UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science

Seminar

Machine Learning for Predictive Maintenance: A Multiple Classifier Approach

by

Mondal, Ambuj Kumar

Supervisor:
Prof. Dr. Theodor Lettmann

Paderborn, March 20, 2022

1 Abstract

Predictive maintenance is an approach to overcome the uncertainties involved with running machines and aim to lower the maintenance cost and operating costs thereby increasing run time availability. Analyzing various machine learning algorithms and their efficiency in the prediction of failures and remaining runtime of machines. This paper is based on the actual work by Susto et. al. [SSP$^+$14] presented in their 2014 paper Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. Using different machine learning algorithms and techniques such as Support Vector Machines (SVM), Kernel methods and clustering approaches such as K-Nearest Neighbors (KNN), the paper evaluates different ways to determine the predictability of errors and failure. Using classifiers to predict failures in machines as a classification problem, this paper proposed to use multiple classifiers that work in parallel and increase the accuracy of the algorithm.

This paper evaluates various approaches of predictive Maintenance concerning the semiconductor manufacturing industry and suggests a multi classifier approach to predict remaining lifetime and potential failures to avoid them. The proposed PdM methodology helps in taking decisions related to maintenance dynamically and the decision rules are adapted while the due course of the process. The PdM approach can be applied to problems related to high dimensional data and censored data. This is observed by training various classification modules differently that provide a balance between the unexpected breakage frequency and unused remaining lifetime of machines. This enables to deduce a maintenance decision system that is based on the policy to reduce the operating cost and the expected costs. The effectiveness of this method is tested using Monte Carlo simulators on datasets generated and collected in a semiconductor manufacturing unit.

Keywords: Machine Learning (ML), preventive Maintenance (PvM), Predictive Maintenance (PdM), Support Vector machines (SVM), K nearest neighbors (k-NN), Multiple Classifier Predictive Maintenance (MC PDM)

2 Introduction

Machines are the backbone of any industry and despite the capabilities of doing repeated tasks nonstop, machines also do tear-out. An unexpected breakdown of a machine or any of its components can severely hamper production units and degrade the quality of product, not to mention the additional cost and time consumption to fix it. Hence, machines need maintenance. Maintenance can be planned to avoid outrages and breakdowns, but it is not always efficient.[DP17] Without scientific evaluation, one cannot predict the state of a machine and if it is about to break. This may lead to premature maintenance or replacement of components when it has capability to serve more. This is again a loss for the business. To mitigate this issue, the need is to somehow predict the future of a machine. By machine, we can assume each component and not just the entirety of it. Prediction needs deep knowledge of the process, especially in manufacturing units.[SG04] Additionally, domain knowledge is crucial that may vary from industry to industry for the same machine. with this expertise, a person can estimate if a machine is running properly or it is facing some issues and how big problem that may cause. This science of prediction is termed Predictive maintenance (PdM).[SBDL12]

Predictive Maintenance (PdM) aims to foresee the early signs of a possible failure and enable proactive maintenance i.e., to mitigate risks of breakdown by acting before an actual failure occurs [Mob02]. This also empowers to predict the remaining lifetime of machines. Predictive Maintenance helps in reducing operating costs in long term and prove better health of machines thereby adding to the quality of products and services [Sel17]. Interconnected network tools have increased the collaborative machine designing and with the advancement of communication technologies [BR10], radio waves and wireless technologies, different devices and components can talk to each other and this leads to the development of data-driven [ZYW19] analysis of health metrics of machines to predict future states and prevent outrages. Machine Learning plays a vital role in Predictive Maintenance.

The knowledge of a domain expert required to successfully run a a machine can be imitated by analyzing the enormous data being produced by the operations and effective machine learning algorithms can find out patterns to predict faults and failures even if there is no imminent danger to ensure that operations work smoothly. Various sensors for pressure, heat, vibration [KB20][SG04], etc. are used in machines to monitor the health of a machine. The volume of data these sensors produce is raw but if correct features are extracted, can result in a realistic prediction of the machine state. Machine Learning (ML) deals with analyzing these data and bringing in helpful insights.

Especially in the manufacturing sector where competition is very high[BR10], businesses target to lower the costs occurring because of downtimes and defective products. This approach can be classified into three categories [SG04]

1. **Run to Failure** (R2F): Often termed as 'Corrective maintenance'[Mob02], this is a simple approach where a team finds out the reason for failure and tries to fix it.

But this can only be done after a failure has already happened. So, in terms of effectiveness, this is not preventing a failure but rectifying a failure. R2F approach can be thought of to be of two different types descriptive and diagnostic

- **Descriptive**: The analysis of a failure to access the impact and its effects. After a failure occurs, the team analyzes what has happened. This is important to find out the root cause
- **Diagnostic**: Once the assessment of a failure is done that what has happened and what are its implications, the team finds out why it happened. This is a diagnostic approach and data plays an important role here.

2. **Preventive Maintenance** (PvM) or timed maintenance are routines and done at periodic intervals n a planned manner. Most industries plan their downtime and operate maintenance activities. The frequency and timeline of planned activities depend on the nature of the industry and the manufacturer's guidance or experience. The planned schedule for maintenance helps to overcome abrupt breakage and failures. But this process is not very efficient as firstly, there is downtime involved and corresponding cost associated with it. Secondly, it leads to unnecessary corrective measures, that might not be needed thus, wasting resources, and increasing operating costs which makes this process inefficient. Another way is to plan the maintenance and operate only certain components that need immediate attention based on some criteria. This is conditional maintenance, but it is not always practical to pinpoint such targets due to a lack of proper knowledge and expertise.

3. **Predictive Maintenance** (PdM): A better way to prevent a failure is to predict it. The primary aim of PdM is failure prevention while the secondary aim is efficient operations [Sel17]. Using continuous monitoring and based on expertise and the data available, PdM predicts what might happen in the future with the current situation. Detection of an anomaly in advance gives ample scope to fix the upcoming problem in a much better way with minimal cost. This method needs empirical knowledge about the system and lots of data from the health monitoring systems such as sensors. This statistical approach is harnessed using the power of machine learning. It can read and access thousands of physical variables such as pressure, vibrations, heat, noise, voltage fluctuations, Current flows, etc. A predictive maintenance approach can drastically limit the cost associated by reducing the maintenance frequency by avoiding unnecessary planned downtimes. In other words, it can be said that PdM is the policy of least maintenance. [Sel17]

The figure 1 as demonstrated by Motahare, Pillai, and Ramachandran in their paper Predictive Maintenance and Architecture [MPR18] shows the relation between the maintenance costs and availability of a system after failure. The corrective or R2F maintenance has a high cost associated because of the downtime. During this phase, the system is unavailable due to fixing or maintenance. While the planned Maintenance is better than an actual failure, it still has a high cost associated. Predictive maintenance

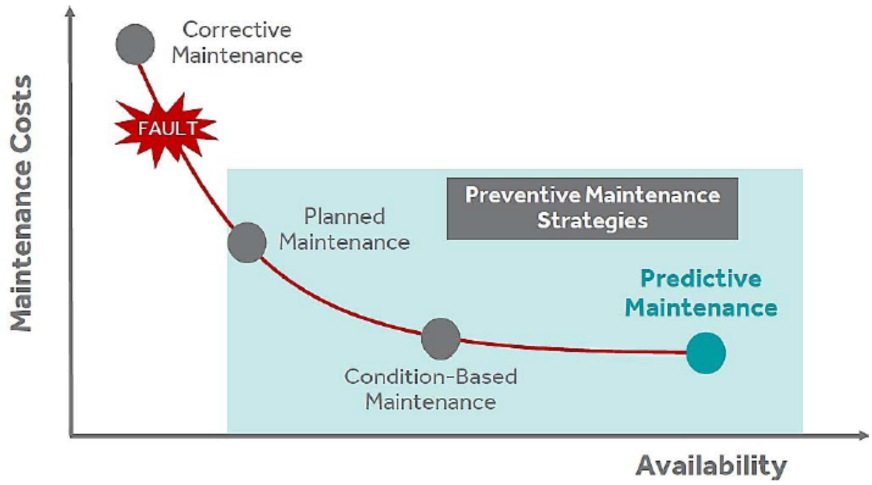

Figure 1: Maintenance cost vs availability
[MPR18]

is the best shot in the long term as it increases the availability of the system and has much less cost associated with maintenance.

The paper machine learning for Predictive Maintenance, 2015 by Susto et.al. [SSP$^+$14] suggests a new method of PdM based on Multiple classifiers termed Multiple classifier Predictive maintenance (MC-PdM). Used for integral fault types [SSP$^+$14] which means the faults occurring due to continuous usage over time or wear and tear faults, MC-PdM can predict faults based on the unbalanced datasets and allows a planned maintenance schedule with the least cost associated.

3 Machine Learning in Maintenance

Machine Learning is a process that uses algorithms and a multitude of data collected from various sources to define a model that can intelligently perform complex tasks and further improve itself with its own experiences. With loads and loads of data being generated, manufacturing Industries harness the power of Machine Learning. Machine learning can be classified into two broad categories Supervised and unsupervised.

Supervised Learning is one of the most used use-cases in the industry is Supervised Learning, where the outcome results are already known. An algorithm is developed for the machine to learn, and train based on the data. Just like a human child is taught based on patterns and can be led by an instructor to follow the correct learning path, in a supervised learning model, the dataset is labeled and already mapped. The model learns the dataset and predicts an outcome based on its experience. This process requires huge human intervention to map the data properly. Data sets are annotated and fed to the algorithm to generalize the input-output relation to a new set of data inputs. For example, assessment of quality as OK or Not OK. Supervised Learning algorithms are mostly used in binary classification and prediction. Prior information of occurrences of failures is available in the modeling dataset. From the perspective of Machine Learning, a supervised approach to PdM needs the availability of labeled dataset S Where the couple $\{x_i, y_i\}$ are the observations containing information related to the i^{th} iteration of the process. Here the vector x_i belongs to a set of real numbers and contains information related to p variables of the process. The output y_i determines the type of algorithm to be used. If y_i is continuous, it will lead to a regression problem and if y_i assumes categorical values, then it becomes a classification problem. further, supervised learning can be bifurcated into two different categories- classification and regression.

$$\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$$

$Classification$ refers to using an algorithm for accurately assigning input data into a specific class. The algorithm recognizes some specifics within the dataset and tries to conclude if the input belongs to one class or another. The most common classification algorithms are Support vector machines (SVM), Decision trees, K-nearest neighbors, random forest, etc. Classification algorithms can be used to classify between healthy and non-healthy states of machines or process iterations based on the processed data, but they may not naturally map for health factors.

Regression refers to the use of an algorithm for understanding the relationship between different dependent and independent entities. This type of learning is mostly used to make some projections and reports such as for a given business, how much is the sales and revenue. Some popular algorithms for regression are Linear Regression, logistic Regression, and polynomial regression. The simplest form of classification algorithm divides the input into two classes by a linear line where one side is class A and the other side is not class A. It divides a plane into two parts and each side represents a yes/No side. For example, an image is of a cat or not.

$Remaining Useful Lifetime$ (RuL) In predictive Maintenance problems, a regression-based approach is used to determine the remaining useful lifetime of equipment or process. Remaining useful lifetime refers to the duration of a machine that it can run further before it faces a breakdown. During Preventive maintenance, an equipment is restored or replaced while it might run longer. This leads to wastage to available run-time of the equipment.[BR10]

$Unsupervised Learning$: The truly intelligent form of machine learning is when data is just given as input and the algorithm finds hidden structures in the data and extracts the important information. It automatically classifies the data and draws a hypothesis that can predict the outcome when provided with further input data. This type of learning is useful in clustering problems and probability density estimation. In Unsupervised Learning, logistic and process information is available, but no maintenance-related data is available.

The quality of data depends on the management of data by the maintenance policies. In the case of a Run to Failure policy, the data between two successive failures are logged and can be analyzed. In such scenarios, supervised learning methods can be easily applied. When a company has preventive Management policies, it may not be possible to observe a full maintenance cycle as the full maintenance is generally avoided. Instead, selective component maintenance data is available. In such scenarios, unsupervised learning methods are helpful to draw a distinct model for prediction. In terms of complexity, Unsupervised models are very complex compared to supervised models. These need much higher processing powers and CPU cores, hence, whenever possible it is advisable to adhere to supervised models.

A major difference between Supervised learning and unsupervised learning is the lack of a labeled data set in the latter. For instance, if there are several objects of varied shapes and colors such as square, rectangle, circle, etc., a supervised learning algorithm can classify these quite easily because it has already been trained with similar inputs. Therefore, image recognition applications mostly use supervised learning algorithms. But the same set of inputs when given to an unsupervised learning algorithm is very difficult for it to classify due to changes in shapes, color, etc. The algorithm does not have any prior knowledge about shapes, and it will define its model. It doesn't know if the shape is a square or a circle but will generate its labels for these and will apply this knowledge to further shaped with a degree of probability to determine it correctly.[MPR18]

In real terms, a solution that can be in between supervised and unsupervised is more practical. This is called semi-supervised learning.

SemiSupervised Learning: An intermediate between supervised and unsupervised, is a semi-supervised learning method. In this, approach, data for a particular state is known and most data for the complete system is not known or labeled. In this approach, the model can detect *outliers* based on deviations from the given state. This is useful in quality assessment and needs proper knowledge about normal production behavior. Examples of usage of Semi-supervised learning are Anomaly detection and One-Class classification problems.

SupportVectorMachines (SVM): a popular supervised learning algorithm that is used to solve both classification and regression problems. It enhances the linear classification by constructing a hyperplane. A hyperplane is a decision boundary classifying two different objects such as oranges and apples on either side of the plane. SVMs are a set of supervised learning methods that are beneficial for classification, regression, and outlier detection. An *outlier* is a data point that differs significantly from other observed points and cannot be classified in a class. Mostly, outliers are because of some recording errors or temporary glitches such as power failure. Such outliers are normally removed or ignored while model processing. SVMs are used to reduce high dimensional data into low dimensional hence, benefitting classification.

The process of using machine learning in any dataset is a standard process which involves data exploreation. Once the raw data is accessible, the data is interpolated and features are extracted. Features depict a mathematical value to the metadata in such a way that it can be used to train the machine learning algorithm. It is also important to select the correct features. with high dimensional datasets having thousands of features, it is important to select the important features to make a balance between the performance and accuracy of the algorithm. These features are passed to the algorithm to get the output. In neural network, the algorithm itself extract features and label the data from the data set.

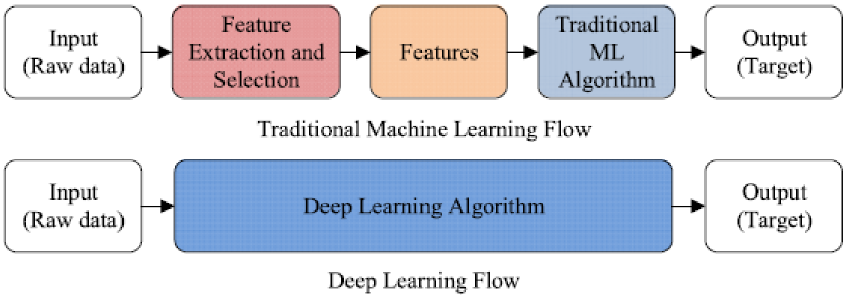

Figure 2: Flow of Ml and DL [ZYW19]

4 MC-PDM or Multiple Classifier Predictive Maintenance

while the classification algorithms are the natural choice to differentiate between healthy and non-healthy states or processes, these do not always lead to health factors for effective Predictive maintenance. So, a multiple classifier approach is introduced to overcome this limitation.

Classification of PdM: Assuming that data for N maintenance cycles are available to access, it can be defined that total machine runs are a summation of all the cycles.

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$$

$$Y(i) = y_i = \begin{cases} \text{F}, & \text{if iteration } i \text{ is faulty} \\ \text{NF}, & \text{if iteration } i \text{ is not faulty.} \end{cases}$$

In the case of R2F, each cycle results in F as the cycle has already failed. Hence the number of instances of F in the dataset is n and for NF it is n-N. A function f(.) is defined as a classification rule and the classifier learns to assign one of the classes PF, NF to each point in the input Space R. This is a weak form of PdM because of two reasons[ZYW19]-

- Only one process per iteration can be classified. This means it is only analyzing the fault and not doing anything to prevent it.
- There is no optimization of operating cost. Every iteration is going to be as costly as the previous one.

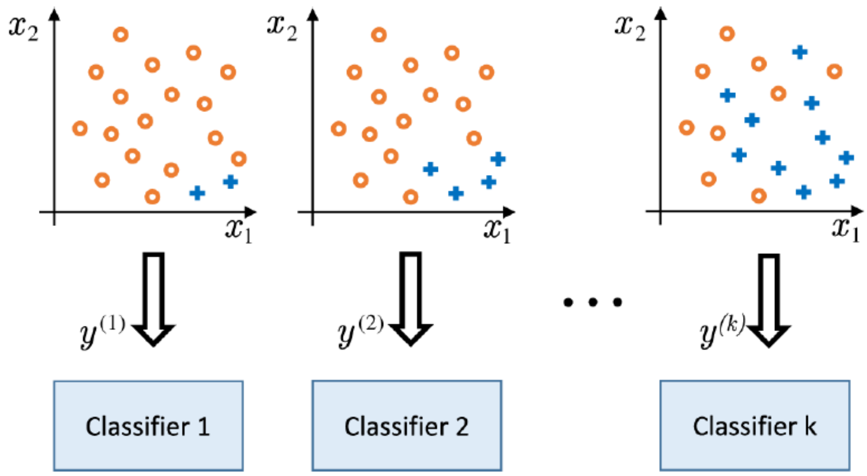

Figure 3: Two dimensional example of k classifiers. [ZYW19]

As depicted in the figure 3, an example of two dimensional data classification using the Multi classifier approach. The orange circles represent points in the class NF(non failure) while the blue crosses represent the data points for class F(Failures). With each iteration, the number of classifiers is increased and the failure horizon is changed. This results in changing the labels from NF to F and vice versa. With k classifiers, after kth iterations, the data points are more stabilized to a particular class. For optimizing the operating costs, two parameters or metrics needs to be considered.[SSP+14]

1. Frequency of unexpected breaks (FuB)- this represents the percentage of failures that were not prevented
2. Amount of unexploited Lifetime (FuL)– this refers to remaining lifetime means an average number of iterations that could have run before actual failure happens provided the preventive maintenance activity has not been carried forward.

The R2F process is initiated only after failures, so, it guarantees that FuB=1 and the FuL=0. Different cost factors associated with FuB and FuL can be assumed as CuB and CuL.

It is evident that the cost for breakage is much higher CuB » CuL as CuB related to unplanned disruptions of the process. It is not feasible to reduce both at the same time but the algorithm tries to find a balance between these two costs and reduce the overall operating cost. A Tradeoff [SSP+14] can be defined using the weighted sum as

$$J = FuB.CuB + FuL.CuL$$

As there is no concrete way to minimize J, it cannot be taken as a policy to prevent failures. Hence it is not ideal for maintenance. Also, it is notable that the variable Y is not balanced, rather it is skewed to a high degree with $NonFailure$ (NF) cases being very very low compared to $Failure(F)$ cases.

Feature Extraction is the process of transforming unprocessed or raw data into some numerical features that can be used by the model to process information while preserving the original information from the actual dataset is termed as feature extraction. Feature extraction involves dimensionality reduction. It is better to transform the raw data and then apply machine learning algorithms, this makes the process faster and improves the accuracy of the model. This also aims to remove unnecessary features that are of no importance for the current scenario. For example, when predicting flight delays, the current weather data is important, but the weather data 3 months ago is redundant. Selecting which features to use for the model is an art of data science. Some of the features that can be used in a machine learning model are as depicted in figure 4 below

- $Maximum$: The data point that poses the maximum mathematical value in the given data set is considered as the maximum value.
- $Minimum$: The value in a dataset that is the least in the given scenario is counted as the minimum value
- *Average*: The arithmetic mean of all the mathematical values of different data points is the average.
- *Standard Deviation*: The amount by which the mathematical value of a data point deviates from the collective mean or average is called standard deviation.
- *Skewness* refers to the distortion or deviation from symmetry when all the data points are presented in a graph. For example, consider a bell curve. If the graph is leaning or lengthened towards the left it is called to have negative skewness. If it is leaning towards the right, it has positive skewness. The default behavior of data is to be symmetric. The distortion tells if the data is improperly distributed.
- *Kurtosis* is a measure of whether the distribution of data points in a graph is heavy-tailed or lightly tailed. Datasets with high Kurtosis have a heavy tail while low kurtosis means light tails. The kurtosis for a standard normal distribution curve is equal to 3.

No.	Name	Equation	No.	Name	Equation
1	Maximum	$s_1 = max(x_i)$	8	Mean square	$s_8 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$
2	Minimum	$s_2 = min(x_i)$	9	Root mean square	$s_9 = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
3	Median	$s_3 = \begin{cases} x_{(N+1)/2}, N\ is\ odd \\ \frac{x_{N/2} + x_{(N+1)/2}}{2}, N\ is\ even \end{cases}$	10	Skewness	$s_{10} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - s_5)^3}{s_7^3}$
4	Peak-to-peak	$s_4 = max\lvert x_i \rvert - min\lvert x_i \rvert$	11	Kurtosis	$s_{11} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - s_5)^4}{s_7^4}$
5	Mean	$s_5 = \frac{1}{N}\sum_{i=1}^{N} x_i$	12	Skewness factor	$s_{12} = \frac{1}{N}\sum_{i=1}^{N} x_i^3 \Big/ \left(\sqrt{s_8}\right)^3$
6	Variance	$s_6 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2$	13	Kurtosis factor	$s_{13} = \frac{1}{N}\sum_{i=1}^{N} x_i^4 \Big/ \left(\sqrt{s_8}\right)^4$
7	Standard deviation	$s_7 = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2}$	14	Mean frequency	$F_1 = \frac{1}{N}\sum_{j=1}^{N} f_j$

Figure 4: features and their formulas [ZYW19]

MC PDM Concept To prevent unexpected failures, a different classification method is used. Instead of only labeling the current iteration, the algorithm tags the last m iterations as F and NF. This allows better analysis and provides to opt for a more detailed maintenance recommendation as to the failure horizon m increases. This helps to reduce the skewness in the dataset we saw in the previous method. So now the statistics are Nm samples for F and n-Nm samples for the class NF. This can be repeated with different values of m, let's say k times. This is the base of the MC PdM approach where k different classifiers run and each iteration deals with a different classification problem. So, it can give a varied result for maintenance outcomes means, varied values of FuB and FuL.

The multiple classifier approach as depicted by algorithm 1 deals with running k different classifiers in parallel. The labels associated with the jth classifier in the ith iteration of the maintenance cycle of total length ni, gives output as F or NF. The performance metric of the jth classifier is represented by FuB(j) and FuL(j). The k classifiers running in parallel are determined based on a decision-making process accessing the costs associated with the operation. If CuB(t) and CuL(t) are the costs at time t, the algorithm suggests a corrective action if the jth classifier outputs as an F class.

Algorithm 1. MC PdM Training

Data: $X, Y, k, q_1, q_2, Q_1, Q_2, \{m^{(j)}\}_{j=1}^{k}$
Result: Classifiers $\{f^{(j)}\}_{j=1}^{k}$ and PdM performances
1. Let $\rho_{\mathrm{UB}} = [\cdot]$ and $\rho_{\mathrm{UL}} = [\cdot]$ (empty vectors)
for $j = 1 to k$ **do**
  2. Compute $Y^{(j)}$ as in Eq. (5) **for** $i = 1 to Q_1$ **do**
    3. Randomly split the maintenance cycles between training and validation samples, keeping the ratio q_1
    4. Compute the MCR of the classifier with different hyper-parameters
  5. Chose the hyper-parameters based on the averaged MCR over the Q_1 simulations
  6. Compute $f^{(j)}$ with the selected hyper-parameters
  for $i = 1 to Q_2$ **do**
    7. Randomly split the maintenance cycles between training and validation, keeping the ratio q_2
    8. Compute $f^{(j)}$ with the selected hyper-parameters
    9. Compute ρ_{UB} and ρ_{UL} of $f^{(j)}$
  10. Compute ρ^{*}_{UB} and ρ^{*}_{UL} as averaged over the Q_2 simulations
  11. $\rho_{\mathrm{UB}} = [\rho_{\mathrm{UB}}; \rho^{*}_{\mathrm{UB}}]$ and $\rho_{\mathrm{UL}} = [\rho_{\mathrm{UL}}; \rho^{*}_{\mathrm{UL}}]$

Figure 5: MC PdM trining algorithm. [ZYW19]

$$y_t^{(j)} = \begin{cases} \mathrm{NF}, & \text{if } t \leq n_i - m^{(j)} \\ \mathrm{F}, & \text{otherwise} \end{cases}$$

The idea is to evaluate the performance of the MC-PDM algorithm in each iteration using random subsampling validation, known as Monte Carlo cross-validation (MCCV).

$Cross - Validation$ is an important part of the generalization of a model. It divides the dataset into multiple fragments and uses one validate while others test. In this way, it validates against the same dataset it came from. Two main types of cross-validations are k fold and Monte Carlo cross-validations

$K - fold\ cross\ validation$: In this approach, a dataset is divided into k equal-sized mutually exclusive chunks of data. the process is run k times, and in each iteration, the kth set is used to test while the remaining k-1 sets are used to train the model. In this process, each set is used once for testing.

$Monte\ Carlo\ Cross\ Validation$ (MCCV): This approach is a little different in terms that the selection of test data is random, without replacement. At random let's say 10% of data is selected to act as test data and the remaining is used for training. This process is repeated multiple times to reach a conclusive state. In terms of performance, K fold test each data set exactly once, so it is more balanced but sometimes, it leads to obvious conclusions, and hence, the Monte Carlo method provides some randomness and derive new insights from the data set. However, it may be the case that all the data is not tested, and this may lead to a biased result.

In the approach of multiple classifiers, the MCCV method is used. Q simulations are performed by splitting the dataset of N maintenance cycles into training dataset and validation dataset

N(TR) = Nq
N(VL) = N(1-q) such that $0 < q < 1$.

It was found that MCCV is consistently pessimistic in predictions of test data compared to full data cross-validation. FuB and FuL are provided by historical data of the system and the costs associated with CuB and CuL are defined by the user and allowed to be changed at each evaluation of the PdM module. Costs are set such that a breakdown during intense production sessions is much more costly than general production days. When an unexpected failure occurs during the operation of a PdM module, it signifies that the maintenance cycle is complete, and the valuable dataset is available for further updating the MC PdM for enhancing its performance.

5 MC PdM Implementation Details

In the k classifier approach, the performance and accuracy of the MC PdM methodology increases with the increasing number of classifiers k, as each classifier provides further information about the health metrics of the process. But k is not the degree of freedom since it has constraints. It is limited by computational power and space capabilities. So, the space and time complexities must be considered to select a balanced value for k.

Failure horizon Selection The k classifiers are selected in such a way that it partitions the dataset into equal chunks and provides a failure horizon for each iteration in such a way that the horizons are equally spaced. The k horizons used can be described as in fig below, Where the first component is the closest integral value then added with 1. M1 is a positive value. M1>0 specifies the maximum failure horizon[SSP+14]. M1 is selected based on the length of the Maintenance cycle and the training dataset. Another

$$m^{(j)} = \left\lfloor \frac{(j-1)(M_1-1)}{k-1} \right\rceil + 1, \quad \text{for } j = 1, \ldots, k$$

approach for selecting the failure horizon is considering the percentage of maintenance cycle in place of a fixed number of iterations. It limits the value M between 0 and 1 so we get a classification between F and NF as where mj is defined as

$$y_t^{(j)} = \begin{cases} \text{NF}, & \text{if } t \leq \lfloor n_i \left(1 - m^{(j)}\right) \rceil \\ F, & \text{otherwise} \end{cases}$$

$$m^{(j)} = \frac{(j-1)M_2}{k-1}, \quad \text{for } j = 1, \ldots, k$$

Classification Algorithms for MC PdM The MC PdM approach does not restrict the classification algorithm to be used. The model is checked with the two most popular classification algorithms, SVM and CNN.

$Support\ Vector\ Machines$ (SVM)are usually the most popular algorithm for classification because of their high accuracy and usefulness for even nonlinear problems. Considering the two classes F and NF in this case, the cases can be linearly separated as faulty and not faulty. Supposing the training dataset S is assigned with values ranging from -1 to +1 such that the output value is -1 for NF and 1 for F. A *hyperplane* is defined such that it separates the two classes with the highest margin.

In real scenarios, classes are not always linearly separable means the two categories have overlapping. In such scenarios, with the liberty of wrong classification of some data points to the other class, SVMs can still be used. So, SVMs are generally combined with Kernel Methods that further enhance the classification of nonlinear data. The computational cost for training with a nonlinear data set is higher and has a complexity of *O(n$\hat{2}$)* to *O(n$\hat{3}$)* based on the algorithm used.

$k\ nearest\ neighbors$(k-NN) is another most used classification algorithm which is also probably the simplest one. In this approach, the distance between two points in the input space is calculated and the neighborhood is determined based on this distance. This approach just needs the computation of distance, for example, the *Euclidian distance* between the points. The only variable parameter is the value k which is defined by the user. Larger values of k can reduce the noise and its effects during classification but may make the boundaries less distinct while a lesser value of k will result in multiple neighborhoods. To make a balance between this, the parameter k needs to be optimized. The computational cost of the k-NN method is relatively very less than SVM and has a complexity of *O(log n)*

6 Use case- MC-PdM in the Semiconductor manufacturing industry

Susto et. Al. tested the efficacy of the MC PdM methodology for replacing tungsten filaments used in ion implantation [] which is one of the most crucial tasks in semiconductor manufacturing fabrication. By injecting doping atoms, the Ion implantation process is used to alter the electrical properties of wafers. This is considered a very tough task and is often described as the “bottleneck” in the production cycle, primarily because of the very high cost of the tool that makes it of utmost importance for the throughput. The components of the implanter tool are as depicted in figure. The filament is part of the source section.

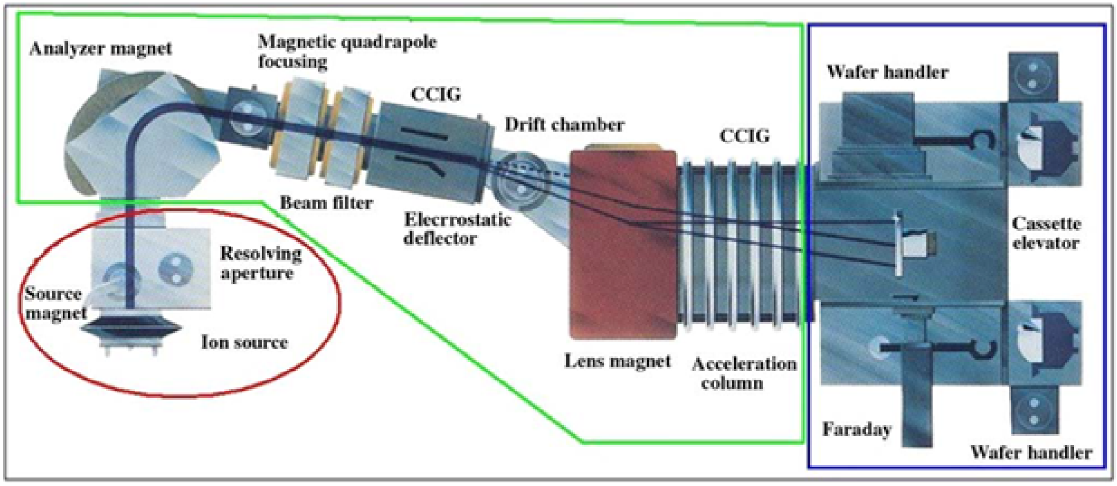

Figure 6: Generic beam ion implementer tool. Red circle: source, green area- beamline area, blue region – end station [SSP+14]

Maintenance Problems During the process, the filament is subjected to heat that results in the electrons boiloff from the heated filament. These electrons are then accelerated via the beamline area and targeted on the wafers located at the end station. Due to this, the tungsten filament needs to be replaced frequently. As the filament is the most important part, while the filament is being replaced, the complete system is down for approximately 3 hours, which demands this process to be highly efficient and cost-effective.

The factors that govern filament health varies from pressure, voltage, current, etc. any fluctuations in these parameters reduce the lifetime of filaments drastically. Other parameters such as cleaning, installation, and degasification also play vital roles in the health and duration of these filaments. So, the requirement was to optimize this process by using data-driven methodologies for predictive maintenance so that the average run time of filaments can be increased, and the operational cost associated with the maintenance can be reduced.

$DataSets for Modelling$ - The data available for modeling consisted of 33 maintenance cycles where the filaments were maintained based on the R2F policy. Every maintenance cycle comprises of the data between the period of installation of a new filament to the point when the filament breaks and needs a replacement for a total of 3671 batches. During the n runs of the maintenance cycle, 31 variables were noted down in a time-series evolution. Some of these variables present a non-uniform sampling rate between different observations. For the model to construct a design matrix that can be used as input for the classifiers, first the features need to be extracted. Using statistical methods, features are extracted that fit the best for prediction. Out of these 31 features, 6 features were further extracted from the time series- maximum, minimum, average, variance, skewness, and kurtosis. After discarding the constants and unnecessary variables, a total of 125 input variables were retained in the dataset.

SVM and k-NN approaches for MC PdM Susto et. al. used both the SVM approach and k-NN approach on the given dataset using the MC- PDM methodology and denote them as MC PDM-svm and MC PDM-knn. In the case of the SVM to implement the nonlinear classification boundaries in the linear framework, they used the Radial Bias Function (RBF) kernels and counted linear SVM implementation.

Simulated PvM policy- Generally, PvM policies are based on the mean and median values of maintenance cycle lengths and an optimum threshold. Once the threshold value is reached, it triggers a Preventive maintenance (PvM) cycle. The PvM policy based on mean is formulated as per algorithm 2

Algorithm 2. PvM module

Data: Y, c_{UB}, c_{UL}
Result: Maintenance Rule (defined by $\tau_\mu *$), ρ^μ_{UB}, ρ^μ_{UL} and $J^\mu(T^\mu)$
1. Compute μ, the mean number of process iterations in a maintenance cycle
2. Define a set of threshold values $T^\mu \in \mathbb{R}^d$
3. Let $\rho^\mu_{UB} = [\,\cdot\,]$ and $\rho^\mu_{UL} = [\,\cdot\,]$ (empty vectors)
for $j = 1 to N$ **do**
    4. Compute ρ^μ_{UB} and ρ^μ_{UL} for all entries in T^μ
    5. $\rho^\mu_{UB} = [\rho^\mu_{UB}; \rho^\mu_{UB}]$ and $\rho^\mu_{UL} = [\rho^\mu_{UL}; \rho^\mu_{UL}]$
6. $\rho^\mu_{UB} = \text{Mean}(\rho^\mu_{UB})$ and $\rho^\mu_{UL} = \text{Mean}(\rho^\mu_{UL})$
7. Compute $J^\mu(\tau^\mu)$ for all entries in T^μ: $J^\mu(T^\mu)$
8. Let $\tau^\mu * = \min_{\tau^\mu \in T^\mu} J^\mu(\tau^\mu)$

Figure 7: PvM module using mean [SSP+14]

SVM classification distance based PdM system: This is based on the suggested MC PdM approach, and it extensively uses the distance from a computed SVM decision boundary [SSP+14]. This is to define a metric that shows the distance of an observed point from a faulty situation. The assumption beneath is that with the increasing stress on the machine, the data points observed are closer to the faulty boundary, and hence, the distance between the two SVM classes decreases. This method suggests that the machine needs a corrective action when the function returns a value lesser than the threshold distance.

Settings and Results : Susto et al. implemented the MC PdM algorithm using the parameters k=10 and M1=85. They selected these values by carefully analyzing the input set [SSP+14]. With the value M1=85, they did not observe any unexpected breaks and with k=10 the model gets enough diversity in the performance in terms of effective FuL and FuB. They found that before the application of the MC PdM approach, the hyperparameters of the MC classifiers need to be tuned using cross-validation techniques. Also, in the case of distance-based SVM clustering techniques. In the case of k-NN, the parameter k which depicts the size of the neighborhood was tuned. They observed that the optimal value of k in MC PdM is equal to 1 which represents the high dimensionality of the problem. They plotted the graphs about the remaining lifetime using different methods and obtained the below results-

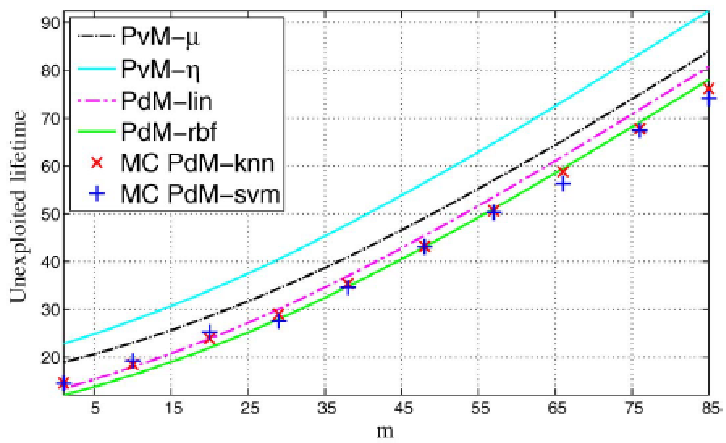

Figure 8: Average remaining lifetime over Monte Carlo simulations with various maintenance processes [SSP+14]

It was observed that the PdM and MC PdM approaches outperform the PvM approach. This means the predictive maintenance model is much better than the preventive model.

- In terms of FuL, the MC PdM achieves similar results to the PdM approaches.
- PdM-rbf method is a little superior for low values of m and MCPdM is superior for higher values of m.
- In terms of PuB, the MC PdM-svm method consistently shows better results in prediction.
- When the fault horizon m increases, then precision decreases which means more false negatives are reported which means more unexploited lifetime, and recall value

increases which means fewer false negatives meaning fewer unexpected breaks.

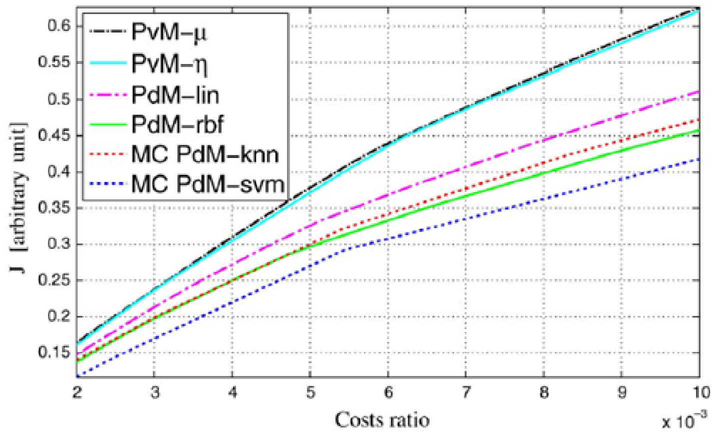

Figure 9: cost ratio of various Pdm techniques [SSP+14]

When comparing the costs factors CuB and CuL, graph 2 depicts the ratio of CuL/CuB averaged over Monte Carlo simulations. J is the minimum total cost achieved with each process and is computed based on average results and presented on the y axis. With the help of process experts, the value of CuB and CuL is computed [SSP+14], and the ratio is presented on the x-axis. The results demonstrate that the MC PdM-svm approach is consistently better than other approaches. The MC PdM-Svm approach achieves the lowest operational cost while also being adaptable to different values of FuB and FuL. MC PdM-svm gave better performance than the MC PdM-knn approach but is much more time-consuming. While computing these values, an iteration with a broken filament is considered as positive and an uninterrupted process is considered a negative condition. Overall, it is observed that the proposed MC PdM approach has a significantly higher performance than other approaches.

7 Limitations and further works

The presented method is a Multiple classifier Predictive Maintenance system for integral type faults. The MC PdM works in parallel to fully exploit the resources and knowledge of the tools at each iteration of the process to help in decision making. But this approach is limited to the availability of datasets that can be labeled. This means it is only applicable to Supervised machine learning algorithms and cannot be used in high dimensional data where the outcome is not known. For such cases, unsupervised machine learning approaches need to be used. Such as Artificial neural networks and Deep Neural networks. Neural network algorithms can train themselves on unlabeled data and formulate their model which gets better over each iteration.

Other ways of predictive maintenance are based on Neural networks and deep machine learning approaches. Algorithms that use decision tree[KB20] and random forests are capable of processing high dimensional data from complex environments to extract useful features that can be used for modelling.

The power of predictive maintenance is now harnessed in every industry and further research areas include e-maintenance, remote maintenance and management, tele-maintenance, and IoT [Sel17]. E-Maintenance is a form of PdM system that provides a solution of predictive maintenance over the internet. Similarly, remote maintenance enables to execution of PdM in potentially hazardous and isolated areas such as aircraft maintenance. Another example is robots being monitored remotely to make a timely decision of maintenance and replacement for reducing the overall higher cost of fixing after breakage or premature maintenance. Like remote maintenance is tele-maintenance mostly observed in corporative works. The maintenance team can monitor and diagnose remotely by taking control and performing repairs from a remote location. PdM can be extended to the Internet of things and is already being used in many complex systems of systems such as automated vehicles and other tools. The monitors collect data based on various sensors and compute the remaining available lifetime and suggest performing certain preventive measures.

The future of Predictive Maintenance is *Prescriptive Maintenance*. Prescriptive maintenance is a system that monitors the process and with zero to minimal human intervention, it can depict the status of the machine. It suggests actions be performed and sometimes is even capable of acting on those suggestions thereby making the system resilient to faults. It can be termed as a self-healing system.

8 Conclusion

Conditions of machines deteriorate over prolonged usage and those need repairs. While breakage of a system involves a costly affair of downtime and damages the brand value and quality of products. So, it is always better to avoid failures. Preventive Maintenance is planning scheduled maintenance over a specific period. While this helps to avoid system failures, it does involve downtime and often leads to unexploited resources and wastage of available runtime of components. The better way is to have a system that can accurately tell when to perform maintenance on exactly which component so that the complete machine failure is avoided at the same time, the cost of maintenance is also reduced. Various machine learning methodologies can be used for this.

In manufacturing industries, mostly data is available in form of Maintenance cycles and hence it classifies under the supervised learning methods. The available datasets are used to train a model that can classify the state of the machine as faulty or not faulty. When this model is presented to new values, it can predict the probability of a failure by classifying it in either of the two classes. A better approach as observed by Susto et. al. is to use multiple classifiers while using supervised learning methodologies. It can be combined with Support vector machines, k-NN methods, and RBF methods to obtain better results in different scenarios. It allows to dynamically change the policies associated with maintenance management and helps the process engineers to adopt a balance between process optimization and operating cost optimization [SSP+14]. Because of various classifiers, the proposed approach can be used as a health factor indicator in PdM approaches. K-classifier allows a better understanding of the maintenance decisions [SSP+14] and enhances the process understanding. The performance of a classifier in the kth iteration in MC PdM approach is strongly affected by the choice of fault horizon. As demonstrated in a semiconductor manufacturing industry where replacing filaments was a very costly affair. MC PdM approach not only helped in reducing the system breakdowns but also helped to minimize the maintenance cost by leveraging on the available life of components based on the data-driven approach. Further with the advancement of AI technologies, machines can be trained to do self evaluation and self healing. This will drastically reduce the operating and maintenance cost and increase run time of machines.

References

[BR10] Shane Butler and John Ringwood. Particle filters for remaining useful life estimation of abatement equipment used in semiconductor manufacturing. In *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 436–441. IEEE, 2010.

[DP17] Jim Daily and Jeff Peterson. Predictive maintenance: How big data analysis can improve maintenance. In *Supply chain integration challenges in commercial aerospace*, pages 267–278. Springer, 2017.

[KB20] Shashidhar Kaparthi and Daniel Bumblauskas. Designing predictive maintenance systems using decision tree-based machine learning techniques. *International Journal of Quality & Reliability Management*, 2020.

[Mob02] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[MPR18] Omkar Motaghare, Anju S Pillai, and KI Ramachandran. Predictive maintenance architecture. In *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–4. IEEE, 2018.

[SBDL12] Gian Antonio Susto, Alessandro Beghi, and Cristina De Luca. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Transactions on Semiconductor Manufacturing*, 25(4):638–649, 2012.

[Sel17] Sule Selcuk. Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9):1670–1679, 2017.

[SG04] Cornelius Scheffer and Paresh Girdhar. *Practical machinery vibration analysis and predictive maintenance*. Elsevier, 2004.

[SSP+14] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE transactions on industrial informatics*, 11(3):812–820, 2014.

[ZYW19] Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019.