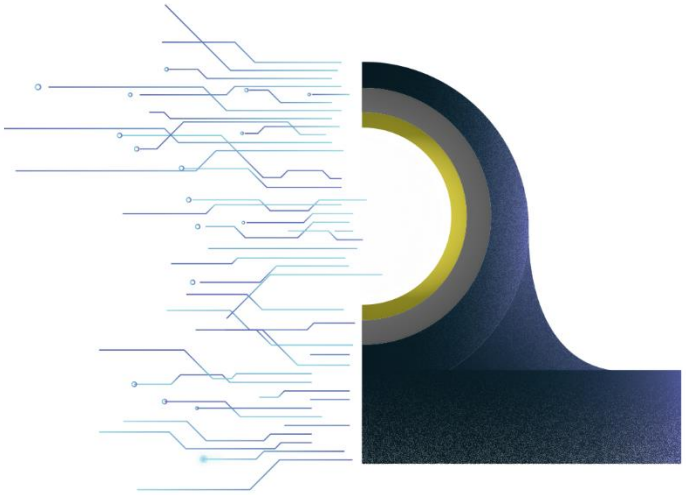
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
DIVISION OF MARINE ENGINEERING

DIPLOMA THESIS

EXPERIMENTAL DETERMINATION OF JOURNAL BEARING CONDITION WITH A MACHINE LEARNING TECHNIQUE



Marios Moschopoulos

Thesis Supervisor: Associate Prof. C. Papadopoulos

Committee Member: Prof. L. Kaiktsis

Committee Member: Assistant Prof. G. Papalamprou

Athens, September 2019

Acknowledgements

I would like to thank the Associate Professor of the School of Naval Architecture and Marine Engineering Mr. Christos Papadopoulos, supervisor of this thesis, for the opportunity he gave to study on such an interesting and fascinating field of engineering. The constant motivation and support helped me overcome obstacles and develop a wider understanding of the subjects at hand.

I would also like to thank Mr. Anastasis Charitopoulos and Mr. George Rossopoulos for their valuable help throughout the experimental and the data-analysis part of this thesis. Their contribution has been of great importance in the development of this machine learning technique. In addition, Mr. Stilianos Tzanetis has been most helpful in achieving this thesis' outcome and I would like to thank him for that.

Finally, I would like to thank my family and friends who stood by me throughout my student years.

Marios Moschopoulos, September 2019

Abstract

Journal and thrust bearings utilize hydrodynamic lubrication to reduce the friction and wear between the shaft and the bearing. The thin lubricant film that is created prevents the contact of the two surfaces and is greatly affected by the load applied from the shaft to the bearing. As a result, knowing the lubrication film thickness or the load is vital for securing the bearings' proper function and maintenance.

The experimental setup used for this project is the Rotor Kit 4 (RK4) of Bently Nevada USA and the measurements were performed in the Laboratory of Marine Engineering (L.M.E.) of the School of Naval Architecture and Marine Engineering of NTUA. The setup includes an electric motor, a speed control box, a journal bearing, a shaft and two rotor mass wheels. The data acquisition was controlled and managed in LabVIEW software.

The goal of this project is to investigate ways to determine the loading of a journal bearing without interfering with the bearings design. Then, develop a methodology that will eventually be integrated into an existing software tool and used to avoid dangerous bearing loading conditions. The efforts are focused on introducing a new methodology that will simplify the procedure mentioned but will have a scientific background strong enough to generalize in all types of journal bearings. The processing of the vibration and acoustic pressure signals leads to the selection and development of an Octave Band analysis technique. The results are then transferred to the feature space of the machine learning modules of Python and fed to several machine learning algorithms for training and testing. Different sets of training and testing data are chosen in each of the case studies in order to define the features' utility.

The method used is validated by applying it to a signal from an identical journal bearing. The results turn out to be very accurate, strengthening the methods credibility. The optimization of the algorithms is performed using cross validation tools available in Python.

Finally, the algorithms developed are trained with data acquired from one journal bearing and tested with data from an identical one in order to examine the generalization potential of the technique.

Keywords:

Journal Bearings, Lubrication theory, Loading condition, RK4 Bently Nevada, LabVIEW, Vibration signal, Acoustic pressure signal, Octave Band signal analysis, Python, Machine learning algorithms.

Σύνοψη

Στα ακτινικά και στα ωστικά έδρανα ολίσθησης χρησιμοποιείται η υδροδυναμική λίπανση για να μειωθεί η τριβή και η φθορά μεταξύ του άξονα και του εδράνου. Το λεπτό φιλμ λιπαντικού που δημιουργείται αποτρέπει την επαφή των δύο επιφανειών και το πάχος του εξαρτάται από το φορτίο που ασκεί ο άξονας στο έδρανο. Ως εκ τούτου, είναι σημαντική η γνώση είτε του πάχους του λεπτού αυτού φιλμ είτε του φορτίου ώστε να εξασφαλιστεί η σωστή λειτουργία και συντήρηση του εδράνου.

Η πειραματική διάταξη που χρησιμοποιήθηκε είναι το Rotor Kit 4 (RK4) της Bently Nevada USA και οι μετρήσεις πραγματοποιήθηκαν στο Εργαστήριο Ναυτικής Μηχανολογίας της Σχολής Ναυπηγών Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Στην πειραματική διάταξη συμπεριλαμβάνονται μεταξύ άλλων ένας ηλεκτροκινητήρας, μία μονάδα ελέγχου της ταχύτητας περιστροφής του κινητήρα, ένα ακτινικό έδρανο ολίσθησης, ένας άξονας και δύο κυλινδρικές μάζες για τον άξονα. Η δειγματοληψία έγινε με χρήση του πακέτου LabVIEW.

Σκοπός της εργασίας αυτής είναι η εύρεση ενός τρόπου καθορισμού του φορτίου που φέρει ένα ακτινικό έδρανο ολισθήσεως χωρις την επέμβαση στην σχεδίαση του σχεδίου του εδράνου. Εν συνεχεία, η ανάπτυξη μιας μεθοδολογίας που να μπορεί εν τέλη να ενσωματωθεί σε ένα υπολογιστικό εργαλείο το οποίο θα χρησιμοποιείται για την αποφυγή επικίνδυνων καταστάσεων φόρτισης των εδράνων. Οι προσπάθειες εστιάζονται στην δημιουργία μιας νέας μεθοδολογίας που να απλοποιεί την διαδικασία υπολογισμού του φορτίου αλλά θα έχει ισχυρή επιστημονική βάση ώστε να μπορεί να χρησιμοποιηθεί σε διαφόρων τύπων ακτινικά έδρανα. Η επεξεργασία των σημάτων ταλαντώσεων και ακουστικής πίεσης οδήγησε στην επιλογή και ανάπτυξη μιας τεχνικής βασισμένης στην Οκταβική ανάλυση. Τα αποτελέσματα μεταφέρονται στο πεδίο των χαρακτηριστικών (feature space) των βιβλιοθηκών μηχανικής μάθησης της Python και τροφοδοτούνται σε διάφορους αλγορίθμους μηχανικής μάθησης, οι οποίοι εκπαιδεύονται και αξιολογούνται.

Για επαλήθευση, η μέθοδος χρησιμοποιείται σε ένα σήμα που παράχθηκε από ένα πανομοιότυπο έδρανο. Τα αποτελέσματα είναι αρκετά ακριβή, ενισχύοντας την αξιοπιστία της μεθόδου. Η βελτιστοποίηση των αλγορίθμων γίνεται με χρήση εργαλείων της Python.

Τέλος, οι αλγόριθμοι που επιλέχθηκαν εκπαιδεύονται με το σήμα που προέρχεται από ένα έδρανο και αξιολογούνται με το σήμα που προέρχεται από ένα πανομοιότυπο έτσι ώστε να ελεγχθεί η δυνατότητα γενίκευσης της τεχνικής αυτής.

Λέξεις κλειδιά:

Ακτινικά έδρανα, Θεωρία υδροδυναμικής λίπανσης, Κατάσταση φόρτισης, RK4 Bently Nevada, LabVIEW, Ταλαντώσεις, Ακουστική πίεση, Οκταβική ανάλυση, Python, Αλγόριθμοι μηχανικής μάθησης.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Symbol	Definition
.CSV	Comma Separated Values
DSL	Domain-specific Language
DTR	Decision Tree Regressor
FFT	Fast Fourier Transformation
GBR	Gradient Boosting Regressor
GPL	General Purpose Language
GUI	Graphical User Interface
KNNC	*k*-Nearest Neighbors Classifier
RFC	Random Forest Classifier
RMS	Root Mean Square

Symbols

Symbol	Definition
A	Feature
c	Radial clearance [m]
C	Set of classes in a dataset
d	Shaft diameter [m]
D	Bearing diameter [m]
e	Shaft eccentricity [m]
E	Young's Modulus [Pa]
f	Friction coefficient
f_u	Upper frequency limit of an Octave band [Hz]
f_0	Center frequency of an Octave band [Hz]
f_1	Lower frequency limit of an Octave band [Hz]
h_0	Minimum oil film thickness
$H(J)$	Entropy of a dataset J

I	Moment of Inertia [m^4]
IG(A)	Information Gain of a feature A for a dataset
k	Number of neighbors in k-Nearest Neighbors algorithm
L_p	Level of power/energy spectrum [dB]
N	Rotational speed [RPM]
p_{ref}	Reference pressure [Pa]
$p(o)$	The ratio of the number of elements of class o to the number of elements in a dataset
P	Bearing Load [N]
r	Shaft radius [m]
R	Bearing radius [m]
R^2	Coefficient of Determination
S	Dataset
SE	Squared Error
T	The subsets of S after splitting on feature A
w	Distribution of weight [kg/m]
β	Angular length of a partial bearing [degrees]
μ	Lubricant dynamic viscosity [cSt]
ρ	Shaft density [kg/m^3]
U	Velocity

1 Introduction

For many years, the selection journal bearings and the proper shaft alignment has been a matter of investment cost. Nowadays, the increase of ships' length has led to an increase of shafts' length thus making the alignment more difficult and the need for a means to predict dangerous situations more profound.

The running condition of a journal bearing is a subject of tribological analysis and is adequately described by the hydrodynamic lubrication theory [1]. A considerable number of papers are published every year over this topic, showing the importance of optimizing mechanical elements with tribological interest. The science of tribology studies the friction, wear and lubrication of mechanical elements. Friction and the consequential wear can be significantly reduced when a thin layer (film) of lubricant is generated between the sliding surfaces. As a result, the main goal of this science is to optimize the tribological behavior of such elements through models that can predict this behavior and suggest more suitable materials for each case studied.

While there is a plethora of software packages that enable the modeling and mapping of a journal bearing's behavior like ANSYS, there is little research on methodologies that enable the condition monitoring of the bearings and the existent focus on specific faulty operation statuses like oil contamination (see [2], [3], [4]). Condition monitoring informs the user in real time about the element's status and operation and enables a faster and more appropriate course of action in case of maintenance or a malfunction [5]. For the better part, as far as journal bearings are concerned, very specific attributes of the bearing's running condition can be monitored with non-invasive techniques (e.g. lubrication oil temperature [6]).

One of the attributes that needs constant monitoring is the force bore by the bearing. The magnitude of this force is very important due to its strong correlation with the thickness of the lubricant's film and its correlation with the load carrying capacity of the bearing. By design, there is a range of values in which the bearing performs satisfactorily and the film maintains its properties, classifying the status of operation as safe [1]. On the other hand, when the load is insufficient the bearing is considered lightly loaded and when the load exceeds a certain level the bearing is considered heavily loaded. Both cases are classified as unsafe. Due to the nature of this attribute, it is quite difficult to directly determine its value without interfering with the bearing's design thus increasing the design's complexity and price. As a result, indirect ways of measurement have drawn attention.

A recent trend in monitoring are the machine learning algorithms. These algorithms have developed significantly over the past few years and have been integrated in many programming languages. With a variety of functionalities, machine learning algorithms can adjust to differently natured problems and, if programmed correctly, perform accurately. The combination of several algorithms is also possible, allowing more customization to the user. Python is one of the programming languages that has a machine learning module with most of the commonly used algorithms and gets regularly updated to meet new and more complex needs [7]. Due to the low computational cost and high computational speed, Python appears to be a simple and trustworthy choice.

1.1 Method Outline

This experimental diploma thesis has taken place in the Laboratory of Marine Engineering (L.M.E.) of NTUA and, therefore, its main interest is to give solutions regarding the proper maintenance of marine journal bearings. The experiments were conducted on the Bently Nevada Rotor Kit 4 of L.M.E. Once the experimental setup is properly assembled and the monitoring system is installed a series of experiments take place. The measurements are transferred through a data acquisition card (DAQ) to the computer (LabView software) and get properly stored.

The data processing used is the One-third Octave analysis and is applied on sections of the data, whose volume depends on the case studied. This processing is performed by a source code written in Python along with every needed adjustment. The frequency band domains produced from this analysis are then used to create the feature space of the machine learning algorithms. The importance of each feature varies depending on the problem at hand. The algorithms' results depend greatly on the information given to the algorithm through the features and the user should pay attention to the selection process.

The cases studied are intended to identify which of these signals and features serve to develop a technique that will be able to determine the charge status of a bearing by measuring the vibrations and acoustic pressure at the bearing base. In short, the main cases that are studied are the following:

- Selection between Vibration and Sound Signals,
- Investigating whether training data from one bearing can help predict the loading condition of a similar bearing and what features enable this function,
- Combining training data from multiple bearings (manufactured from the same design plan) can produce an algorithm that can make accurate predictions for all bearings that are manufactured using this bearing design plan.

1.2 Thesis Outline

This thesis report is divided into 6 chapters. Chapter 1 includes the introduction, goal of the project and method outline. Chapter 2 describes the theoretical background of journal bearings and condition monitoring, including the hydrodynamic lubrication theory. In chapter 3, the theoretical background of machine learning is presented along with Python and its machine learning module. Chapter 4 describes the experimental setup and the experimental procedure, from the preparation of the experiments to the data acquisition. In chapter 5, the postprocessing of the raw data carried out and the Octave Band analysis are presented, as well as the case studies. Finally, chapter 6 outlines the conclusions of this thesis and notes suggestions for future work.

2 Journal Bearings

2.1 Hydrodynamic Lubrication Theory

The main goal of lubrication is to reduce friction and wear of machine elements with relative motion. When inserted between the two surfaces the lubricant can reach this goal and also act as a coolant. In a journal bearing the essential motion is sliding and the theory that describes this lubrication phenomenon is that of the hydrodynamic lubrication. Apart from the reduction of friction and wear, the lubricant allows the transfer of forces without the direct contact of the sliding surfaces creating a vast field of application for journal bearings. In order to achieve hydrodynamic lubrication, no pressurized fluid is required, though that may occur; but the requirement for continuous lubricant inflow is definite. The pressure created is due to the supply of energy from moving surface of the shaft to the lubricant. After a certain critical value is reached, the pressure generated can separate the surfaces and support the shaft loading. This phenomenon was firstly investigated by Beauchamp Tower (1880s) and based on his results Osborne Reynolds proposed some equations which are presently used to interpret hydrodynamic lubrication. [1]

Hydrodynamic lubrication conditions:

1. There should be relative movement between the two surfaces with a value high enough to create a lubrication film that carries bare loads,
2. An inclination between the sliding surfaces is mandatory for the creation of the hydrodynamic wedge. If the two surfaces are parallel then the creation of the pressure profile is not possible.

In **Figure 2.1** the mechanism of the creation of the hydrodynamic pressure between two surfaces with an inclination is shown. One surface moves at a constant velocity U while the other surface is stationary. There is always a sufficient amount of lubricant between the two surfaces, making sure that both conditions mentioned above are met. By moving the upper surface, the lubricant is forced to enter the converging geometry (wedge), resulting in a pressure build up. Pressure has a positive derivative at the inlet of the wedge, controlling the lubricant's inflow, and a negative derivative at the outlet, thereby allowing the outflow of the lubricant. The velocity profile strictly depends on the pressure distribution and more specifically on the special derivative of the pressure. Therefore, the distribution profile at the inlet rotates the concave downward while it rotates the concave upward at the outlet, as shown in **Figure 2.1**. [1]

The geometry of the plane slider shown in **Figure 2.1** is similar to the hydrodynamic wedge created inside a journal bearing. The hydrodynamic wedge created between the journal (shaft) and the bushing (bearing) is shown in **Figure 2.2**. An appropriate mathematical model can be used in order to calculate the value of the hydrodynamic pressure and accurately predict the operating characteristics of a bearing.

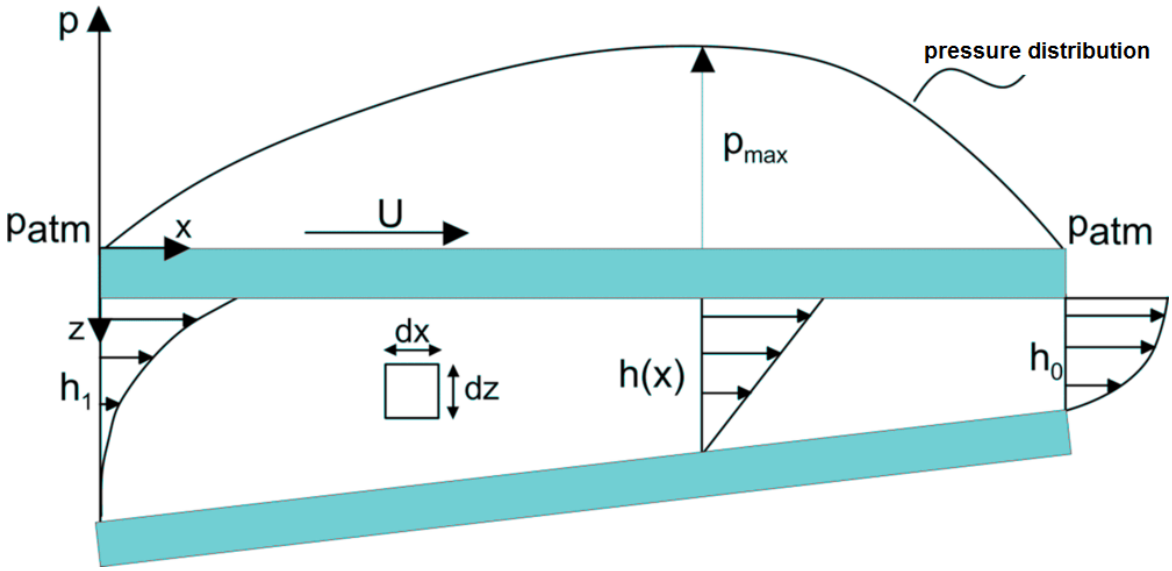


Figure 2.1 Hydrodynamic pressure creation in the case of a plane slider bearing. The upper surface moves horizontally at speed U, while the lower surface is stationary [1]

The nomenclature of the partial journal bearing in **Figure 2.2:**

- Bearing diameter, *D*
- Bearing's radius, *R*
- Shaft diameter, *d*
- Shaft radius, *r*
- Radial clearance, *c* (the difference in the radii of the bearing and the shaft)
- Rotational speed, *N*
- Lubricant dynamic viscosity, μ
- Minimum film thickness, *ho*
- Eccentricity of the shaft, *e*
- Angular length of a partial bearing, β

Figure 2.2 shows a section of a journal bearing and a shaft that rotates clockwise. Suppose there is a sufficient amount of lubricant inside the bearing. The rotation of the shaft drives the lubricant into the clearance around the shaft in the direction of the rotation. As the lubricant enters this geometry, it pushes the shaft to the opposite side of the bearing. This creates a lubricant layer with a minimum film thickness *ho* at a certain position which is not at the bottom of the bearing, but a position displaced along the axis of rotation. This position is a result of the balance of forces on the vertical and horizontal axis.

In **Figure 2.2**, the center of the shaft is point O while the center of the bearing is point O'. The minimum gap *ho* is located at line created by the two centers, OO'. The eccentricity *e* of the axis relative to the center of the bearing is also shown in **Figure 2.2** and the radial grace *c*, which is defined as $c = R-r$, where *R* is the radius of the bearing and *r* the radius of the shaft. *ho* should be greater than 0 and less than *c*. When there is no rotation, the center of the shaft O is under the center of the bearing O' and has no horizontal displacement. On the other hand, when the shaft rotates at a very high speed, due to the high hydrodynamic speed, the two centers coincide.

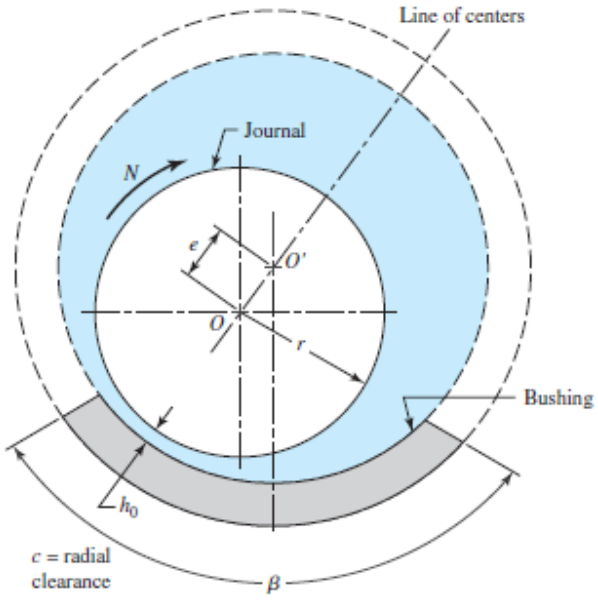


Figure 2.2 Cross-section of a partial journal bearing in operation

2.2 Bearing Loading Condition

The journal bearing creates a hydrodynamic film in order to carry the load of the shaft. When there is a variation in the loading, the bearing assembly reacts and changes its eccentricity and film thickness until the load is carried. The actual minimum permitted value of *ho* is affected by the surface roughness, the specifications of the lubricant and the bearing dimensions.

In hydrodynamic lubrication theory, there are three distinct operation conditions (**Figure 2.3**):

1. Full film hydrodynamic lubrication,
2. Mixed film lubrication,
3. Boundary lubrication.

The first condition describes the safe mode where there is enough oil and the contact of the two surfaces is prevented. No wear is observed and the oil film thickness is much greater than the surface roughness.

The cases where the lubrication film is too thin to separate the surfaces are called mixed film lubrication or boundary lubrication. In these cases, the lubricant can only receive a part of the shaft's load and thus there is contact between the surfaces. The mixed film lubrication refers to the transition from boundary lubrication to full film lubrication. There is asperity to asperity contact of peaks and wear is inevitable if there is prolonged operation in this condition.

Boundary lubrication refers to the condition where the two surfaces are not separated by the lubrication film and there is asperity contact. The oil film thickness is normally less than the surfaces' roughness and wear damage is significant.

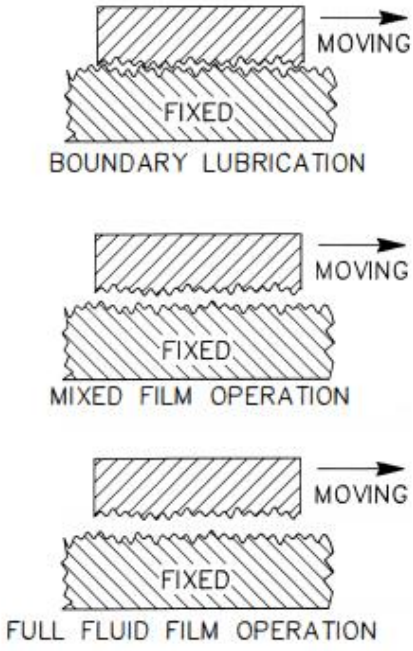


Figure 2.3 Three Lubrication Conditions [8].

McKee brothers have obtained the plot presented in **Figure 2.4** after carrying out a test of friction and show how the coefficient of friction *f* changes depending on the bearings characteristic $\mu N/P$. As described by *Petroff's equation,* $f = 2\pi^2(\mu N/P)\,(r/c)$, with P being the load.

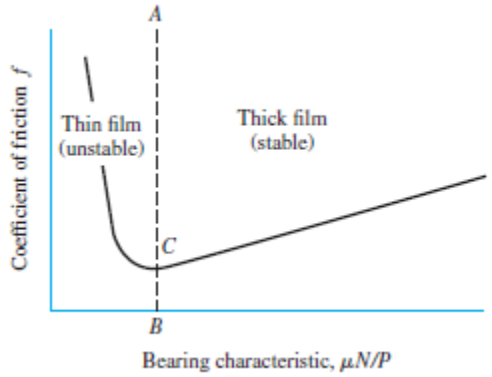


Figure 2.4 Variation of the coefficient of friction [1]

In **Figure 2.4** there are two regions, one on the left of AB (unstable) and one the right of AB (stable). When operating on the stable side, if for example a rise in temperature of the lubricant occurs then its viscosity decreases hence $\mu N/P$ decreases. As a result, the coefficient of friction decreases too, thus not too much heat through shearing the lubricant is generated and the temperature of the lubricant eventually drops. When operating on the unstable side, the same rise in temperature would have the opposite effect on the coefficient of friction, rising the temperature even more. A lubricant is usually consisted of hydrocarbons and reacts to the rise of temperature

by vaporizing lighter components. While this process takes time, the viscosity of the lubricant constantly increases which leads to a rise of the heat generation rate that elevates lubricant temperature even more, entering a vicious circle leading to failure.

2.2.1 Insufficiently loaded bearings

Insufficiently loaded bearings are called lightly loaded. Such bearings are susceptible to subsynchronous vibrations called "oil whirl". The pressure of the lubricant film is higher than that needed to support the shaft, which lifts the shaft up to a certain point and then drops it down. The vibrations produced can cause damage to the bond of e.g. the Babbitt layer with the steel and lead to failure.

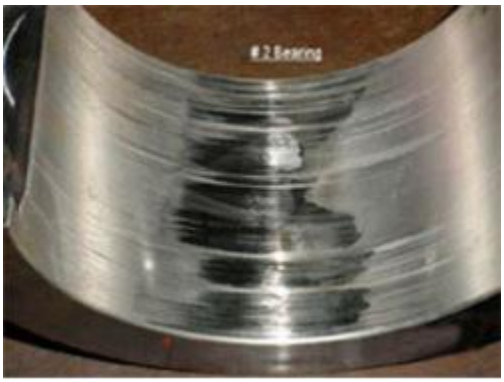


Figure 2.3 Result of metal-to-metal contact due to 'oil-whirl' induced vibration [9]

Moreover, it is inherently difficult for a journal bearing to bear big dynamic loads and avoid damage. It is more difficult when the properties of the oil film are unstable because of the insufficient loading of the bearing [1]. If the oil is modeled as a spring and a damper, then the insufficient loading decreases the resistance against deformations of the spring (K) and the damping properties of the damper (C).

2.2.2 Overloaded bearings

If plotted, the timeseries of the loading condition of a naval journal bearing would have many fluctuations. That is explained by the nature of a ship's construction materials, which is mostly steel, and the ship's deformation principally caused by its inertia, cargo and buoyancy. The shaft is deformed as well causing a change in the load imposed on the bearing. The force applied on the bearing has a direct relation to the oil film thickness developed in the bushing. Oil film thickness decreases proportionally with the increase of the load.

Suppose that both oil temperature and the rotational speed of the shaft are constant. When there is a rise in the load, as described above, the bearing assembly will react and change the oil film thickness and its eccentricity. If this load is too high then, as mentioned earlier, the temperature will rise, leading to failure due to oil overheating [10]. Overloading can also lead to wiping. Some of the symptoms are heavy scoring, circumferential movement of while metal, re-solidification of whitemetal deposited in oil grooves. Oil contamination is also present if there is metal-to-metal contact and material is.

As mentioned above, there are many dynamic loads met in the sea. Excessive dynamic loads lead to metal fatigue which causes cracks in the white area surface around the application area of the load. These cracks develop sideways and can lead to a whitemetal detaching. Big dynamic loads can have an upward direction and cases have been reported where the housing of the bearing has been damaged.

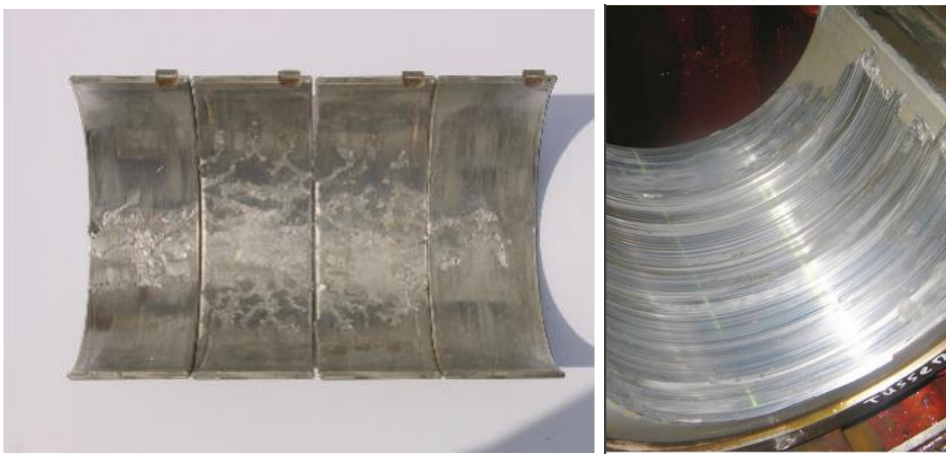


Figure 2.4 Bearing overloading: Metal fatigue (left) and excessive wiping (right)

2.3 Condition monitoring and Predictive maintenance

A very common type of surface damage is wear. It usually arises from the interaction of surfaces and is connected with material properties, geometry, lubrication and operating status. Predicting the rate of wear of a product is important in determining its life span and thus dealing with the repair or replacement needed. Engineers can nowadays make accurate predictions using materials science and modeling/simulation software regarding a component's e.g. strength, but find it difficult to address wear-related issues due to the dynamic nature of the phenomenon. [11]

Condition monitoring is the process followed by an operator of constantly monitoring the value of a parameter in machinery. The goal is to observe significant and meaningful changes in that value which could indicate a possible malfunction. By doing so it is possible in an early stage to prevent conditions that could lead to an emergency breakdown in an early stage and, if needed, schedule maintenance in a more convenient manner. [5]

A term that is closely related to condition monitoring is predictive maintenance. This term describes the techniques used to determine the actual condition of operating equipment so as to roughly calculate when maintenance should be carried out. These methods have a positive financial impact for the owner of the machinery monitored, a positive environmental impact due to less accidents that could lead to environmental issues and material consumption and, finally, promote safety for the staff. Condition monitoring is one of the techniques used in predictive maintenance. [12]

In the field of journal bearings, it is quite important to monitor a components status and wear for both safety and financial reasons. Rotating machines are of high interest for the condition monitoring industry and, in more detail, vibration measurements are the most common technique [3]. The measurements acquired through accelerometers and microphones are then processed in respect with the type of the machinery and interpreted by specialists in the field. Research in sensors technology has provided a wide range of equipment that can be used to monitor a variety of running condition attributes.

3 Machine Learning Algorithms

3.1 Introduction to Machine learning algorithms

Machine learning, also known as predictive analytics or statistical learning, is a research field combining statistics, artificial intelligence and computer science and its goal is to produce knowledge from specific data [7]. Nowadays, it is very common to interact with machine learning models because of the trending user-customization-friendly online platforms and websites. Handcoded decision-making algorithms are faced with the disadvantage that they are task- or domain-specific and that a constant need for a human expert on the field would arise due to the importance of understanding how the decision process works. However, in machine learning, no such need is present.

One of the most successful kinds of machine learning algorithms are the supervised learning ones. The process requires that the user provides a set of known input and output data to the algorithm and the algorithm uses these sets for training its decision-making ability. Having enough training data is a prerequisite for such an algorithm to work and defining enough can sometimes be tricky for the user. [7]

The two major types of supervised learning problems are called classification and regression. In classification, the goal is to predict a certain class label from a predefined set of attributes. When there are two classes the case is called binary classification and when there are more than two it is called multiclass classification. In classification algorithms, the accuracy of the algorithm is measured by the number of correct class predictions that the algorithm has managed.

In the case of regression, the task is to predict a floating-point number in programming terms or a real number in mathematical terms. A simple regression task would be the prediction of a rotor's rotational speed. The output values have an obvious continuity in the positive integer numbers spectrum. A regression algorithm's accuracy is defined by the coefficient of determination or R^2:

$$\boldsymbol{R^2 = 1 - \frac{SE\hat{y}}{SE\bar{y}}}$$

Where

- $SE\hat{y} = \sum_i (f_i - y_i)^2$ is the squared error of the training data to the regression line,
- $SE\bar{y} = \sum_i (y_i - \bar{y})^2$ is the squared error of the training data to the mean y.

The other kind of machine learning algorithms is called unsupervised. In this type of algorithms only the input data is known and no output data is defined. Although there are many cases for which these algorithms become useful, their use is limited because of the difficulty to understand and evaluate them. [7]

Unsupervised algorithms are divided into two major categories. Transformations and clustering. Transformation algorithms create a new representation of the data, depending on the user's needs, by feature processing. For example, a five-dimensional problem can become two-dimensional for plotting purposes.

On the other hand, clustering algorithms subdivide data into well-defined groups of same/similar items. These algorithms are frequently used for tasks, like segmenting customers into groups

depending on their preferences, where the number or the "properties" of the groups are not known in advance.

3.2 Types of Supervised Machine learning algorithms

In this paragraph, the main algorithms that are used for this thesis will be presented. Each algorithm functions and performs differently from the others due to the approach of each developer. Some algorithms are based on existent mathematical models. Other algorithms are created in order to correct the weaknesses of older algorithms or combine the advantages of several algorithms into one.

3.2.1 *k*-Nearest Neighbors

k-Nearest Neighbors algorithms used for classification are simple and only require the storage of the training dataset. They do not create in internal model to aid with the prediction. When given a new data point, the algorithm searches for the closest point in the dataset. *k* represents the number of closest points that will participate in the majority vote used to classify the new data point. The class assigned to the point is that with the most representatives in the *k*-nearest neighbors.

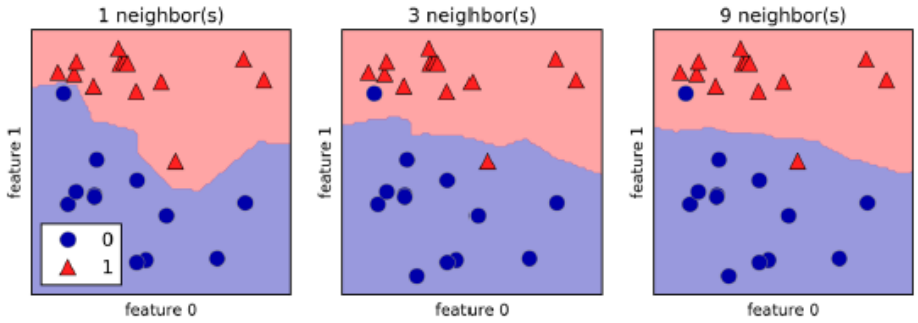


Figure 3.1 Decision boundaries of the nearest neighbors algorithm for different values of *k*

In its simplest form, the majority vote involves computing the distances between all pairs of data points: for N samples and D dimensions this form scales as $O[DN^2]$. Several distance metrics are available:

Distance Functions

Euclidean $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{n}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{n}(|x_i - y_i|)^q\right)^{1/q}$

Where :

n = no of dimensions
x = datapoint from dataset
y = new data point

Figure 3.2 Distance Metrics

When the sample is relatively small the algorithm is performing well but when the samples becomes too big then the nearest neighbors algorithms become impractical. This computation setting is called brute-force

Different and computationally cheaper techniques involve indirect estimation of distance between two points with the use of tree-based data structures (Tree structures are mentioned in paragraph 3.2.2). K-D Tree is a binary tree which generalizes two-dimensional Quad-trees or three-dimensional Oct-trees to a random number of dimensions. In this way, K-D crates data-populated regions without calculating the multi-dimensional distance. [7]

3.2.2 Decision Trees

The decision trees are models that create an order of if/else questions that ultimately lead to a prediction of the value of the target variable. The training is performed with the data features. In **Figure 3.3** there is a series of boxes called nodes. Nodes can either be decision nodes that lead to another set of nodes ('Can Fly?' node, **Figure 3.3**) or a prediction node that terminates the procedure ('Hawk' node, **Figure 3.3**). The prediction nodes can also be called leaf nodes.

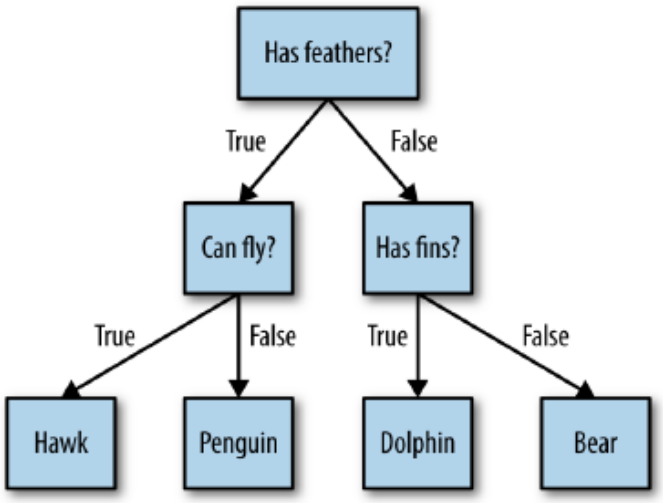


Figure 3.3 A simple decision tree to distinguish animals [7]

The first step in tree building is to find a feature that is most informative about the output variable. There are several criteria that can be used to perform this choice [13] [14]:

I. Entropy ***H(S)*** measures the amount of uncertainty in the dataset ***S*** in the following way

$$\boldsymbol{H}(\boldsymbol{S}) = \sum_{\boldsymbol{c} \in \boldsymbol{C}} -p(c) \log_2 p(c)$$

Where,

- ***S***, the dataset for which the entropy is calculated for

- **C**, set of classes in ***S***
- **p(*c*)**, the ratio of the number of elements in class ***c*** to the number of elements in dataset ***S***

The entropy is calculated for each feature and the feature with the smallest entropy is used to split the dataset. When ***H(S)***=0 then the dataset is thoroughly classified.

II. Information gain ***IG(A)*** is defined as the difference in entropy between before and after splitting the dataset ***S*** on a feature ***A***.

$$IG(S,A) = H(S) - \sum_{t \epsilon T} p(t)H(t)$$

Where,

- ***H(S)***, the entropy of set ***S***
- **T**, the subsets of ***S*** created by splitting on feature ***A*** such that $S = \cup_{t \epsilon T} t$
- **p(*t*)**, the ratio of the number of elements in class ***t*** to the number of elements in dataset ***S***
- ***H(t)***, the entropy of subset ***t***

The information gain is calculated for all the features and the feature with the largest information gain is used to split the dataset ***S***.

Using this information, the data (root node) is split into two newly formed nodes (depth=1). Although a single split might be adequate, in many cases the percentage of purity of the nodes is not high enough. By repeating the previous process, a more accurate model is created (depth=2,3…) with each node rising its purity (**Figure 3.4**). If data partitioning continues then every leaf node eventually contains only one regression value/class and is called pure.

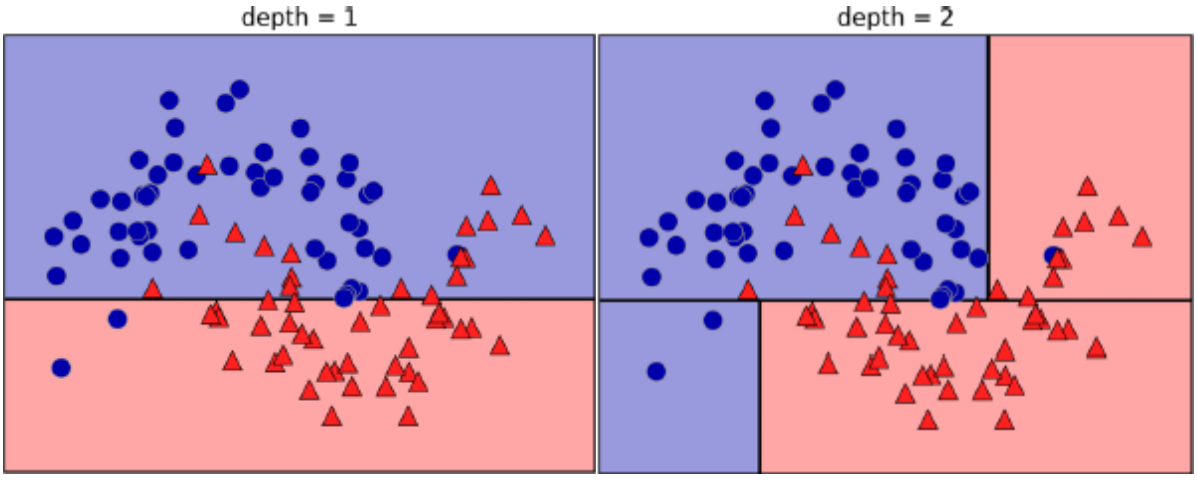


Figure 3.4 Decision boundaries of decision trees with depth=1 (left) and depth=2 (right) [7]

According to the values of its features, a new data point lies in a specific region of the partition produced from the training process. The prediction results from the majority target, if the leaves are not pure, or from the single target, if the leaves are pure, of that region.

3.2.3 Ensemble methods

In ensemble methods, the models created are a combination of different/many machine learning models that create a more powerful and effective algorithm with a higher predictive performance. Each individual model is called a base learner. The major representatives of this category of algorithms are the Random Forests and the Gradient Boosted methods that have as a base learner the classic Decision Trees.

3.2.3.1 Random Forest

Random Forests are a way to solve some inherent drawbacks of Decision Trees. A random forest is a set of many random trees that differentiate from each other in two ways: the data points used to build the tree and the features used for each split. The algorithm starts (b=1) by drawing a sample from the whole training dataset and creating a tree (*T)* according to a set of features drawn from the available feature space. Once the minimum node size n_{min} is reached, the algorithm creates the next random tree until $b = B$, where *B* is the number of estimators (random-forest trees) defined by the user. The end of the training results in an ensemble of trees $\{T\}_1^B$.

After the random-forest trees are created the algorithm makes a prediction for each tree. If it is a regression problem, the algorithm averages the results to produce a prediction for the new data point *x* [14]:

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

If it is a classification problem then the algorithm creates a voting strategy where every tree provides a probability for each class and then all the probabilities are averaged so as to find the highest one [14]:

$$\hat{C}_{rf}^{B}(x) = majority\ vote\{\hat{C}_b(x)\}_1^B$$

Where,

- $\hat{C}_b(x)$, the class prediction of the *b*th random-forest tree.

3.2.3.2 Gradient Tree Boosting

In Gradient Tree Boosting methods, the algorithm generates trees in a "serial" way and each new tree attempts to correct the mistakes of the one previously created. The user defines the trees' size and aims to create shallow trees. This is called pre-pruning. These shallow trees are called weak learners and their depth usually variates between two and five. As more trees (m) are added, the performance of the algorithm improves until the max number of trees is reached (M) or performance is not improving.

The figure below shows the exact methodology followed.

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Figure 3.5 Gradient Tree Boosting Algorithm [14].

Step 1 of **Figure 3.5** shows the first prediction leaf created by the algorithm. It is the lowest value of the sum of the loss function $L(y_i, \gamma)$, where y_i is the actual value of the dataset and γ is the predicted value, with respect to γ. The loss function can be chosen from many different formulas but the most common is the following: $L = \frac{1}{2}[y_i - f(x_i)]^2$

In step 2, an iteration is performed for the number of trees M. Firstly (a), the pseudo residual $r_{i,m}$ is calculated as the derivative of the loss function for i = 1, 2…, N, where N is the number of samples in the dataset. Then (b) a regression tree is fir to the pseudo residuals $r_{i,m}$ with terminal regions (leaves) $R_{j,m}$, where J is the number of leaves in the tree. The output value $\gamma_{j,m}$ for each leaf is then calculated and it is the value for γ that minimizes the summation in 2.c of **Figure 3.5**. In part (d) the new prediction $f_m(x)$ for each sample is calculated. This prediction is taking into consideration the previous prediction $f_{m-1}(x)$ and the summation of the output values $\gamma_{j,m}$. Step 2 is repeated until $m=M$.

Step 3 is the final summation of the M trees created in step 2.

3.3 Software and features

3.3.1 Python

Python is a broadly applicable programming language that combines the features of a general-purpose language (GPL) and of a domain-specific language (DSL). This characteristic makes it suitable for data science applications like machine learning, statistics, image processing, visualization, neural networks and more. In addition, Python enables the user to directly interact with the code by using terminals like Jupyter Notebook and helps produce a compact and readable code. It is widely used in industrial applications because of its adaptiveness to existing systems and the ability to produce elaborate graphical user interfaces (GUIs) [15]. Python is also supported by a vast community of users and developers that provide constant feedback concerning bugs and needs for the majority of the official libraries, allowing for a continual improvement of each package. Finally, Python has a great number of different libraries that support simple to computationally-intensive tasks. In the case of machine learning, the developers of Python provide the package called scikit-learn.

3.3.2 scikit-learn

Scikit-learn is the machine learning library of Python. It is an open source project that is free and its source code is available to see and edit. As mentioned above, scikit-learn belongs to the part of Python that has a big active community and is being constantly improved and expanded by its developers. This library contains some of the most well-known and powerful algorithms along with well-written and comprehensive documentation about each of them. It also provides cross-validation tools to optimize the parameters of the algorithms, a procedure called Hyper-Parameter tuning. Apart from the official documentation found online, due to the big community, there are plenty of tutorials and code snippets for a new user to get acquainted with the library [16].

4 Experimental setup and description of experimental procedure

4.1 Experimental setup

4.1.1 Bently Nevada

The Marine Engineering Laboratory features the Bently Nevada Rotor Kit Model RK4. The device is designed and manufactured in the United States of America. It consists of a long, sturdy steel base, at one end of which has a special position for mounting a radial bearing. At the other end is a small electric motor with a maximum rotational speed of 10000 RPM. The main features of the base and shaft are shown in **Table 4.1**:

Base dimensions		**Rotor dimensions**	
Length	**780 mm**	**Length**	**45.7 mm**
Width	**340 mm**	**Diameter**	**24.95 mm, 10 mm**
Height	**165 mm**	**Weight**	**0.3626 kg**
Weight	**14.5 kg**		

Table 4.1 Base and Rotor Dimensions

The diameter of the shaft is 10mm throughout its length except for a small 25.4 mm part at the end with a diameter of 24.5 mm, designed this way so as to operate along with the bearing.

The device also has additional weights which can be used to modify axle loading. Specifically, there are two cylindrical masses 75 mm in diameter, weighing 0.800 kg each, with a length of 25.0 mm [17] [18].

4.1.2 Electric motor & Speed control unit

The rotor is driven by an electric motor whose main characteristics are shown in **Table 4.2** that follows.

Electric motor characteristics	
Max. rotational speed	**10000 rpm**
Weight	**14.5 kg**

Table 4.2 Electric Motor Characteristics

The motor speed is adjusted with the help of the RK4 Speed Control Unit, which has a display to indicate the speed. The operator can monitor the current rotational speed of the device or, by using appropriate buttons, set the desired operating speeds. The controller is informed about the shaft rpm with the help of a proximity probe mounted on a properly configured gear wheel. The controller can modify the axle speed and acceleration (or deceleration) to achieve the desired speed.

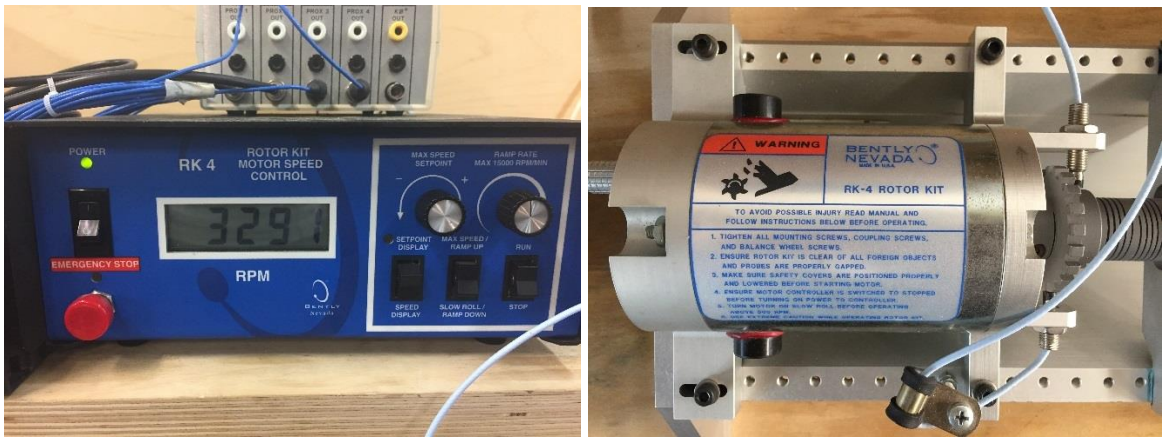


Figure 4.1 Bently Nevada RK-4 Speed Control unit (left) & Electric Motor (right)

4.1.3 Oil pump

The oil pump of the assembly is the Bently Nevada RK-4 Rotor Kit Oil Pump. This pump features a small, analogue type indicator to indicate oil pressure in either psi or kPa. On the other side, it has two sockets, on which the oil supply to and from the bearing is located.

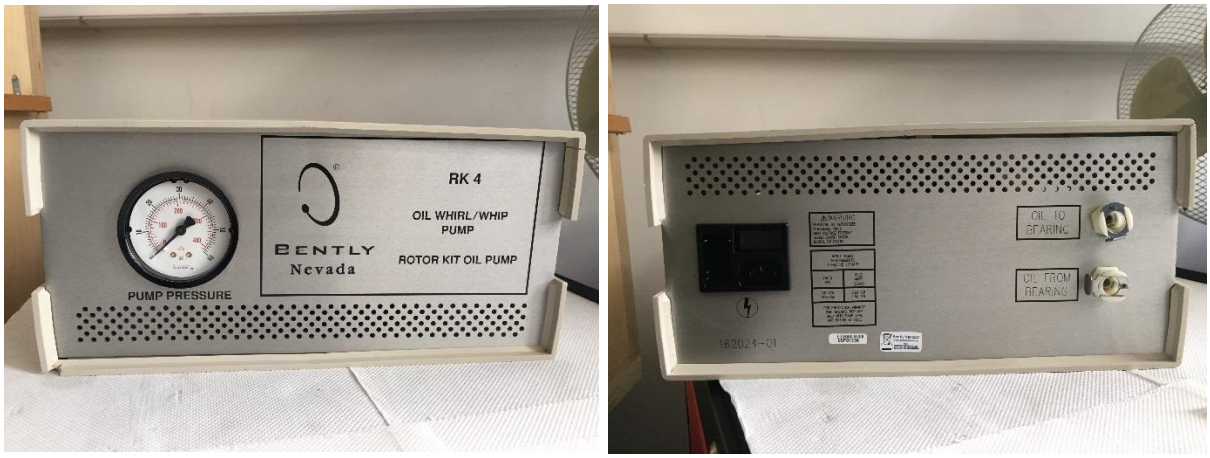


Figure 4.2 Bently Nevada RK-4 Oil Pump front side (left) and back side (right)

4.1.4 Journal Bearings

The first bearing used was ServoFluid Control Bearing which is designed, manufactured and assembled by Bently Nevada.

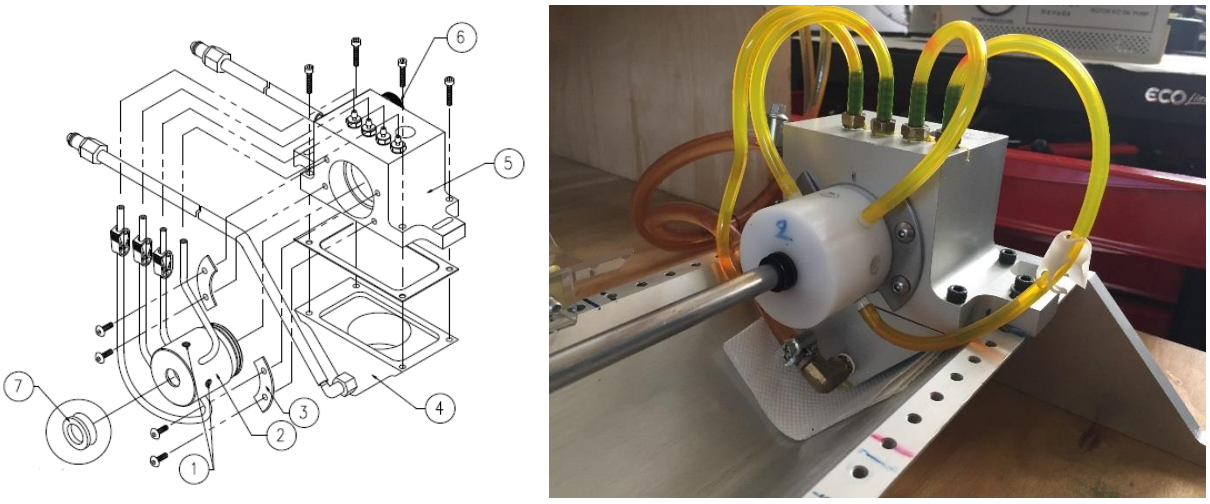


Figure 4.3 Bently Nevada RK-4 ServoFluid Control Bearing [17]

1) Proximity probes Mounting holes
2) Fluid Film Bearing Support
3) Bearing Retainer
4) Oil Reservoir
5) Bearing Support
6) Main Pressure Valve
7) Oil Bearing Seal

It should be noted that the bearing is made of Plexiglas plastic (Poly methyl methacrylate, PMMA) and its base, including the oil reservoir, is metallic. The major geometric features of the bearing are presented in the following table:

ServoFluid Control Bearing Bently Nevada: Nominal Dimensions

Inner diameter	**25.43 mm**
Length	**25 mm**
Clearance (c=R-r)	**0.225**

Table 4.3 Bently Nevada RK-4 ServoFluid Control Bearing Nominal Dimensions

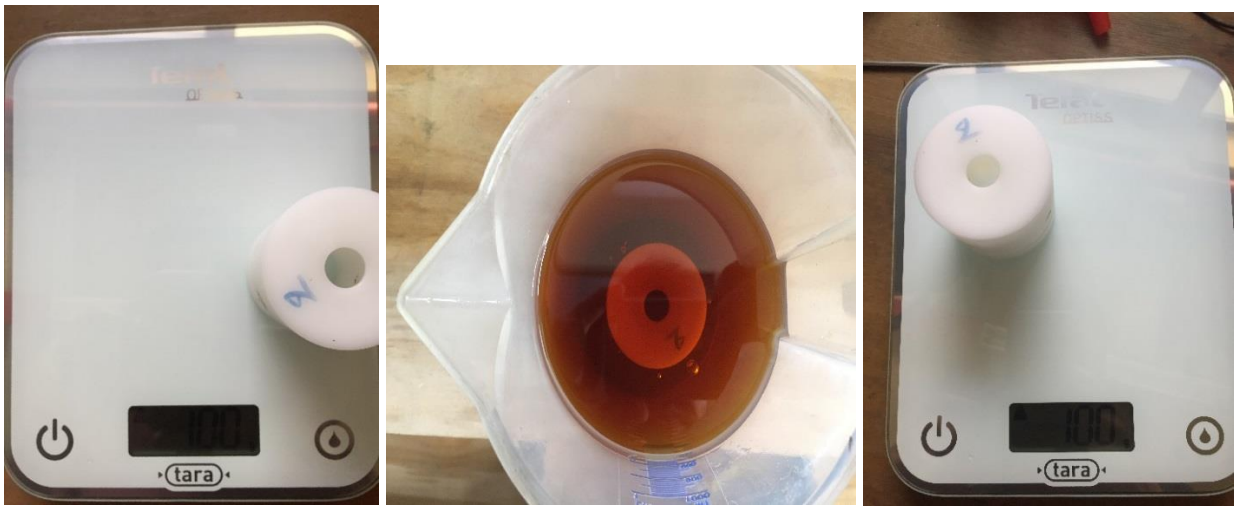
The second bearing that was used was manufactured according to the design plans of Bently Nevada, as described in Appendix A. The inner diameter of this bearing is measured with a three points internal micrometer (accuracy of 0.005mm) and is found slightly conical so the value shown in **Table 4.4** is the inner diameter in the center of the shaft. The material used was ACETAL (Polyoxymethylene, POM) and its oil resistance to many types of oil is high [19].The oil resistance

properties were also tested, confirming that the dimensions and properties of the bearing would not change throughout the experimental procedure. A TEFAL Optics scale with accuracy of 1 gram was used in order to determine the absorption of lubrication oil by the bearing. The bearing was weighted (100 grams) (**Figure 4.4, a**), fully immersed in lubrication oil (**Figure 4.4, b**) and the cleaned and weighed again (100 grams) (**Figure 4.4, c**). The second weighting confirmed the material properties found in the bibliography. The dimensions of the bearing were also measured and found unchanged. The experiment could induce wear to the bearing and it could be weighed again in order to determine the extent of the wear. However, the duration of the experiments was small so no such action was needed.

Custom ACETAL Bearing: Nominal Dimensions

Inner diameter	**25.43 mm**
Length	**25 mm**
Clearance (c=R-r)	**0.225**

Table 4.4 Custom Bearing Nominal Dimensions



(a) **(b)** **(c)**

Figure 4.4 Bearing weighting test

4.1.5 Soundproof Cover

In order to acquire better and more accurate results, a soundproof cover was designed and put over the bearing. This cover works beneficially in two ways:

- it prevents the sound pressure waves of external sources from scrambling the useful waves produced by the bearing assembly and
- it absorbs the waves that are produced by the assembly, not letting them get reflected and return as noise in the microphone.

Figure 4.5 Soundproof cover

4.1.6 Triaxial Accelerometer and Microphone

The triaxial accelerometer is a ICP® Model 356A02 with a hexagonal base (**Figure 4.6**). Its frequency range (±10%) spans between 0.5 and 6000 Hz and has a measurement range of ±500 g pk [20]. The hexagonal base of the accelerometer is mounted on the surface with the instant adhesive Loctite 454 and the accelerometer is then secured to the base.

The microphone is the ICP® 130D21 Array Microphone, a prepolarized condenser microphone coupled with a ICP® sensor powered preamp. Its frequency response (-2 to 5 dB) is 20 to 15000 Hz [21].

The calibration data for both sensors are found in Appendix A

Figure 4.6 ICP® Microphone (left) and Triaxial Accelerometer (right)

4.2 Conduct of the experiment

4.2.1 Preparation for the experiment

Before performing the experiments, it is necessary to properly prepare the experimental setup and the computer that will receive the results of the measurements of each experiment. In the previous paragraphs, parts of the Bently Nevada Rotor Kit 4 were described. The process of assembling the experimental setup as well as acquiring signals will be described in this section.

At one end of the assembly lies an electric motor which rotates the rotor. A simple radial bearing follows, then two cylindrical weights (the number depends on the desirable loading condition of the bearing) are adjusted, and at the other end of the shaft there is the radial sliding bearing. The motor is controlled by the RK4 Speed control unit, which enables it to determine the rotational speed and the axle acceleration / deceleration. There is a triaxial accelerometer mounted on the bearing support (number 1, **Figure 4.7**) located at the end of the shaft and a microphone mounted on the soundproof cover (number 2, **Figure 4.7**) above the bearing (number 3, **Figure 4.7**).

Figure 4.7 Bearing and Soundproof cover mounted with the sensors

The accelerometer and the microphone are connected to a Model 482A22 ICP® Sensor Signal Conditioner shown in the left side of **Figure 4.8** with four signal inputs and outputs, as shown on the right of **Figure 4.8**. The signal is transferred through BCN-type cables. On top of the left picture in **Figure 4.8** there is an indicator with three colors; yellow, green and red and the words OPEN, OK and SHORT respectively. Bellow there is a channel controller for each of the four channel inputs of the unit. While the conditioner is plugged in and the power switch is on, a vital check of the signal coming from the accelerometer and the microphone is performed. All four channels are checked on the three-colored indicator and the expected result is color green – OK [22]. The conditioner is then connected to the IoTech DaqBook 2000 that can gather signals from different signal conditioners and simultaneously send them to the data acquisition card.

Figure 4.8 ICP® Sensor Signal Conditioner

The data acquisition card used is the IoTech DaqBoard 2001 and constitutes the input of the analog signal in the computer. The software installed on the computer is the NI LabView 2017. The data acquisition is performed in Single-ended mode and refers to the circuits set-up in which voltage is measured between one signal line and common ground voltage (Vcm) [23]. The preparation of the computer is as follows:

1) Installation of the DaqBoard on the motherboard,
2) Installation of IoTech software for the correct card configuration (**Figure 4.9**),
3) Installation of NI LabView,
4) Identification of DaqBoard in LabView [24] (**Figure 4.10**)
5) Coding of the sampling process of the experiment in LabView

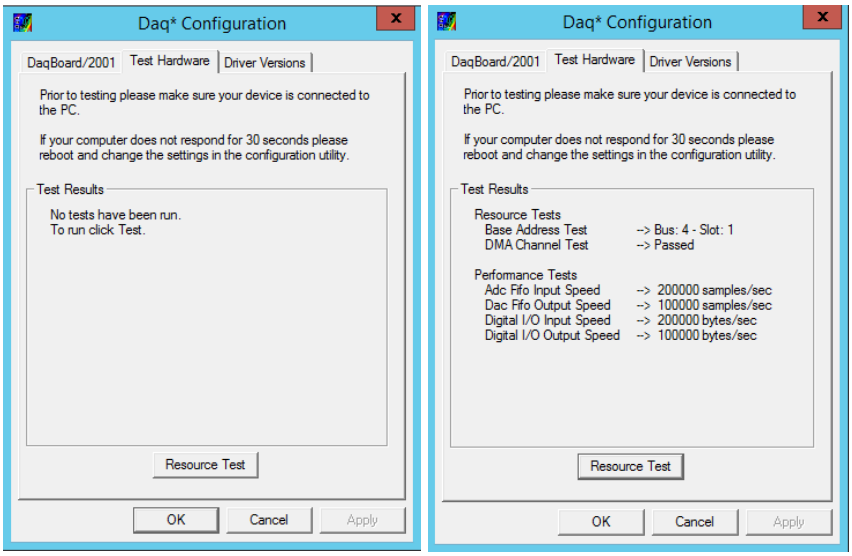


Figure 4.9 DaqBoard Configuration

In **Figure 4.10** is the Block Diagram developed in LabView. The basic sections to an acquisition, as defined by the LabView manual [25], are the following:

1) Initialize the Acquisition,
2) Configure Channels,
3) Define Trigger Method,
4) Configure Scan Properties,
5) Arm the Acquisition,
6) Read Scans,
7) Close the Acquisition.

These sections are all present in the Block Diagram shown in **Figure 4.10** with the same numbers as the list above and the blocks of code used were provided by DaqBoard 2001 LabView Support. The remaining numbered blocks were used for:

8) Time Delay,
9) Acquisition Duration control,
10) Signal Data Storage.

Apart from the numbered blocks, there are several indicators that are managed in the Front Panel of LabView and allow to determine whichever parameter of the experiment like the type of file the data will be stored in or the delimiter used to separate the data inside that file.

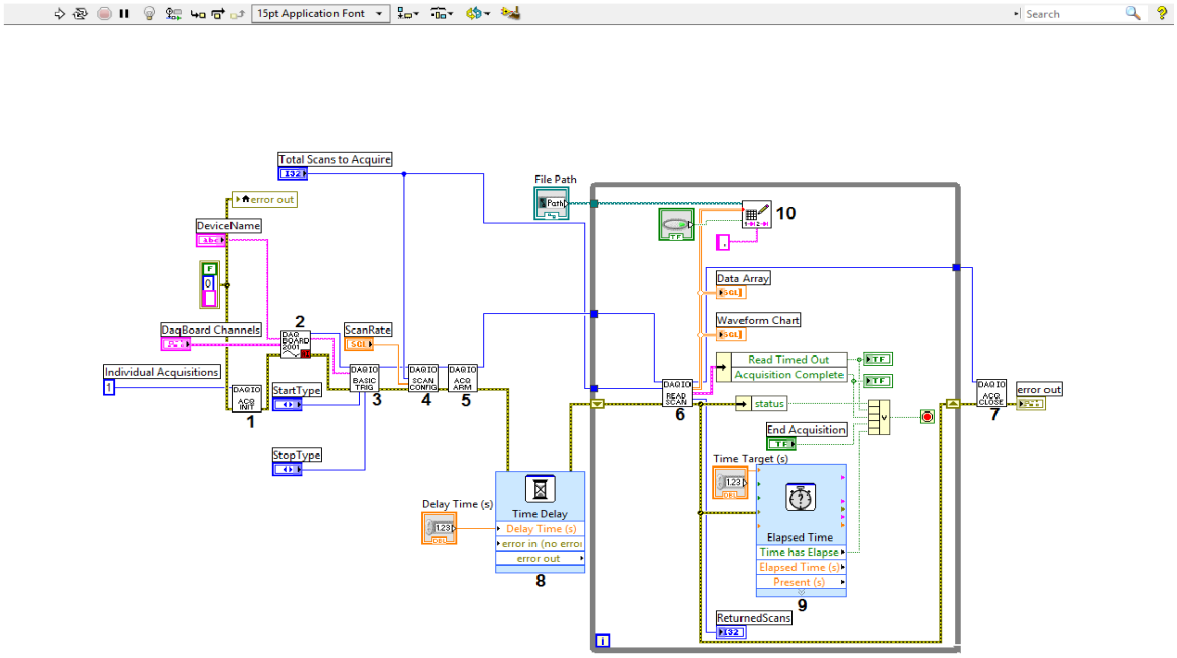


Figure 4.10 LabView Block Diagram

The sampling rate of each experiment is 1000 samples per second and the duration of the experiment is thirty seconds. Both of these values are set in the Front Panel of LabView. There is an example of the Front Panel of this experiment shown in **Figure 4.11** where the Scan Rate, the Start and Stop Types, the Configuration of the Channels and the Measurements Waveform Chart are visible.

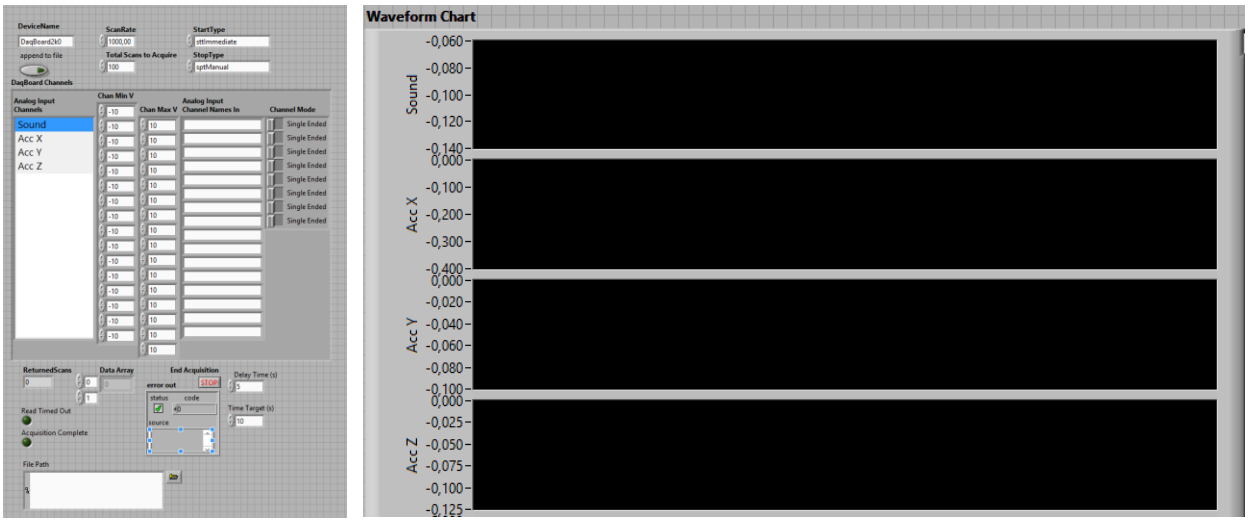


Figure 4.11 LabView Front Panel

4.2.2 Experimental procedure

The measurements are acquired from the microphone and the accelerometer with a sampling rate of 1000 samples/sec which are placed on the soundproof cover and the bearing support accordingly. The signal received by the sensors is forwarded through a data acquisition card to the computer and stored. **Figure 4.12** shows a typical series of acceleration z measurements. On the y-axis is the value of the signal in millivolts (mV) and on the x-axis is the time in seconds (s). The duration of this experiment is ten seconds so there are 10000 samples in this figure.

$$\text{Number of samples} = t \cdot 1000,\ t \text{ in sec}$$

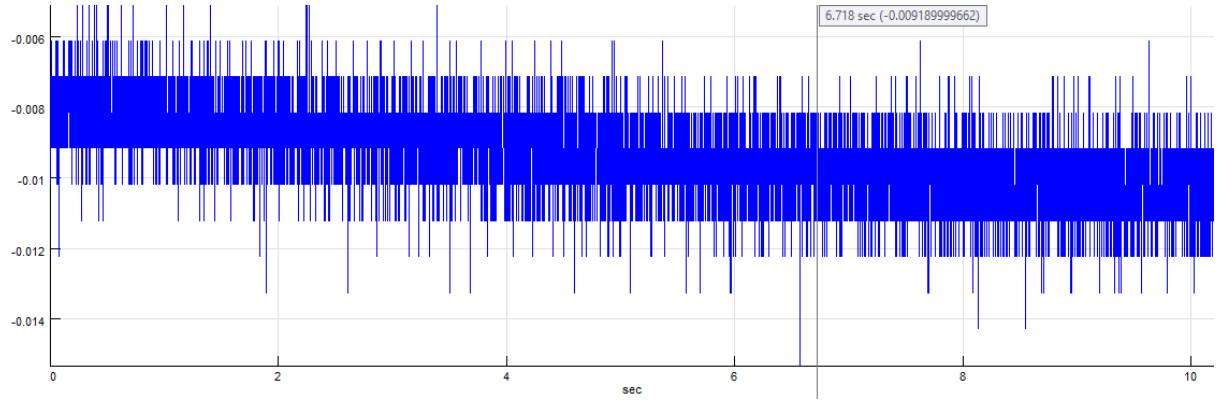


Figure 4.12 Typical acceleration z measurements

The first step of the experiment, before switching on every device, is to double-check every screw on the set-up and set the protection covers in place. Because of the high rotational speeds, a safety issue arises. After switching on every device, the test of the sensors as described in paragraph 4.2.1 takes place and a short test on LabView is run so as to make sure that the signals are coming through. If there is no problem then the soundproof cover along with the sensors are put in place. The oil pump motor is switched on so that the lubrication oil reaches the bearing and shaft and after making sure that there is a proper oil circulation the soundproof cover lid is placed. A slow rotational speed is set in the Speed control unit to reach a better lubrication of the bearing and shaft and to check for a possible leakage or other problems. In the case that no unexpected error occurs the actual experiments begins. It should be mentioned that the bearing at the other side of the shaft also needs lubrication and that is achieved with a plain lubrication spray.

The rotational speed of the shaft changes about every 2 minutes, so the shaft balances on the bearing at different values of dimensional eccentricity and angle of behavior. This affects the horizontal and vertical values of vibrations and the levels of the acoustic pressure produced. In particular, the rotational speed is increased from zero to 4500 RPM, moving the shaft closer to the center of the bearing and changing the measurements of the accelerometer and the microphone. The specific rotational speeds used for the experiments are 500 RPM, 1000 RPM, 1800 RPM, 2500 RPM, 3300 RPM, 4000 RPM, 4500 RPM. There is a small variation of the speed during each experiment of ± 10 RPM. The cylindrical masses mentioned in paragraph 4.1.1 and shown in **Figure 4.13**.

Figure 4.13 Cylindrical mass

To determine the mean load from the bearing, the Shaft Alignment Tool of the Marine Engineering Laboratory (L.M.E.) is used. Three loading conditions are modeled: no cylindrical mass added (**Figure 4.16**), one cylindrical mass (**Figure 4.17**) and two cylindrical masses added (**Figure 4.18**). The following steps are needed for the proper modeling of the shaft:

- Selection of the number of beams that combined create the shaft,
- Determination of beam properties, (**Figure 4.14**),
- Determination of support nodes and external forces application nodes
- Apply forces and specify support points (**Figure 4.15**)

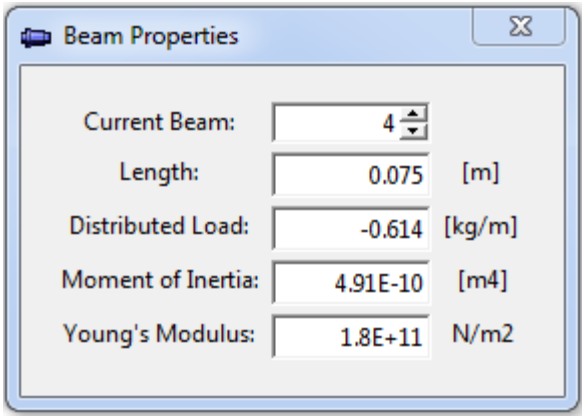


Figure 4.14 Shaft Alignment: Beam Properties

In **Figure 4.14**, Current Beam refers to the part of the shaft that is being defined, Length refers to the length of this part, Moment of Inertia refers to I [m^4] and Young's Modulus refers to E [N/m^2]. The calculation of the Distributed Load section is done in the following way:

$$w = \frac{\pi \cdot D^2}{4} p$$

Where,

- w, the distributed weight of the shaft
- D, the shaft diameter
- *p*, the shaft density

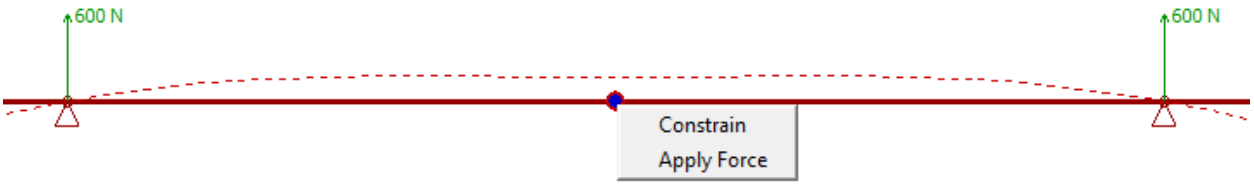


Figure 4.15 Shaft Alignment: Force Application and Support Points Specification (Constrain)

It should be noted that while modeling with the help of the Shaft Alignment program, the following assumption was made: The axial support point position inside the sliding bearing was taken half the length of the bearing. This assumption applies only if the shaft is fully aligned with the bearing. In practice, however, due to the bending arrow of the shaft, there is a shaft misalignment inside the bearing, which is characterized by the relative angle of the misalignment γ. Therefore, the pressure development in the radial sliding bearing will not be symmetrical, resulting in the position of the bearing in which the resultant support force is exerted not to be identical to the center of the bearing along its length.

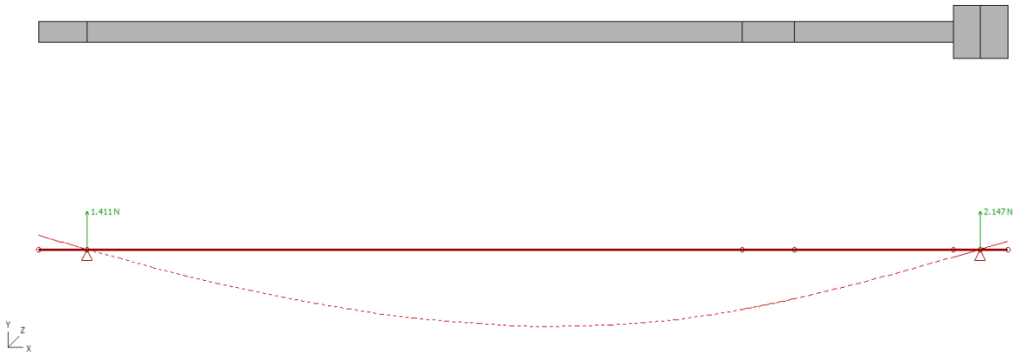


Figure 4.16 Shaft Alignment: Modeling of the unloaded shaft

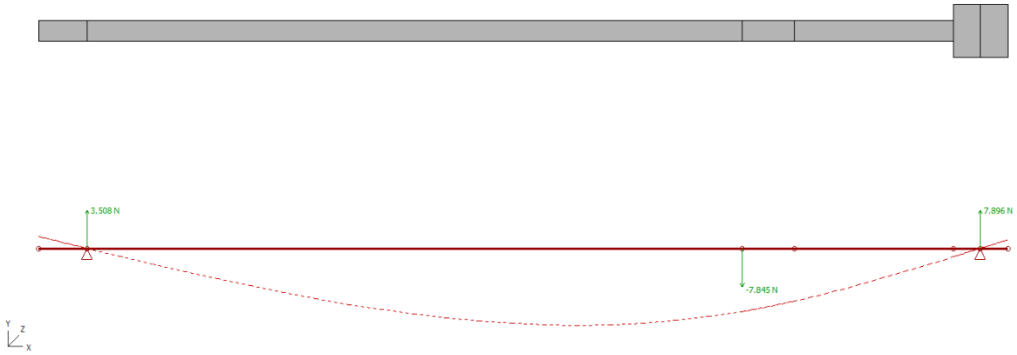


Figure 4.17 Shaft Alignment: Modeling of the single-loaded shaft

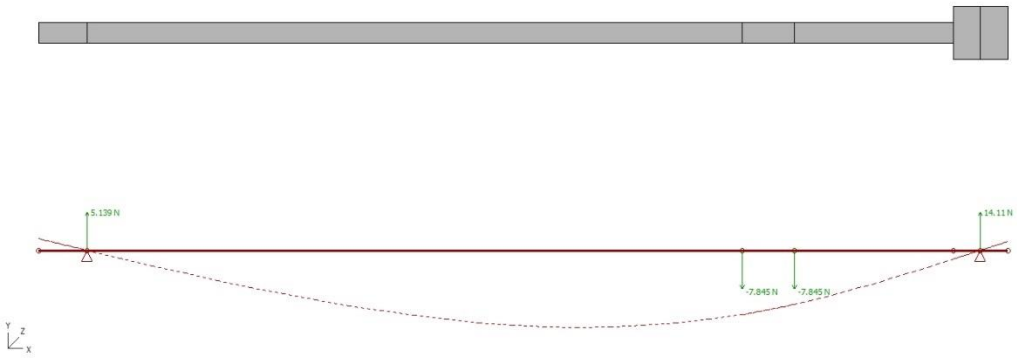


Figure 4.18 Shaft Alignment: Modeling of the double-loaded shaft

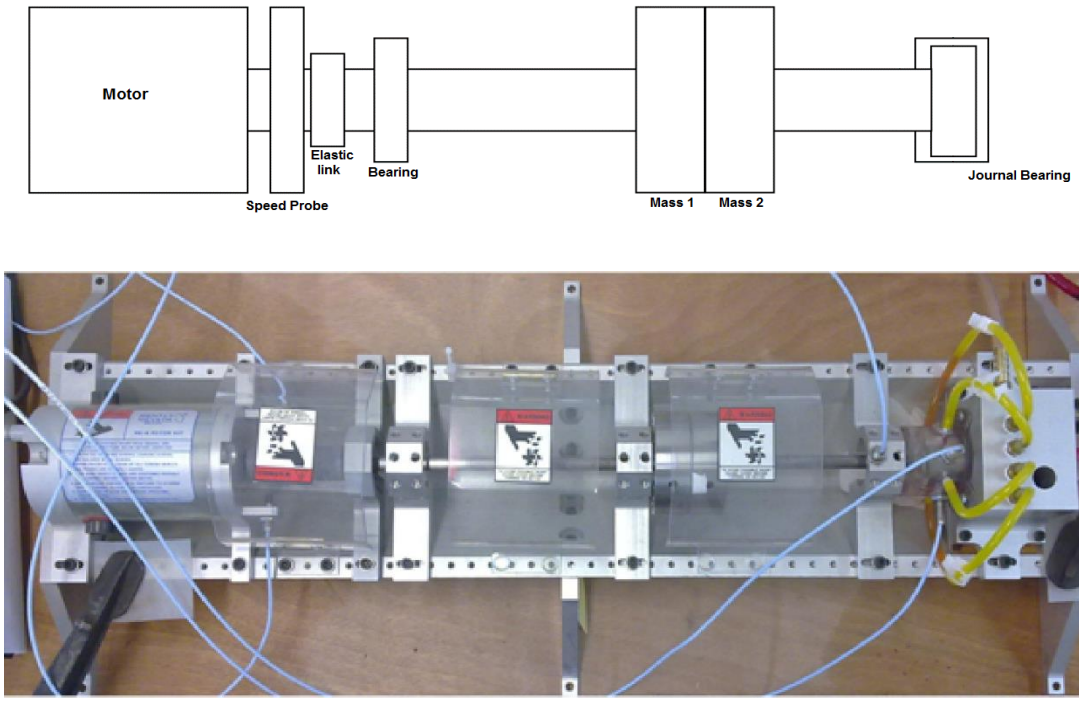


Figure 4.19 Bently Nevada Rotor Kit 4 Assembly

The file produced after each experiment is a comma delimited values file (.csv) with every line containing an instance with five values; sound, acceleration x, acceleration y, acceleration z in mV and rotational speed in RPM. The succession of the five values is determined in the Front Panel along with a title, if needed, for each column.

5 Data analysis - Results

5.1 Raw data processing and Octave Band analysis

Appropriate processing is performed for each set of experimental results to determine the loading condition of the bearing. The first stage of this processing includes the conversion of the raw data acquired through the acquisition card and stored in the .csv files. This data, as mentioned above, are measured in mV. According to the Calibration Certificate of each axis of the triaxial accelerometer and of the microphone, the following multiplication should be conducted:

Calibration Data

X axis	**1.002 mV/m/s^2**
Y axis	**0.990 mV/m/s^2**
Z axis	**0.979 mV/m/s^2**
Microphone	**33.8 mV/Pa**

Table 5.1 Microphone and Accelerometer Calibration Data

As the data files contain an enormous number of instances and there are many data files as well, a small Python script is prepared to deal with these conversions. Once a file is read, another file is produced with the proper data. The latter file is then fed to a second Python script that cuts small samples of one, three of five seconds in duration of one value e.g. acceleration x. These different values of duration aim to find the minimum amount of data needed for the machine learning algorithm training. This minimum value is not known and constitutes a highly important parameter of the data processing and will be shown in more detail in paragraph 5.2 and 5.3.

The next step of the processing is requiring a choice between many signal processing methods. In many papers, the RMS or the Peak-to-Peak values of the signal are used as features to build the training dataset of the algorithms. However, in other techniques, noises and vibration waves are broken up into sine waves and transferred to the frequency domain [3]. In this study, the octave-band type analysis is used to filter the acceleration and sound pressure signals. This type of analysis is chosen for two main reasons:

1) The frequency domain reveals frequency components and their individual amplitudes,
2) It can be easily combined with machine learning (in comparison to simple FFT analysis).

Vibration signals of interest can extend between frequencies from near 0 Hz to around 70 Hz and noise signals to can reach very high frequencies depending on the application (e.g. aircrafts generate high frequency noise) [26]. A complex harmonic signal might have many frequency components and in the real world that is normally the case. Industrial noise and vibration are either complex signals or random signals and should be analyzed in frequency bands. Out of many types of frequency bands, the octave bands are the most widely used for frequency analysis and are concerned with halving or doubling the frequency. In this study, a narrower band analysis is required and thus the one-third-octave bands are used.

If the center frequency of a band is defined as f_0, the upper frequency limit of the band as f_u and the lower frequency limit as f_1, then [26]:

$$f_u = 2^n f_1,$$

where n can be any number and equals to 1/3 in the case of the one-third-octave bands. The center frequency is the geometric mean of the upper and lower limit so:

$$f_0 = (f_1 f_u)^{1/2},$$

$$f_0 = \sqrt{2}\, f_1 = f_u / \sqrt{2}$$

After the bands are created, the signal passes through bandpass filters that correspond to each band. The power/energy spectrum level is then calculated for each band by placing a logarithm on the mean-squared pressure of the band (p_{i^2}) divided by the squared reference pressure (p_{ref}^2) as shown below:

$$L_{\text{p band}} = 10 \log_{10} (p_1^2 / p_{ref}^2)$$

where p_{ref}^2 is an internationally accepted value equal to $2x10^{-5}\ N\ m^{-2}$.

Each band will now be presented as a feature for the training process of the machine learning algorithms. The following tables show the match between the bands (central frequencies) and the features' names. Because there is overlap between sound and vibration bands, some bands will have double names. This correction is necessary for the case studies where both sound and vibration signals are used.

Vibration Features nomenclature

Feature	Nominal Band Center Frequency (Hz)
1	**1.00**
2	**1.25**
3	**1.60**
4	**2.00**
5	**2.50**
6	**3.15**
7	**4.00**
8	**5.00**
9	**6.30**
10	**8.00**
11	**10.0**
12	**12.5**

13	**16.0**
14	**20.0**
15	**25.0**
16	**31.5**
17	**40.0**
18	**50.0**
19	**63.0**

Sound Features nomenclature

Feature	**Nominal Band Center Frequency (Hz)**
20	**31.5**
21	**40.0**
22	**50.0**
23	**63.0**
24	**80.0**
25	**100**
26	**125**
27	**160**
28	**200**
29	**250**
30	**315**
31	**400**

Table 5.2 Vibration and Sound Feature Nomenclature

The frequency of a sine wave can only be defined if at least two samples of it are included in one sampling cycle. As a result, the upper frequency limit that can be safely defined is half of the sampling rate. This frequency is referred to as Nyquist cutoff frequency and the procedure of applying this sort of lowpass filter is called anti-aliasing [26]. The band with 400 Hz as a center frequency has an upper limit of 447 Hz. The next band would extend from 447 Hz to 562 Hz so the Nyquist frequency would be surpassed and the results would change. In **Figure 5.1** is an example of the one-third octave analysis for a sound signal printed with Python matplotlib. The nominal band center frequencies are on the x-axis and the acoustic level is on y-axis.

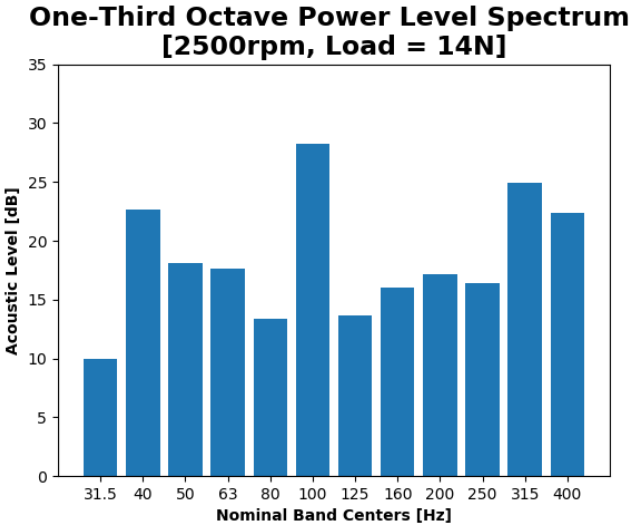


Figure 5.1 One-Third Octave Power Level Spectrum of a sound signal example

5.2 Bearing loading condition determination algorithm

The actual algorithm is a combination of all the steps that were described above. The time series signal is converted to the frequency domain with the one-third octave filter. Then, the bands are transferred to the feature space as shown in **Table 5.1**. At this point, the training data set is ready to be fed to the algorithm.

However, the quantity of the training data needed is still unknown and should be examined. The experiments last for thirty seconds and several samples should be extracted from each experiment dataset before the one-third octave filter is applied. If the duration of the sample is not long enough then the low frequencies will be filtered out because those sine waves will not manage to appear at least two times in the sample and thus not be noticed. These low frequencies are needed for the vibration signal only. One second in a sample's duration corresponds to a signal frequency of 2 Hz; two seconds correspond to 1 Hz and three seconds correspond to 0.67 Hz.

The number of these samples is different in each case study and is defined at the beginning of the case. Finally, it should be mentioned that the testing data are divided into two categories. In the first category, the testing data is randomly chosen and removed from the training dataset by a cross validation tool of Python. This data is then used to evaluate the algorithm's performance. The second category includes the first one but has an extra particularity. An extra set of testing data is used coming from a completely unknown part of the experimental dataset e.g. an unknown combination of rotational speed – load or an unknown part of a rotational speed's dataset or a different bearing's experimental dataset.

5.3 Case Studies

In this paragraph, all the different processing scenarios and case studies will be presented. The aim is to highlight the strong characteristics of this technique and its potential. The results' structure varies depending on the type of the algorithm used and their scientific content. There will be two tables in the beginning of each case study that will explain which the training and testing data are and what the feature space consists of. **Figure 5.2** shows the training and testing data mapping and **Figure 5.3** shows how the combination of the two will be used from here on. The second table appears in **Figure 5.4** and **Figure 5.5**. The former shows that feature 10,11 and 12 are used and the latter shows that feature 25, 26 and 27 are used. It should be noted that each algorithm was run multiple times in order to strengthen the reliability of the results.

Training Data

RPM / Load (N)	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Testing Data

RPM / Load (N)	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Figure 5.2 Training Data and Testing Data Mapping

Training & Testing Data

RPM / Load	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Figure 5.3 Training and Testing Data Mapping Combination

Vibration Feature Space

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 5.4 Vibration Feature Space Mapping

Sound Feature Space

Feature	20	21	22	23	24	25	26	27	28	29	30	31

Figure 5.5 Sound Feature Space Mapping

A confusion matrix is a table that is used in machine learning in order to visualize the performance of a classification algorithm when the actual values of the labels of the tested data is known. In the diagonal of the table are the correct answers (white color) and in the sum.col and sum.lin are overall percentages of the columns and lines respectively (the right answers are green and the

wrong are red). **Figure 5.6** shows an example of a confusion matrix. The y-axis shows the predicted classes and the x-axis shows the actual values. The example has an accuracy of 90.48%. Class A was wrongly predicted 2 times in the place of Class B.

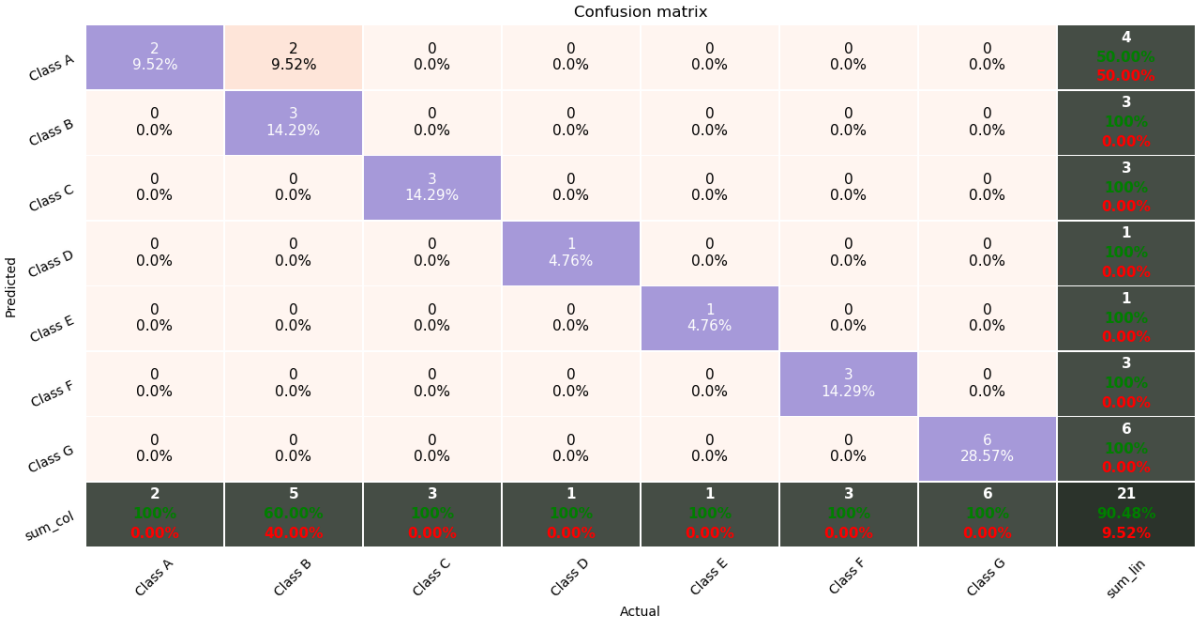


Figure 5.6 Confusion Matrix Example

5.3.1 Case Study #1, Acceleration X, Y, Z and Sound Evaluation

In order to determine which of the four signals (acceleration x, y, z and sound) performs better for the task at hand, a simple test is run that uses as input one of these values every time. The label is the mean load of the bearing. After the hyper-parameter tuning is applied, the algorithms are run and the results are evaluated. The features and RPM-Load combinations used for each value are the same so they will be illustrated once.

Training & Testing Data

RPM / Load (N)	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Sound Feature Space

Feature	20	21	22	23	24	25	26	27	28	29	30	31

Acceleration x, y, z Feature Space

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 5.7 Training & Testing Data and Acceleration & Sound Feature Space Mapping #1

The score for each classification and regression algorithm are shown in **Table 5.3**. It is easily noticed that the sound signal produces better results without overfitting to the data. As expected, the acceleration z signal comes second best and acceleration x and y follow. The algorithms used are the Random Forest Classifier (RFC), the *k*-Nearest Neighbors Classifier (KNNC) and the Gradient Boosting Regressor (GBR). It should be noted that the acceleration x and y results are unstable and inconsistent.

Vibration & Sound Accuracy

	AccX	AccY	AccZ	Sound
RFC	70%	80%	85%	98%
KNNC	80%	80%	85%	98%
GBR	66%	53%	67%	99%

Table 5.3 Vibration and Sound Accuracy

The case studies that follow use the sound and acceleration z signals as an input for the algorithms' training. In some cases, acceleration z signals perform adequately but in some cases, they do not so in these cases a combination of the two signals will be used in order to examine the results.

The next column charts show the acoustic pressure level after applying the One-third Octave filter. The most important features for the algorithms are feature 25, feature 28 and feature 31. If examined, the following charts can illustrate why the algorithms choose this features as the most important in the decision-making process.

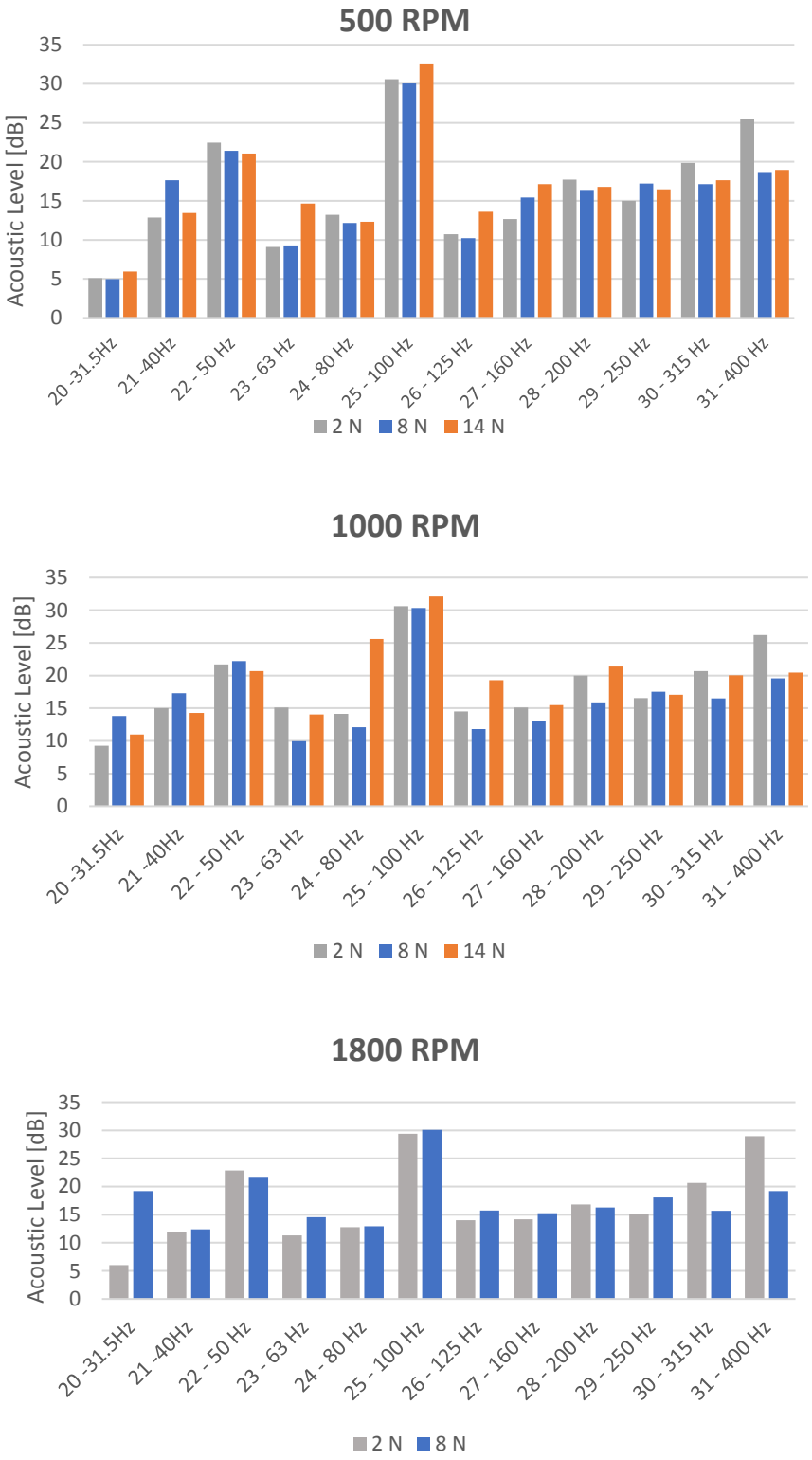


Figure 5.8 One-Third Octave Power Level Spectrum of a sound signal #1

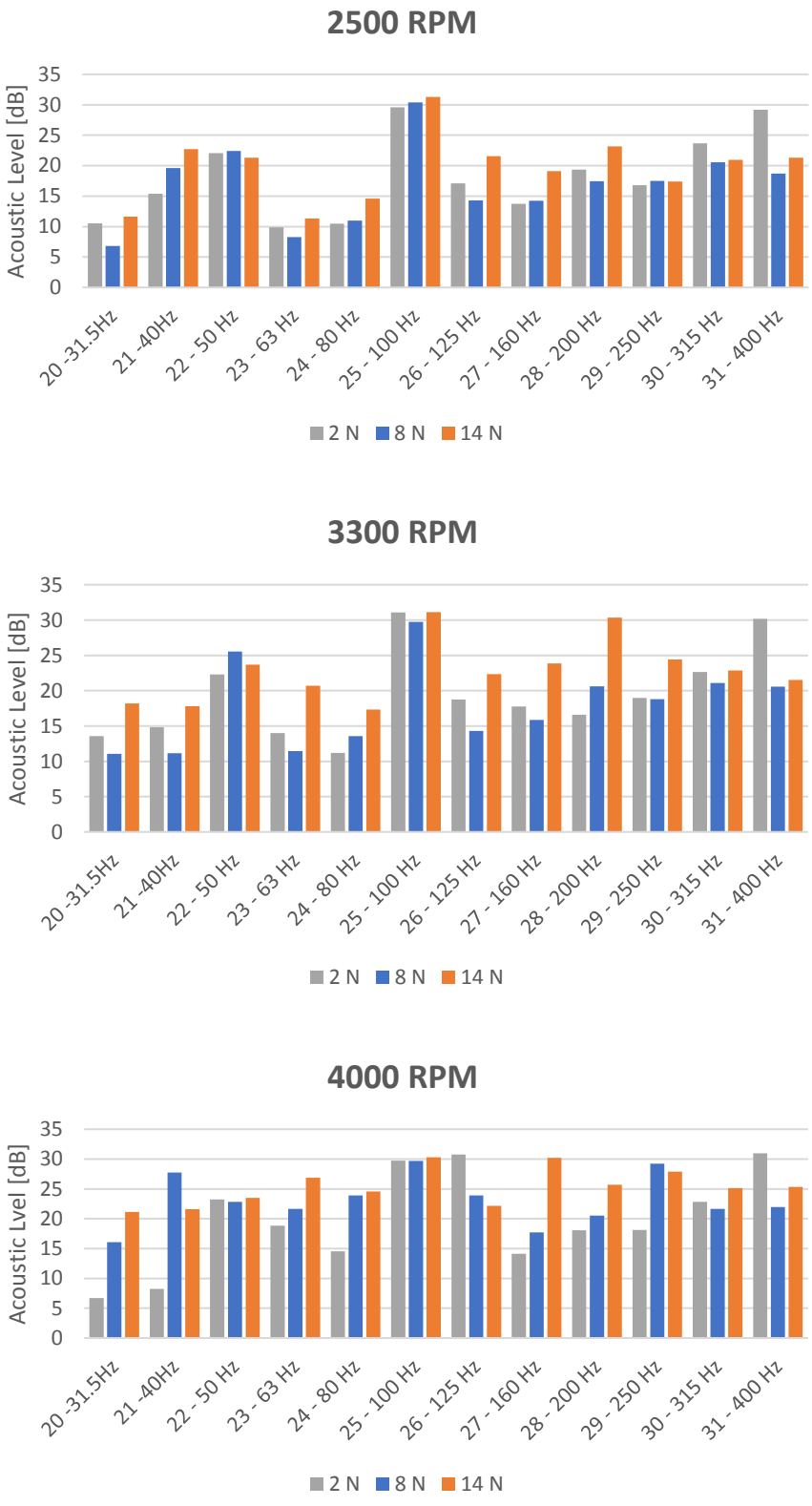


Figure 5.9 One-Third Octave Power Level Spectrum of a sound signal #1

5.3.2 Case Study #2, RPM Determination

This case study aims to determine the rotational speed of the shaft by using the sound signal produced by the bearing. The training samples have a one second duration and thirty samples have been used for each rotational speed. The testing data used are randomly chosen from the training data pool and constitutes the 10% of the testing data volume. The actual RPM value and the algorithm's prediction are then presented in order to evaluate its accuracy.

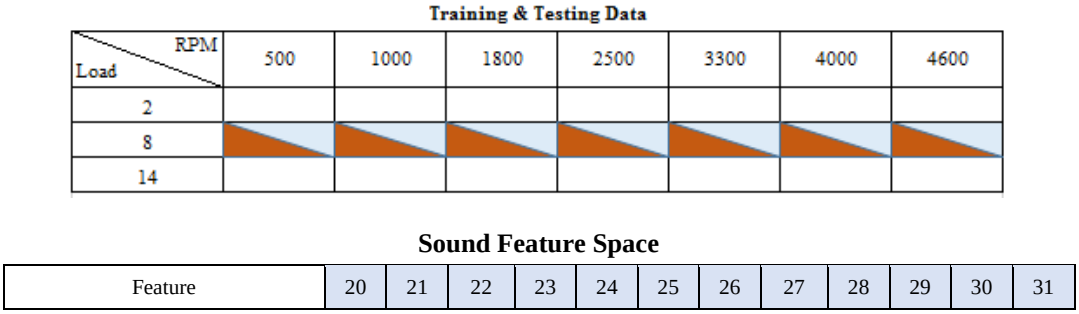


Figure 5.10 Training & Testing Data and Sound Feature Space Mapping #2

It is obvious from **Figure 5.11** that the frequency signature of each rotational speed differs; thus, the results of this case study were expected. **Figure 5.12** is a chart that shows the importance of each feature in the decision-making processes of the algorithms. The algorithms used are the RFC and the GBR. The difference in the training process becomes clear; RFC distributes the importance to all features, whereas GBR only to few. For both algorithms, the max tree depth equals 2 and the number of estimators (trees) equals to 50 in order to avoid overfitting.

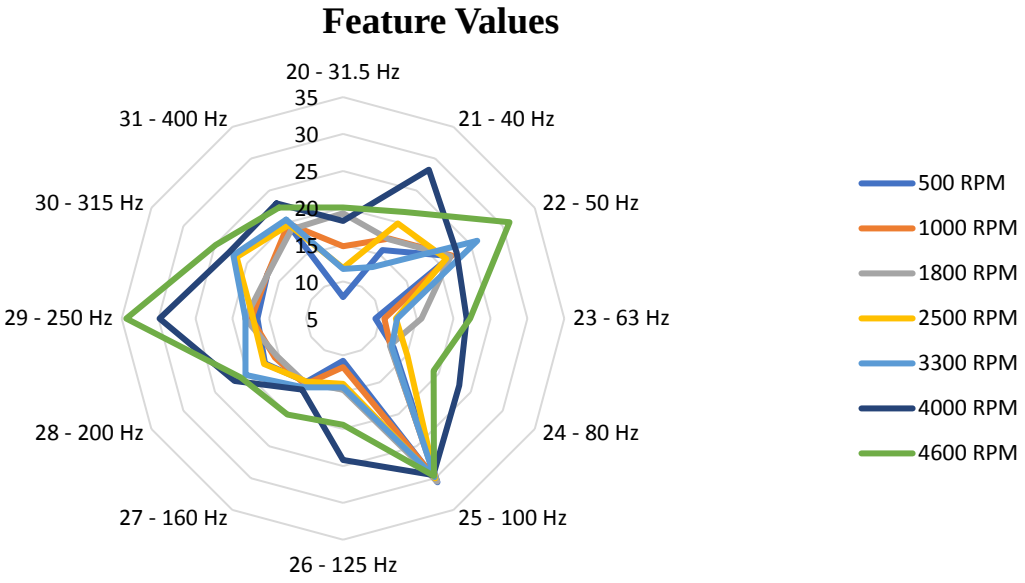


Figure 5.11 Feature Values

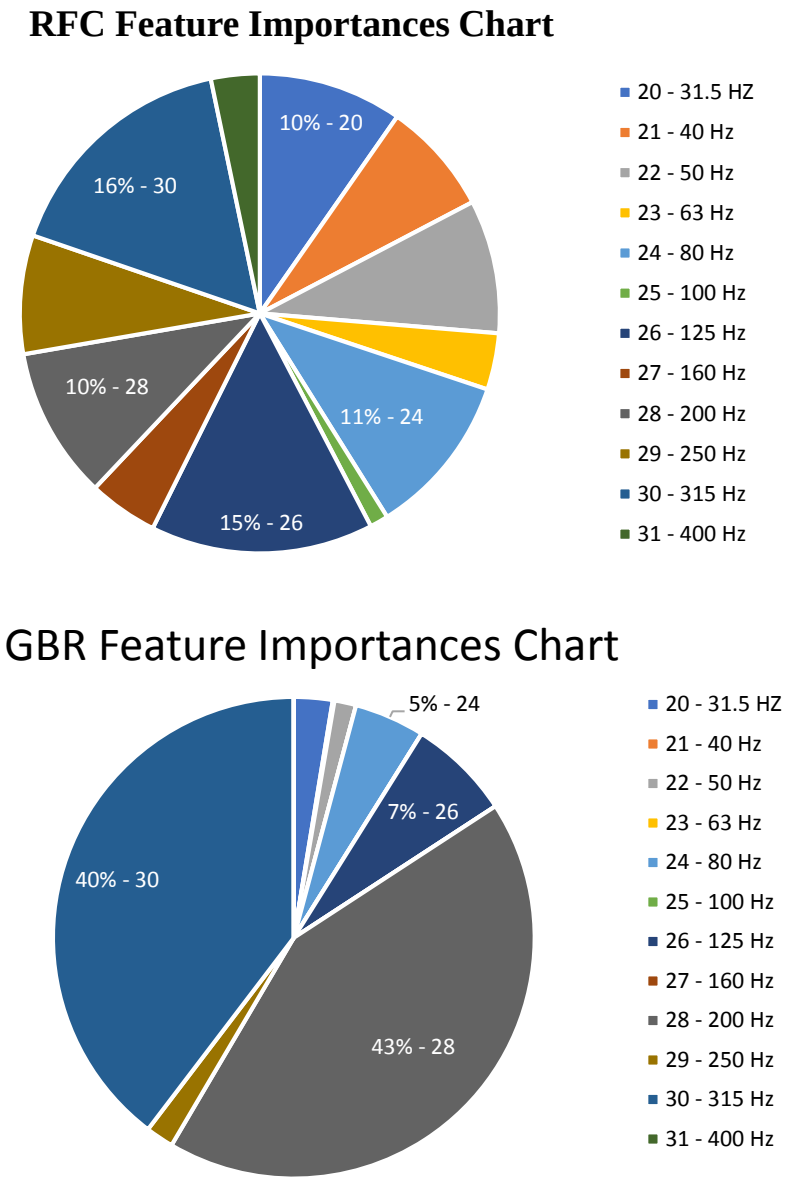


Figure 5.12 Feature Importance of RFC and GBR #2

The confusion matrix in **Figure 5.13** shows the results of the RFC (95.24% accuracy) and the **Table 5.4** shows the results of the GBR (98.9% accuracy).

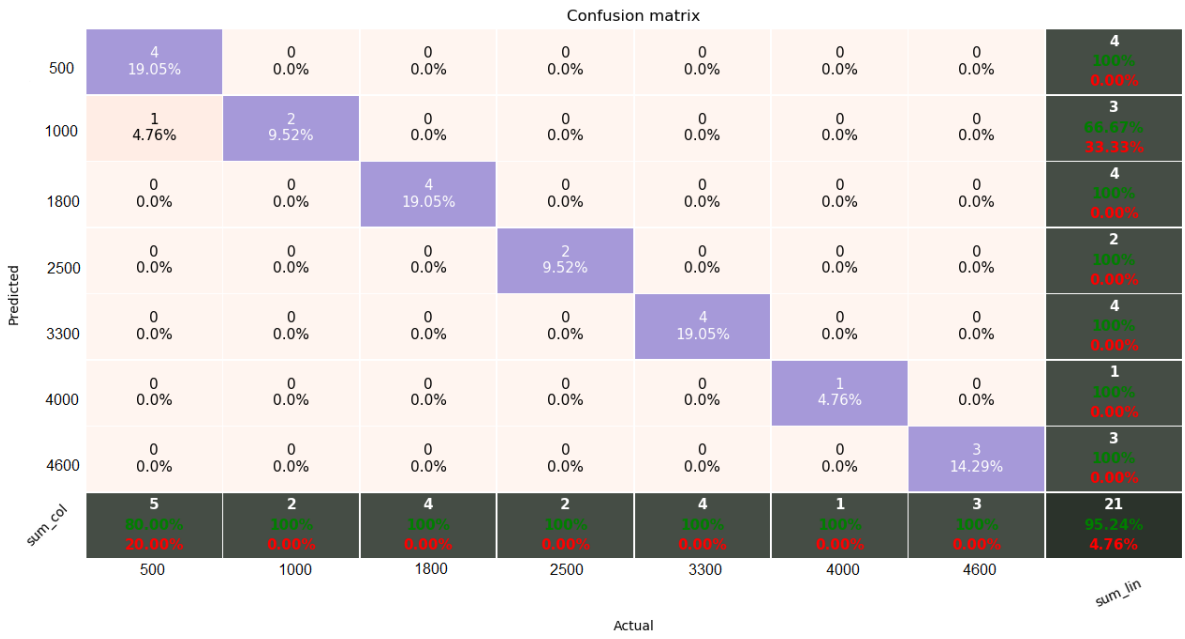


Figure 5.13 RFC Confusion Matrix #2

RPM Prediction Table

Actual	500	500	500	500	500	1000	1000
Predicted	546	515	560	542	514	1013	1012
Actual	1800	1800	1800	1800	2500	2500	3300
Predicted	1803	1788	1815	1764	2504	2553	3296
Actual	3300	3300	3300	4000	4600	4600	4600
Predicted	3299	3299	3308	3993	4572	4572	4572

Table 5.4 GBR Prediction Table #2

The classification problem has no obvious usage but was very accurate, as predicted by observing the feature values. The regression problem has high variance in the 500 RPM. The 4600 RPM is predicted three times as 4572 RPM which, after analyzing the data, is due to the similar values of features 28 and 30 of the data at hand and the high importance of these features. Because of the high precision equipment that already exists in measuring the rotational speed of a shaft, most of the next case studies will use the RPM as a feature.

5.3.3 Case Study #3, Load Determination via Acceleration Z signals

In this case study, the goal is to determine the loading condition of the bearing by using the acceleration z signal. The thirty samples per load per rotational speed used for the training have a duration of 3 seconds. The testing data for the first sub-case is randomly selected from the pool of the training data and constitute the 20% of the training data volume. The testing data of the second sub-case is produced from the last three experimental seconds of the 4000 RPM rotational speed with a mean load of 2N. These three seconds of instances are not part of the experimental data used to create the training data pool. The algorithms used are the RFC, the KNNC and the GBR.

Figure 5.14 shows the training and testing data mapping. **Figure 5.15** shows the feature importance for RFC and GBR. The feature values will not be illustrated due to the difficult simultaneous plotting of all the features of all the loading cases at once. However, **Figure 5.16** shows a 3D visualization of the multi-dimensional samples, where the three different loading cases are distinguished through color. Feature 12 and 14 are chosen because of the high importance they have for RFC.

Training & Testing Data

Load (N) \ RPM	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Acceleration z Feature Space

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 5.14 Training & Testing Data and Vibration Feature Space Mapping #3

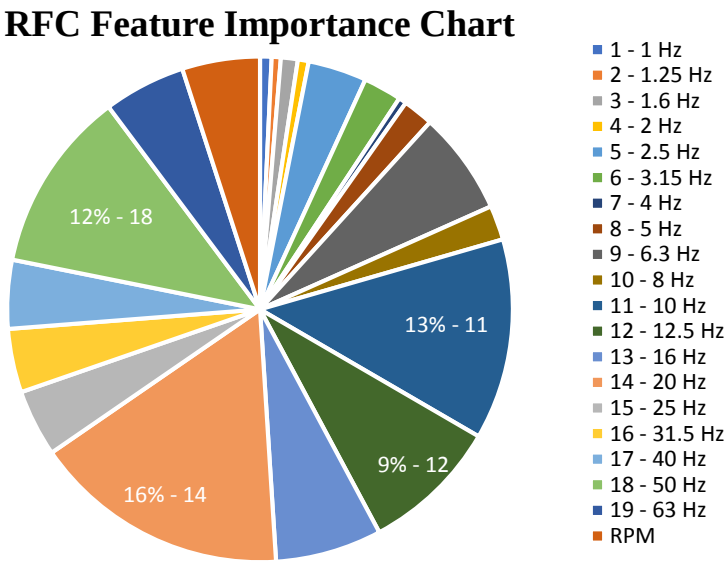


Figure 5.15 Feature Importance of RFC #3

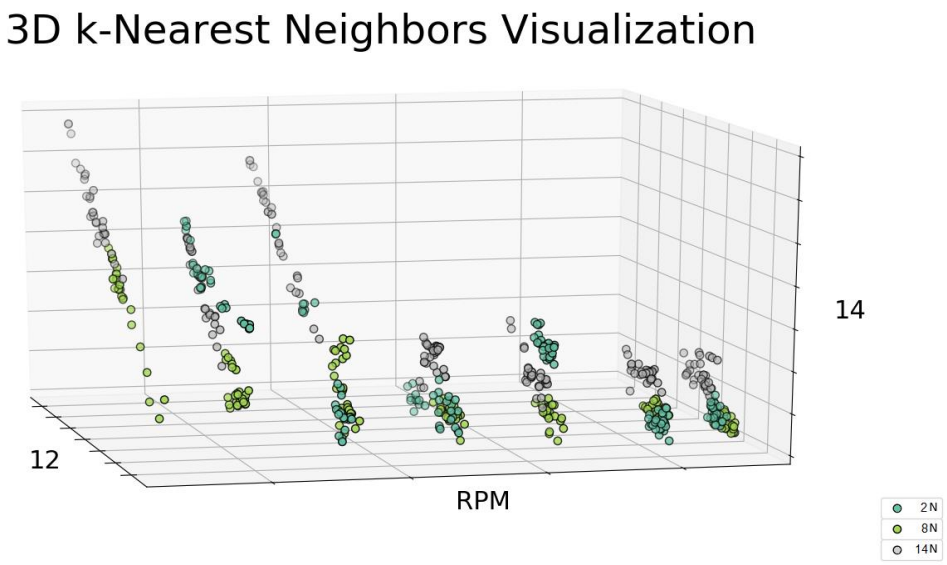


Figure 5.16 3D KNNC Visualization of RPM, Feature 12 and Feature 14 #3

In **Figure 5.16** the three dimensions out of the 20-dimensional mapping produced by the KNNC are shown. As described above, the value of *k* effects significantly the results of the algorithm and should be paid the appropriate attention during the training. In this case study, the *k* equals to 4.

The results produced by each algorithm are shown in **Figure 5.17** and **Figure 5.18**. KNNC has an accuracy from 93 to 97 % and RFC from 77 to 83 %. The results do not vary if some of the features are excluded from the algorithms' training and testing and this feature selection procedure will be examined in the forthcoming case studies. GBR is not presented due to the low accuracy and high variance of the results.

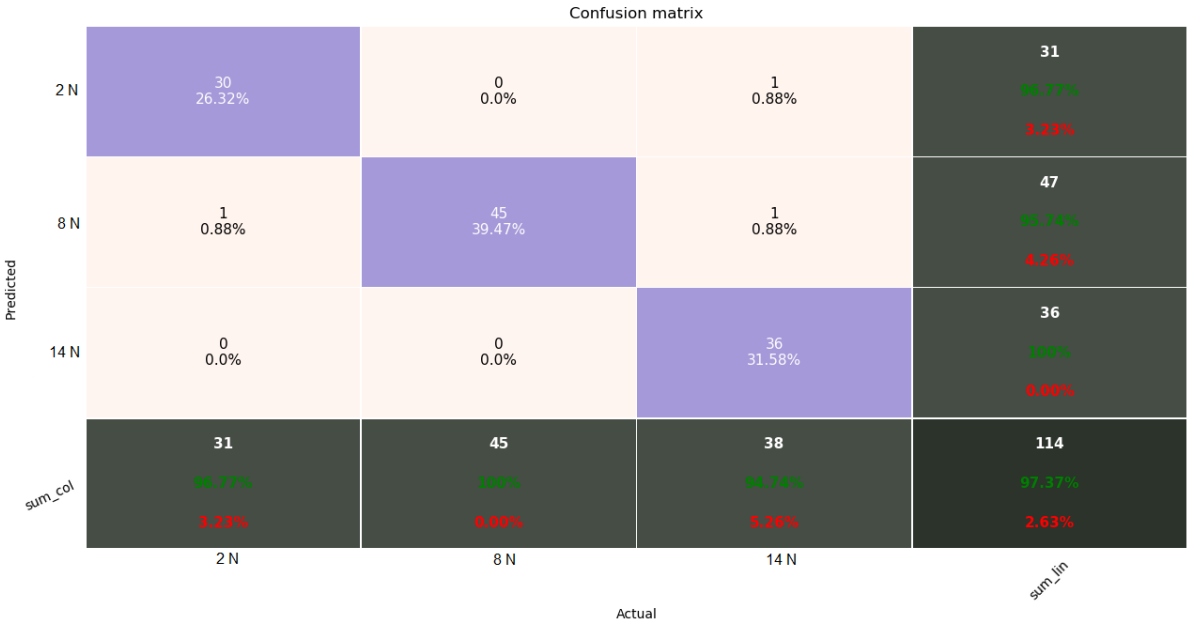


Figure 5.17 KNNC Confusion Matrix #3

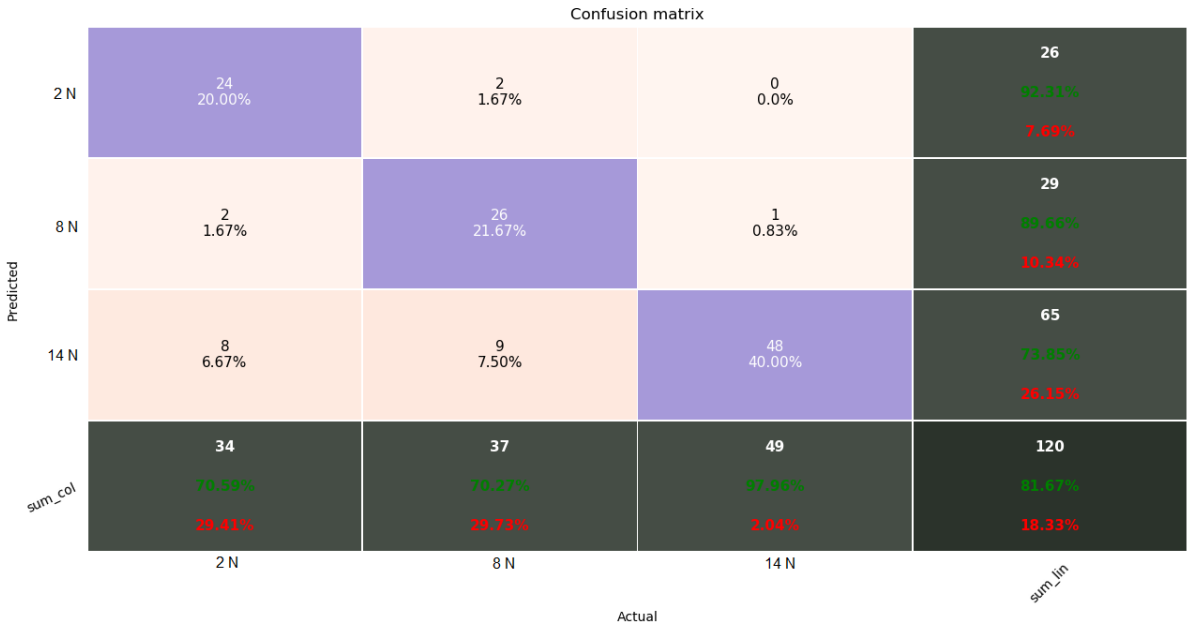


Figure 5.18 RFC Confusion Matrix #3

The second sub-case did not produce any solid results. The actual value was 2N and the algorithms gave a prediction of 8N or 14N in most of the cases. This showed that a better mapping of the bearing's operation is needed in order to explore the full potential of this technique when using acceleration z signals.

5.3.4 Case Study #4, Load Determination via Sound signals

This case study aims to determine the loading condition of the bearing by using the sound signals acquired in the experimental procedure. The data format is same as in case study #3:

- The thirty samples per load per rotational speed used for the training have a duration of 3 seconds
- The testing data for the first sub-case is randomly selected from the pool of the training data and constitute the 20% of the training data volume
- The testing data of the second sub-case is a produced from the last three experimental seconds of the 4000 RPM rotational speed with a mean load of 2N, with these three seconds of instances not being part of the training data pool
- The algorithms used are the RFC, the KNNC and the GBR.

The training and testing data as well as the sound features are shown in **Figure 5.19**. In **Figure 5.20** appears the feature importance chart for each algorithm. While the RFC, due to its inherent ability to divide the importance between the features, has given at least a small percent of importance to all the features, it has given the highest percentage to features 25, 28 and 31, the same features the GBR has highlighted as the most important. In should be noted that in the GBR Feature importance chat only the three most important features are visible because the rest have a very low percentage of importance.

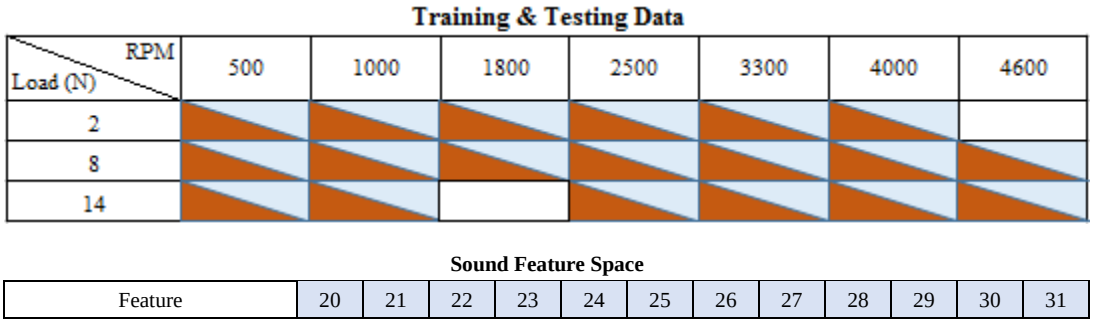


Figure 5.19 Training & Testing Data and Sound Feature Space Mapping #4

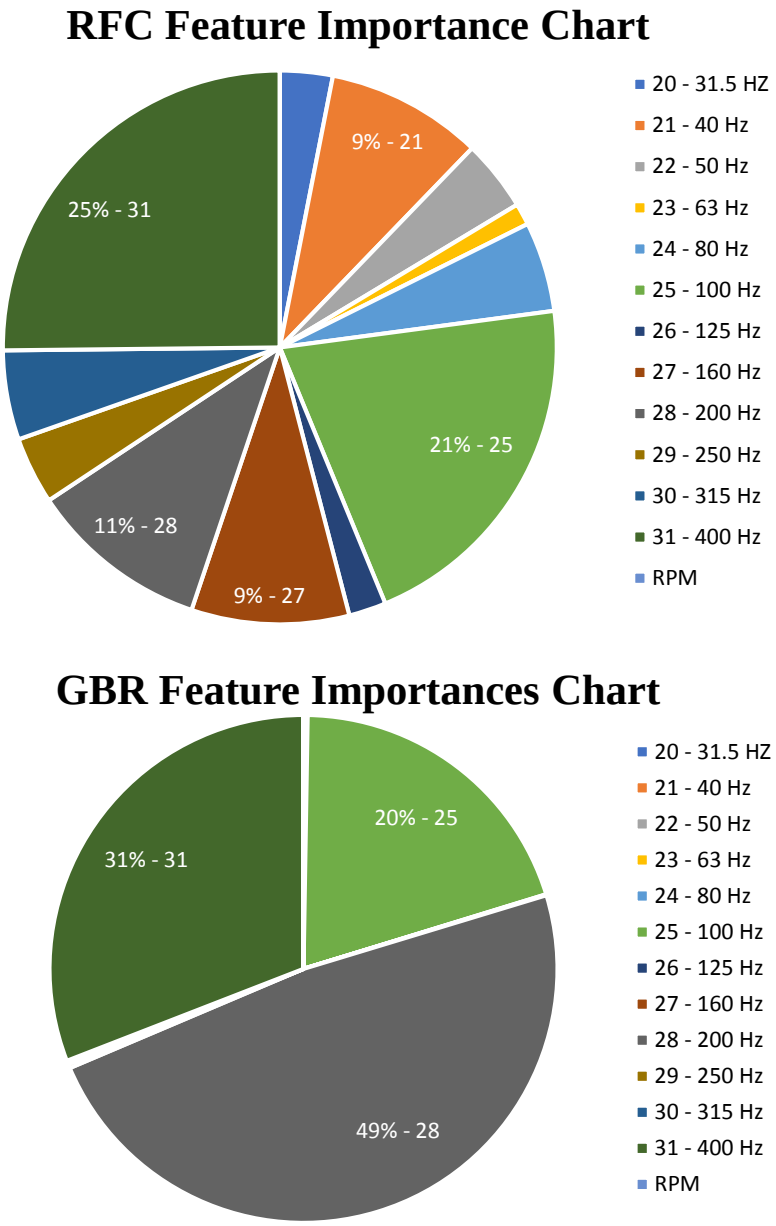


Figure 5.20 RFC and GBR Feature Importance Charts #4

The results of the features importance can be explained by **Figure 5.21**. In this figure, it is obvious how the three loading conditions are creating three separate areas of operation. This makes it easier for the algorithm to predict the mean load of the bearing and for the user to visualize the results and understand the usefulness of the Octave analysis for this technique. The lower color intensity indicates that the point is further away in the 3D plot.

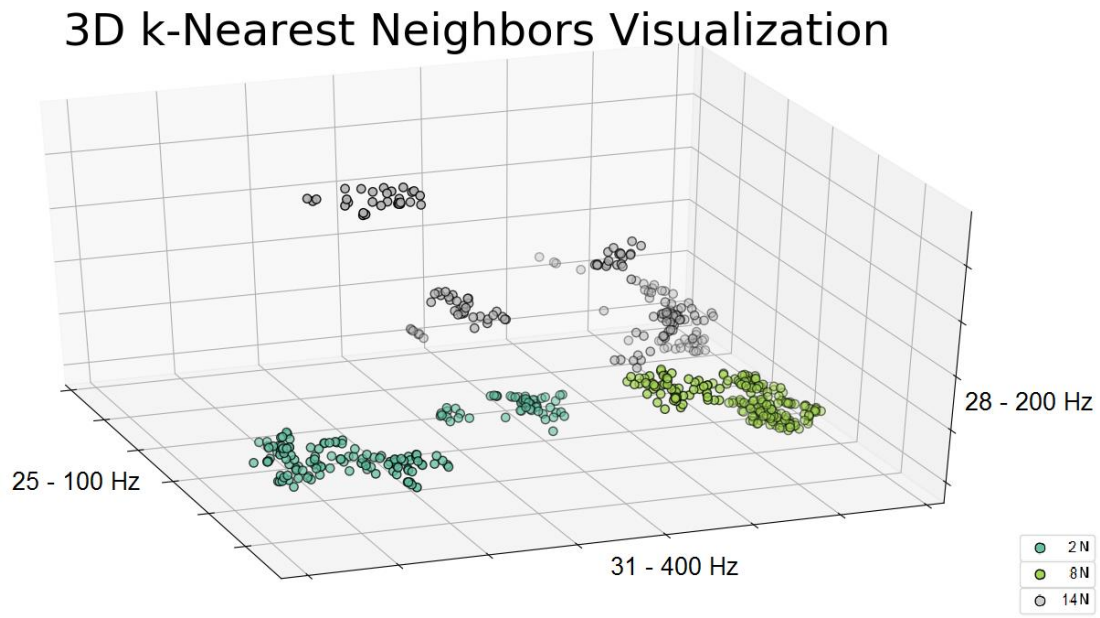


Figure 5.21 3D KNNC Visualization of Feature 25, Feature 28 and Feature 31 #4

For the first sub-case, the results for the RFC are shown in **Figure 5.22** and for the KNNC are shown in **Figure 5.23**. The accuracy of both classifiers does not drop under 99% with the overfitting being avoided through proper hyper-parameter tuning. The results of GBR are visualized via a boxplot in **Figure 5.24**. The boxplots show the low dispersion of the data and the high accuracy of algorithm with some outliers still being existent.

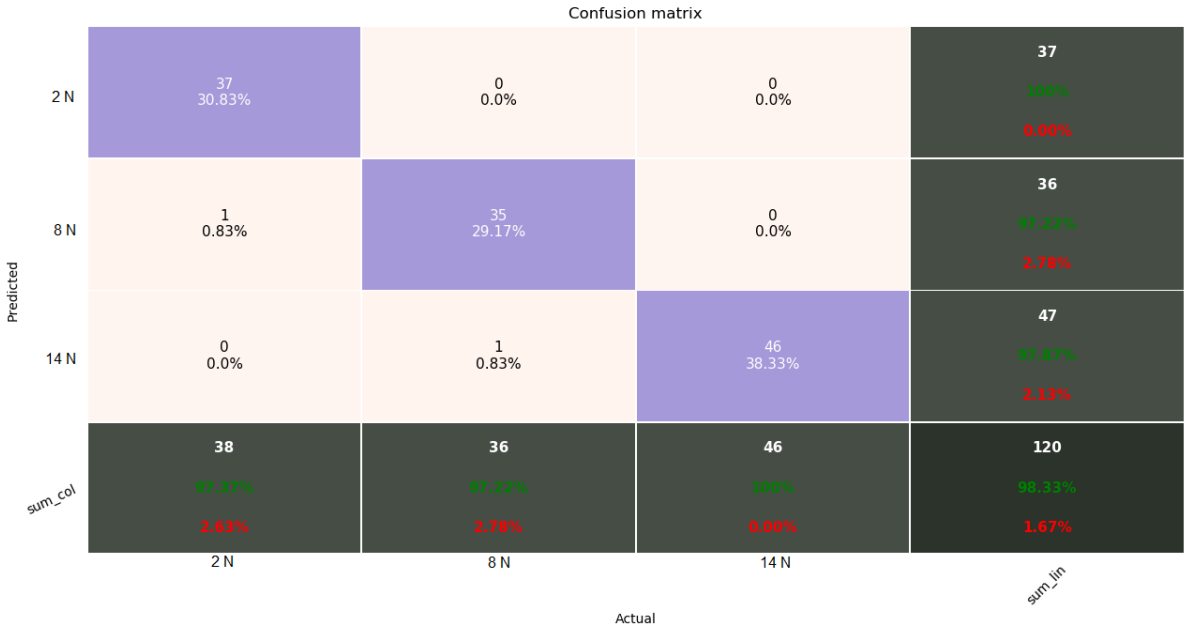


Figure 5.22 RFC Confusion Matrix #4

Figure 5.23 KNNC Confusion Matrix #4

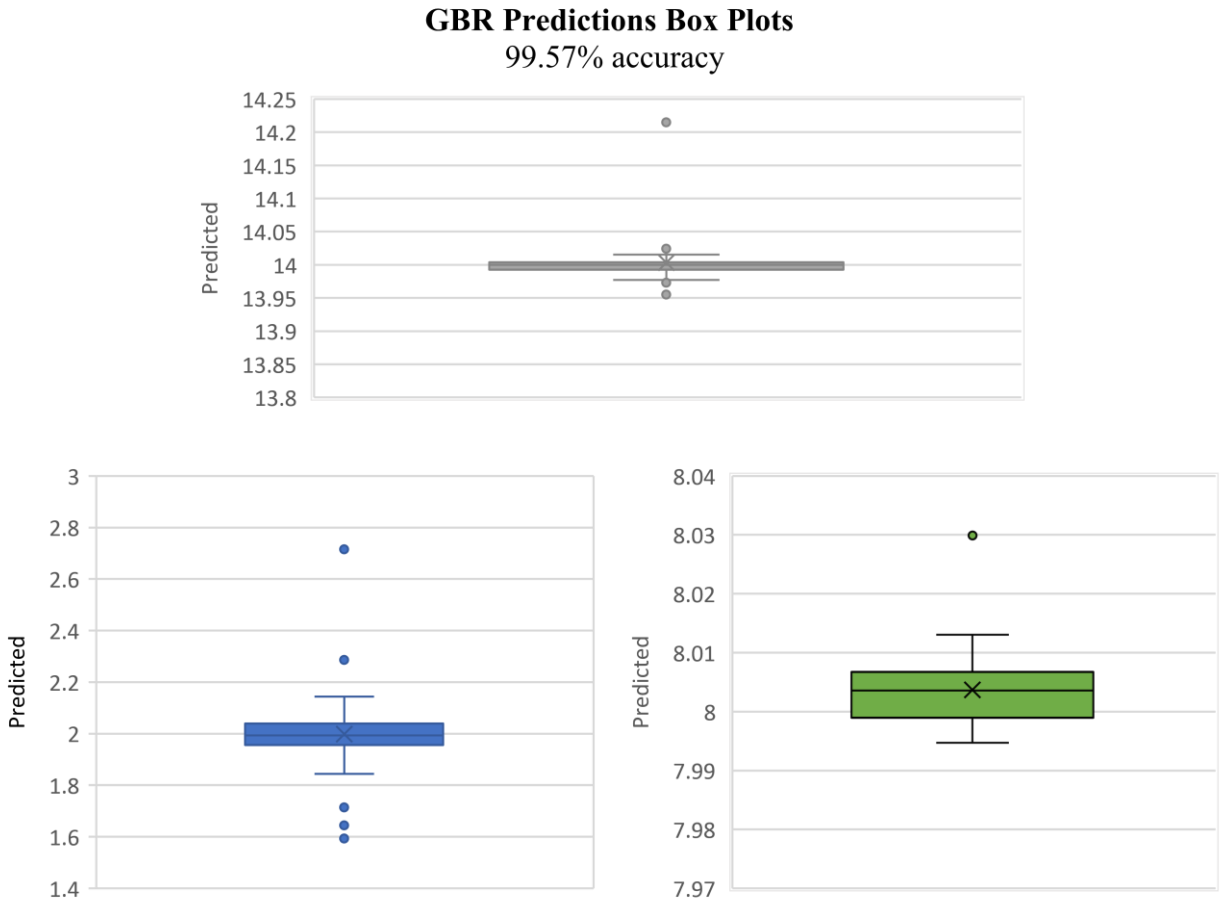


Figure 5.24 GBR Boxplots, Grey-14N, Blue-2N, Green-8N #4

The second sub-case, where the testing data are from a part of the experimental data that were not used for creating the training pool, is also successful as all three algorithms have predicted correctly the actual value of the load which was 2N. **Table 5.5** shows the exact results. This difference in the prediction accuracy with the second subcase of case study #3 verifies once again the results of case study #1; the sound signals can produce better and more accurate predictions than vibration signals. GBR algorithm predicts a load between 1.98 and 2.03 N which is a very low standard deviation.

Algorithm Predictions

RFC	2 N
KNNC	2 N
GBR	1.98-2.03 N

Table 5.5 Algorithm Predictions #4

In the third sub-case, there is a change in the number of samples of RPM-Load combinations that are used to create the training pool. The number drops from thirty to ten in order to determine a minimum data volume needed for the algorithms to function. No actual data volume is suggested in machine learning bibliography so this was a subject of investigation. The exact results of the algorithms are shown in **Table 5.6**. After evaluation, the number of samples is chosen to be thirty per load per rotational speed in order to create a better mapping of the bearings operation.

Algorithm Performance

	Accuracy	**Prediction**
RFC	99 – 100 %	2 N
KNNC	99 – 100 %	2 N
GBR	96 – 99 %	2.004 N

Table 5.6 Algorithm Performance #4

5.3.5 Case Study #5, Load Determination via Sound signals for unknown RPM-Load combinations

The training pools used in the following sub-cases do not include a RPM-Load combination and the excluded combination is different in each subcase and used for to generate the testing data. The sub-cases will be named after the combination that was not used for the training process.

In the sub-cases that follow, the results will be displayed in the form of tables. The tables will have information about the accuracy (in predicting the values from the excluded combinations). The algorithms used are the RFC, the KNNC, the GBR and the Decision Tree Regressor (DTR). Other algorithms have also been tested but these four have the higher overall accuracy. All sound signal features are used in every sub-case.

I. Load 8N, 500 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	99 – 100 %	8 N
KNNC	99 – 100 %	8 N
GBR	97 – 99 %	8 ± 0.15 N

Table 5.7 Algorithm Performance, Sub-case I #5

II. Load 8N, 2500 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	99 – 100 %	8 N
GBR	97 – 99 %	8 ± 0.1 N

Table 5.8 Algorithm Performance, Sub-case II #5

III. Load 2N, 500 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	Unstable	Unstable
KNNC	Unstable	Unstable
GBR	Unstable	Unstable

Table 5.9 Algorithm Performance, Sub-case III #5

IV. Load 2N, 1000 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	99 – 100 %	2 N
KNNC	99 – 100 %	2 N
GBR	97 – 99 %	2 ± 0.13 N

Table 5.10 Algorithm Performance, Sub-case IV #5

V. Load 14N, 4600 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	99 – 100 %	14 N
GBR	Unstable	Unstable
DTC	95 – 99 %	14.0 N

Table 5.11 Algorithm Performance, Sub-case V #5

VI. Load 14N, 4000 RPM

Algorithm Performance

	Accuracy	Prediction
RFC	99 – 100 %	14 N

Table 5.12 Algorithm Performance, Sub-case VI #5

The aim of this case study was to learn how the algorithms handle new information that belong to different parts of the data pool. That is the reason why in sub-case III and V the combinations of RPM-Load chosen are the extreme values of the rotational speed and loading condition. While the algorithms maintain their performance for the intermediary combinations of RPM-Load, the accuracy decreases dramatically when reaching the edge of the data pool. This implies that a better mapping of the bearing's operation is needed. An instability like the one observed above could indicate that the testing input belongs to an unmapped part of the bearing's operation and suggest an error condition.

5.3.6 Case Study #6, Load Determination via Sound signals with training on the ACETAL Bearing and testing on the Plexiglass Bearing (2500 RPM)

The sound signal Octave analysis technique for the creation of the feature space of a bearing's loading determination algorithm has been successful so far, with the classification and the regression problems succeeding in most of the cases. Training and testing data were both mined from the experimental data of a single journal bearing. In this case study, the training data will be extracted from the data measured on the ACETAL bearing and the testing data will be extracted from the data acquired from the Plexiglass bearing. It should be noted that the previous case studies have been tested for both bearings and the results converge.

For the first attempt, the training and testing data and the sound feature space used appear in **Figure 5.25**. The feature space also includes the rotational speed of the rotor. The results of the RFC and the KNNC (k = 5) appear in **Figure 5.26**. While evaluating the inaccurate results of the GBR, it was observed that the features that had high importance could be excluded from the training process and predictions' accuracy raised up until an average of 90% but were unstable and got rejected.

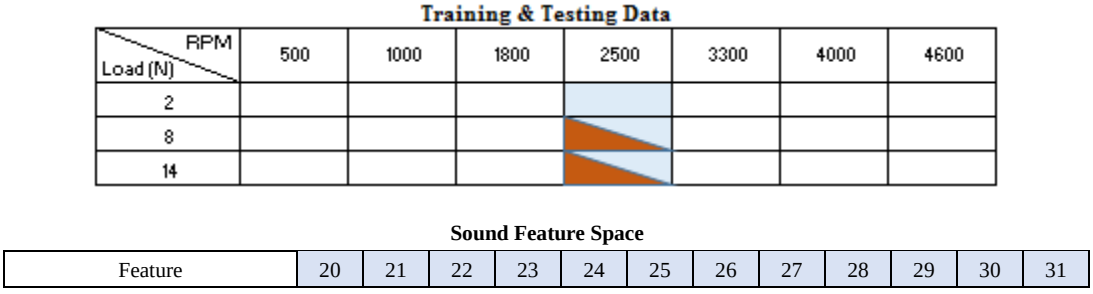


Figure 5.25 Training & Testing Data and Sound Feature Space Mapping #6

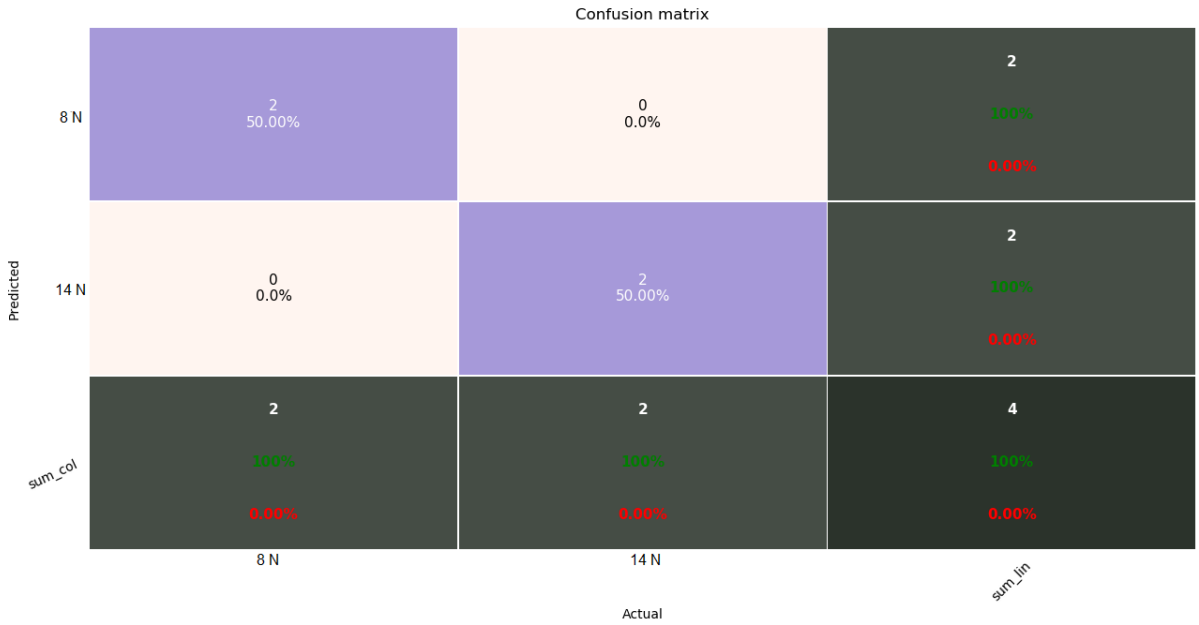


Figure 5.26 RFC and KNNC Confusion Matrix #6

5.3.7 Case Study #7, Load Determination via Sound signals with training on the ACETAL Bearing and testing on the Plexiglass Bearing (4000 RPM)

A different approach was used for this study. The goal is to find the minimum number of features needed to predict the Plexiglass bearing's loading condition. In order to achieve this goal, it is important to find the features that hold the information needed. On the one hand, the importance of the features, when having training and testing data from the same bearing, is known. On the other hand, this feature importance changes when the training and testing data originate from separate bearings. So, the way to choose the order of the features that are added in each run is to build a dataset that consists of training and testing data generated from both bearings and apprehend the feature importance. The rotational speed is always 4000 RPM.

Figure 5.27 shows the training and testing data mapping. The KNNC algorithm is used in this case study. In **Table 5.13** appear the features that are added in each run and the accuracy of the predictions. The results show that feature space selection is an area of investigation and should be given enough attention in order to determine the useful and useless data. This selection varies from problem to problem and from dataset to dataset, depending on the experimental procedure.

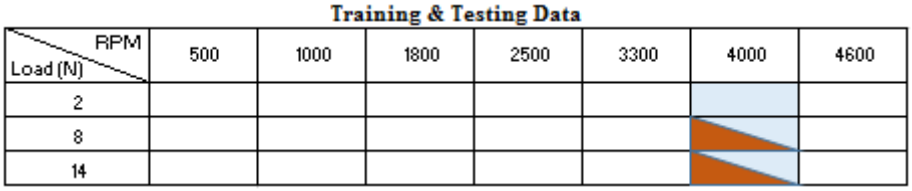


Figure 5.27 Training & Testing Data Mapping #7

Feature Selection Table

Run	Features Added	Accuracy
1st	23	42.86%
2nd	20	71.43%
3rd	31	85.71%
4th	29	100%

Table 5.13 Feature Selection 4000 RPM #7

5.3.8 Case Study #8, Combination of Case Study #6 and Case Study # 7

The combination of the two case studies aims to broaden the perception of how information is stored in each feature. For example, some features carry information regarding the rotational speed of the shaft while other features may carry information about the smoothness of the shaft's surface. This suggests that the values of the features depend on the experiment's execution, the experimental equipment etc.

This case study combines the two rotation speeds in order to examine if the feature selection of case study #7 can produce results of the same accuracy as rotational speed is 2500 RPM. The two rotational speeds are then combined. **Figure 5.28** shows the training and testing data mapping. **Table 5.14** shows the accuracy of KNNC as to the feature selection. The default features are feature 23,20,31 and 29.

Feature Selection Table

Run	Features Added	Accuracy
1st	23, 20, 31, 29	81.82%
2nd	26	100%

Table 5.14 Feature Selection 2500 RPM #8

The rotational speed of 2500 RPM needs a different set of features in order to reach high prediction accuracy. This is expected due to the different sound frequency signatures of each operational condition. 4000 RPM and 2500 RPM data are combined as shown in **Figure 5.28** and the results are shown in **Figure 5.29**.

Training & Testing Data

RPM / Load (N)	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Sound Feature Space

Feature	20	21	22	23	24	25	26	27	28	29	30	31

Figure 5.28 Training & Testing Data and Sound Features Mapping #8

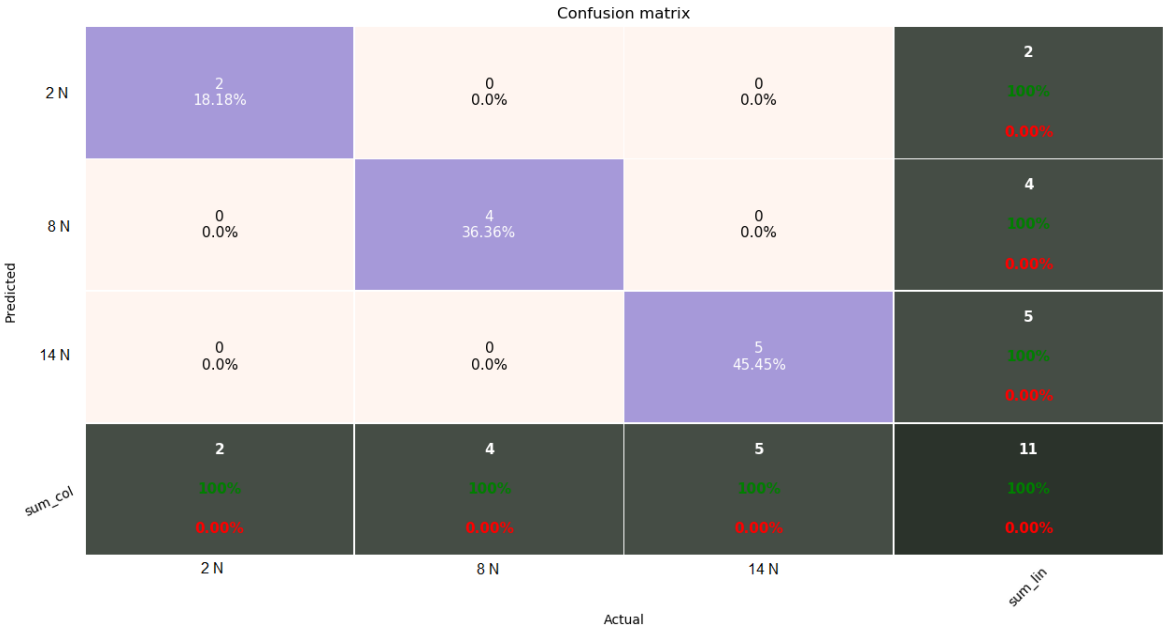


Figure 5.29 KNNC Confusion Matrix #8

5.3.9 Case Study #9, Training with ACETAL and Plexiglass Bearing Sound Signal

In this last case study, the aim is to examine the algorithms' accuracy on training and testing data that come from both the ACETAL and the Plexiglass bearings. **Figure 5.30** shows the data used in this case study. The first sub-case deals with classifying which of the two bearings is the tested one. When dealing with this problem, the algorithms will highlight the features that best distinguish the two bearings. The algorithms used are the RFC and the KNNC. The feature importance appears in **Figure 5.31** and the results appear in **Table 5.15**.

Training & Testing Data ACETAL Bearing

Load (N) \ RPM	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Training & Testing Data Plexiglass Bearing

Load (N) \ RPM	500	1000	1800	2500	3300	4000	4600
2							
8							
14							

Sound Feature Space

Feature	20	21	22	23	24	25	26	27	28	29	30	31

Figure 5.30 Training & Testing Data and Sound Features Mapping #9

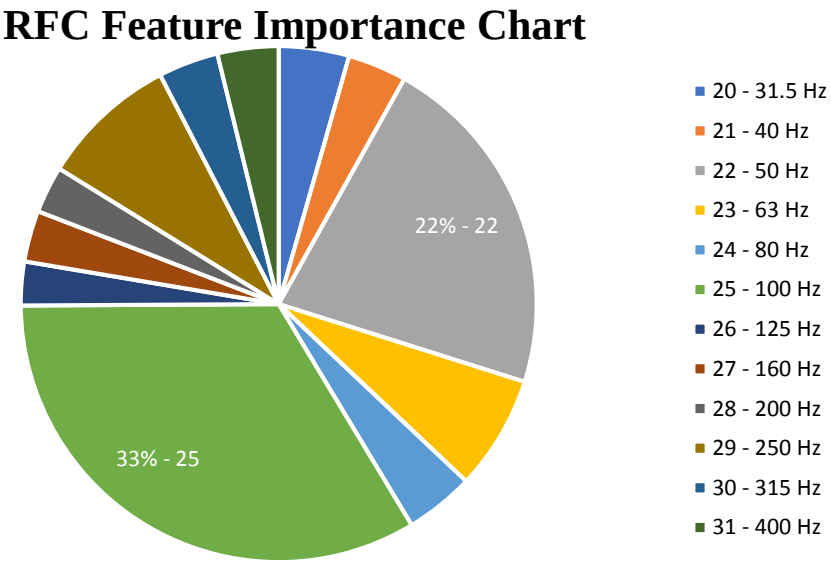


Figure 5.31 RFC Feature Importance Chart #9

Algorithm Accuracy

	Accuracy
RFC	100 %
KNNC	100 %

Table 5.15 Algorithm Accuracy #9

In the second sub-case, the label of the algorithms is the load. The algorithms used are the Random Forest Classifier and the Gradient Boosting Regressor and the feature importance of both is shown in **Figure 5.32**. The features that hold most of the common information for the two bearings are features 25 and 31 followed by feature 22. The results of the RFC and the GBR are shown **Figure 5.33** and **Table 5.16** respectively. The high accuracy percentage of both algorithms suggests that a dataset created from multiple bearings (manufactured from the same design plan) can have a wider application regarding this specific bearing.

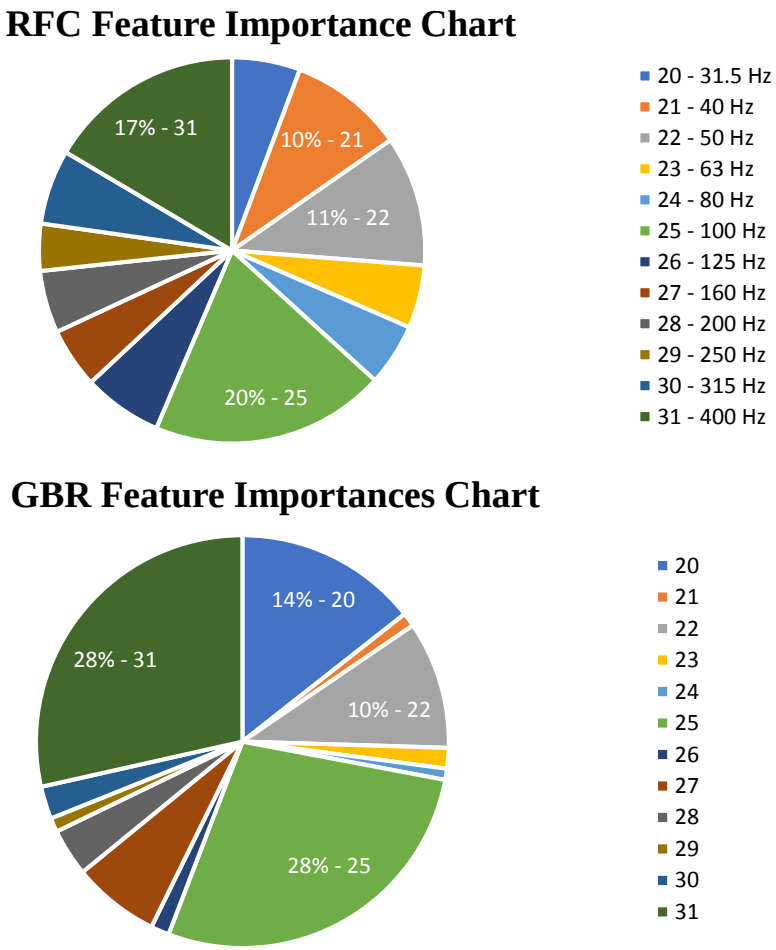


Figure 5.32 RFC and GBR Feature Importance Chart #9

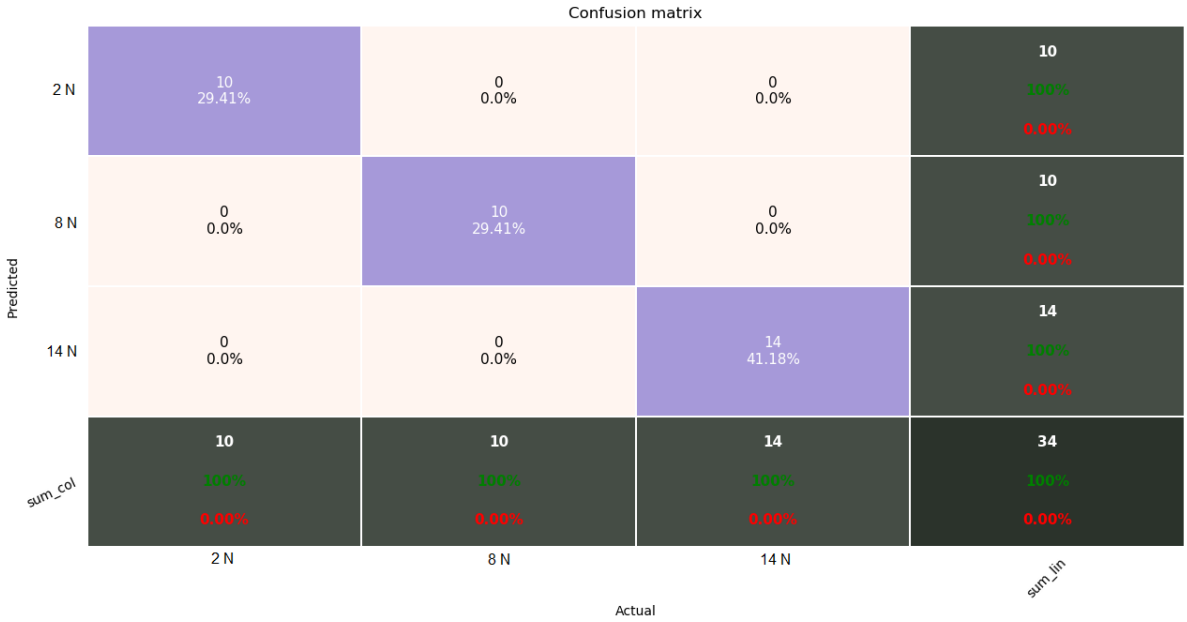


Figure 5.33 RFC Confusion Matrix #9

GBR Prediction Table
(93 – 96 % accuracy)

Actual	2	2	2	2	2	2	2	2	8	8
Predicted	2.10	2.11	2.53	1.86	2.79	2.16	1.74	1.79	8.43	8.00
Actual	8	8	8	8	8	8	8	14	14	14
Predicted	8.21	7.97	8.01	8.24	8.06	8.67	8.25	13.85	14.00	13.83
Actual	14	14	14	14	14	14	14	14	14	14
Predicted	14.04	14.03	13.98	14.00	14.05	13.90	12.10	10.16	14.26	13.86

Table 5.16 GBR Load Prediction #9

6 Conclusion and Suggestions for future work

6.1 Conclusion

The goal of the present work is to develop a machine learning technique that predicts the loading condition of a journal bearing. The data used are acquired through an experiment performed on the Bently Nevada Rotor Kit 4 in the Laboratory of Marine Engineering (L.M.E) of NTUA. The data processing source code and the machine learning algorithm is written in Python.

The experimental procedure included assembling the RK4 assembly, adjusting the microphone and the triaxial accelerometer, placing the soundproof cover, regulating the rotational speed and oil supply subsystems, connecting the sensors with the data acquisition card (DAQ) and developing a block diagram in LabView in order to acquire and store the experimental data. A second bearing was manufactured using the design plan of the ServoFluid bearing of RK4.

After preparing the experimental setup, a series of measurements are performed with different rotational speeds, in a range of 500 up until 4600 RPM, and loading conditions of the bearing, taking the values of 2, 8 and 14 N. The microphone and the triaxial accelerometer are used to measure acoustical pressure and vibration signals generated by the rotor's operation. A One-Third Octave filter is then applied to the signal.

The filtered signal is cut into smaller, in duration, samples and fed to the machine learning algorithms existent in the scikit learn module of Python. The data used to determine the algorithms' prediction ability can either constitute a part of the algorithms' training pool or be generated from a different set of experimental data.

The signal that performs better is the processed acoustic pressure signal and the acceleration z signal is second best. This conclusion is very important due to the simplicity of a microphone's installation on a bearing. A variety of scenarios are examined and the prediction ability of the algorithms is adequate in many cases. The algorithms' performance variates depending on the testing data; the algorithms perform better if the testing data belongs to an intermediary training RPM-Load combination than if the testing data belongs to an extreme combination.

In an attempt to investigate on the different ways the algorithms' training data set could be created, the data from two different bearings is used. The algorithms' prediction ability is found adequate with many data combinations. The results suggest that applying this procedure for every bearing produced from the same design plan could be possible.

6.2 Suggestions for future work

While three loading case scenarios worked well enough for this thesis and constitute a satisfying mapping of the bearing's operation, a wider set of data, regarding the loading condition and the rotational speed of the shaft, could be used in order to conclude in safer results.

Another suggestion, as mentioned in the paragraph 6.1, is to gather experimental data from multiple bearings that were manufactured from the same design plan. This way, the factors that originate from manufacturing errors could be eliminated. This thesis project has used two bearings to maintain this concept but more could produce a better result. Furthermore, an analysis of the features' importance could be performed in order to determine the bearing operation characteristics that are involved.

In addition, programming-wise, there is a large number of machine learning algorithms that could be tested in regards this task. The source code of these algorithms could be modified to satisfy the needs of this project.

Finally, while this thesis discusses how to determine a journal bearing's loading condition, there are other attributes of the bearing's condition that could be determined in the same or a similar way. Moreover, there are many signal analysis methodologies that could improve this technique and broaden its application potential to different types of bearings and gear.

References

[1] J. K. N. Richard G. Budynas, Shigley's Mechanical Engineering Design, Tenth Edition.

[2] P. N. K.M. Saridakis, "Fault Diagnosis of Journal Bearings Based on Artificial Neural Networks and Measurements of Bearing Performance Characteristics," in *Ninth International Conference on Computational Structures Technology*, Stirlingshire, 2008.

[3] N. T. Surojit Poddar, "Detection of particle contamination in journal bearing using acoustic emission and vibration monitoring techniques," *Tribology International*, vol. 134, pp. 154-164, 2019.

[4] A. A. Narendiranath Badu T., "Automatic Fault Classification for Journal Bearings Using ANN and DNN," *ARCHIVES OF ACOUSTICS*, vol. 43, pp. 727-738, 2018.

[5] R. A. Collacott, Mechanical Fault Diagnosis and Condition Monitoring, 1977.

[6] E. R. B. Michael M. Khonsari, Applied Tribology, John Wiley & Sons, Ltd, 2008.

[7] S. G. Andreas C. Muller, Introuction to Machine Learning with Python, 2017.

[8] B. S. H. Fouad Y. Zeidan, "FLUID FILM BEARING FUNDAMENTALS AND FAILURE ANALYSIS," in *PROCEEDINGS OF THE TWENTIETH TURBOMACHINERY SYMPOSIUM* .

[9] M. Ojaghi, "Oil-Whirl Fault Modeling, Simulation and Detection in Sleeve Bearings of Squirrel Cage Inductor Motors," *IEEE Transactions on Energy Conversion*, 2015.

[10] Malcolm E. Leader, "Understanding Journal Bearings," [Online]. Available: http://edge.rit.edu.

[11] G. Straffelini, Friction and Wear: Methodologies for Design and Control, Springer International Publishing, 2015.

[12] R. K. Mobley, An Introduction to Predictive Maintenance, Butterworth-Heinemann, 2002.

[13] R. M. Gray, Entropy and Information Theory, Springer, 2011.

[14] R. T. J. F. Trevor Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction., Springer, 2013.

[15] A. S. Peter Norton, Beginning Python, 2005.

[16] G. V. Fabian Pedregosa, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[17] B. N. General Electric, Rotor Kit (10mm) Model RK4 Operation and Maintenance Manual, 2015.

[18] B. N. General Electric, Rotor Kit Oil Whirl/Whip Option Model RK4 Operational Manual, 2015.

[19] K. R. Erwin Baur, Chemical Resistance of Engineering Phermoplastics, William Andrew, 2016.

[20] P. Piezotronics, Model 356A02 ICP Accelerometer Installation and Operating Manual, 2015.

[21] P. Piezotronics, Model 130D21 ICP Array Microphone Installation and Operating Manual, 2015.

[22] P. Piezotronics, Model 482A22 ICP Sensor Signal COnditioner Installation and Operating Manual, 2015.

[23] IoTech, DaqBoard/1000 and /2000 Series User's Manual, 2005.

[24] I. Measurement Computing, NI LabVIEW Support Driver Support Enhancements.

[25] I. Measurement Computing, DaqIO for NI LabVIEW Support VIs.

[26] A. Brandt, Noise and Vibration Analysis: Signal ANalysis and Experimental Procedures, 2011.

Appendix A

This appendix contains parts of the data processing. It also contains some of the coding of the algorithms as shown in Python IDLE 3.6 and the algorithms' result printed in the Python Shell.

There are three discrete steps to get the data from their raw condition ready for training the machine learning algorithms. **Figure I** shows the data as acquired from LabView.

```
1  -0,028.-0,008.-0,009.-0,009.
2  -0,027.-0,008.-0,008.-0,008.
3  -0,026.-0,006.-0,007.-0,008.
4  -0,022.-0,009.-0,008.-0,009.
5  -0,023.-0,008.-0,009.-0,007.
6  -0,021.-0,008.-0,008.-0,008.
7  -0,020.-0,007.-0,008.-0,008.
8  -0,023.-0,009.-0,009.-0,010.
9  -0,023.-0,007.-0,008.-0,009.
10 -0,028.-0,007.-0,006.-0,007.
11 -0,028.-0,008.-0,008.-0,008.
12 -0,026.-0,008.-0,008.-0,009.
13 -0,027.-0,008.-0,007.-0,008.
14 -0,024.-0,007.-0,008.-0,008.
15 -0,023.-0,008.-0,008.-0,008.
16 -0,023.-0,007.-0,006.-0,007.
17 -0,022.-0,009.-0,009.-0,009.
18 -0,025.-0,008.-0,008.-0,007.
19 -0,028.-0,008.-0,008.-0,006.
20 -0,026.-0,008.-0,009.-0,009.
```

Figure I

The first step is to use the calibration data and transform the data from mV to acceleration and sound pressure values. **Figure II** shows the converted data.

1	RPM	Sound	AccX	AccY	AccZ
2	500	0.00083	-0.00798	-0.00909	-0.00919
3	500	0.00080	-0.00798	-0.00808	-0.00817
4	500	0.00077	-0.00599	-0.00707	-0.00817
5	500	0.00065	-0.00898	-0.00808	-0.00919
6	500	0.00068	-0.00798	-0.00909	-0.00715
7	500	0.00062	-0.00798	-0.00808	-0.00817
8	500	0.00059	-0.00699	-0.00808	-0.00817
9	500	0.00068	-0.00898	-0.00909	-0.01021
10	500	0.00068	-0.00699	-0.00808	-0.00919
11	500	0.00083	-0.00699	-0.00606	-0.00715
12	500	0.00083	-0.00798	-0.00808	-0.00817
13	500	0.00077	-0.00798	-0.00808	-0.00919
14	500	0.00080	-0.00798	-0.00707	-0.00817
15	500	0.00071	-0.00699	-0.00808	-0.00817
16	500	0.00068	-0.00798	-0.00808	-0.00817
17	500	0.00068	-0.00699	-0.00606	-0.00715
18	500	0.00065	-0.00898	-0.00909	-0.00919
19	500	0.00074	-0.00798	-0.00808	-0.00715
20	500	0.00083	-0.00798	-0.00808	-0.00613

Figure II

The second step is to create a number of samples of specific duration from the dataset and store them separately so as to apply the One-Third Octave filter to each sample and create the training

pool. **Figure III** shows ten acceleration x samples with duration of three seconds, which mean that there are three thousand instances in each sample. Only the last instances are visible in the figure.

	DJR	DJS	DJT	DJU	DJV	DJW	DJX	DJY	DJZ	DKA	DKB	DKC	DKD	DKE	DKF	DKG	DKH	DKI	DKJ	DKK
1	-0.01098	-0.00898	-0.00998	-0.01098	-0.01198	-0.00898	-0.00898	-0.00998	-0.00998	-0.01098	-0.01098	-0.00898	-0.00898	-0.00898	-0.00898	-0.01098	-0.01098	-0.00898	-0.00898	500]
2	-0.00998	-0.01098	-0.00998	-0.01198	-0.01098	-0.00998	-0.01098	-0.00998	-0.01098	-0.00998	-0.01098	-0.01198	-0.00898	-0.00898	-0.01297	-0.01198	-0.01198	-0.00998	-0.01098	500]
3	-0.01098	-0.01198	-0.00998	-0.00998	-0.01098	-0.01098	-0.00998	-0.01098	-0.00998	-0.00998	-0.01098	-0.01098	-0.00998	-0.00998	-0.01098	-0.00998	-0.00798	-0.01297	-0.01098	500]
4	-0.01098	-0.00898	-0.00898	-0.01098	-0.00898	-0.00898	-0.00998	-0.00998	-0.00898	-0.01098	-0.01098	-0.00998	-0.00998	-0.00998	-0.00998	-0.01098	-0.00998	-0.00898	-0.01098	500]
5	-0.01198	-0.01297	-0.01297	-0.01198	-0.01098	-0.01098	-0.01297	-0.01198	-0.01198	-0.01098	-0.01098	-0.00798	-0.01098	-0.01198	-0.01198	-0.01198	-0.01198	-0.01198	-0.01098	500]
6	-0.01098	-0.01198	-0.00998	-0.01198	-0.01098	-0.01198	-0.00898	-0.01198	-0.01098	-0.01198	-0.01098	-0.00998	-0.01098	-0.01098	-0.01098	-0.01198	-0.00998	-0.01198	-0.00998	500]
7	-0.01098	-0.01297	-0.01098	-0.01098	-0.00998	-0.01297	-0.01198	-0.01098	-0.01098	-0.01198	-0.01098	-0.01198	-0.01297	-0.01098	-0.01297	-0.00998	-0.01198	-0.01297	-0.01198	500]
8	-0.01098	-0.01198	-0.01198	-0.01198	-0.01198	-0.01198	-0.01198	-0.01098	-0.01198	-0.01198	-0.00998	-0.01198	-0.01098	-0.01198	-0.01198	-0.01098	-0.01198	-0.01098	-0.01098	500]
9	-0.01098	-0.01098	-0.01098	-0.01098	-0.00998	-0.01198	-0.01198	-0.01297	-0.01098	-0.01098	-0.01098	-0.01098	-0.01098	-0.01198	-0.01198	-0.00898	-0.01098	-0.01198	-0.01297	500]
10	-0.01098	-0.01098	-0.00898	-0.01098	-0.00998	-0.01098	-0.01297	-0.00898	-0.00998	-0.00998	-0.00998	-0.01098	-0.01098	-0.01198	-0.01198	-0.00998	-0.01098	-0.01098	-0.01098	500]

Figure III

Lastly, the One-Third Octave filter is applied on each sample. **Figure IV** shows the file that will be fed to the machine learning algorithms. The first row are the features and labels.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	RPM	Load
2	27.97182	27.67345	24.07383	25.27395	26.85273	25.14821	24.80418	20.60555	26.3612	27.1514	26.68041	25.23261	27.40806	27.16759	28.17688	31.46923	37.28856	39.09954	35.77964	500	8
3	26.91511	26.20964	26.50296	20.14074	27.132	19.40243	22.52987	19.95986	24.10589	29.13516	26.67755	27.16563	25.43778	28.74498	29.75737	31.61749	38.0606	39.32258	35.72408	500	8
4	29.96398	25.61497	25.3559	22.13844	27.95722	20.41193	24.51569	21.48156	27.58055	27.81773	27.31123	27.04139	26.64959	28.732	30.62302	31.61304	37.76535	39.09899	36.334	500	8
5	30.75025	24.99303	19.6463	18.02998	26.09635	20.13667	20.45198	23.40315	23.5444	27.58685	24.76354	26.2725	27.87998	29.59584	31.47457	31.74428	37.1776	39.26846	36.2181	500	8
6	25.19331	25.66034	24.75643	19.80852	27.74414	25.38346	25.42576	21.30741	23.4569	28.03315	27.37718	26.17241	25.55606	28.81821	30.72212	31.40607	37.81867	38.89152	36.41617	500	8
7	31.67753	26.72864	24.91872	24.83047	25.92235	25.95268	22.63291	25.20444	25.87741	26.51136	27.61963	25.0813	27.72946	28.90493	29.84599	31.44247	37.20306	39.26836	35.93455	500	8
8	23.71063	25.49191	23.42996	19.83469	25.74312	21.08927	23.11873	24.16693	25.2895	27.50667	26.2087	26.64878	25.88172	28.685	30.86637	31.43639	37.85903	38.85002	36.45273	500	8
9	30.95365	24.81662	26.24468	26.2594	21.30328	23.73824	24.77672	22.29079	23.64605	28.68632	26.59054	27.91868	25.76708	28.88333	29.84971	31.61951	38.15021	39.52493	35.65863	500	8
10	30.58281	29.02925	27.45256	20.73009	24.13998	22.46891	22.03829	22.46296	25.24302	29.31102	27.2986	25.94508	27.15847	28.6383	28.80859	31.25216	37.8458	39.6141	35.86971	500	8

Figure IV

The Random Forest Classifier appears in **Figure V**. Some of the available parameters are set in the beginning in order to show how the module works. Then, the cross-validation tool RandomizedSearchCV of scikit-learn is used to tune these parameters and find the most efficient combination of values. Several other cross-validation tools are available in the module. The combinations tested are set to 20. The number of parameters is chosen by the user. The user defines the limits of the search and evaluates the results. **Figure VI** shows the Python Shell results.

```
#                    RANDOM FOREST CLASSIFIER

clf = RandomForestClassifier(n_estimators = 100, n_jobs = -1, bootstrap = True,
                             criterion = 'entropy',max_depth = None,
                             min_samples_split = 11, max_features = 1)
param_dist = {"max_depth": [3, None],
              "max_features": randint(1, 4),
              "min_samples_split": randint(2, 15),
              "bootstrap": [True, False],
              "criterion": ["gini", "entropy"]}
n_iter_search = 20
random_search = RandomizedSearchCV(clf, param_distributions=param_dist,
                                   n_iter=n_iter_search, cv=5)
start = time()
random_search.fit(X, y)
print("RandomizedSearchCV took %.2f seconds for %d candidates"
      " parameter settings." % ((time() - start), n_iter_search))
report(random_search.cv_results_)

#                                --
```

Figure V

```
RandomizedSearchCV took 65.89 seconds for 20 candidates parameter settings.
Model with rank: 1
Mean validation score: 0.616 (std: 0.083)
Parameters: {'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 1, 'min_samples_split': 7}

Model with rank: 2
Mean validation score: 0.566 (std: 0.091)
Parameters: {'bootstrap': True, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 8}

Model with rank: 3
Mean validation score: 0.561 (std: 0.082)
Parameters: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': None, 'max_features': 2, 'min_samples_split': 5}
```

Figure VI

If the parameters of the problem have been figured out then the next runs will not need a cross-validation tool. When the algorithm is trained, there is no need to retrain it. The option of saving it is available and is performed in the way shown in **Figure VII**.

```
#               Classifier Save

import pickle
with open('RandomForestClassifier.pickle', 'wb') as f:
    pickle.dump(clf,f)

#                   --

#               Classifier Load

pickle_in = open('RandomForestClassifier.pickle','rb')
clf_new = pickle.load(pickle_in)

#                   --
```

Figure VII