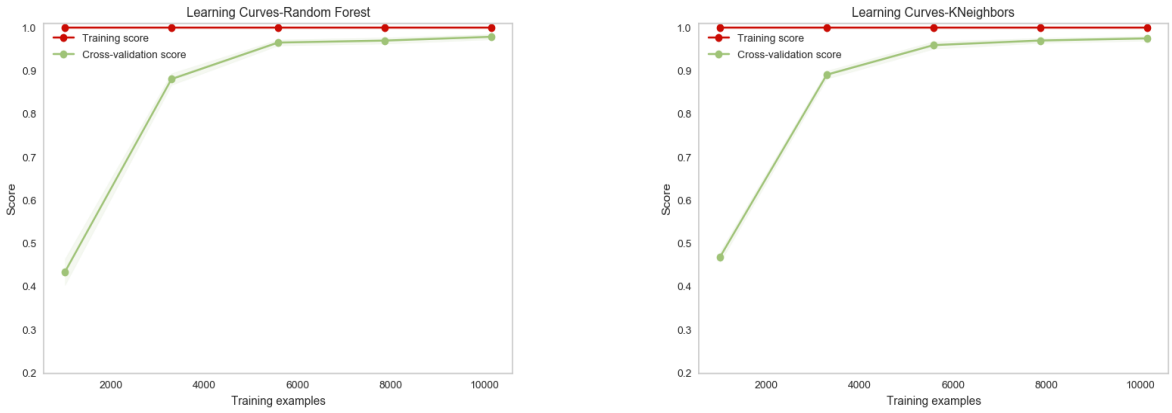

Figure 27: Random forest and KNNeighbors learning curves for step No.1

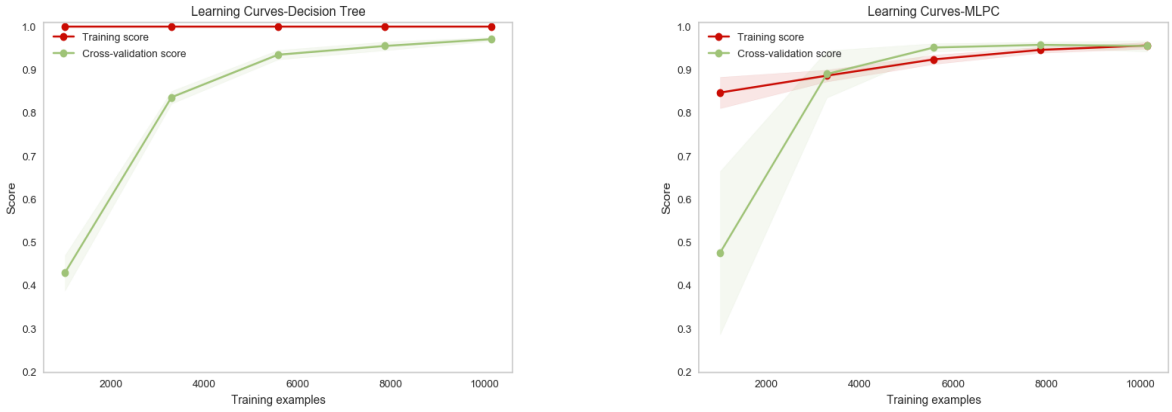

Figure 28: Decision tree and Multi-layer Perceptron learning curves for step No.1

Figures 27 and 28 show, that the benefit from adding more training data to the current dataset is insignificant. The accuracy is already high so a increase in training samples would result in a insignificant increase of algorithms' score.

7.3.3 Parameter Optimization-Step No.1

At this point the most important parameters of the algorithms are found using the GridsearchCV module of python, in order to maximize the score of the model for the specified feature space. The optimization is done by splitting dataset to folds and their evaluation for the different values of the parameters is based on the accuracy score. The influence that a single hyperparameter can have on the training and validation score is presented at the bellow figure for the random forest algorithm and for various values of the parameter *max_features*, which are the maximum number of features the algorithm is allowed to try in individual tree.

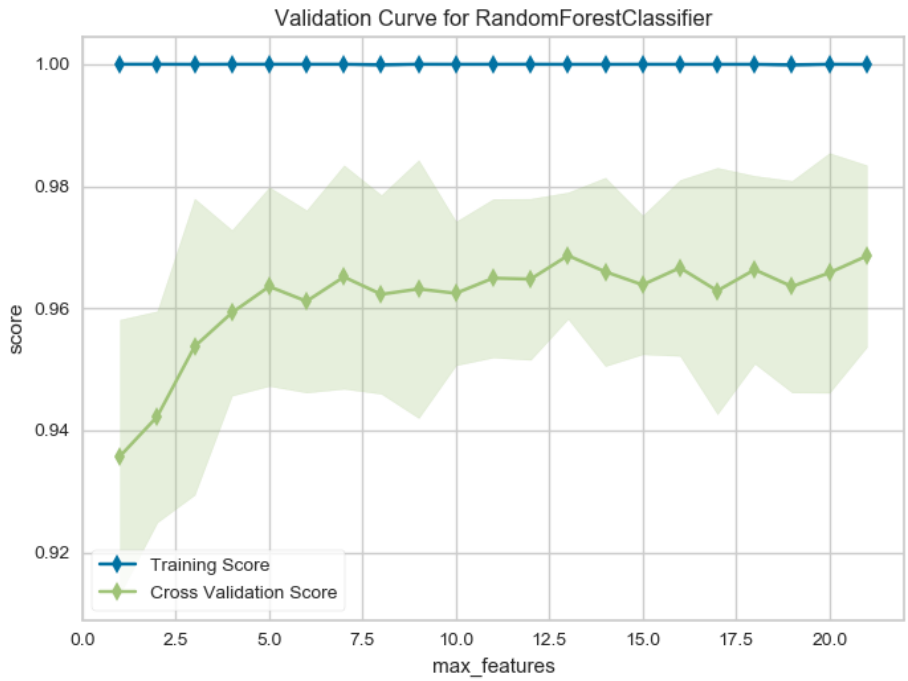

Figure 29: Example of random forest hyperparameter tuning for step No.1

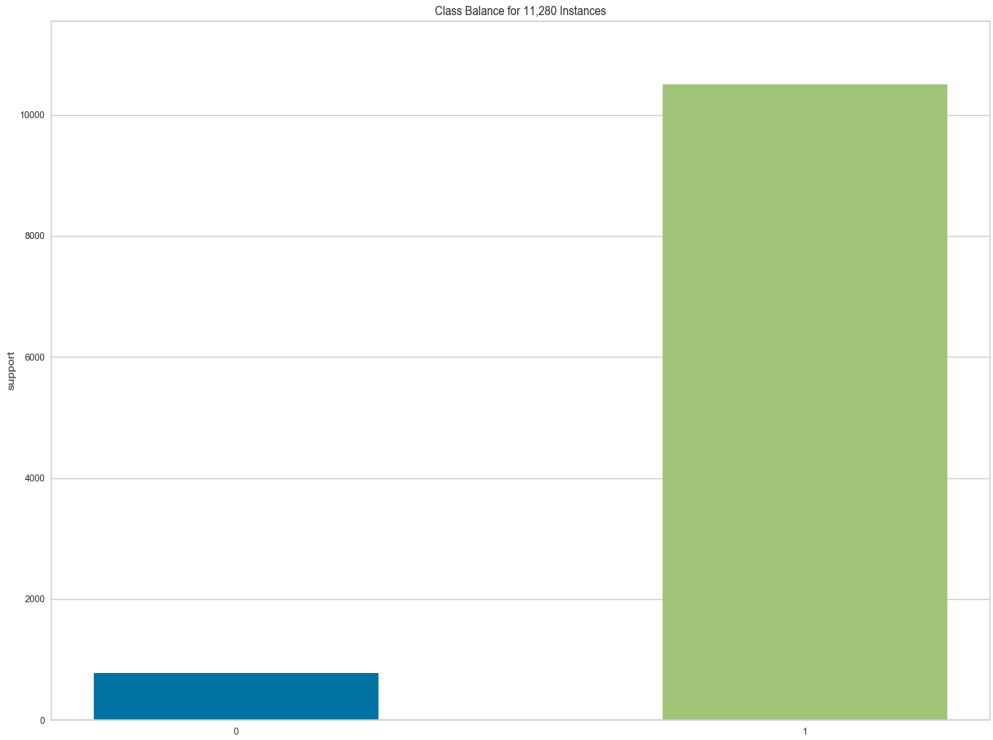

Figure 30: Class balance for step No.1

From figure 29 it's concluded that when selecting a small value for *max_features*, the training and validation score are both low and the model will be underfitting. Instead, when selecting a higher value for *max_features*, the training score is high and the validation score is lower, but not too low so that the model is overfitting. In addition, the parameter *class_weight* will be used, due to the fact that the classes are unbalanced, figure 30, and otherwise model would be biased. Moreover, all of the initial selected features are considered important for the algorithms' performance. figure 31.

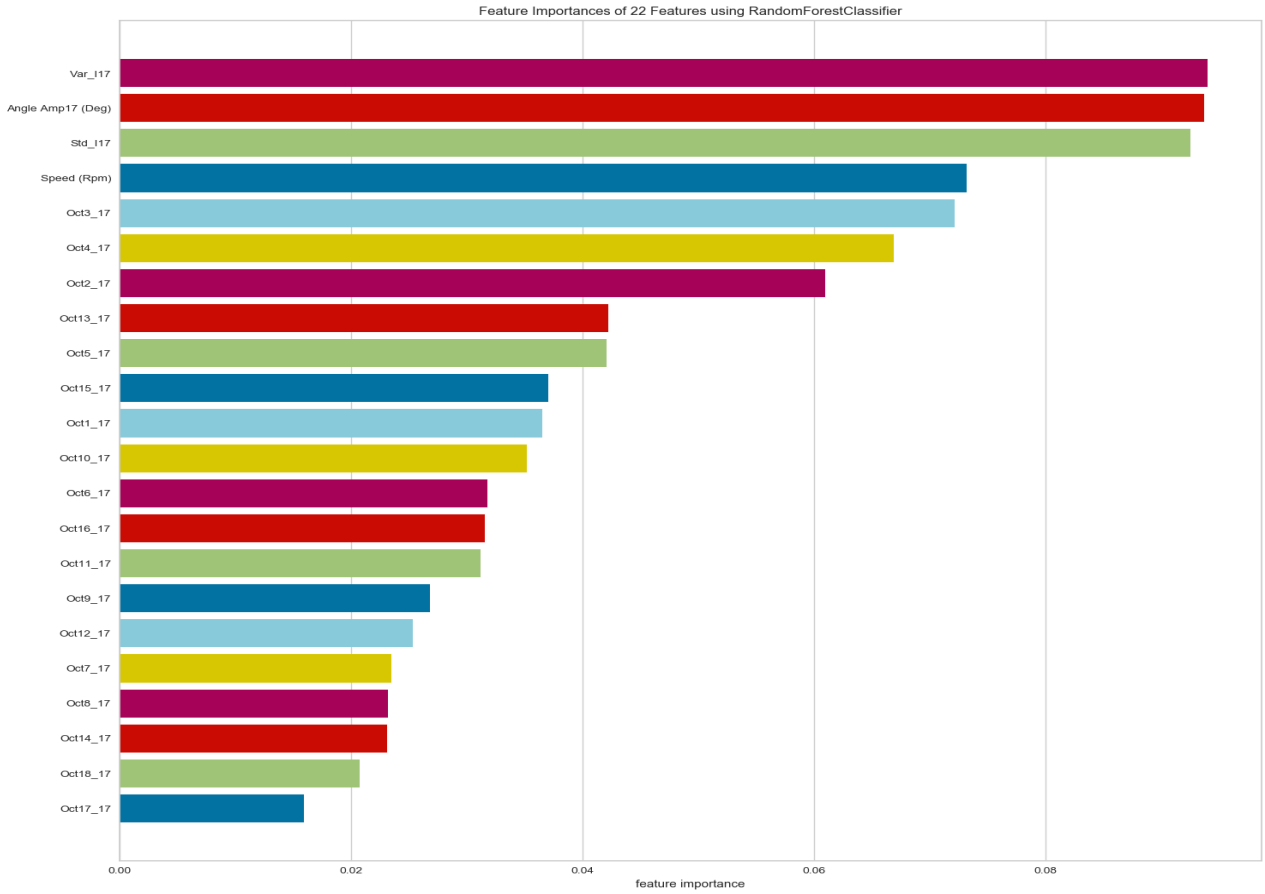

Figure 31: Feature importance for step No.1

Figure 31 illustrates the feature importances for the random forest classifier. At this point all algorithms were tested in order to find the most important features. These features that were later used for the training dataset did not provide any significant improvement to the overall performance. Also, based on the fact that this procedure is computationally expensive, the initial dataset including all features is used instead, which provides equal performance.

7.3.4 Algorithm Evaluation Results After Parameter Tuning-Step No.1

The evaluation for step No.1 will be conducted again including the parameter optimizations.

- **Classification report**

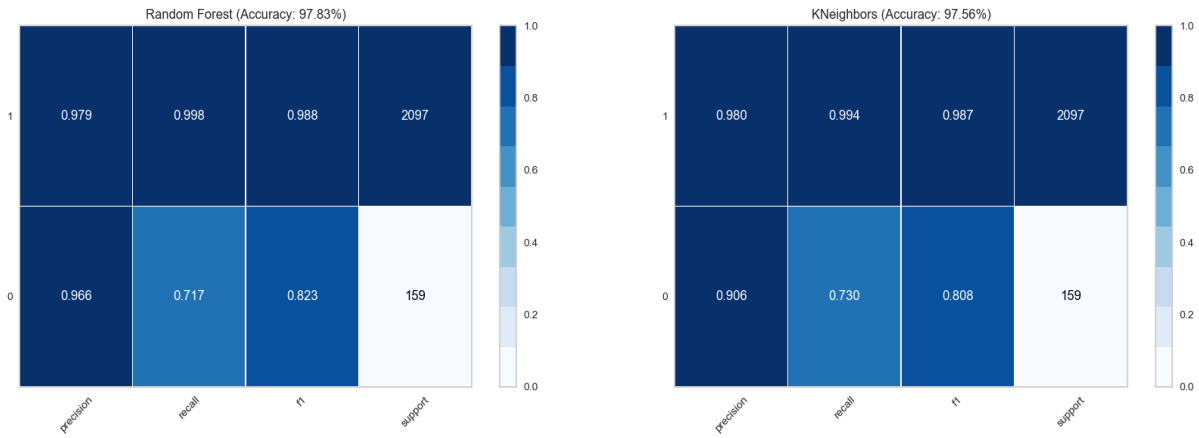

Figure 32: Random forest and KNNeighbors classification report for step No.1

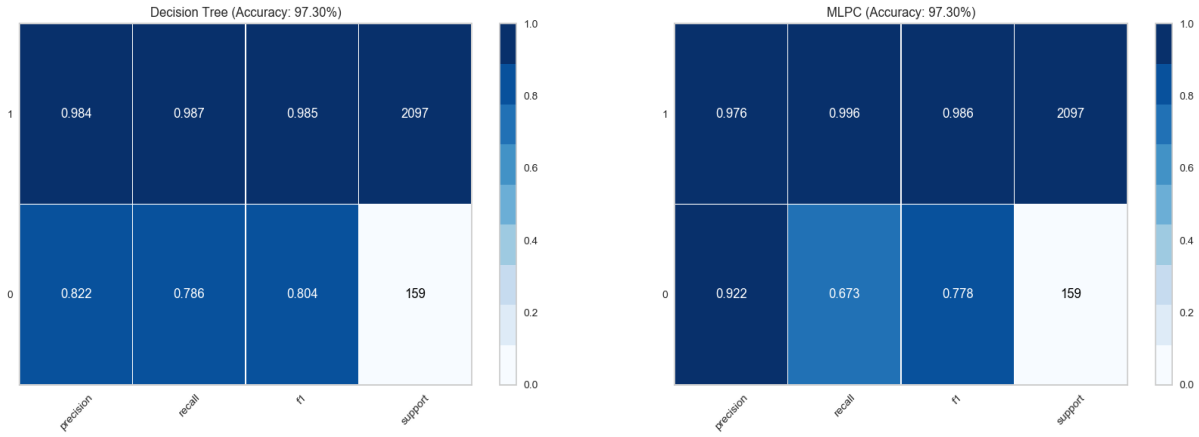

Figure 33: Decision tree and Multi-layer Perceptron classification report for step No.1

A high accuracy score is displayed for all algorithms. In addition, the precision score is high, but as before the recall seems to be the one lowering the F1 score. That means that all algorithms have classify correctly most of their prediction but don't perform as well at finding all positive instances of the class. Despite that, the accuracy and the recall have increased after the parameter optimization.

- **Confusion Matrix**

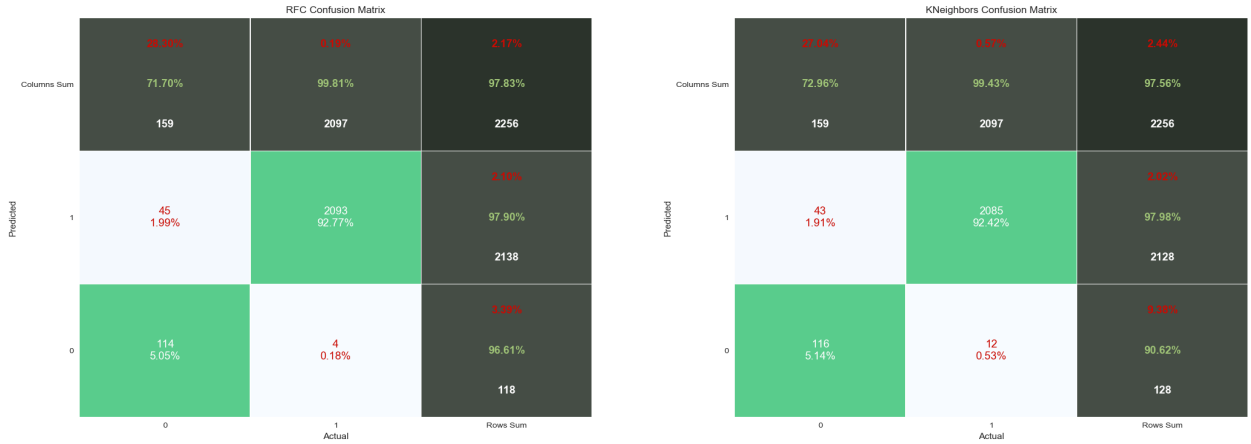

Figure 34: Random forest and KNNeighbors confusion matrices for step No.1

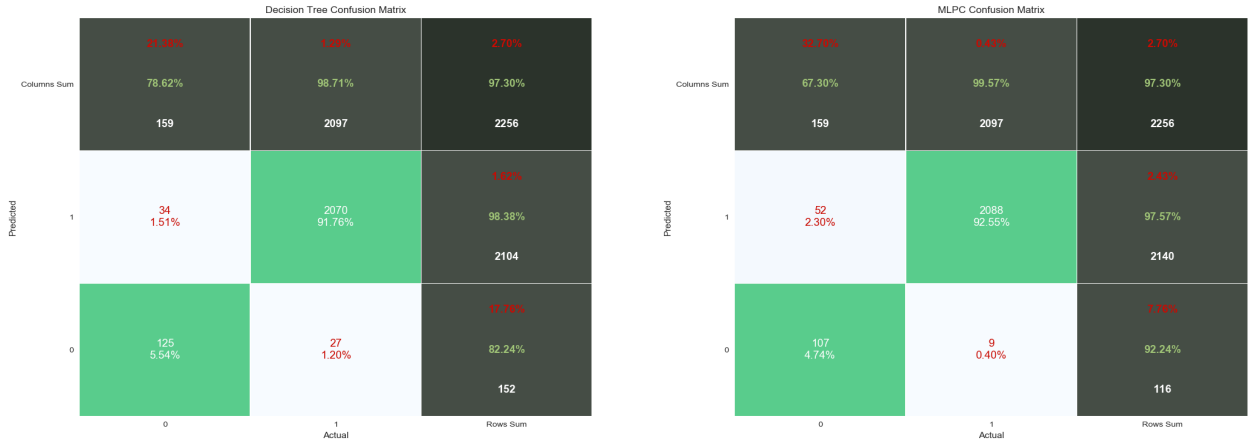

Figure 35: Decision tree and Multi-layer Perceptron confusion matrices for step No.1

The confusion matrix confirms the high precision and lower recall scores. Lower recall score is due to algorithms tendency to classify samples as fault condition, which is due to the class imbalance. After the parameter optimization all algorithms seem to be less effected from the class imbalance and show less confusion.

- **Learning curves**

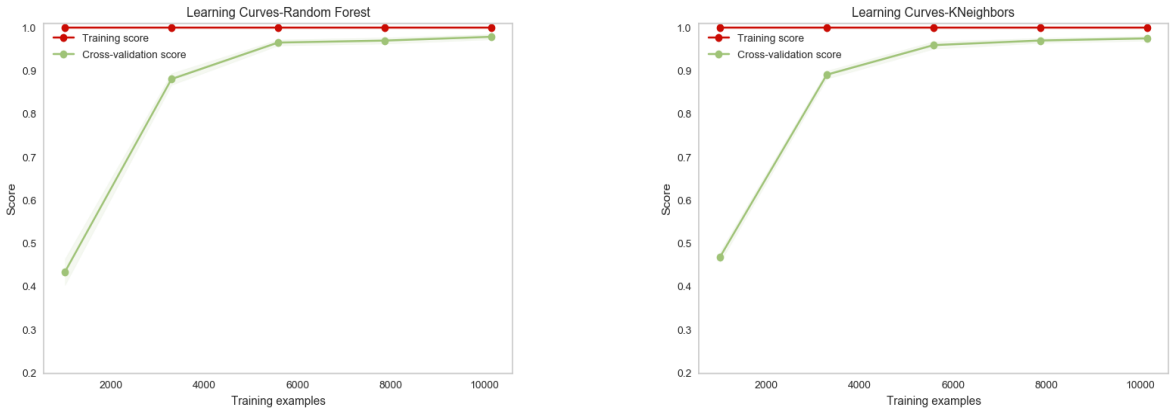

Figure 36: Random forest and KNNeighbors learning curves for step No.1

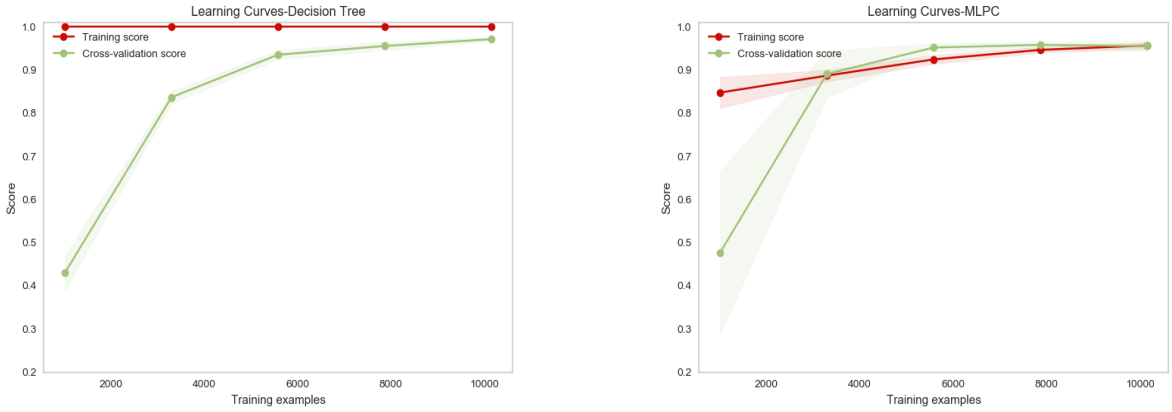

Figure 37: Decision tree and Multi-layer Perceptron learning curves for step No.1

Figures 36 and 37 show, that the benefit from adding more training data to the current dataset is insignificant. The accuracy is already high so a increase in training samples would result in a insignificant increase of algorithms' score. Parameter optimization didn't have a notable effect on the score of learning curves.

7.3.5 Algorithm Initial Evaluation Results-Step No.2

The same evaluation method is used for the selected algorithms for step no.2 which is the fault location problem. At the bellow figures the correlation between all features, as well as the correlation of all features with the class *Cylinder* are presented respectively.

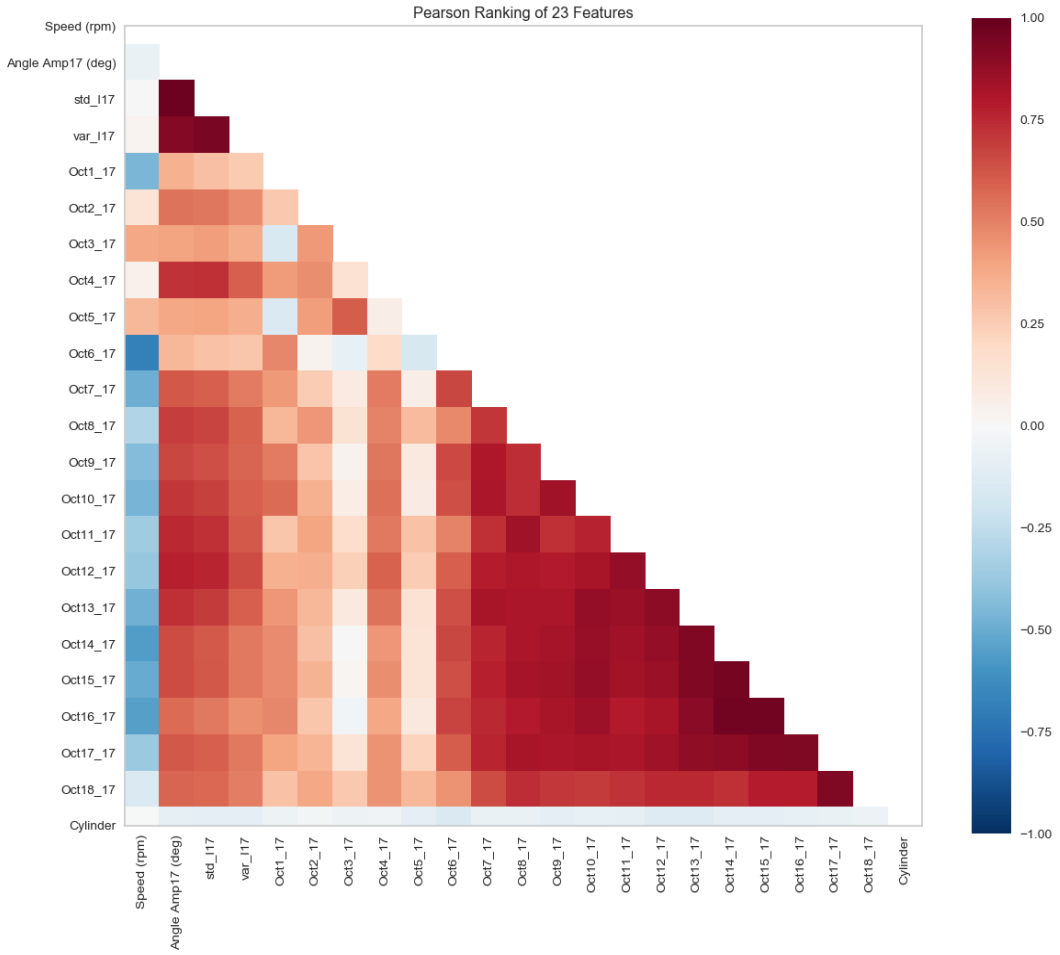

Figure 38: Correlation heatmap between features

Figure 38 visualizes the strong correlation between all selected features and the correlation of all features with the output class *cylinder*.

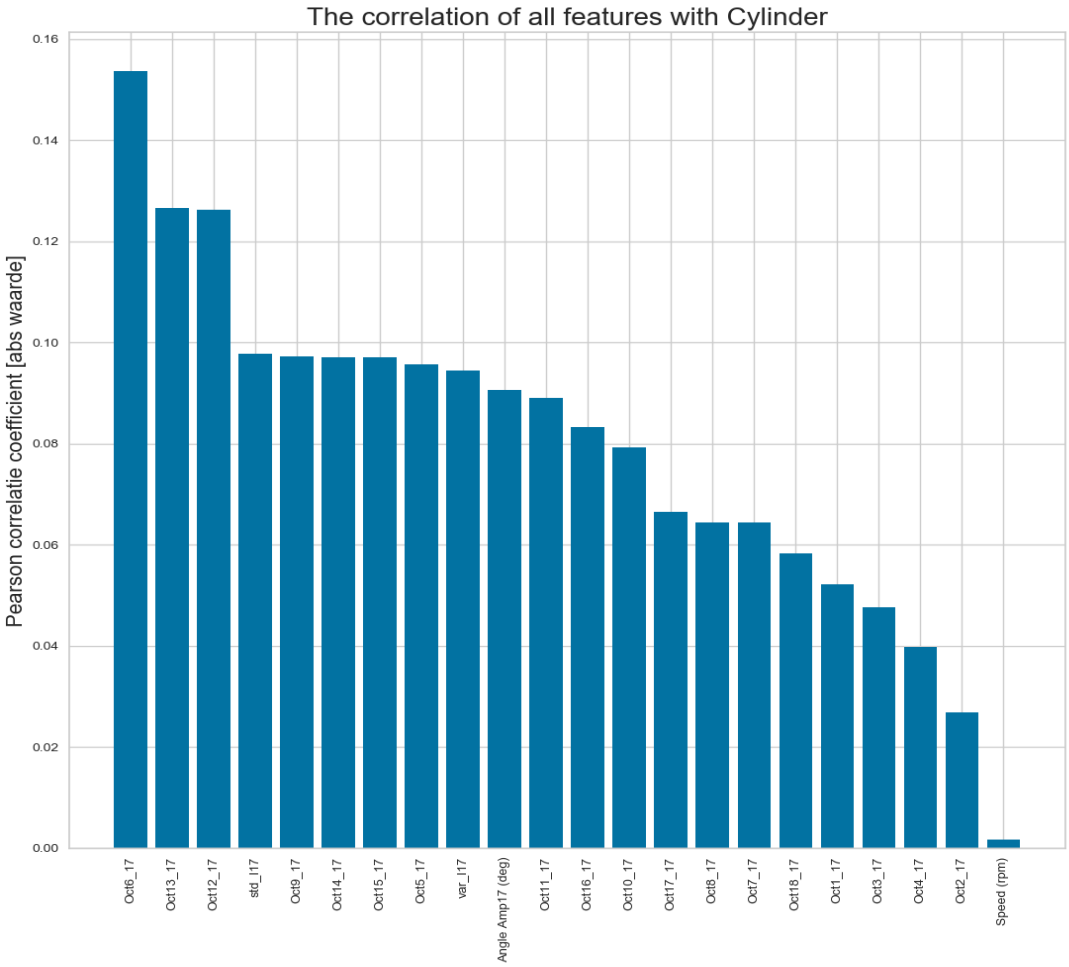

Figure 39: Correlation heatmap of features with class

Figure 39 visualizes the correlation using the Pearson's Correlation Coefficient as before, ranking the correlation of all features with the output class *cylinder*. In figure 39 the mean speed value has the least correlation and the addition of that feature may interfere with accuracy. In general, when dealing with large datasets with many features, non-correlated features should be ignored, in order to avoid noise and increased complexity of the system. The most important feature in this step is the sixth one third Octave. Lastly, for a dataset as that of this case study a small coefficient it's not necessarily bad, since the dataset might have a large statistically significant correlation and further investigation is needed.

- **Classification report** A low accuracy score is displayed for all algorithms, with

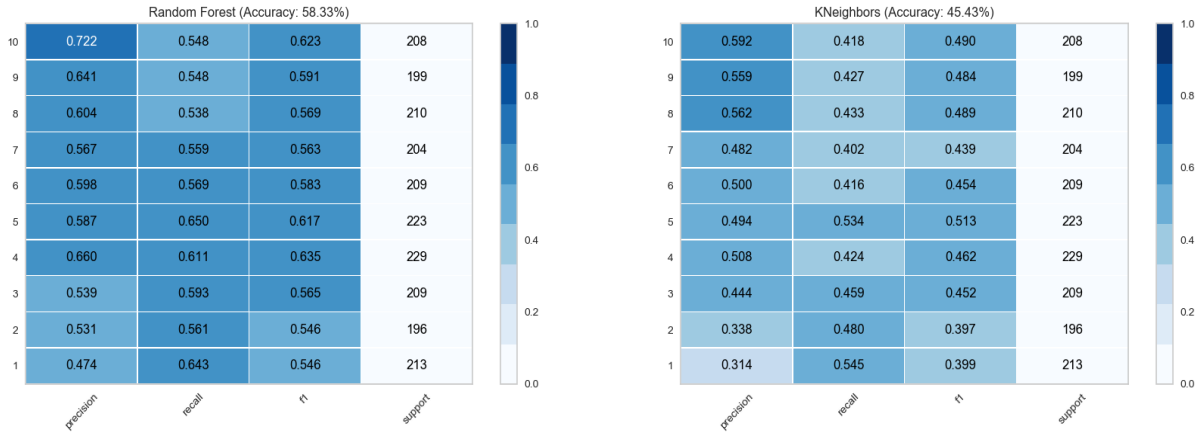

Figure 40: Random forest and KNNeighbors classification report for step No.2

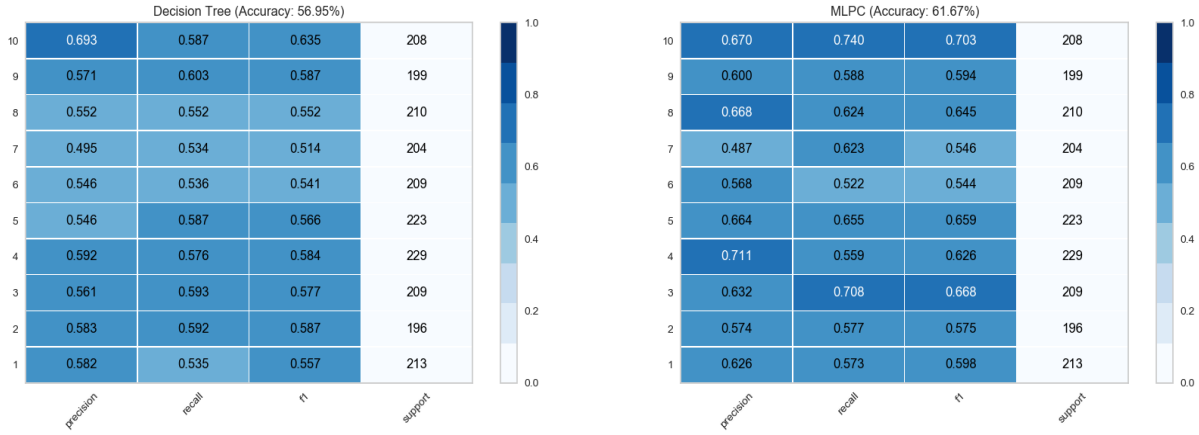

Figure 41: Decision tree and Multi-layer Perceptron classification report for step No.2

KNeighbors showing the worst performance and MLPC the best. In addition, the precision score is close to the recall score. At this point the algorithms' performance is not acceptable leading to further optimizations.

- **Confusion Matrix**

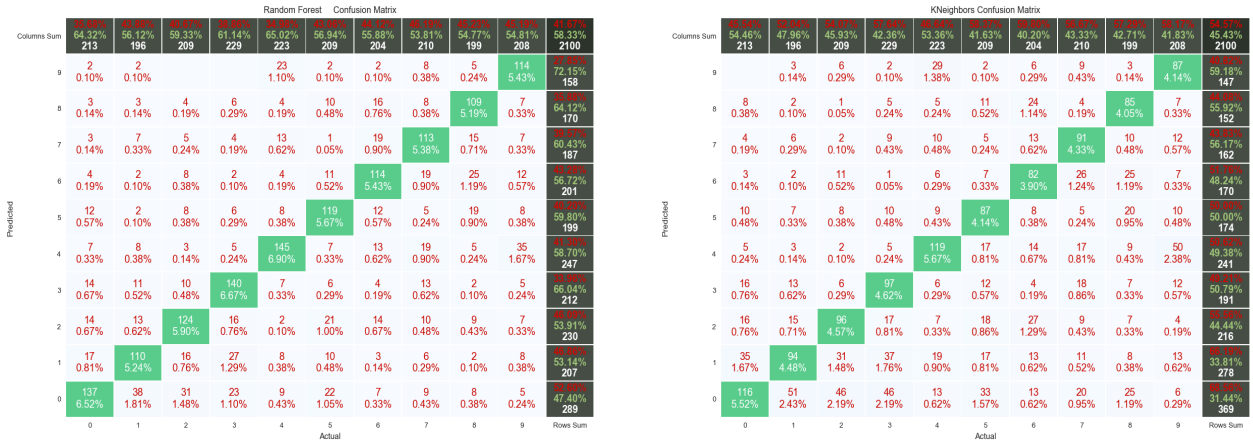

Figure 42: Random forest and KNNeighbors confusion matrices for step No.2

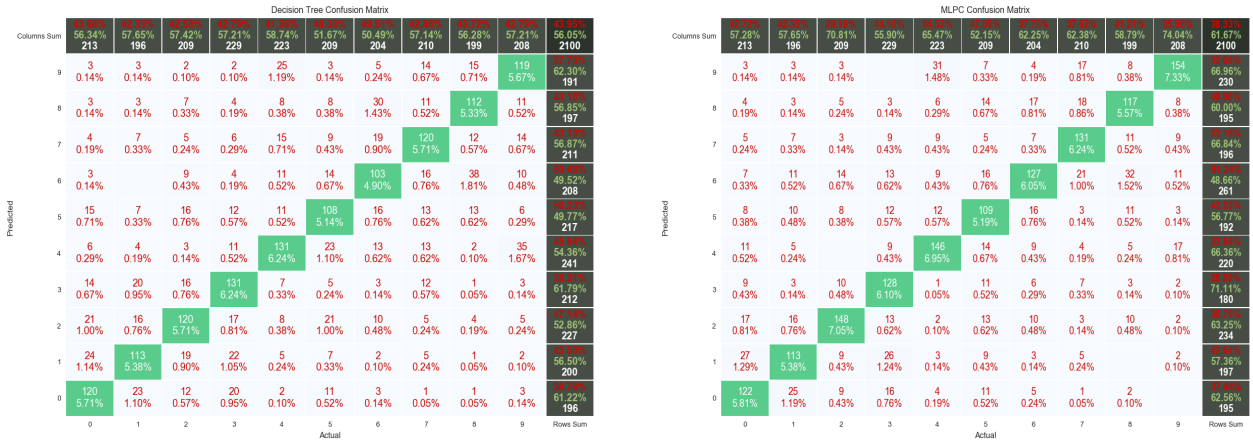

Figure 43: Decision tree and Multi-layer Perceptron confusion matrices for step No.2

The confusion matrix confirms the algorithms' low performance, showing the difficulty to distinguish the possible fault location. It is worth mentioning that the algorithms confuse the true fault location as a fault to other cylinders which are close rather than cylinder further from the true one.

- **Learning curves**

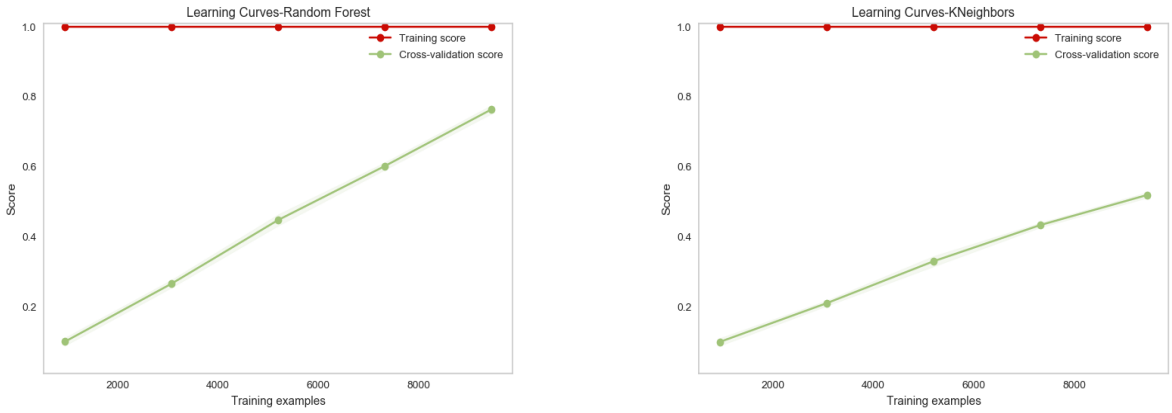

Figure 44: Random forest and KNNeighbors learning curves for step No.2

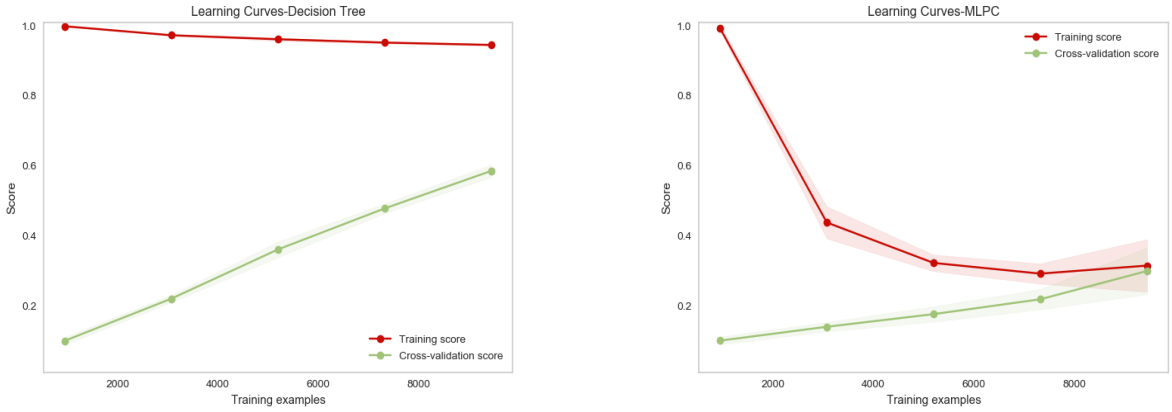

Figure 45: Decision tree and Multi-layer Perceptron learning curves for step No.2

Figures 44 and 45 show, that algorithms would benefit from adding more training data to the current dataset. Thus a higher accuracy could be achieved by an increase in training samples. Moreover, the lower training score of MLPC could also improve by increasing the number of training samples.

7.3.6 Algorithm Evaluation Results After Parameter Tuning-Step No.2

The evaluation for step No.2 will be conducted again including the parameter optimizations as with step no.1.

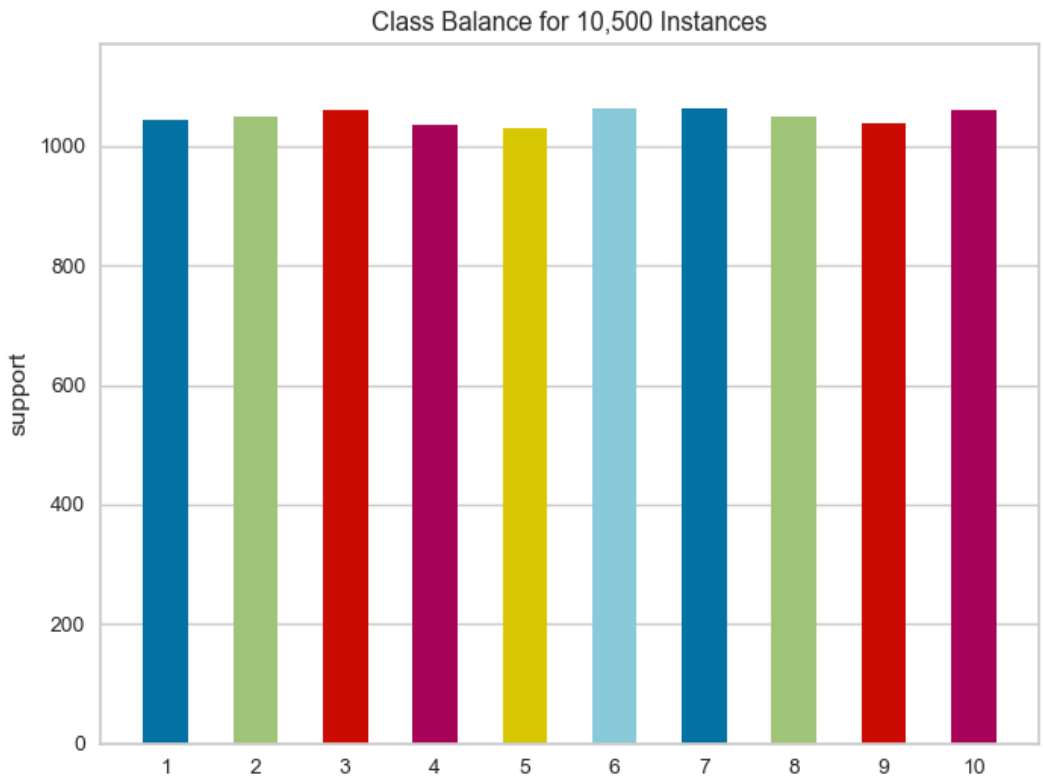

Figure 46: Class balance for step No.2

From figure 46 is concluded that at this step there is no problem due to class imbalances.

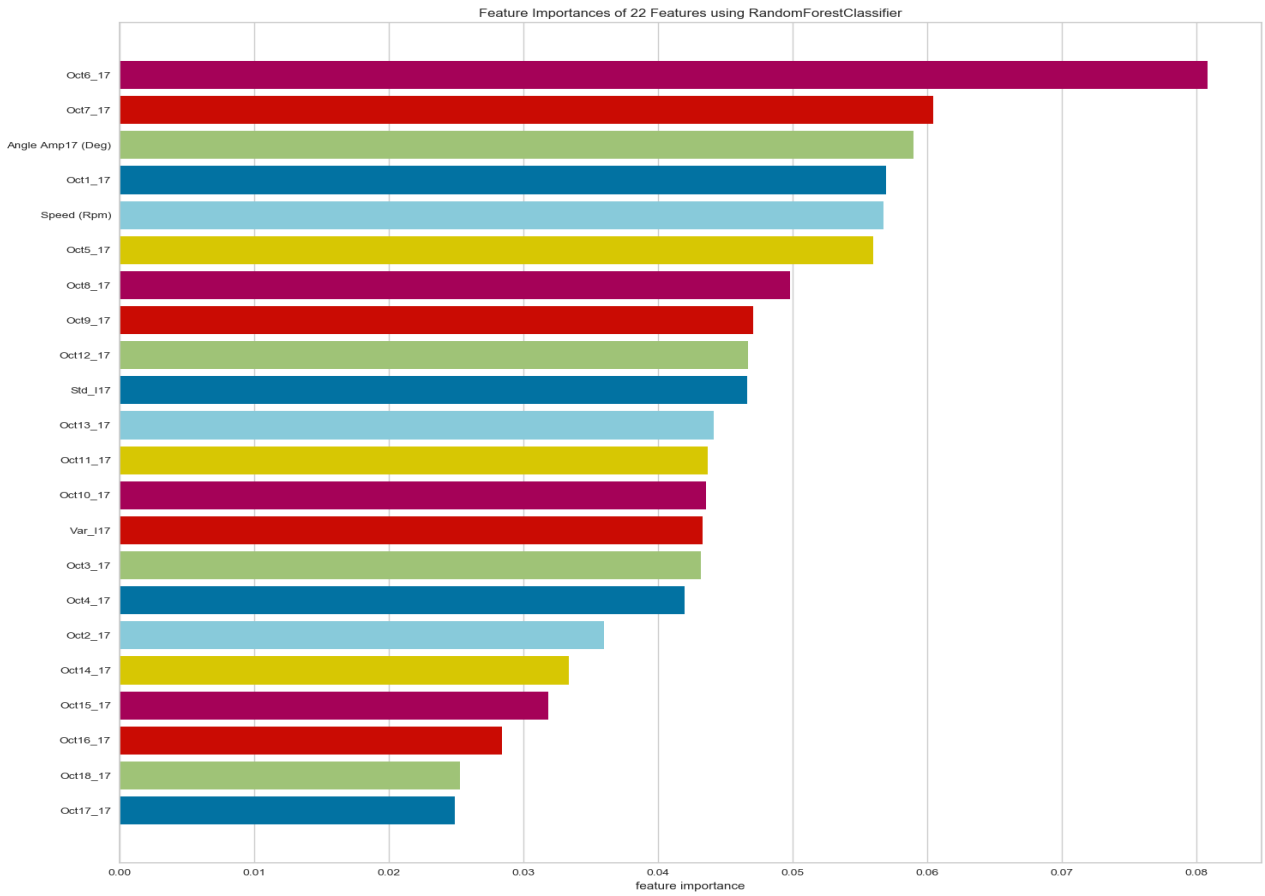

Figure 47: Feature importance for step No.2

Figure 47 illustrates the feature importances for the random forest classifier. At this point all algorithms were tested in order to find the most important features. These features that were later used for the training dataset did not provide any significant improvement to the overall performance. Also, based on the fact that this procedure is computationally expensive, the initial dataset including all features is used instead, which provides equal performance.

- **Classification report**

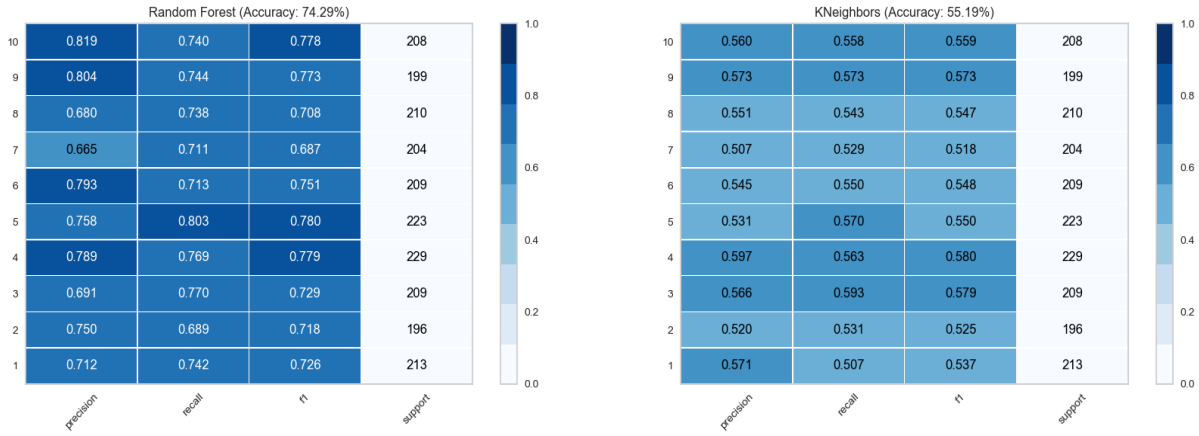

Figure 48: Random forest and KNNeighbors classification report for step No.2

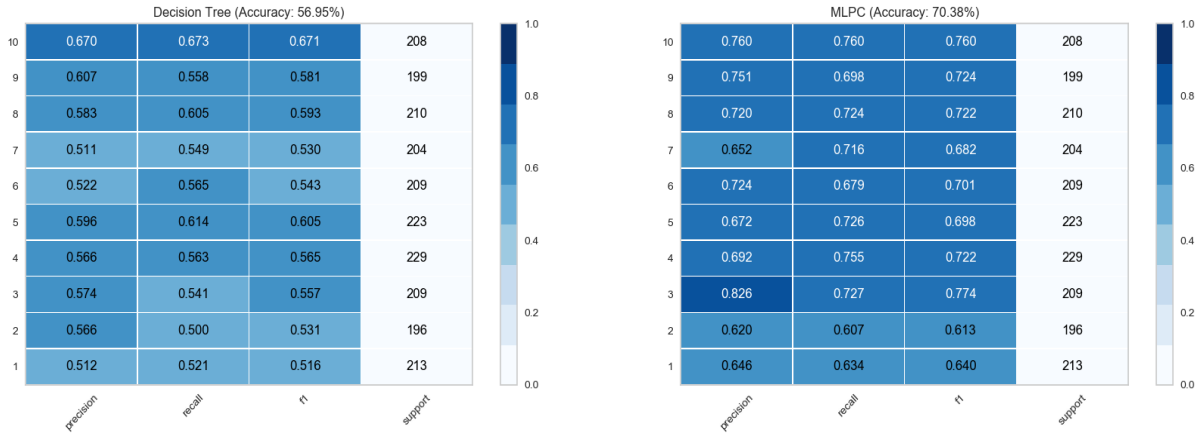

Figure 49: Decision tree and Multi-layer Perceptron classification report for step No.2

Random forest and MLPC show an acceptable accuracy score for this step. In contrast the performance of the decision tree and of the KNeighbors is not satisfying even after the parameter optimization. In addition, the precision score is close to the recall score and all algorithms improved after the parameter optimization.

- **Confusion Matrix**

Figure 50: Random forest and KNNeighbors confusion matrices for step No.2

Figure 51: Decision tree and Multi-layer Perceptron confusion matrices for step No.2

After the parameter optimization all algorithms seem to perform better on distinguishing between the different classes.

- **Learning curves**

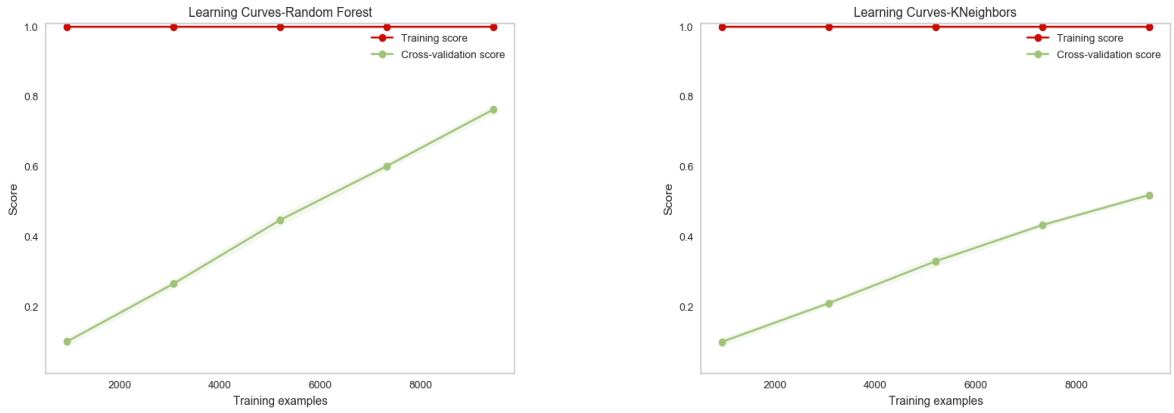

Figure 52: Random forest and KNNeighbors learning curves for step No.2

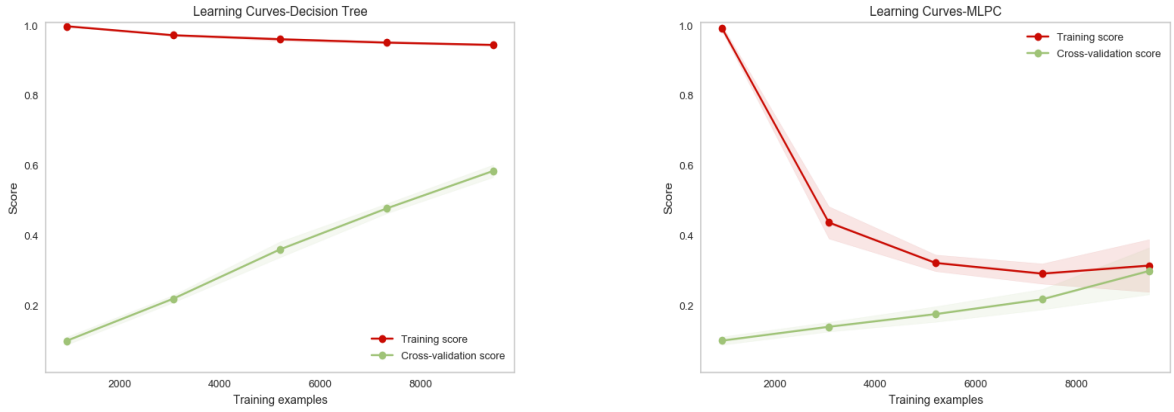

Figure 53: Decision tree and Multi-layer Perceptron learning curves for step No.2

Parameter optimization didn't have a notable effect on the score of learning curves. Figures 52 and 53 show, that algorithms would benefit from adding more training data to the current dataset. Thus a higher accuracy could be achieved by an increase in training samples. Moreover, the lower training score of MLPC could also improve by increasing the number of training samples.

7.3.7 Algorithm Initial Evaluation Results-Step No.3

As before same evaluation method is used for the selected algorithms for step no.3 which is the fault severity problem. At the bellow figures the correlation between all features, as well as the correlation of all features with the class *misfire* are presented respectively.

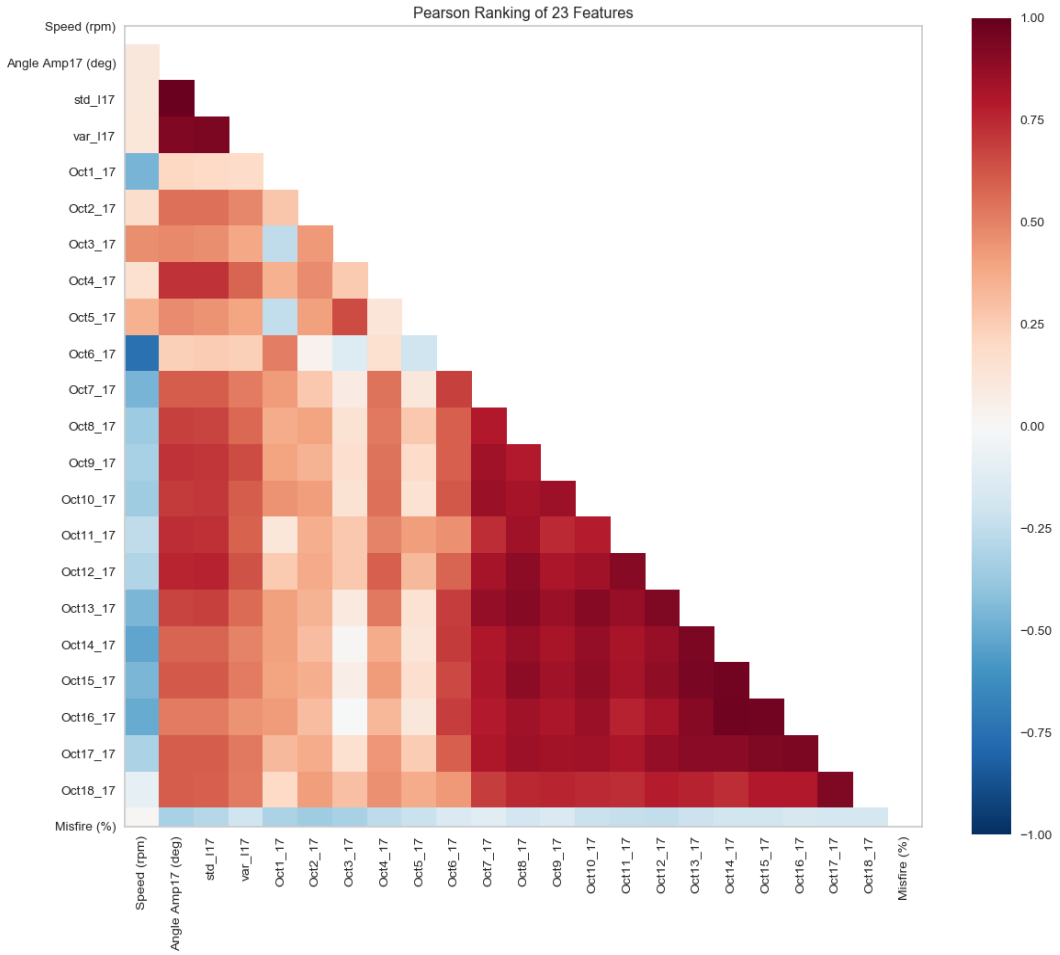

Figure 54: Correlation heatmap between features

Figure 54 visualizes the strong correlation between all selected features and the correlation of all features with the output class *misfire*.

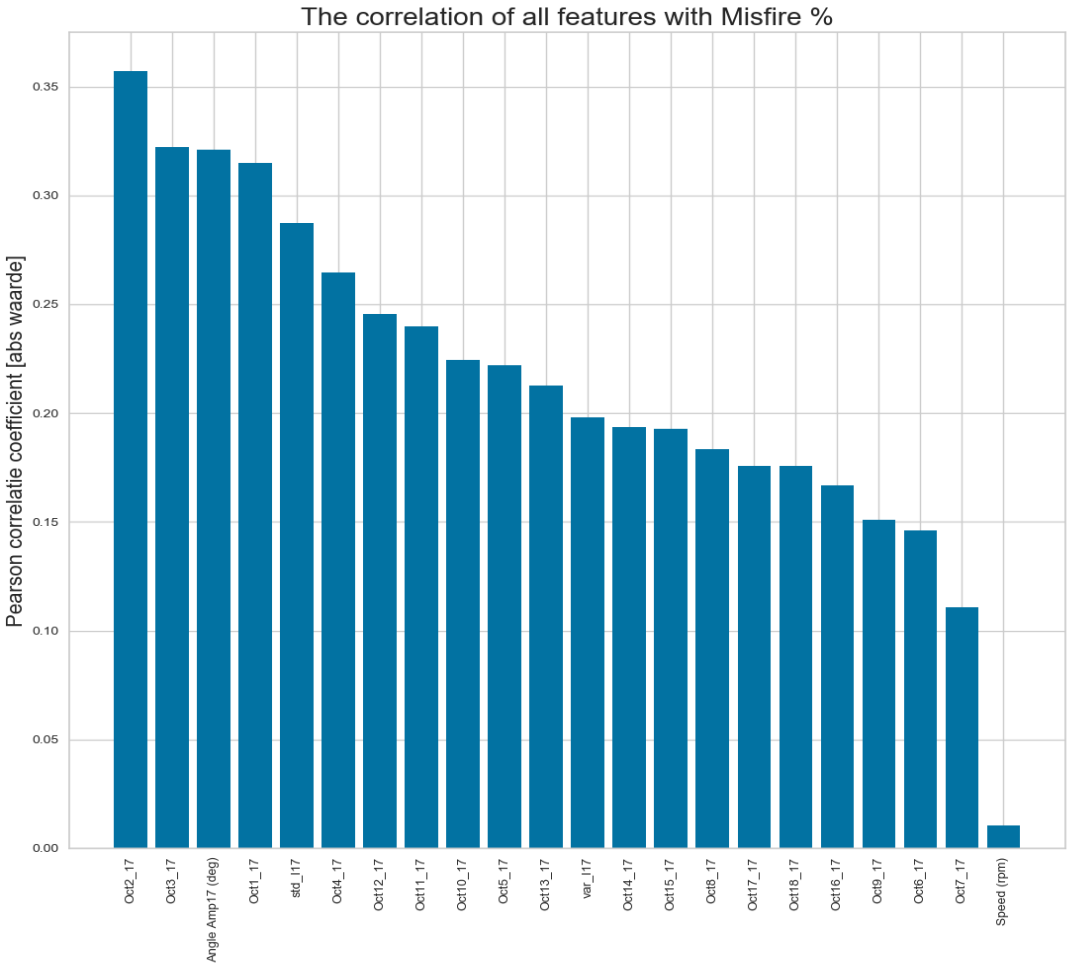

Figure 55: Correlation heatmap of features with class

Figure 55 visualizes the correlation using the Pearson's Correlation Coefficient as before, ranking the correlation of all features with the output class *cylinder*. In figure 55 the mean speed value has the least correlation and the addition of that feature may interfere with accuracy. In general, when dealing with large datasets with many features, non-correlated features should be ignored, in order to avoid noise and increased complexity of the system. The most important feature in this step is the second one third Octave. Lastly, for a dataset as that of this case study a small coefficient it's not necessarily bad, since the dataset might have a large statistically significant correlation and further investigation is needed.

- **Classification report**

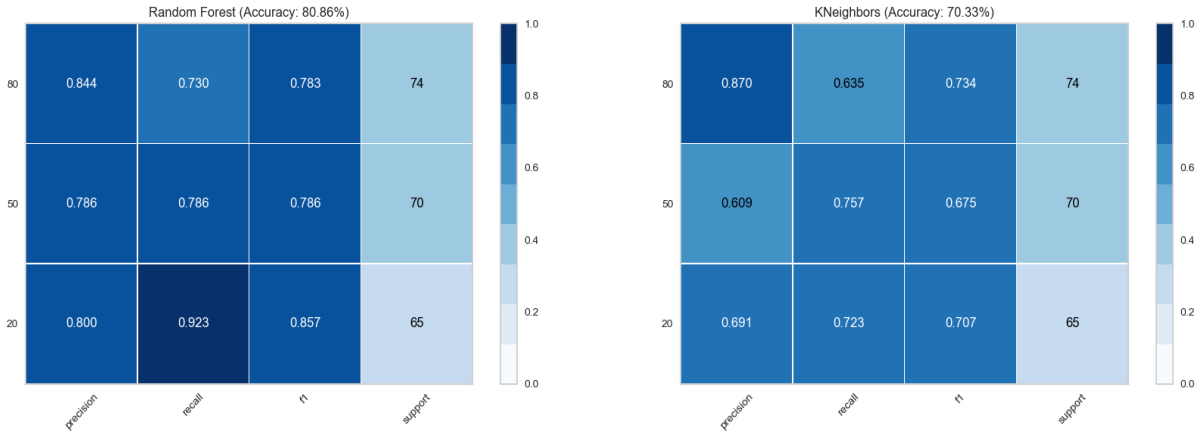

Figure 56: Random forest and KNNeighbors classification report for step No.3

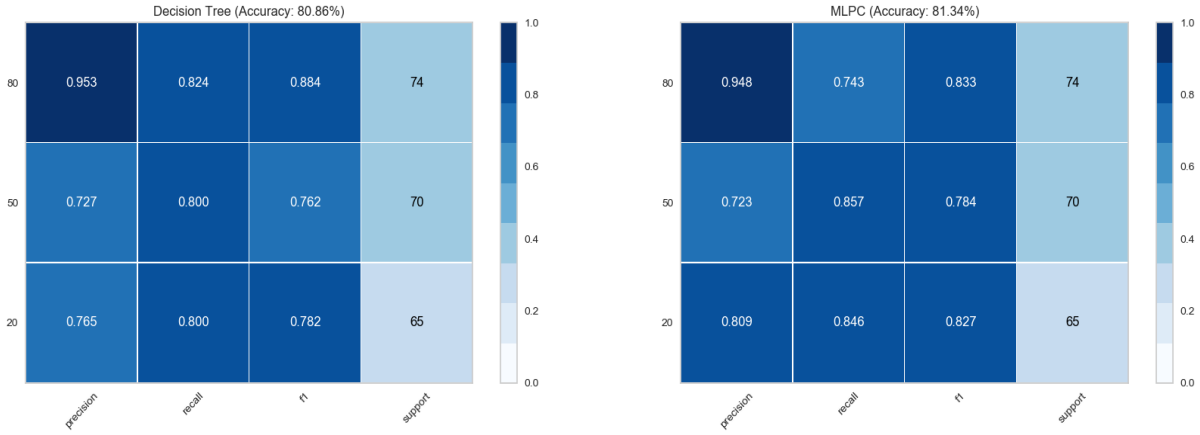

Figure 57: Decision tree and Multi-layer Perceptron classification report for step No.3

An acceptable score is displayed for all algorithms at this step.

- **Confusion Matrix**

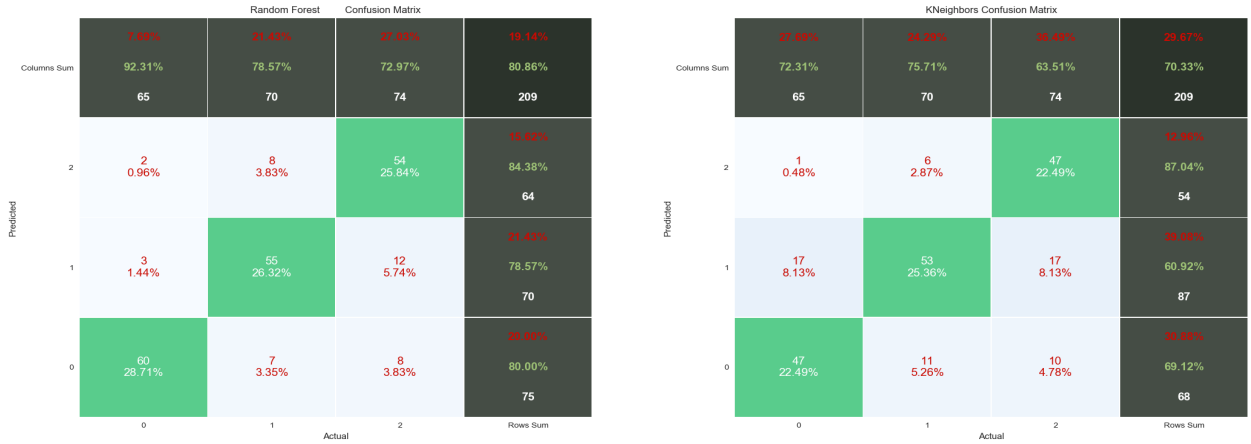

Figure 58: Random forest and KNNeighbors confusion matrices for step No.3

Figure 59: Decision tree and Multi-layer Perceptron confusion matrices for step No.3

The confusion matrix confirms the algorithms' good performance, showing the adequate ability to distinguish the possible misfire percentage. It is worth mentioning that the algorithms confuse the true misfire percentage as a misfire percentage closer to the true value.

- **Learning curves**

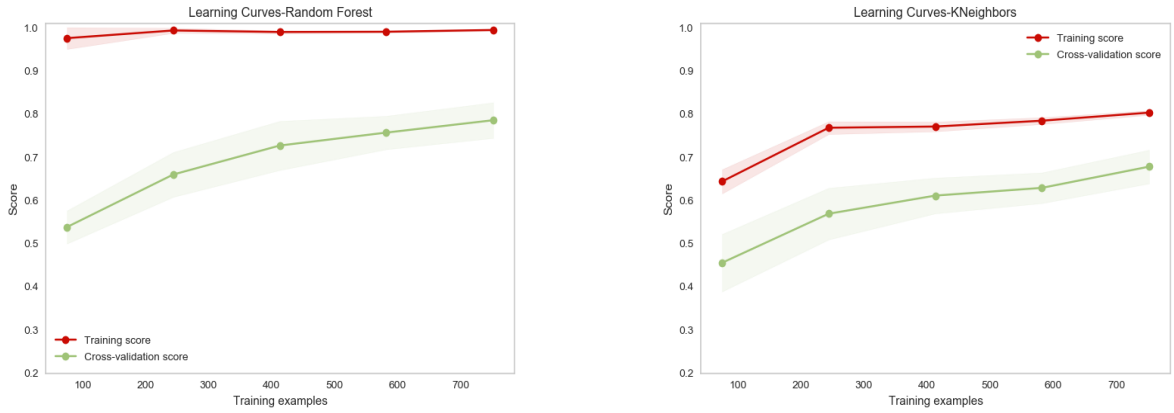

Figure 60: Random forest and KNNeighbors learning curves for step No.3

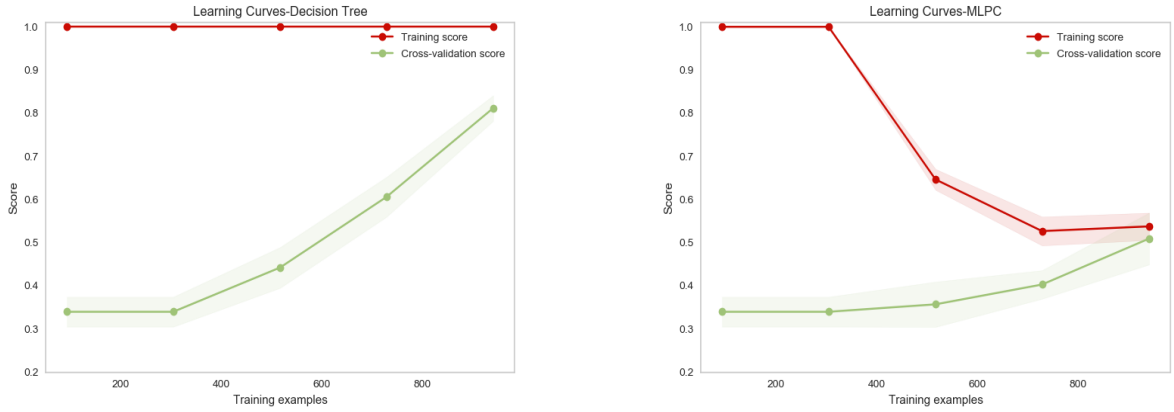

Figure 61: Decision tree and Multi-layer Perceptron learning curves for step No.3

Figures 60 and 61 show, that algorithms would benefit from adding more training data to the current dataset, with the decision tree being the one that would benefit the most. Thus a higher accuracy could be achieved by an increase in training samples. Moreover, the lower training score of MLPC could also improve by increasing the number of training samples.

7.3.8 Algorithm Evaluation Results After Parameter Tuning-Step No.3

The evaluation for step No.3 will be conducted again including the parameter optimizations as with steps no.1 and no.2.

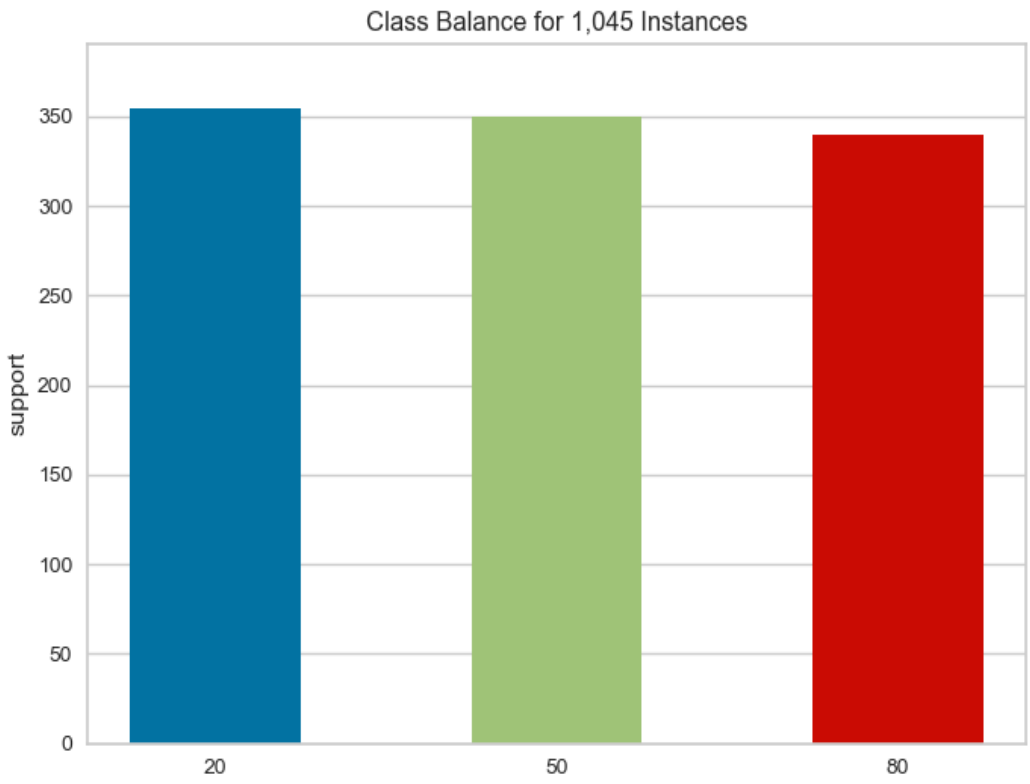

Figure 62: Class balance for step No.3

From figure 62 is concluded that at this step there is no problem due to class imbalances.

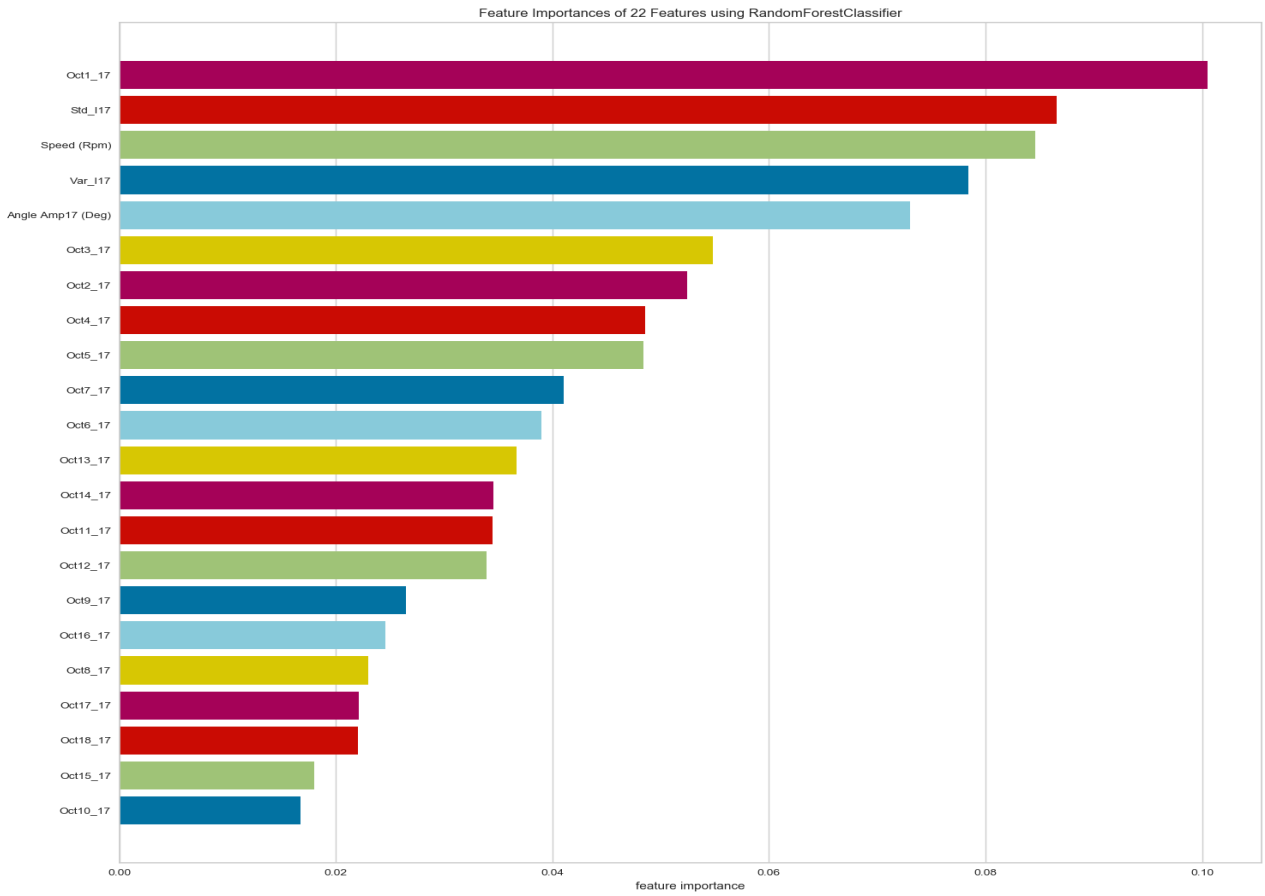

Figure 63: Feature importance for step No.3

Figure 63 illustrates the feature importances for the random forest classifier. At this point all algorithms were tested in order to find the most important features. These features that were later used for the training dataset did not provide any significant improvement to the overall performance. Also, based on the fact that this procedure is computationally expensive, the initial dataset including all features is used instead, which provides equal performance.

- **Classification report**

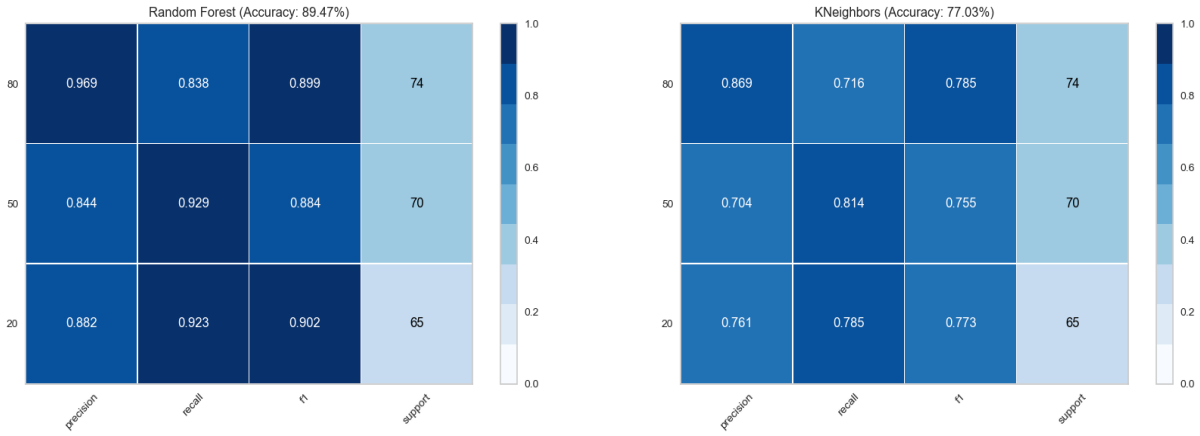

Figure 64: Random forest and KNNeighbors classification report for step No.3

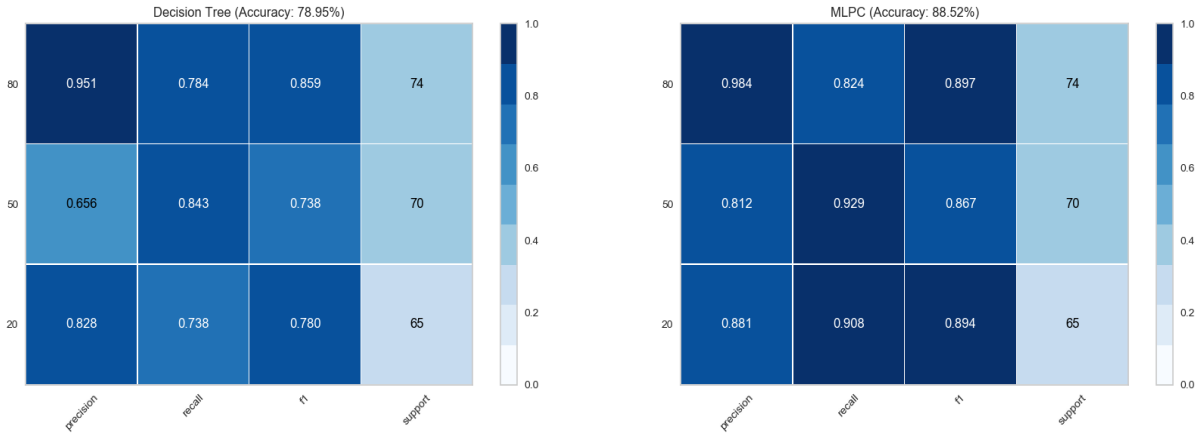

Figure 65: Decision tree and Multi-layer Perceptron classification report for step No.3

Random forest and MLPC show an increased and very satisfying score for this step after the parameter optimization. In contrast the performance of the decision tree and of the KNeighbors are lower, but still acceptable. In addition, the precision score is close to the recall score and all algorithms improved after the parameter optimization.

- **Confusion Matrix**

Figure 66: Random forest and KNNeighbors confusion matrices for step No.3

Figure 67: Decision tree and Multi-layer Perceptron confusion matrices for step No.3

After the parameter optimization all algorithms seem to perform better on distinguishing between the different classes.

- **Learning curves**

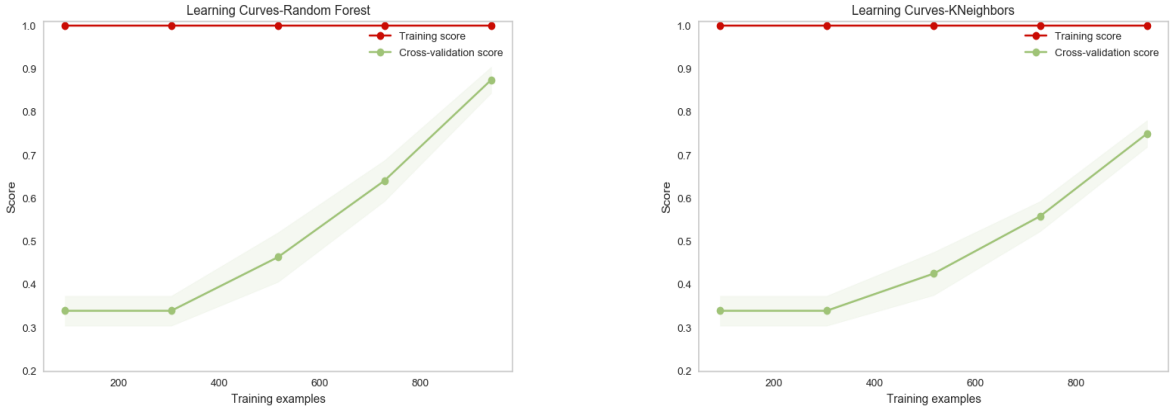

Figure 68: Random forest and KNNeighbors learning curves for step No.3

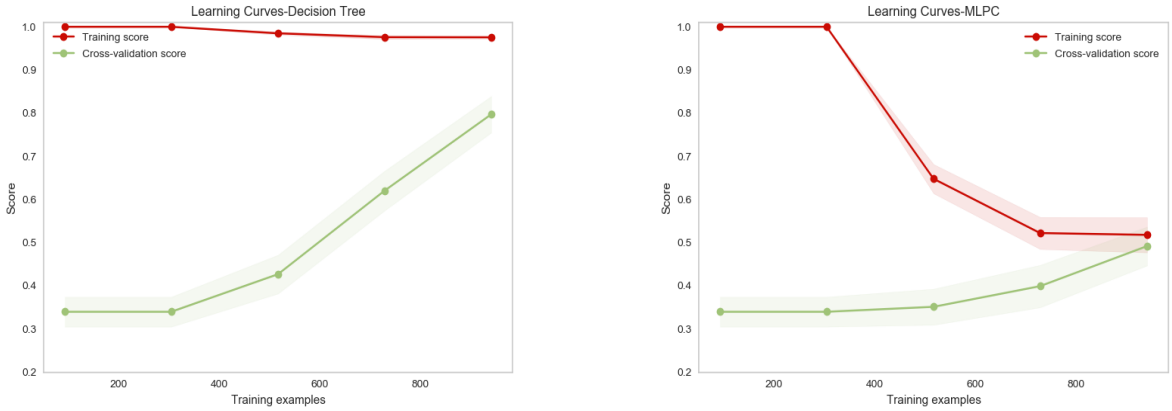

Figure 69: Decision tree and Multi-layer Perceptron learning curves for step No.3

Parameter optimization affected the score of the learning curves, indicating that after optimization an increase in samples would result in an even greater increase in performance. Figures 68 and 69 show, that all algorithms would benefit from adding more training data to the current dataset. Moreover, the lower training score of MLPC could also improve by increasing the number of training samples.

7.3.9 Making Predictions-Case Study

Previous sections, indicate that the selected algorithms perform reasonably well. All algorithms show great performance at the first classification step predicting between fault and intact conditions. The performance is lower for the second step when predicting the fault's location, while the performance raises again at the last step predicting between the different misfiring levels. In this section the trained Random Forest algorithm presented earlier is used for predicting new test data, that is not previously included in the train neither the validation set. The reason of this test is to demonstrate the results that someone would obtain if he would use the proposed method for making predictions for the engine's condition.

The data that will be used for making prediction is again generated using the OpenModelica model. Samples are obtained for various speeds at the engine's speed range, ignoring simulations at speeds used earlier at the evaluation section. The samples of these simulations represent the measurements that would be made onboard a vessel. Bellow are screenshots from the command line window showing the results after running the *Python* script.

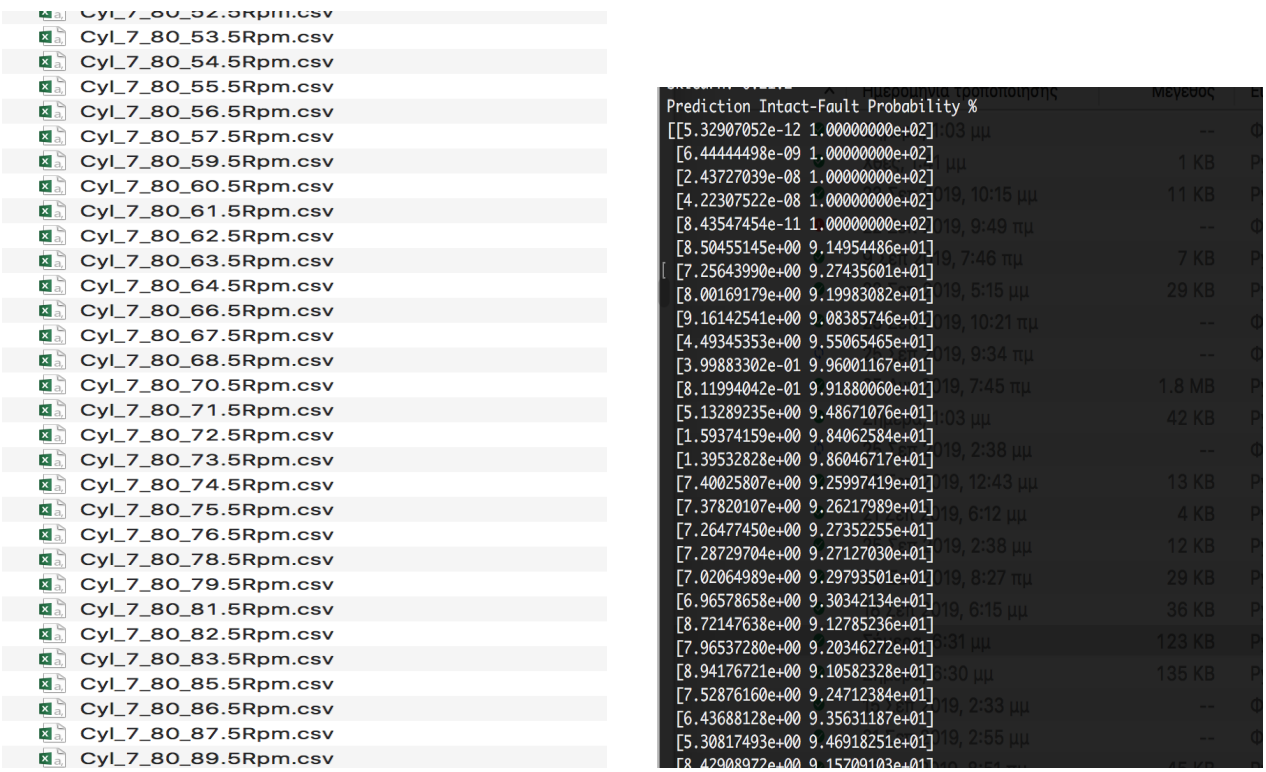

Figure 70: Insight of folder with .csv files and classes' probabilities for step No.1

```
 [1.16228131e-05 9.99999884e+01]
 [1.64832715e-04 9.99998352e+01]
 [1.69534406e-04 9.99998305e+01]]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Fault Detected
Locating Fault
Faulty Cylinder Propabilities %
[[0.06333333 0.02333333 0.06666667 ... 0.08       0.13333333 0.07      ]
 [0.04333333 0.03333333 0.05333333 ... 0.08333333 0.28333333 0.20333333]
 [0.07       0.05333333 0.02333333 ... 0.04333333 0.36666667 0.15333333]
 ...
 [0.06       0.04333333 0.05666667 ... 0.17       0.18       0.05333333]
 [0.08666667 0.07333333 0.04333333 ... 0.16666667 0.17333333 0.05      ]
 [0.07       0.05666667 0.04666667 ... 0.18333333 0.19666667 0.05      ]]
Faulty Cylinder Predictions
[ 6  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10  8 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10  9 10  9 10 10  9  9  9  9  9 10  8 10  8 10 10 10
 10 10 10  3  3  3  4  5 10 10 10 10 10 10  8 10 10 10  8 10  8 10  8 10
  8 10 10  5 10 10  8 10 10  7  7  6  7  8  6  4  6  8  4  6  6  6  6  6
 10 10  5 10 10 10 10 10 10  5  5  9  7  9  9  9 10  8  6  7 10  9  9  7
  7  9 10  9  9  9  5  7  8  9  8  6  5  6  3  7  7  7  7  7  7  7  7  7
```

```
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Fault Detected
Locating Fault
Faulty Cylinder Propabilities %
[[0.06333333 0.03666667 0.05666667 ... 0.05666667 0.18666667 0.06333333]
 [0.06       0.02333333 0.04333333 ... 0.05666667 0.27666667 0.22333333]
 [0.07       0.05       0.04333333 ... 0.04666667 0.34333333 0.16666667]
 ...
 [0.05666667 0.06333333 0.04666667 ... 0.12666667 0.19666667 0.04666667]
 [0.07       0.05       0.03666667 ... 0.13333333 0.17666667 0.07333333]
 [0.07666667 0.04333333 0.03       ... 0.22666667 0.18       0.06666667]]
Faulty Cylinder Predictions
[ 6  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10  9 10  9 10 10  9  7  9  9  9 10 10 10  5  8  3  3
 10 10 10  3  3  3  4  2 10  7 10 10  7 10  8 10 10 10  8 10  8 10  8 10
  8  8  8  5 10 10  8 10 10  7  7  9  7  7  8  1  8  8  4  6  6  6  6  6
 10 10  5 10 10 10 10 10 10 10  5  1  6  9  9  9 10  8  6  9 10  7  9  7
  7  9 10  9  9  9  5  7  8  9  8  6  5  6  1  7  7  7  7  7  7  7  7  7
  7  7  7  7  8  7  7  7  7  7  7  7  7  8  7  7  7  7  7  7  7  8  9  7
  7  7  7  7  9  7  9  7  9  7  7  7  7  7  7  7  7  7  7  7  7  7  7  8
  8  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  9
  9  7  7  7  6  7  6  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  9  9  8  7
  8  9  7  8  7  9  8]
Misfire Detected At Cylinder No.7
```

Figure 71: Step No.1 predictions and step No.2 probabilities and predictions

The binary results at left screenshot of figure 71, illustrate the predictions made for step No.1 classifying between intact and faulty condition, 0 and 1 respectively. All the samples examined in this case concern the scenario of cylinder no.7 misfiring at 80% compared to intact condition. It is easily understood that the algorithm classified almost all samples as fault condition, validating the high accuracy score during the validation procedure.

The results with values ranging from 1 to 10 at the right screenshot of figure 71, illustrate the predictions made for step No.2 classifying the cylinder, where the fault is located. Again, the lower accuracy score observed during the validation procedure can be validated. Noting that the illustrated results are in an ascending order based on their rotational speed, another important observation can be made. It is observed that correct predictions were made for samples with higher speeds and to be more precise for close to *MCR*.

```
Misfire Percentage Propabilities
[[0.32333333 0.46       0.21666667]
 [0.25666667 0.56       0.18333333]
 [0.28       0.61666667 0.10333333]
 [0.28333333 0.59333333 0.12333333]
 [0.26666667 0.63       0.10333333]
 [0.08666667 0.23       0.68333333]
 [0.1        0.22       0.68      ]
 [0.11       0.26       0.63      ]
 [0.07       0.22333333 0.70666667]
 [0.04       0.20666667 0.75333333]
 [0.12       0.62333333 0.25666667]
 [0.11666667 0.60666667 0.27666667]
 [0.06666667 0.60333333 0.33      ]
 [0.09666667 0.54666667 0.35666667]
 [0.10333333 0.57333333 0.32333333]
 [0.13666667 0.24666667 0.61666667]
 [0.13666667 0.24666667 0.61666667]
 [0.13333333 0.24666667 0.62      ]
 [0.13       0.25333333 0.61666667]
 [0.13333333 0.24666667 0.62      ]
 [0.07       0.20666667 0.72333333]
```

```
 [0.17666667 0.38666667 0.43666667]
 [0.16333333 0.39       0.44666667]
 [0.15       0.36666667 0.48333333]
 [0.20333333 0.39666667 0.4       ]
 [0.19333333 0.40666667 0.4       ]
 [0.18666667 0.4        0.41333333]
 [0.2        0.39666667 0.40333333]
 [0.19333333 0.4        0.40666667]]
Misfire Percentage Prediction
[50 50 50 50 50 80 80 80 80 80 50 50 50 50 50 80 80 80 80 80 80 80 80 80
 80 80 80 80 80 80 50 50 50 50 80 80 80 80 80 80 80 80 80 80 80 80 80 80
 80 80 80 80 80 80 80 50 50 50 50 50 50 50 50 80 80 80 80 80 80 50
 80 80 80 80 80 80 80 80 80 50 80 80 50 80 80 50 50 80 80 50 80 50 50
 50 50 50 80 50 50 50 50 50 50 50 50 50 50 50 50 50 50 80 20 80 20 80
 20 20 20 20 20 50 20 50 50 50 50 50 50 50 50 80 80 80 80 80 80 80 80
 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
 80 80 80 50 80 80 50 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
 80 80 50 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 50
 80 50 50 80 80 50 80 50 50 80 80 80 80 80 80 80 80 80 80 80 80 80 80
 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 50 80 80
 80 80 80 50 80 80 80]
Misfire Percentage:80.000000
```

Figure 72: Step No.3 probabilities and predictions

The results with values $20, 50$ and 80 at the right screenshot of figure 72, illustrate the predictions made for step No.3 classifying the level of the misfire. Also, in this case the good performance of the algorithm observed during the validation procedure can be validated. Similar observation as for step no.2, about the correct predictions can be made, but at less extend, with higher true positive rates for samples of higher speeds

8 Conclusions and Future Work

8.1 Conclusions

The present work proves that failure diagnostics of Main Engine , through monitoring of torsional vibrations can be achieved. The literature review demonstrates the significant research and development steps on diagnostic systems and their potential use for marine diesel engines.

The torsional vibrations of the shafting system can be modelled using the Modelica language, to conduct both steady state and transient simulations. The case study conducted, in which the propulsion system of a container ship vessel was modelled using Modelica, proves that such a model is able to describe accurately the system's dynamics. The aforementioned model was able to perform simulations for various steady state conditions, including cases with all cylinders at intact condition and for cylinder misfiring.

The data acquired from the simulations were processed and used for training machine learning algorithms, using Python, in order to predict operating conditions. The time domain features consisting of the mean value, peak amplitude, standard deviation and the variance and the frequency domain features of the one third octave band analysis provide sufficient information regarding the transient signal characteristics of the dynamic system.

The classification procedure consists of three steps classifying data initially, as data corresponding to intact condition or as data corresponding to cylinder misfiring. The second step utilizes only data corresponding to cylinder misfiring. predicting the most probable fault location. In the third and last step, the algorithms are trained only with data corresponding to the probable cylinder misfiring, indicating the most probable percentage of misfiring. Every algorithm showed great performance on the first step, which means that any abnormal signal can easily be detected. Taking advantage of this prediction unplanned preventive maintenance inspections can be made to locate the fault location, but additional information would drastically minimise the required work . At the second step machine learning algorithms showed satisfying performance, meaning that utilizing such a tool inspections would only be carried out on the most probable cylinder. At the third step, performance is again satisfying and close to that of step one. Knowing the percentage of misfire is critical, since that would allow to schedule future maintenance. It is also concluded that the parameter optimization and feature selection at each step, is crucial in order to increase the algorithm's performance. The random forest algorithm showed the best performance on all three steps, compared to the rest of the studied algorithms, while MLPC showed same level of performance.

The predicted results indicated that the algorithms tend to make correct predictions at speeds closer to MCR. At that speed range, the effect of the oscillations due to resonance with the system's natural frequencies is not as strong as at the lower speed range. Thus, the algorithms can better distinguish the various fault conditions at speed closer to MCR. It is concluded that the developed method shows promising results for further research on a predictive maintenance tool that can be used for marine diesel engines through monitoring of torsional vibrations. Furthermore, the combination of additional performance data could provide more accurate and precise predictions.

8.2 Future Work

The presented prediction technique, produced satisfying results that could be further improved. Algorithms' classifying performance could be increased by investigating the use of additional features. Additional features from the time and frequency domain could be used, such as kurtosis value, skewness value, crest factor, shape factor, root mean square frequency and Root variance frequency [14].

Moreover, within this thesis only conditions of on cylinder misfiring at specific misfiring percentages were simulated, therefore combinations of multiple cylinders misfiring at various percentages should also be researched. Furthermore, additional features could be extracted from the torsional vibration model or from the addition of performance data. The Modelica torsional vibration model could be combined with other domain models and create a full model of the main engine and main engine systems and simulate vibration characteristics of the most common faults. Finally, the Modelica model could be developed and validated based on experimental result acquired from real time operation measurements of the main engine.

.

References

[1] BANKS J., HINES J., LEBOLD M., CAMPBELL R, BEGG COLIN, BYINGTON C., *Failure modes and predictive diagnostics consideration for Diesel engines*, 2001

[2] RUBIO E., JAUREGUI J.C., *Time-Frequency Analysis for Rotor-Rubbing Diagnosis*, CIATEQ A.C., Centro de Tecnologia Avanzada Mexico, 2011

[3] CHEN J., RANDALL R., FENG N., PEETERS B., HERMAN VAN DER AUWERAER, *Automated diagnostics of internal combustion engines using vibration simulation*, School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney Australia, 2013

[4] WILLIAMS J., *An overview of misfiring cylinder engine diagnostic techniques based on crankshaft angular velocity measurements*, SAE Press, 1996

[5] DESBAZEILLE M., RANDALL R. B.,GUILLET F., BADAOUI M., HOISNAD C., *Model-based diagnosis of large diesel engines based on angular speed variations of the crankshaft* , Mechanical Systems and Signal Processing, vol. 24, no.5, Academic Press, 2010

[6] ZHANG Y. RANDALL R. B., *The In-Cylinder Pressure Reconstruction and Indicated Torque Estimation Based on Instantaneous Engine Speed and one Measured In-Cylinder Pressure*,Comadem Conference, Faro, Portugal, 2007

[7] MORO, DAVIDE AND CAVINA, NICOLO AND PONTI, FABRIZIO, *In-Cylinder Pressure Reconstruction Based on Instantaneous Engine Speed Signal*, Journal of Engineering for Gas Turbines and Power-transactions of The Asme - J ENG GAS TURB POWER-T ASME, 2002

[8] CRUPI V., GUGLIELMINO E.,MILAZZO G., *Neural-network-based system for novel fault detection in rotating machinery* , Journal of Vibration and Control, 2004

[9] NAHVI H., ESFAHANIAN M., *Fault identification in rotating machinery using artificial neural networks*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2005

[10] UNITED NATIONS, *Review of Maritime Transport*, United Nations Conference on Trade and Development, 2015.

[11] GKEREKOS C., LAZAKIS I.,THEOTOKATOS G., *Ship machinery condition monitoring using performance data through supervised learning* , University of Strathclyde, Glasgow, UK

[12] CHANDROTH G., *Condition monitoring: the case for integrating data from independent sources* , Journal of Marine Engineering & Technology, 2004

[13] PENG LI, LEI LIU, HAIXIA G., *The Research of the Intelligent Fault Diagnosis Optimized by ACA for Marine Diesel Engine*, Advancing Computing, Communication, Control and Management, 2010

[14] GKEREKOS C., *Ship Machinery Condition Monitoring using Vibration Data through Supervised Learning*, International Conference on Maritime Safety and Operations, Glasgow, UK, 2016.

[15] HESARI A. H., ANSARI-RAD S., SHIRAZI F. A., AYATI M., *Fault detection and diagnosis of a 12-cylinder transient diesel engine based on vibration signature analysis and neural network*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2018

[16] LIANG XINGYU, SHU GEGUN, DONG LIHUI, WANG BIN, YANG KANG, *Progress and Recent Trends in the Torsional Vibration of Internal Combustion Engine*, State Key Laboratory of Engines, Tianjin University

[17] ZHIXIONG L., XINPING Y.,CHENGGING Y., ZHONGZIAO P., *Intelligent fault diagnosis method for marine diesel engines using instantaneous angular speed*,Reliability Engineering Institute, School of Energy and Power Engineering, Key Laboratory of Marine Power Engineering & Technology, Ministry of Transportation, Wuhan University of Technology, School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, Australia, 2012

[18] HSING-CHIA KUO, LI-JEN WU, JUN-HORN CHEN, *Neural-fuzzy fault diagnosis in a marine propulsion shaft system*, Department of Naval Architecture and Marine Engineering, National Cheng Kung University, Taiwan, 2000

[19] GIRDHAR P., *Practical Machinery Vibrations Analysis and Predictive Maintenance*, 2004

[20] VULIC N., DOBROTA D., KOMAR I., *Damping and excitation in the torsional vibrations calculation of ship propulsion systems*, Faculty of Maritime Studies, University of Split, Croatia, 2016

[21] MOSCHOPOULOS M., *Experimental journal bearing loading condition determination with a machine learning technique*, Diploma Thesis, National Technical University of Athens, School of Naval Architecture & Marine Engineering, 2019

[22] PETROVSKY N., *Marine Internal Combustion Engines*, Moskow: Mir., 2004

[23] COLLACOTT R. A., *Mechanical Fault Diagnosis and Condition Monitoring* 1977.

[24] ANDREAS S. G., MULLER C., *Introduction to Machine Learning with Python* 2017.

[25] FABIAN PEDREGOSA G. V., *Scikit-learn: Machine Learning in Python* , Journal of Machine Learning Research, vol. 12

[26] RAO S., GRIFFIN P., *Mechanical vibrations*, Harlow: Pearson, 2018