Tampere University

Iiris Keinonen

MOBILE MACHINE MAINTENANCE USING AI-BASED CONDITION MONITORING

Bachelor's Thesis
Faculty of Engineering and Natural Sciences
Examiner: Prof. Tatiana Minav
March 2025

ABSTRACT

Iiris Keinonen: Mobile Machine Maintenance Using AI-based Condition Monitoring
Bachelor’s Thesis
Tampere University
Bachelor’s Programme of Engineering Sciences
March 2025

Maintenance of any machinery or system is essential to ensure reliability, profitability, long useful life and safety of operations. The maintenance strategies that are mostly used in the field of mobile machinery are falling behind in deploying the latest advanced approaches. To enhance the maintenance further and detect when components are approaching failure, predictive methods are being developed. The aim of this work is to utilize machine learning (ML) -based models to predict health conditions of hydraulic components. ML-based models were selected because of their simple and easy usability.

The work introduces the challenges and special characteristics in the maintenance of mobile machinery and provides an overview of maintenance strategies in general. The methodology of implementing predictive maintenance using condition monitoring is presented, and three selected classification models, Decision Tree (DT), K Nearest Neighbors (KNN), and Stochastic Gradient Descent (SGD), are trained to classify health states of two hydraulic components, pump and accumulator. A publicly available experimentally obtained dataset of hydraulic component health conditions is used for the training. The data is pre-processed including re-sampling, feature extraction, and scaling. The classification results between the selected models and components are compared, and a feature selection method is used to decrease the amount of data needed for training.

DT and KNN performed well with fewer features. 15 most relevant features were identified for the classification of pump health states, and a validation accuracy of 100% was achieved with the DT and KNN models. For the hydraulic accumulator case, 10 most relevant features were used for the and in this case as well, the DT and KNN had a validation accuracy of 100%. The benefits of condition monitoring were recognized and demonstrated that if there is data available, the implementation for a simple application of health state classification can be done with little resources.

Keywords: maintenance strategies; predictive maintenance; condition monitoring; machine learning

The originality of this thesis has been checked using the Turnitin Originality Check service.

TIIVISTELMÄ

Iiris Keinonen: Mobile Machine Maintenance Using AI-based Condition Monitoring
Kandidaatintutkielma
Tampereen yliopisto
Teknisten tieteiden kandidaatin tutkinto-ohjelma
Maaliskuu 2025

Jokaisen koneen tai järjestelmän ylläpito on välttämätöntä luotettavuuden, kannattavuuden, pitkän käyttöiän ja käytön turvallisuuden varmistamiseksi. Liikkuvien työkoneiden alalla eniten käytetyt huoltostrategiat ovat jäämässä jälkeen uusimpien ja edistyneempien lähestymistapojen käyttöönotossa. Ennakoivia menetelmiä ollaan kehittämässä, jotta ylläpitoa voitaisiin tehostaa entisestään ja havaita, milloin koneen komponentit lähestyvät vikatilaa. Tämän työn tavoitteena on hyödyntää koneoppimiseen pohjautuvia malleja hydraulisten komponenttien terveydentilan ennustamiseen. Koneoppimiseen pohjautuvat mallit valittiin niiden yksinkertaisuuden ja helpon käytettävyyden ansiosta.

Työssä esitellään liikkuvien työkoneiden kunnossapidon haasteita ja erityispiirteitä sekä annetaan yleinen kuvaus olemassa olevista huoltostrategioista. Menetelmät ennakoivan huollon toteuttamiseksi esitetään ja valitaan kolme luokittelumallia koulutettavaksi. Valitut mallit ovat Decision Tree (DT), K Nearest Neighbors (KNN) ja Stochastic Gradient Descent (SGD). Mallit koulutetaan luokittelemaan kahden komponentin, pumpun ja hydraulisen toimilaitteen vikatiloja. Mallien kouluttamiseen käytetään avoimesti saatavilla olevaa kokeellisesti mitattua aineistoa, joka sisältää mittauksia usean komponentin vikatiloihin liittyen. Käytettävät mittaukset esikäsitellään, johon sisältyy uudelleennäytteistys, tilastollisten ominaisuuksien laskeminen ja skaalaus. Valittujen mallien ja komponenttien välisiä luokittelutuloksia verrataan ja ominaisuuksien valintamenetelmää hyödynnetään vähentämään koulutuksessa tarvittavan aineiston määrää.

DT ja KNN ennustivat vikatilat hyvin, myös kun käytettävän aineiston määrää vähennettiin. Pumpun kunnon luokitusta varten tunnistettiin 15 merkittävintä aineistosta laskettua ominaisuutta ja DT- ja KNN-malleilla saavutettiin 100%:n validointitarkkuus. Hydraulisen toimilaitteen tapauksessa käytettiin koulutukseen 10 merkittävintä ominaisuutta ja tässäkin tapauksessa DT- ja KNN-mallien validointitarkkuus oli 100%.

Avainsanat: kunnossapitostrategiat; ennakoiva huolto; kunnonvalvonta; koneoppiminen

PREFACE

Writing this thesis started off slowly as I had trouble defining and deciding what I actually want to do with it. Eventually I reached the point where the goal became clear, and I enjoyed diving more deeply into the topics of this thesis. I want to thank my wonderful supervisor Tatiana Minav for her support and guidance during the writing process. I also want to thank my friends in the Zonal Hydraulics research group who provided support on this journey. Special thanks to Abid Abdul Azeez for his mentoring on condition monitoring and machine learning during my time in the research group, which sparked my interest to write my thesis on those topics.

Tampere, 3 March 2025

Iiris Keinonen

CONTENTS

LIST OF FIGURES AND TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

DT	Decision Tree
KNN	K-Nearest Neighbors
LGBM	Light Gradient-Boosting Machine
ML	Machine learning
mRMR	Minimum Redundancy – Maximum Relevance
OEM	Original machine manufacturer
PdM	Predictive maintenance
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Oversampling Technique

1. INTRODUCTION

Mobile working machines are widely used in several industries such as construction, agriculture, forestry, and mining. These machines are designed to perform diverse tasks and have a wide range of functions. The operations are implemented with a combination of mechanical, electrical, and hydraulic components. Hydraulic systems are most widely used in mobile working machines because of their high power-to-weight ratio and the availability of large forces already at lower speeds [1]. With their extensive functionality and an increasing amount of automated operations mobile machines have become more complex systems requiring advanced control mechanisms and advanced hardware [2].

As these machines operate in demanding environments, they are at risk to component failures, which can also damage other components and result in costly repairs and downtime [3], [4]. Faults in hydraulic components, such as wear, leakages, and lags in valve switching are examples of issues that can affect performance. For instance, leakages can affect the system performance and controllability by causing a drop in pressure, which in turn reduces controllability and cause severe damage. Proper maintenance is required to reduce the downtime and losses caused by faults and component failures, and to ensure the safety, reliability, and productivity. [5], [6], [7]

Traditional maintenance strategies, reactive and preventive maintenance, have for long been the standard for mobile machines. In these approaches, problems are addressed after they have already been observed, or maintenance is conducted based on observations done in the past. [8] Both approaches have their limitations, especially considering the increasing complexity of mobile machines.

To address the shortcomings of the traditional methods, predictive maintenance has become a promising solution to complement them. Predictive maintenance utilizes the data collected from the sensors of the machine and aims to predict failures before they occur. The advancements in artificial intelligence (AI) and machine learning (ML) have enabled the analysis of larger datasets in predicting health conditions. This approach prevents unnecessary maintenance, minimizes downtime, and optimizes maintenance costs.

Despite its adoption in other fields, such as manufacturing, wind energy, and aviation, continuous research on predictive maintenance is conducted extensively [4], [9], [10], predictive maintenance is not yet commonly deployed in mobile machinery. Many companies are not yet providing services that take advantage of the predictive maintenance methods. Leading manufacturers, including John Deere, Komatsu, and Caterpillar offer maintenance services like oil or cooler fluid analysis and fleet management [11], [12], [13]. Research papers related to specifically the predictive maintenance of mobile machinery are also clearly lower in numbers. This presents an opportunity to research and demonstrate the benefits of predictive maintenance strategies.

The objective of this work is to showcase a possible implementation of condition monitoring, which is one approach for predictive maintenance. Using experimental data, ML-based models are trained to predict health conditions of hydraulic components. The remainder of the document is structured as follows. Chapter 2 provides an overview of maintenance and maintenance strategies in general with additional case examples relating to mobile machinery or other fields. Chapter 3 introduces the methodology and applications of predictive maintenance by condition monitoring. In chapter 4, the theory from chapter 3 is deployed and the process of implementing multiclass classification of health conditions is presented.

2. OVERVIEW OF MAINTENANCE

The definition of maintenance according to the SFS standard [14] is a “function which includes all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function”. Usually, maintenance is divided into categories depending on the set of rules which are used to determine the triggering events for maintenance actions, and those categories are referred to as maintenance strategies [15].

A commonly used division in different industries is to divide maintenance into the following three categories: reactive maintenance, preventive maintenance, and predictive maintenance (PdM). These strategies can also be divided into unplanned and planned maintenance. In literature, the same strategies can be referred to with multiple terms. For example, reactive maintenance might also be called corrective maintenance, and preventive maintenance could be called proactive maintenance. [8], [16] Different categorizations also exist, but the division to these three categories is used in this work because of its wide use in literature. Figure 1 presents a summary of the maintenance strategies which are also introduced in this chapter.

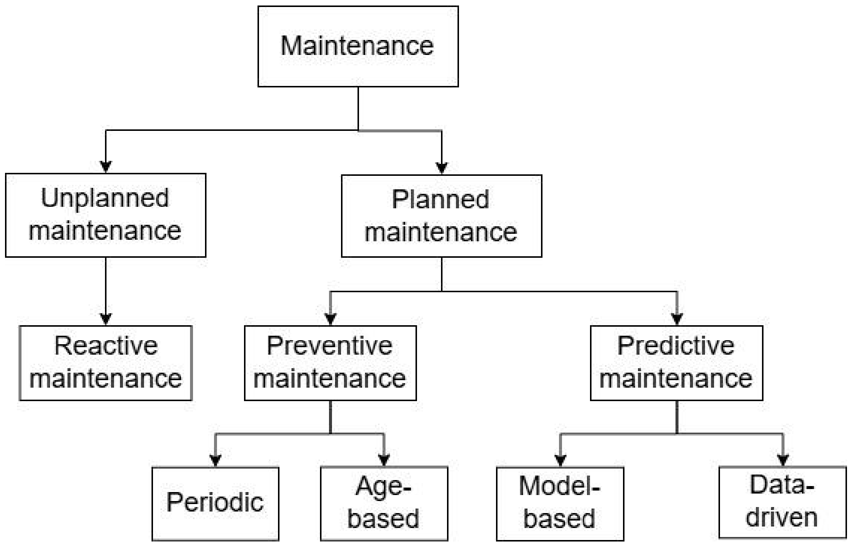

Figure 1. *Maintenance strategies. Adapted from [17], [18].*

In the remainder of this chapter, an overview of the maintenance strategies – reactive, preventive and predictive – is presented, and examples of their utilization for mobile machinery as well as other applications are provided. The strategies are also compared in terms of costs, and their ability to prevent damage and downtime.

2.1 Reactive maintenance

As the simplest maintenance strategy, reactive maintenance is straightforward. The principle is that repairs are done when they are needed, in other words when something is already broken. For the machine to continue operating as soon as possible, sometimes only temporary repairs might be completed. [19] This can leave the machine vulnerable to further breakdowns and additional downtime.

For this strategy, maintenance costs are not predictable. The cost depends, for example, on the amount of damage to the system, and the number and type of components in need of repairs or replacement. Furthermore, because the need for spare components is not predictable, there may be delays in obtaining the materials needed to complete the repairs. A total failure of a component can also cause collateral damage to other components, causing a longer downtime and increased repair costs [8].

2.2 Preventive maintenance

The objective of preventive maintenance – which is one of the planned maintenance strategies (Figure 1) – is to reduce component breakdowns or failures, improve the machine quality, and most importantly extend the useful life of the machine. The principle is to schedule the maintenance activities and inspections by utilizing knowledge of breakdown frequencies. They can be determined based on the component's age, or other statistics of the components and the machine. [8] A typical lifetime for a component is usually estimated by the manufacturers. However, it is important to note that the environment and the way those components are used and strained in practice, affect their lifetime [20] and need to be taken into account.

Based on previous experience of the breakdowns, the needed repairs can be identified with frequent inspections, and components can be changed before one of them fails completely. The early detection of defects decreases the costs compared to reactive maintenance. In [21], performance indicators of maintenance for off-road mining trucks were analyzed with a particular focus on the impact of preventive maintenance hours. It was demonstrated that neglecting the plan of scheduled maintenance resulted in an increase of machinery failures. However, when the maintenance activities took longer than planned, the productivity and availability of the machinery suffered resulting in production losses.

Therefore, the planned approach to maintenance is beneficial only if the timing is done well. Otherwise, there is a possibility that a component is changed even if it is not yet at the end of its life cycle. Risks of this kind have led to the development of PdM. [16], [22]

2.3 Predictive maintenance

In the case of preventive maintenance, it is not guaranteed that component failures would not occur between the scheduled maintenance events. PdM aims to solve this problem by continuous real-time monitoring of the machine, and by detecting early signs of failures which may not be noticeable for the operator, even though many machines do provide measurements during operation [23].

The implementation of PdM is based on historical data, models, and domain knowledge. As presented in Figure 1, it can be divided into data-driven and model-based approaches. In the model-based method, the system and its components are described by physical equations which are then used to forecast performance degradation. A popular implementation for a model-based approach is digital twin, where the equations are used to build a virtual model of the physical system. The data-driven approach on the other hand is solely based on data without needing knowledge about the physical models describing the system, and typically AI-based methods are used. Generally, the model-based approach is more costly. [24]

The historical data used in the data-driven approach is ideally measured during operation. Additionally, to reliably identify the health state of a machine, data must be collected in various health conditions, which are also individual for each machine type. Acquiring the data is the most crucial part of implementing data-driven PdM. The predictions of the machine condition are done by finding patterns and trends from the data and linking those to the known health conditions. [25] When the data-driven approach to PdM is based on continuous real-time monitoring of the machine state, it is referred to as condition monitoring.

With PdM, components can be ordered and maintenance scheduled in advance as signs of approaching failure are detected early. In fact, using a PdM strategy can produce "25-30% reduction in maintenance costs, a 70-75% reduction in breakdowns, and a 35-45% reduction in downtime" [26]. The advantages of condition monitoring compared to preventive maintenance include reducing the downtime, as failures can be predicted anytime during operation, and cutting the costs of maintenance. However, the initial investment for this type of maintenance is large, since the real-time monitoring requires multiple sensors, and suitable hardware to process all the measurements, and to perform necessary calculations or decision-making to determine the current health condition.

2.4 Summary

The main differences between the maintenance strategies are presented in Table 1. Each approach has its own advantages and drawbacks, and more than one of them is needed to have a reliable maintenance plan. As the above example demonstrated, large amounts of data is required to produce reliable statistics of the occurrence and causes of failure for a certain type of machine. The same also applies to PdM, for which acquiring the needed data is the main challenge.

Table 1. *Summary of maintenance strategies.*

Attribute	**Reactive**	**Preventive**	**Predictive**
Initial investment	None	Large	Large
Data	None	Requires	Requires continuously
Costs	Unpredictable	Can be estimated	Can be estimated
Unscheduled downtime	Increases	Reduces	Reduces
Damage	Cannot prevent	Can prevent	Can predict
Maintenance costs	High	Reduces	Reduces

As the machines are becoming more intelligent with an increasing number of sensors and processing power, the implementation of condition-based PdM becomes more approachable. Taking advantage of the existing system, the initial costs can be reduced at least in terms of hardware and data acquisition. Condition monitoring could provide additional support for scheduling the maintenance events when they are needed and would reduce unscheduled downtime further. Given that historical data on different health states of system components has been collected, implementation of the predictive model used in condition monitoring is not so complicated. In the next chapter, the steps of implementing a simple AI-based classification model to identify the current health state for two selected hydraulic components are presented.

3. IMPLEMENTATION

This chapter presents the process of using ML-based methods to classify health states of hydraulic components based on publicly available experimental data. A classification problem is a situation where each data point has a label, and the task is to use this knowledge to create a model that is able to predict an unknown label for a new data point [27] p. 38-39.

The basic phases of the workflow are adapted from [28], and consist of preprocessing the data, choosing the used ML-based models, training those models, and performing feature evaluation and selection to reduce the amount of needed data. The process used in this work is represented as a simple flowchart in Figure 2.

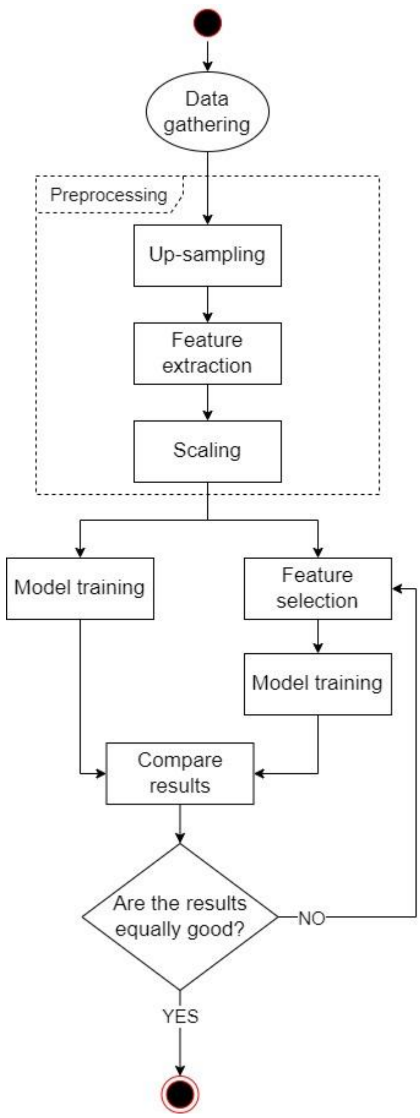

Figure 2. *Implementation flowchart. Adapted from [28].*

Data preprocessing is divided into three stages. First, to have an equal number of samples on each signal, the data is up-sampled. Next, statistical features are calculated for each signal individually, and finally, the data is scaled. The classification of the health states is conducted separately for each component. The implementation is done using Python in Jupyter Notebook environment.

3.1 Data collection

A publicly available dataset [29] is used in this work, and, therefore, a data-driven implementation is the only option. The data has been experimentally obtained with a hydraulic test rig and contains information on health conditions of four hydraulic components. The available health conditions and components are presented in Table 2.

Table 2. *Representation of the dataset [29].*

Component	Condition	Value / Label	Instances
Cooler	Full efficiency	100 %	741
	Reduced efficiency	20 %	732
	Close to total failure	3 %	732
Valve	Optimal switching behaviour	100 %	1125
	Small lag	90 %	360
	Severe lag	80 %	360
	Close to total failure	73 %	360
Pump	No leakage	0	1221
	Weak leakage	1	492
	Severe leakage	2	492
Hydraulic accumulator	Optimal pressure	130 bar	599
	Slightly reduced pressure	115 bar	399
	Severely reduced pressure	100 bar	399
	Close to total failure	90 bar	808

The dataset provides three conditions of cooler efficiency, four conditions of valve switching, three conditions of internal pump leakage, and four conditions of hydraulic accumulator pressure. Each component has measurements in an optimal condition, in a severely compromised condition, and a varying number of intermediate conditions. The pump is the only component without data during a state close to total failure. Since all components have more than two recorded health states, multiclass classification is chosen as the classification method [30]. Considering the scope of this work, data of only two components is used. The chosen components are the pump and hydraulic accumulator since leakage and reduced pressure can cause severe damage to a system, and therefore predicting the occurrence of those faults is important to avoid costly repairs and downtime.

The experiment utilized 17 sensors measuring pressure, motor power, volume flow, temperature, vibration, cooling efficiency, cooling power, and efficiency factor. In total, 2205 cycles were recorded, and each has a duration of 60 seconds. The sensors used sampling frequencies of 100 Hz, 10 Hz, and 1 Hz resulting in 6000, 600, and 60 samples per cycle respectively.

3.2 Data processing

Preprocessing the data is an important step in using ML-based models as it enhances and increases the information provided by the raw data, and it is done before any model training. Usual steps in the preprocessing depending on the used data are filtering, up- or down-sampling, feature extraction, and scaling. [31] The remainder of this section introduces the preprocessing steps and techniques applied in this work.

3.2.1 Re-sampling

Re-sampling can be used to either increase or reduce the number of samples, also known as up-sampling and down-sampling, or oversampling and undersampling respectively. The simplest way of re-sampling is multiplication or removal of existing samples. However, the multiplication does not provide any new information of the system conditions, and removing samples can result in losing valuable information and a decreased prediction accuracy. [32]

Multiplication of samples can often lead to a phenomenon called overfitting, where the model becomes too complex, learning the specific characteristics of the training data. For this reason, an overfit model performs well with the data used for training, but poorly with new data. Removal of samples on the other hand, results in underfitting, where the model is too simple to learn the data patterns, and it will not perform well on any data. [33] To achieve more beneficial results, other methods for up-sampling exist.

Widely used techniques interpolate samples between the existing ones. In practice, two samples are chosen, and a new sample is created at a random point on the line connecting the two existing samples. One commonly used method using interpolation is Synthetic Minority Oversampling Technique (SMOTE). [33]

As there is a significant difference between the number of samples recorded by the sensors due to the different sampling frequencies, the signals contain different amounts of information. To provide enough information for ML-based algorithms, the data can be resampled so that all signals match the 100 Hz sampling frequency [34]. This way the loss of information is also avoided.

Before up-sampling, the first few samples were inspected, and an unusual peak was noticed on some signals. The error from the peak values is amplified by the up-sampling since it would try to interpolate several values between the two samples with a great difference and would not accurately reflect the system behavior. To avoid this, the first few samples were removed from all signals.

The up-sampling is performed using a signal processing method from Python's SciPy library. The resample-method uses Fourier method to interpolate additional values between the samples. As shown in Program 1, the arguments for the method are the original signal to resample and the total number of samples desired in the resampled signal.

```
upsampled_signal = scipy.signal.resample(original_signal,
                                         desired_number_of_samples)
```

Program 1. *Data up-sampling.*

The up-sampling was performed for each signal using the number of samples in the 100 Hz signal as a reference for the up-sampling.

3.2.2 Feature extraction

After the up-sampling, statistical features presented in Table 3 were determined for each signal using a sliding window method. In this method, each recorded signal is divided into overlapping fixed-sized windows, and for each window, the statistical features are calculated. The sliding window method is visualized in Figure 3. The features for which an equation representation is not applicable, are determined by choosing the relative position of the maximum or minimum value in the window.

Table 3. *Determined statistical features and their formulas.*

Feature	Equation	Feature	Equation
Minimum	$\min(a)$	First location of minimum	Not applicable
Maximum	$\max(a)$	First location of maximum	Not applicable
Mean	$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$	Last location of minimum	Not applicable
Median	$median = \frac{d_{\frac{n}{2}} + d_{\frac{n}{2}+1}}{2}$	Last location of maximum	Not applicable
Standard deviation	$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n-1}}$	Sum of values	$\sum_{i=1}^{n} x_i$
Variance	$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$	Absolute energy	$\underline{b} \cdot \underline{b} = \sum_{i=1}^{n} b_i b_i$
Skewness	$\tilde{\mu}^3 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{(n-1)\sigma^3}$	Absolute sum of changes	$\sum_{i=1}^{n} \mathrm{diff}_1(a)$
Kurtosis	$\sum_{i=1}^{n} \frac{\frac{x_i - \bar{x}}{n}}{\sigma^4} - 3$	Mean absolute change	$\frac{\lvert \mathrm{diff}_1(a) \rvert}{n}$

In the formulas of Table 3, a is an array of the data points in a window, n is the size of the window, x_i is the i^{th} data point in the window, d is the sorted array of the data points in the window, $\underline{b}$ is the data points of a window as a vector, and $\mathrm{diff}_1(a)$ is the first-order difference defined as

$$\mathrm{diff}_1(a) = [a_2 - a_1,\ a_3 - a_2, \ldots, a_{n-1} - a_{n-2},\ a_n - a_{n-1}\ ].$$

The column with data points $a_1, a_2, \ldots, a_n$ in Figure 3 represents one of the measured signals obtained from the experimental test rig, such as pressure, temperature, or flow. A window with a size of 3, and step size 2 are used as an example.

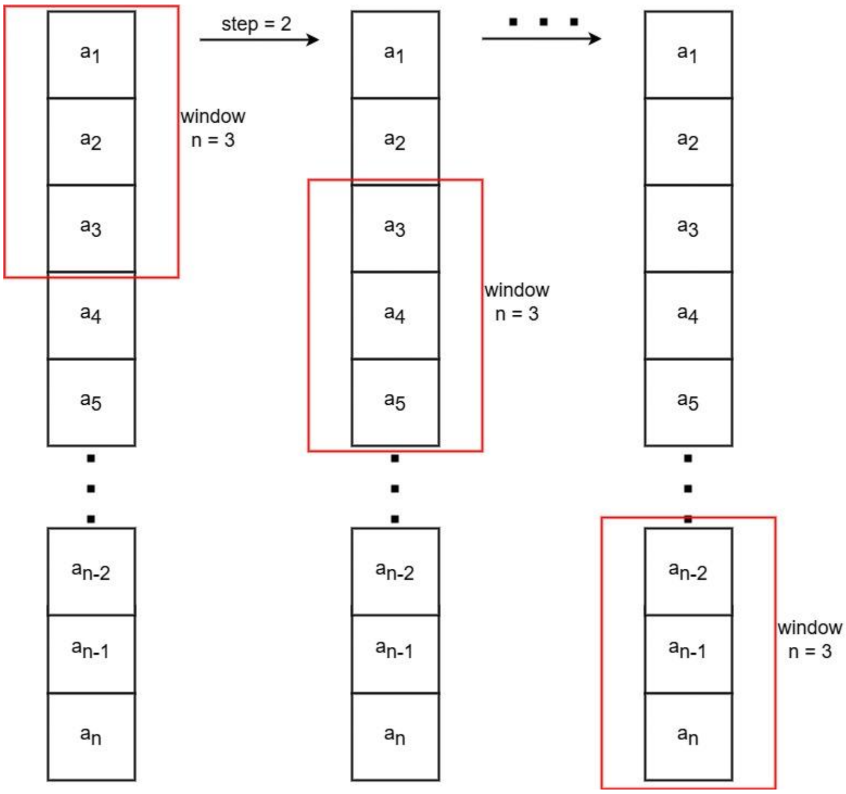

Figure 3. *Sliding window with an overlap.*

All data points in the measurement are iterated through moving the window for each iteration. From each window with a different set of data points, the statistical features are calculated. As there are 16 features extracted from each of the 17 signals, a total number of features after the feature extraction is 272.

3.2.3 Scaling

When data from physical systems is used, typically the magnitude of the measured variables can differ hugely. For example, pressure might be represented in pascals where typical values are in hundreds of thousands, and for temperature in Celsius, the values are in tens. The algorithms used to train the models only use information of the value as it is and are not able to consider the meaning or unit of the value. Therefore, some algorithms might determine the larger values as more important, which is not always true and can decrease the model’s ability to find relevant patterns from the data. This problem is countered by scaling the data, usually between -1 and 1 or between 0 and 1. With scaling, each value is put in context and data originally in different units becomes comparable for the algorithm. [35]

3.3 Feature selection

From the selected data, 272 features were extracted in Section 3.2.2. As was demonstrated, it is not difficult to obtain many features even by simply calculating statistical features from the data. Openly available feature extraction tools also exist, for example in Scikit-learn library [35], and they can be used to extract a huge amount of information. If a large amount of data is used, extracting features and training the models with them becomes more demanding in terms of processing power. Furthermore, not all the available features are significant in the model training and good results may be achieved with less features [36]. For these reasons, feature selection methods are used.

In the case of classification problems, the purpose of feature selection is to identify a subset of features which can discriminate samples belonging to different classes. Usually, features are divided into relevant, irrelevant and redundant. Relevant features increase a model's ability to learn the differences between classes, and irrelevant features decrease this ability. Features that provide the same information as some other features are called redundant. Therefore, the goal would be to identify the relevant features and minimize the irrelevant and redundant features. [37]

The method used in this work is called Minimum Redundancy – Maximum Relevance (mRMR) first introduced in [38]. The mRMR iterates through the feature set and selects the most relevant feature in each iteration and removes it from the set for the next iteration. When a feature has been selected, it is compared to the other selected features. If it is determined that the new feature does not provide additional information, meaning it is redundant, it is discarded. At the end, a subset of relevant features with a low redundancy has been produced.

```
selected_features = mrmr_classif(X = the dataset,
                                 y = the classes of each sample,
                                 K = the number of features to select)
```

Program 2. *Usage of the mRMR method in Python.*

Considering the scope of this work, the feature selection in practice is done using a ready to use a Python package, mrmr-selection. Program 2 presents the syntax of the method. The readers are free to familiarize themselves with the detailed equations and definitions of the mRMR presented in [38].

3.4 AI-based models in classification

AI-based models are widely used for classification problems. For this work, three models, specifically machine learning -based (ML), available in the Scikit-learn library [35] are used: K-Nearest Neighbours (KNN), Stochastic Gradient Descent (SGD) and Decision Tree (DT). As the purpose is to demonstrate how an ML-based model could be used to classify component health states, only the basic logic of each model is described here. Hyperparameter tuning is also out of the scope of this work. The following model definitions are based on the descriptions provided in [35].

The KNN is the simplest of the considered models as it does not require any actual learning from the model as it simply saves all the given training data. When the model is asked to predict the class of a new sample, it finds a given number of the data points that are closest to the new sample from the training data. Usually, Euclidean distance is used to calculate the difference between two samples. From those nearest data points, a class is assigned based on what is the majority class among them. Generally, the KNN performs well on small data sets where the classes are clearly separated. For example, by plotting all data points it could be estimated whether this model will work on a specific dataset based on how the data points for each class differ from each other.

Another simple model is the SGD classifier, which consists of linear classifiers with SGD training, meaning that the chosen linear model is optimized by minimizing the loss function. This is done iteratively, updating the model parameters each round according to the estimated gradient of loss. SGD has also been applied to large-scale ML-problems in text classification and natural language processing.

The DT is a model that produces a tree-like structure of decision rules based on the data features and uses those to predict the classes of new samples. Some benefits include that DTs are simple to understand as the formed decision trees can be visualized, they are computationally efficient even with larger datasets.

4. RESULTS AND DISCUSSION

This chapter presents the results obtained deploying the methods introduced in the previous chapter. The classification results of each model are compared between the health different health states of the components and between using all extracted features versus only selecting a few of them to be used. The presented results used a window size of 100 and a step size of 10 for the calculation of the statistical features.

In addition to classification accuracy, confusion matrices and recall values are used to present the results of the classification as they provide complimenting information about the model performance. The confusion matrices have the true labels on the y-axis and the predicted labels on the x-axis. The numerical values in the matrix represent the number of predictions corresponding to each actual and predicted label pair. The labels correspond to those introduced in Chapter 3.1 in Table 2.

The recall value is defined as the proportion of true positive predictions to all positive predictions as

$$recall = \frac{TP}{TP + FN} \tag{1}$$

Where TP is the number of true positives and FN is the number of false negatives. A high recall value indicates that a classifier has the ability to find a high number of the positive samples, value of 1 being the best and 0 the worst. [3]

4.1 Results using all features

In this section, the classification results of models trained with all the 272 extracted features. Figures 4-6 present the results for the pump health states classification. All models predicted the healthiest state 0, which represents no leakage, with a perfect score. However, differentiating the states 1 - weak leakage, and 2 - severe leakage, proved to be a little more challenging for some of the models, resulting in varying number of false predictions between the models. Decision Tree had the lowest number of false predictions.

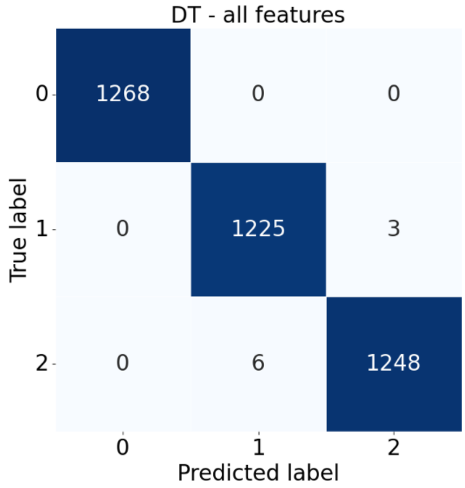

Figure 4: *Confusion matrix – pump case, Decision Tree, all features.*

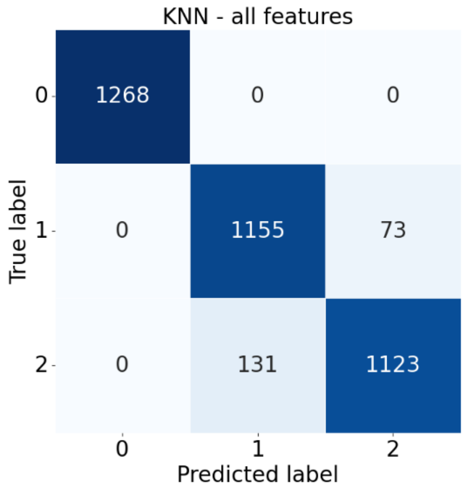

Figure 5: *Confusion matrix – pump case, K Nearest Neighbors, all features.*

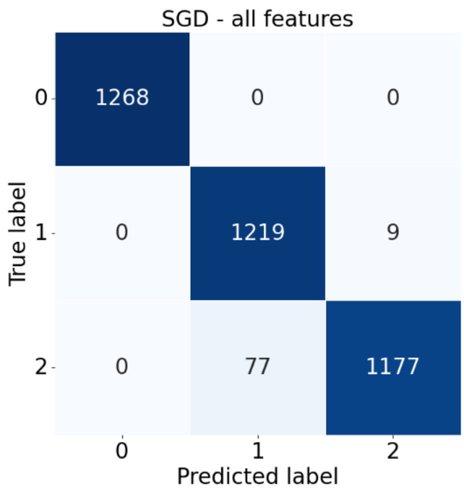

Figure 6: *Confusion matrix – pump case, Stochastic Gradient Descent, all features.*

The recall values of the pump case for each health state according to used model are presented in Table 4. As expected based on the number of false predictions for each model, the recall values are fairly high indicating a good success rate for the classification.

Table 4. *Recall values for the pump case.*

Health state	DT	KNN	SGD
0	1.00	1.00	1.00
1	0.99	0.91	0.96
2	0.99	0.91	0.96

The results for the accumulator health state classification are presented in Figures 7-9. Also in this case, the models produced the highest number of correct predictions for the healthy state - 130 bar, which was predicted correctly by all models. The Decision Tree had the highest performance with zero incorrect predictions, and SGD only had a total of two incorrect predictions. The KNN on the other hand provided a much larger number of false predictions for the three faulty states. It had most difficulty differentiating the states 90 and 100 bar.

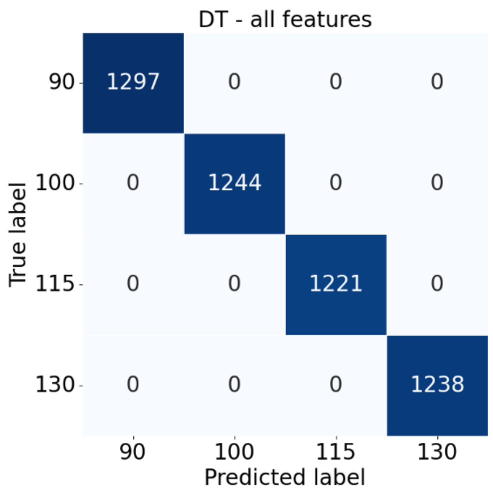

Figure 7: *Confusion matrix – hydraulic accumulator case, Decision Tree, all features.*

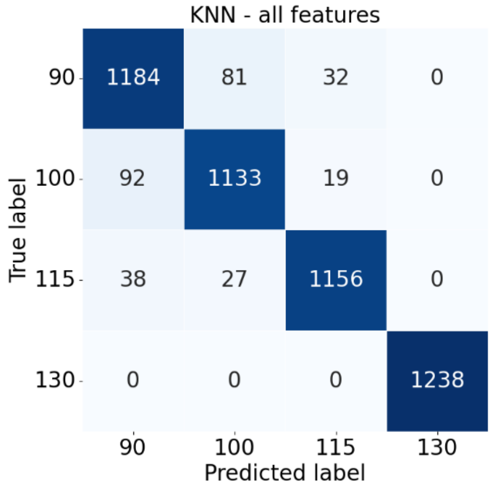

Figure 8: *Confusion matrix – hydraulic accumulator case, K Nearest Neighbors, all features.*

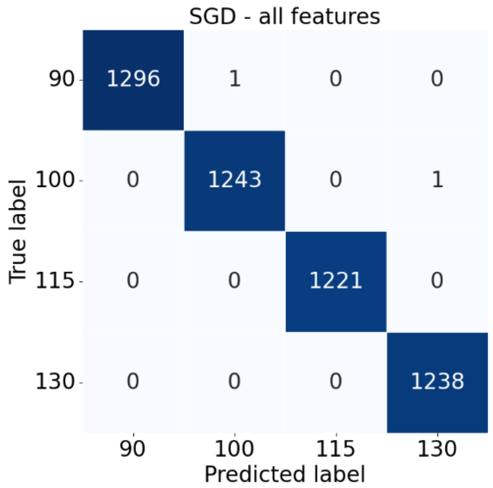

Figure 9: *Confusion matrix – hydraulic accumulator case, Stochastic Gradient Descent, all features.*

The recall values of the pump case for each health state according to used model are presented in Table 5. Although the KNN had many false predictions compared to the other models, the recall values remain above 0.9.

Table 5. *Recall values – hydraulic accumulator case.*

Health state	DT	KNN	SGD
130 bar	1.00	1.00	1.00
115 bar	1.00	0.94	1.00
100 bar	1.00	0.91	0.99
90 bar	1.00	0.91	0.99

In the case of both components, the best performing model was the Decision Tree. In the pump case, it reached an average validation accuracy of 99%, and in the accumulator case 100%. Next, the results using feature selection are presented.

4.2 Results with feature selection

Using the mRMR feature selection method introduced in Chapter 3.3, 8 feature sets containing 10 to 45 features were selected separately to be used for the pump and accumulator health state classification. To get a general idea of the performance depending on the number of features used, they are first compared in terms of accuracy. In the case of both components, a high validation accuracy was achieved with a low number of features.

Table 6 presents the average accuracy of each feature set and model for the pump. After 15 features, the accuracy does not significantly increase. Therefore, 15 features are used for the pump case in the next step.

Table 6. *Pump - average validation accuracy [%] of each feature set.*

Number of features	DT	KNN	SGD
10	98	98	95
15	99	99	95
20	99	99	93
25	99	99	94
30	99	99	92
35	99	99	95
40	99	99	96
45	99	99	96

Similarly, Table 7 presents the average accuracy of each feature set and model for the hydraulic accumulator. The same phenomenon can be observed as for the pump. In the case of the accumulator, the lowest number of features after which the accuracy does not significantly increase is 10. For the accumulator case, 10 features are used in the next step.

Table 7. Hydraulic accumulator - average validation accuracy [%] of each feature set.

Number of features	DT	KNN	SGD
10	100	100	95
15	100	100	97
20	100	100	99
25	100	100	99
30	100	100	99
35	100	100	98
40	100	100	95
45	100	100	99

Figures 10-12 present the classification results for the pump health states, when the models were trained using 15 features. In this case, the performance of Decision Tree and KNN are nearly equal, and KNN performed better. SGD had the most incorrect predictions. However, once again the healthy state was the easiest for the models to predict, having no incorrect predictions.

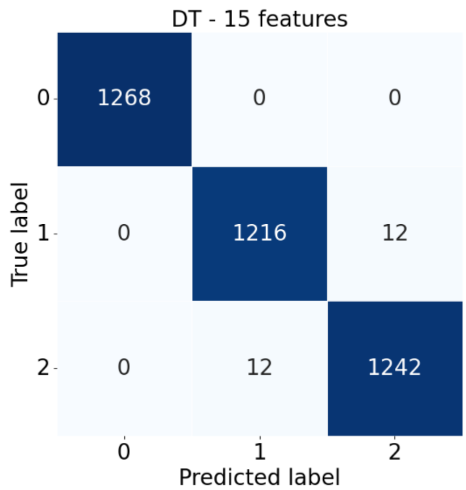

Figure 10: *Confusion matrix – pump case, Decision Tree, 15 features.*

Compared to the results when all features were used, the Decision Tree (Figure 10) doubled the number of incorrect predictions. Since their number compared to the total predictions remains still very low, the model performed practically the same. The KNN, presented in Figure 11, on the other hand improved its performance and the number of incorrect predictions decreased significantly compared to the previous results. The results are consistent with the fact that KNN models usually perform better with smaller datasets. Since the results were also good with all features, improved results with fewer features could have been expected.

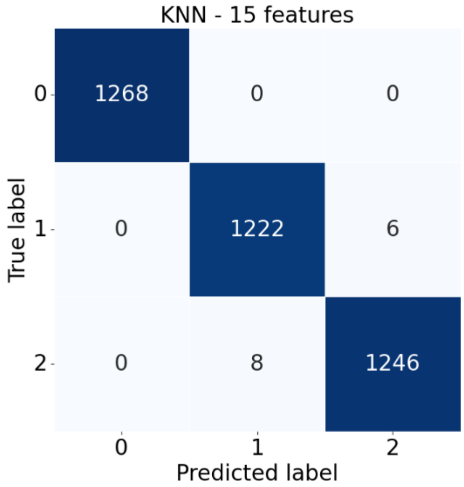

Figure 11: *Confusion matrix – pump case, K Nearest Neighbors, 15 features.*

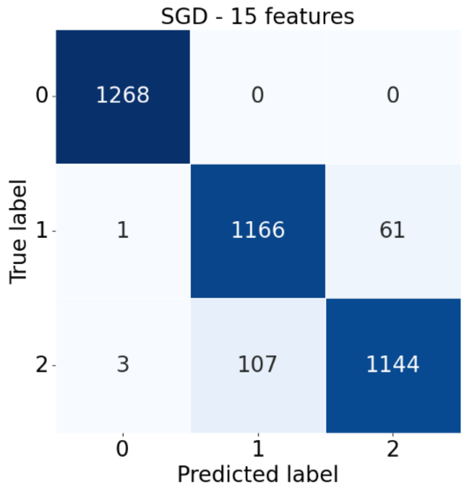

Figure 12: *Confusion matrix – pump case, Stochastic Gradient Descent, 15 features.*

SGD, in Figure 12, performed worse with the decreased number of features and increased the number of false predictions. In three cases, the worst health state 2 was predicted to be healthy when previously only the faulty health states were confused with each other. The recall values of the pump case using 15 features are presented in Table 8. The values for the Decision Tree remain the same as with all features. For the KNN, the recall values of the faulty health state 1 and 2 increased. As the SGD predicted a few of the faulty states as healthy and the incorrect predictions increased overall, the recall values decreased. The Decision Tree and KNN reached a validation accuracy of 100%.

Table 8. Recall values – pump case, 15 features.

Health state	DT	KNN	SGD
0	1.00	1.00	1.00
1	0.99	0.99	0.94
2	0.99	0.99	0.91

Also in the hydraulic accumulator case, Decision Tree and KNN performed well despite only using 10 features in the training. The results are presented in Figures 13-15. The performance of the Decision Tree, in Figure 13, was not affected by the decrease in features and again predicted all samples correctly. As for the pump case, the performance of the KNN was significantly improved. Figure 14 presents the classification results of the KNN in the accumulator case using 10 selected features. In this case, it predicted all samples correctly.

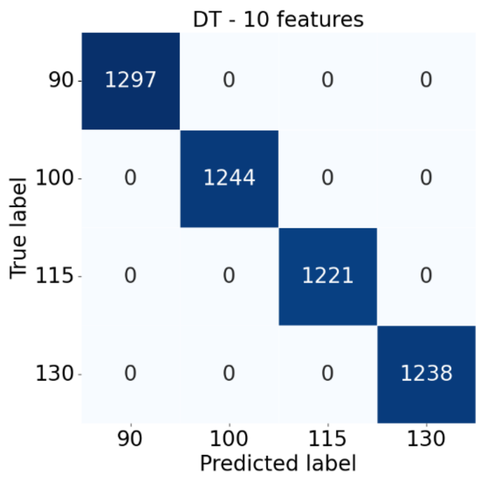

Figure 13: *Confusion matrix – hydraulic accumulator case, Decision Tree, 10 features.*

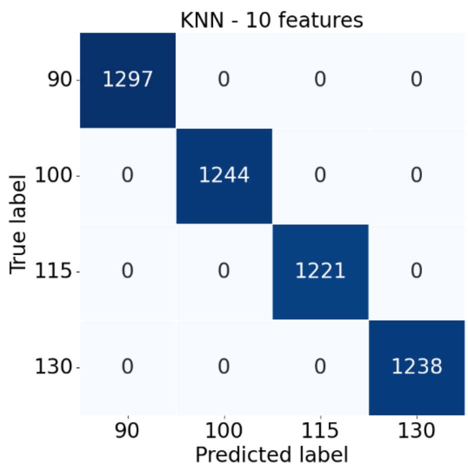

Figure 14: *Confusion matrix – hydraulic accumulator case, K Nearest Neighbors, 10 features.*

The SGD did not perform as well with the limited training data. It provided an increased number of false predictions mainly for the 100 and 115 bar states but predicted all healthy samples correctly.

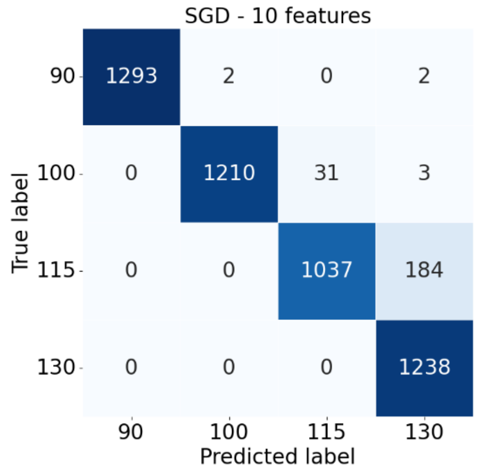

Figure 15: *Confusion matrix – hydraulic accumulator case, Stochastic Gradient Descent, 10 features.*

The recall values for the accumulator case using 10 features are listed in Table 9. According to the changes in the classification results, the recall values follow those changes. In this case both the Decision Tree and KNN reached a validation accuracy of 100%.

Table 9. *Recall values – hydraulic accumulator case, 10 features.*

Health state	DT	KNN	SGD
130 bar	1.00	1.00	1.00
115 bar	1.00	1.00	0.87
100 bar	1.00	1.00	0.96
90 bar	1.00	1.00	0.98

The top features selected with mRMR for both the pump and accumulator are presented in Table 8. The ranking represents the order of importance determined by the mRMR algorithm. For the sake of comparison, 15 top features for each component are listed, even though only 10 were used for the accumulator. All selected features are from the data provided by volume flow (FS) and temperature (TS) sensors, and almost all of them are the same for both components, although not exactly in the same order of importance.

Table 10. *Top features for the pump and hydraulic accumulator – mRMR.*

Ranking	Pump	Hydraulic accumulator
1.	TS3 absolute energy	TS3 absolute energy
2.	FS2 minimum	FS2 minimum
3.	FS2 median	FS2 sum of values
4.	FS2 mean	FS2 mean
5.	FS2 sum of values	FS2 median
6.	FS2 maximum	FS2 maximum
7.	FS2 absolute energy	FS2 absolute energy
8.	TS2 absolute energy	TS2 absolute energy
9.	TS3 mean	TS3 mean
10.	TS3 sum of values	TS3 sum of values
11.	TS3 median	TS3 minimum
12.	TS3 minimum	TS3 median
13.	TS3 maximum	TS3 maximum
14.	TS1 absolute energy	TS1 absolute energy
15.	TS2 mean	TS2 mean

To further inspect the information that is obtained from the selected features, Figure 4 presents the values of the highest-ranking feature, TS3 absolute energy, corresponding to each pump health state. As the figure shows, the healthy state without any leakage is clear to differentiate from the other two health states with a human eye. This justifies the high classification accuracy for this health state across all used models, even the simplest ones.

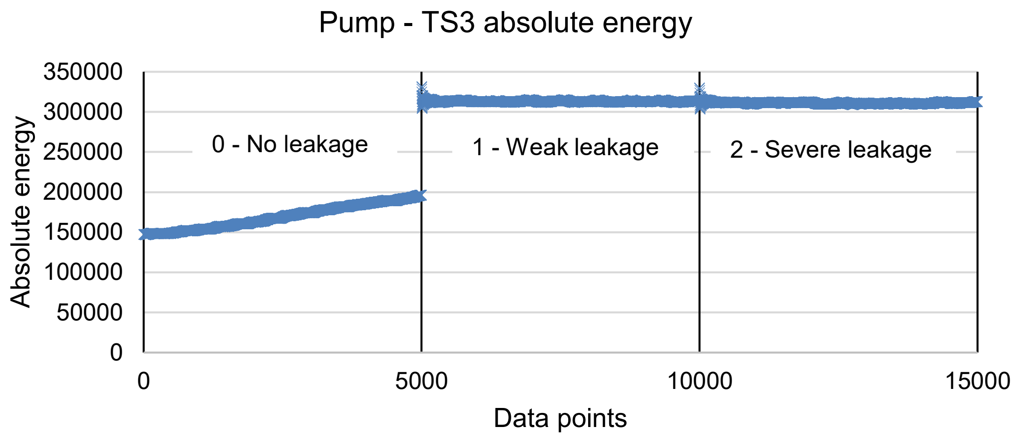

Figure 16: *Health states of the pump and calculated absolute energy.*

Additionally, Figure 5 presents the TS3 absolute energy values for each of the hydraulic accumulator health states. Similarly to figure 4, the healthy state at 130 bar is clearly different from the other states. In this example, however, the differences between all the health states with decreased pressure are noticeable.

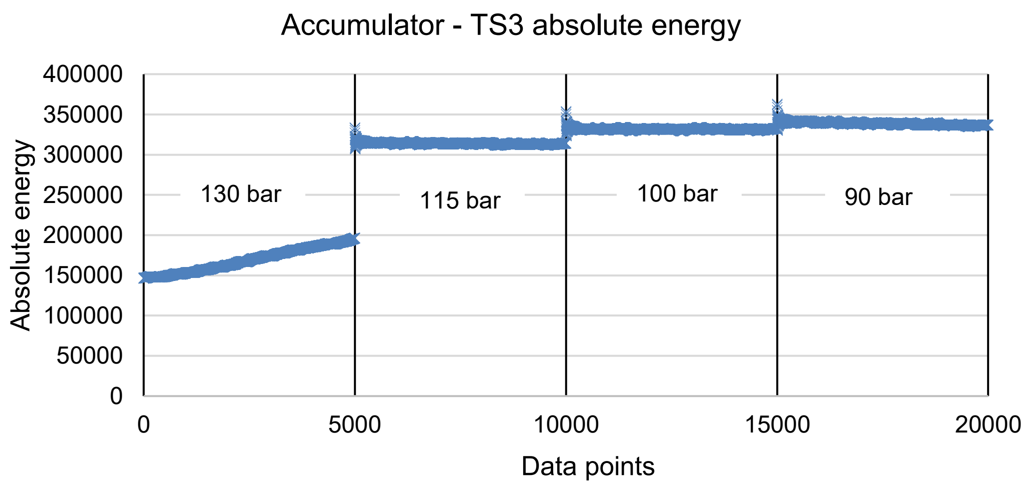

Figure 17: *Health states of the accumulator and calculated absolute energy.*

It is also worth noting that the sensors providing the top features for each component were the ones that have been re-sampled to increase the number of data points. The temperature sensors (TS) originally used a sampling frequency of 1 Hz, and the flow sensors (FS) used a sampling frequency of 10 Hz. As the data used is experimental, it does not have as much noise as there could be if it was obtained from a real-life system. Because of this, generation of such a high number of additional samples might not be as risky, since the original data is fairly noise free.

5. CONCLUSIONS

Mobile machinery maintenance is in a turning point in deploying predictive maintenance methods. In this work, the benefits of predictive maintenance compared to the other maintenance strategies were introduced and concluded that they can bring additional value when applied together with preventive and reactive maintenance. One of the possible approaches to predictive maintenance is condition monitoring. In this work, publicly available data of health states of hydraulic components was used as learning material for ML-based classification models.

The performance of three models, Decision Tree, KNN, and SGD, were compared. The models were trained to individually predict the health states of a pump and hydraulic accumulator. When all features were included in the model training, the classification of the pump health states achieved a validation accuracy of 99% and accumulator case 100%. Decreasing the number of features from 272 to 15 and 10 did not result in a decrease in the validation accuracy. With 15 features used in the training, both Decision Tree and KNN achieved a validation accuracy of 100%. They achieved the same accuracy for the accumulator case where only 10 features were used in the training.

Such high accuracy scores may have been the result of the fairly low number of data points and the initial method used in the measurements. Although the data is from an experimental setup, it does not reflect the real operation of a mobile machine. The setup lacks the variation related to operating in different environments and executing different work functions as it only repeats constant load cycles. Accuracies as high would not be achievable if a large number of measurements from actual operation was used in the training.

The workflow used in this work could be applied to a real case if data related to component health states would be available. However, the preprocessing of such data may require more steps, for example filtering. In addition, a larger amount of data would be needed to represent the complexity of the machine work functions. The future steps would include gathering of data and applying the processing chain in a work machine to make the health state predictions during the operation.

REFERENCES

[1] A. Akers, M. Gassman, and R. Smith, Hydraulic Power System Analysis, CRC Press, Boca Raton, 2006.

[2] A. Ahonen, M. de Koning, T. Machado, R. Ghabcheloo, O. Sievi-Korte, An exploratory study of software engineering in heavy-duty mobile machine automation, Robotics and Autonomous Systems, Vol. 165, 2023, p. 104424.

[3] H. A. Khan, U. Bhatti, K. Kamal, M. Alkahtani, M. H. Abidi, S. Mathavan, Fault Classification for Cooling System of Hydraulic Machinery Using AI, Sensors, Vol. 23, No. 16, 2023, p. 7152.

[4] P. Zhang, W. Hu, W. Cao, L. Chen, M. Wu, Multi-fault diagnosis and fault degree identification in hydraulic systems based on fully convolutional networks and deep feature fusion, Neural Computing and Applications, 2024.

[5] G. Wrat, P. Ranjan, and J. Das, Leakage and Its Effects In Various Hydraulics Driven System.

[6] G. Xu, C. Ma, Z. Gao, X. Hu, Y. Luo, Modeling and simulation of aero-hydraulic pump wear failure, in 2017 Prognostics and System Health Management Conference (PHM-Harbin), Harbin, China: IEEE, 2017, pp. 1–7.

[7] X. Zhao, S. Zhang, C. Zhou, Z. Hu, R. Li, J. Jiang, Experimental study of hydraulic cylinder leakage and fault feature extraction based on wavelet packet analysis, Computers & Fluids, Vol. 106, 2015, pp. 33–40.

[8] J. F. Bateman, Preventive maintenance: Stand alone manufacturing compared w, Industrial Management, Vol. 37, No. 1, 1995, p. 19.

[9] J. Carroll*, S. Koukoura, A.McDonald, A. Charalambous, S. Weiss and S. McArthur, Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques, Wind Energy, Chichester, England, vol 22, no 3, 2019, pp. 360-75.

[10] R. Meissner, A. Rahn, K. Wicke, Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making, Reliability Engineering & System Safety, Vol. 214, 2021, p. 107812.

[11] A. AG, S.O.S-näytteenotto-ohjelma Cat-koneellesi, website. Available (accessed on 15.5.2024): https://www.avesco-cat.fi/fi/palvelut/huolto-ja-korjaus-palvelut/sossm-oljynaytteiden-ottaminen.

[12] Metsäkoneiden huolto, John Deere FI, website. Available (accessed on 15.5.2024): https://www.deere.fi/fi/varaosat-ja-huolto/huolto/.

[13] Huolto ja kunnossapito, Komatsu, website. Available (accessed on 15.5.2024): https://www.komatsuforest.fi/palvelut/huolto-ja-kunnossapito.

[14] SFS-EN 13306:2017:en Maintenance. Maintenance terminology.

[15] Complex System Maintenance Handbook, in Springer Series in Reliability Engineering, London: Springer London, 2008.

[16] P. Coandă, M. Avram, V. Constantin, A state of the art of predictive maintenance techniques, IOP Conference Series: Materials Science and Engineering, Vol. 997, No. 1, 2020, p. 012039.

[17] R. Khodabakhshian, A review of maintenance management of tractors and agricultural machinery: preventive maintenance systems, Agricultural Engineering International: CIGR Journal, Vol. 15, No. 4, 2013, pp. 147–159.

[18] A. Rachad, L. Gaiz, K. Bouragba, M. Ouzzif, Predictive Maintenance-as-a-Service (PdMaaS) in Industry 4.0 using Blockchain, in 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), 2023, pp. 1–6.

[19] K. F. Gallimore and R. J. Penlesky, A Framework For Developing Maintenance Strategies, Production and Inventory Management Journal, Vol. 29, No. 1, 1988, p. 16.

[20] R. J. Shipley, B. A. Miller, R. J. Parrington, Introduction to Failure Analysis and Prevention, Journal of Failure Analysis and Prevention, Vol. 22, No. 1, 2022, pp. 9–41.

[21] A. F. S. Silva, E. C. S. Santos, R. M. do N. Luz, R. da S. Fernandes, Analysis of preventive maintenance strategy in off-road trucks, Gest. Produção, Vol. 30, 2023, p. e5923.

[22] S. Choubey, R. Benton, T. Johnsten, Prescriptive Equipment Maintenance: A Framework, in 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA: IEEE, 2019, pp. 4366–4374.

[23] M. M. Hassan, C. Boehl, M. Kuruppu, Predictive Maintenance as an Integral Part of Asset Life Cycle Maintenance Model, in Asset Intelligence through Integration and Interoperability and Contemporary Vibration Engineering Technologies, J. Mathew, C. W. Lim, L. Ma, D. Sands, M. E. Cholette, and P. Borghesani, Eds., Cham: Springer International Publishing, 2019, pp. 229–237.

[24] J. Gama, R. P. Ribeiro, B. Veloso, Data-Driven Predictive Maintenance, IEEE Intelligent Systems, Vol. 37, No. 4, 2022, pp. 27–29.

[25] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, G. P. Li, Predictive maintenance in the Industry 4.0: A systematic literature review, Computers & Industrial Engineering, Vol. 150, 2020, p. 106889.

[26] Predictive Maintenance Market Size, Report 2023 To 2032, website. Available (accessed on 26.5.2024): https://www.precedenceresearch.com/predictive-maintenance-market.

[27] L. Pedro Coelho, W. Richert, M. Brucher, Building Machine Learning Systems with Python: Explore Machine Learning and Deep Learning Techniques for Building Intelligent Systems Using Scikit-Learn and TensorFlow, 3rd Edition, Birmingham, UNITED KINGDOM: Packt Publishing, Limited, 2018

[28] B. Cui, Y. Weng, N. Zhang, A feature extraction and machine learning framework for bearing fault diagnosis, Renew. Energy, Vol. 191, 2022, pp. 987–997.

[29] N. Helwig, E. Pignanelli, A. Schütze, Condition monitoring of a complex hydraulic system using multivariate statistics, in 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2015, pp. 210–215.

[30] O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, L. Abu-aligah, Multiclass feature selection with metaheuristic optimization algorithms: a review, Neural Compututing and Applications, Vol. 34, No. 22, 2022, pp. 19751–19790.

[31] K. Maharana, S. Mondal, B. Nemade, A review: Data pre-processing and data augmentation techniques, Global Transitions Proceedings, Vol. 3, No. 1, 2022, pp. 91–99.

[32] F. Alahmari, A Comparison of Resampling Techniques for Medical Data Using Machine Learning, Journal of Information & Knowledge Management, Vol. 19, No. 01, 2020, p. 2040016.

[33] Machine Learning for Imbalanced Data, website. Available (accessed on 3.8.2024): https://learning.oreilly.com/library/view/machine-learning-for/9781801070836/.

[34] R. Shafique et al., Breast Cancer Prediction Using Fine Needle Aspiration Features and Upsampling with Supervised Machine Learning, Cancers, Vol. 15, No. 3, 2023.

[35] F. Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, Vol. 12, No. 85, 2011, pp. 2825–2830.

[36] Z. Zhao, R. Anand, M. Wang, Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform, in 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2019, pp. 442–452.

[37] Aggarwal, C. Charu, Feature Selection for Classification: A Revie', Data Classification, Chapman and Hall/CRC, 2015, pp. 65–92.

[38] Hanchuan Peng, Fuhui Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, 2005, pp. 1226–1238.