Industrial AI Challenge
Final Report

Team 4

1 Outline

ITG Tecnologie is involved in an innovation process that has a core step the introduction of a maintenance AI solution for its product. The Challenge lead having a first team of experts in AI techniques in order to start analyzing the problem. The results of the work, confirming the business choice made, lead to:

- Help directing data acquisition and improve AI knowledge of the enterprise
- A first problem analysis, feasibility study and solutions researches involving possible improvements and techniques to be experimented (symbolized by this report and in particular synthesized in the bibliography attached)
- An agent implementation of a system that satisfies objectives such as
  - Prevention of faults
  - Estimating future degradation
  - Reduction of intervention cost

  with capability of reflecting real operational situation (applicable without modification with minimum changes) but mainly, given the intention displayed by company at the start of the challenge, to be used as reliable base for future research, development and experimentation
- The model proposed is strictly coupled with a comprehensive suggested architecture that was followed to be applied in a real context of a new application of industrial predictive maintenance that could also be reemployed in other similar projects by the company and to lead experiments and innovation

2 The team

The key person involved in the challenge are the following:

- 5 students form the University of Trento as solvers
- 2 mentors, as reference for a new startup company specialized in processing visual data and providing additional video services and solutions for Industry 4.0
- 4 people for the company ITG Tecnologie

The team members representatives are the following. The solvers:

- Davide Cappellaro, Data Engineer, master student in Data Science, bachelor degree in Interfacce e tecnologie della comunicazione
- Dimitri Vinci, master student in Artificial Intelligence System, bachelor degree in Informatics
- Giorgia Rossato, master student in Mathematics and Statistics for Life and Social Sciences
- Jhonny Hueller, master student in Artificial Intelligence System, bachelor degree in Informatics
- Marco Di Francesco, Junior Data Scientist in FBK, bachelor student in Computer Science

The mentors:

- Candida Palmera, head of R&D of TR2 company
- Mauro Venanzi, CEO of TR2 Company

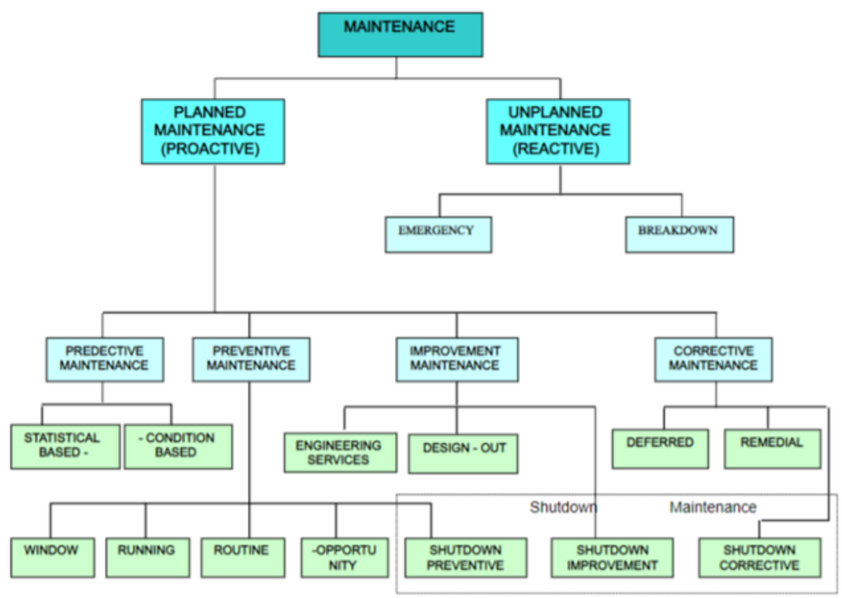
From ITG Tecnologie:

- Tancredi Lolli, CEO
- Lorenzo Beatrici, head of the Technical dep.
- Christian Girardi, head of the Operation
- Lorenzo Giordanino, external consultant

3 Description of the Challenge

ITG Tecnologie is a company based in Polo Meccatronica placed in Rovereto and active in the field of innovative solutions for lifting equipment. The product range comprises magnets motor for lifting purposes that is the reference for the entire work .ITG Tecnologie would like to develop a predictive maintenance and monitoring system. In this market segment the motor is normally installed in the new lift and then putted in operation where there is not the possibility to check constantly the product to avoid all problems. When a critical issue occurs, the lift is stopped due to the fault of the motor. After this, the lifting company starts to organize the maintenance work to solve the problem and restart the lift which cause a really important lack of service.

Typically the machine has to be changed after a fault increasing also the stop time of the plant (Corrective maintenance approach, repairing after failure or run to failure approach, and Preventive maintenance, repair at predetermined schedule). It must be underlined that ITG Tecnologie is not responsible and do not take in account personally the cost of changing the engine but it is all a user necessity.



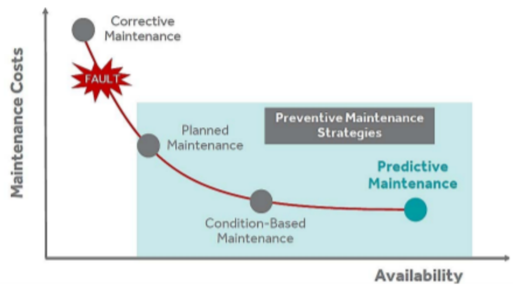
The enterprise wanted to

- Study and implement as commercial solution a monitoring tool of engines giving the opportunity to the final client to monitor the characteristic data of the machine
- Develop a proprietary platform for data acquisition that use the monitoring tool (approach suggested also by [120])

This is called predictive/proactive maintenance [22][178][28][1][112]
Given typical maintenance objectives

- increase machine availability
- ensure safe operation [1]

To contextualize, typical costs of different maintenance strategies are depicted here



Predictive maintenance reduces maintenance cost and increases availability. This is done anticipating event criticality

[112]Basically it brakes the tradeoff between exploitation of machine and anticipating fault [160]

For ITG

- generating more added value to all product life cycle creating a usage smooth process
- introducing an Iot and AI solution in the lift market (it is worth mentioning that for predictive maintenance use an external service is the preferred approach for companies)

For the customer of the elevator manufacturer the value added of this system could be

- obtain maximum quality need flexibility in a production process. In this context manage of assets like the elevator itself is fundamental. A wait to break mentality is not sustainable (maintenance cost for enterprise account for 60% of total production costs. Predictive maintenance increases uptime of 20%, reduce costs of 10% and maintenance planning time from 20 to 50 %) [111]
- For elevators in production process all phases of production process are affected by a fault
- In critical situation like for hospitals a fault in elevator means a life risk for a patient, and preventing it for a hospital is of primary importance

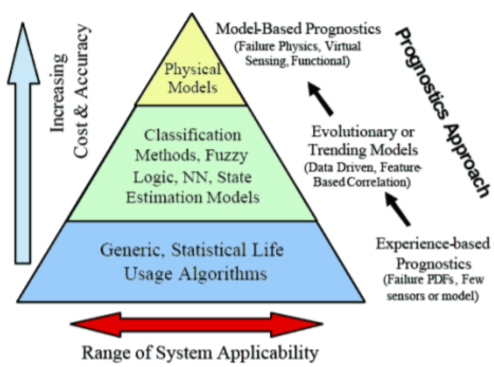
- Prevent faults for elevator in a building means increasing comfort

At both levels, it helps financial accounting to increase accuracy of the budget plan [24]

Typically predictive maintenance models are classified in

- physical model of machine, have a limitations in flexibility and feasibility [181][99]
- data based methodology, a typical machine learning solution [182]
- simple statistical algorithms, typical models used if data usage capability is low (for example setting a predefined threshold of time to do inspections via SCADA systems[111])

Given that ITG wants to monitor different engines and prospects a continuous flux of data, a flexible approach like data based model will be most useful. In fact a physical model lead to superior performances only if updated regularly and completely adherent to the subject monitored.



[6] But the core advantage of data based model is the possibility to constantly improve predictive maintenance via data in an autonomous way [1]

In our context the enterprise wanted to have a predictive maintenance solution that will be the basis of future development for a commercial product. In particular intentions exposed in candidature to AI challenge were formulated as ”Analysis of measures related to the product with aim of

- Predictive maintenance
- Resource expenditure reduction for maintenance

With a particular attention of variation of characteristic parameter of the engine in the time.

Last sentence suggests as we are realizing a condition-based predictive maintenance as defined by [98] (in fact typically more useful than a policy based maintenance [127][1] [131]) And in particular we are at the level Health assessment and prognosis of an health management approach[6] Given the fact that time of faults of components is the principal aim of enterprise final product we are in the context of Reliability based maintenance and following that disciple we decided to focus only on a particular important equipment and take in account limitations in resources. Starting the challenge an assess of means and actual enterprises desires (in particular determination of most important failure modes as suggested by ISO 13381-1) shows that for our scope the needs was related in particular with the possibility of using monitoring data to predict existence of the fault and if it was possible the origin of the fault itself related to

- Bearings
- Magnets
- lack of grounding connection from the control panel

to produce a basis for future development.

It was proposed later in the project by the team to give the possibility for the enterprises to include in the "basic product" the ability to

- Predict remaining useful file of magnets, given the fact that it was possible to collect a degrading like information about that component

Include a way to take into account maintenance cost in predicting way to intervene. Servitization of the product implies the possible use of this tool by users in a maintenance framework, leading to have to take in account not only faults but also costs of the maintenance[1], help management in reducing budgets, minizing roa, reduce capital budgets via intelligent maintenance scheduling and help gut and feel decision making of enterprise users of the product [24].

4 Description of the industrial process from which the data stem from

Predictive maintenance needs a high quantity of data regarding the specific domain. In our case although there was no such data promptly available, putting us in a less than ideal scenario. ITG Tecnologie built a testing environment collected data on an engine creating artificial damage to measure characteristics parameter during different situation of motor operability.

Figure 2: Test bench with the machine under test

Figure 3: Rear view of the motor with the Asystom sensor (red circle)

Figure 1: Typical motor used in lifting application. The main parts are the traction sheave (yellow component), the motor (orange part) and the brake.

Considering that the market volumes are concentrated to low and medium loads, a motor range with payloads form 225 to 320 kg (typical homelift, 4 people), was taken in to consideration for the test. All the data was collected with AsystomSentinel, an intelligent multi-sensor device able to autonomously analyze sensor signals and transmit them on a private cloud server (asystom platform and related dashboard), making data available to the ITG and the team in real time. The system was directly connected to motor as shown in figures 2 and 3. The machine is a three-phase AC synchronous motor with permanent magnets, with a nominal supply voltage, frequency and velocity of 380 Volts, 20 Hz and 75 rpm respectively.

The device is able to analyze several types of signals from the device it's attached to: vibration, acoustic and temperature signals.

In our experiments we used a time-triggered measurement process: the sensor measured the motor once at each time interval, which in our case it has been set to its lower bound of 1 minute in a steady state response under different circumstances. This type of tests were similar approach for example to [178] and was determined to obtain a fairly good dataset for preliminary analysis.

Difference from this case and our case is that for time constraint limitations we could not afford to lead machine to naturally fault but we have to simulate the fault ourself by damaging each component of the machine externally before collecting data. The data

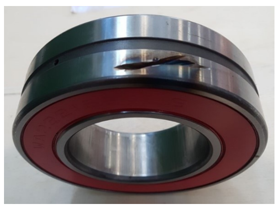


Figure 4: Bearing with radial defect

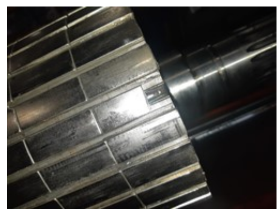


Figure 5: Rotor without 30% of the magnet

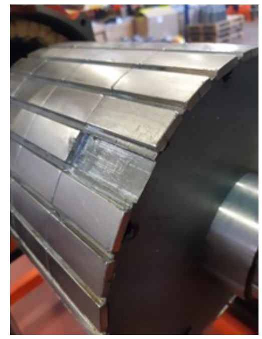


Figure 6: Rotor without 1 magnet

collection followed the cases:

- Experiments on a new motor
- Experiments with degraded bearings
- Experiments with partially demagnetized motor removing from one to two entire segment of permanent magnets.
- Experiments with a faulty power supply

5 Dataset

The data registered by the sensors is: vibration in 3 axes, ultrasound, ambient temperature, surface temperature, and motor current. Stating in a mathematical approach our raw data for an instant time i, $X_1(i), \ldots, X_T(i)$ is a set of T sensors measurements. Each $X_t(i)$ is a set of measures. [164] This is what is technically called field level asset which offered typically too much parameters to be measured [24] and lead to select a subset to be monitored.

Vibration spectrogram was taken as reference by suggestion by ITG engeneers in our team for predictive analysis to also predict defects due to electrical condition.

Figure 7: Asystom platform with some data collected and related graphical view

The table shows all the data provided by AsystomSentinel sensors. Note that not all these labels are relevant.

Signal label	Description
learning_step	Learning iteration (used during the setup of the sensor)
mileage	Masks machine on/off state, learning status and learning session count
$s_{00}, \ldots, s_{09}$	10 vibration frequency bands linearly spaced between 0 and 2 kHz, measured in dB, with 0dB representing the sensor full scale equivalent to 16g.
$s_{10}, \ldots, s_{19}$	10 sound frequency bands, linearly spaced between 0 and 80 kHz, measured in dB, with 0dB representing the sensor full scale equivalent to 120dB SPL.
sonic_rmslog	RMS value for ultrasonic data in the ultrasonic band 30-80 kHz, measured in dB.
vib_x_vel vib$_{x_{\text{vel}}}$	X-Axis RMS level of vibration velocity, measured in mm/s
vib_x_acc	X-Axis RMS acceleration value, measured in g
vib_x_peak	X-Axis absolute peak acceleration level, measured in g
vib_x_kurt	X-Axis kurtosis
vib_x_root	X-Axis root frequency, expressed in RPM
vib_x_f1	X-Axis RMS velocity around root frequency, measured in mm/s
vib_x_f2	X-Axis RMS velocity around 1st harmonic, measured in mm/s
vib_x_f3	X-Axis RMS velocity around 2nd harmonic, measured in mm/s
vib_y_[...]	Y-Axis parameter of all the X-axis ones
vib_z_[...]	Z-Axis parameter of all the X-axis ones
temp	surface temperature
sonic_custom	reserved by creator for future scopes
vibra_custom	reserved by creator for future scopes

6 Description of the activities

For the Miro Board see the following url: `https://miro.com/app/board/o9J_lwDaEU0=/`.

6.1 Starting situation of challenge

Starting from the historical expertise of ITG Tecnologie the suggestion for the teams that was accepted by company was to simulate the typical and most frequent type of faults on the test bench.

6.2 Processes involved in the challenge

Prioritize problem areas, analyse how affect profitability, evaluate starting analytics quotient of the enterprise [24] were immediate goals of team. It was suddenly evident that ITG wanted to pass from a novice data usage profile to builder data usage profile learn how produce a viable results in the task they wanted to accomplish with AI. So it was necessary to manage for the team two processes:

- Data gathering process: at the beginning of the challenge (24th September 2021) the enterprise did not have data to use for a predictive model. Given the maturity of data usage of enterprise was necessary to plan and follow all together data collection. For data collection we tried following an agile approach. Established a possible schedule, more important data were collected at first, make able to reschedule and change subject of data collection during the challenge itself if it was necessary to pursue objectives.

- Solution development process, the solution process adopted regarding three objective/key activities.

  - Documentation: it must be underlined that no member of team was expert or have opportunity to cope previously with predictive maintenance and business process improvement
  - Exploration (engage with company to learn about their desires, constraints, acceptable solutions), the company had a very loose aim at the start that must be defined and was updated in an agile fashion during the process(when an objective was reached another was added)
  - Code development (that includes all phases in development of a machine learning model) note that some aspects like data analysis were repeated all time because of continual arrival of data

Summary of activities
Week 1

- Documentation: analysis of pursued business, brief analysis of possible competitors (meaning enterprises in the elevator market which offer predictive maintenance services), analysis of structure and typical defects of the type of engine involved

- Exploration: setting of prediction objectives with enterprises and base results (capability of identify a fault in the elevator), discussion of intended pursued business model

- Data gathering: installation and setting up Asystom system

Weeks 2-3-4-5

- Documentation: about previous solution founded in scientific literatures for the entire problem of predictive maintenance, construction of intended architecture

- Exploration: obtaining a prospect of typical faults with supposed probabilities (in particular determination of most important failure modes as suggested by ISO 13381-1) to learn where to focus and schedule better data gathering, validation of proposed architecture and determination of constraints for a solution, obtaining possible extra objectives could be added if the time is sufficient

- Code development: reverse engineering platform to agile collect data when needed in a format able to be used by our data science instrument, data analysis (number of data, type of data, class separation of data, noise of the data)
- Data gathering: starting preliminary testing with the motor without defects. The first testing campaign was performed with starts and stop cycles in order to simulate the typical running condition of a lift. Considering the time frame of the Asystom sensors (1 measure every 1 minute , not able to statistically intercept acceleration and stopping phase of the motor, without any added value for the analysis) and the time available for the testing campaign, a continuous running cycle was decided to use for collecting data.

Weeks 6-7-8

- Documentation: how reduce cost of maintenance, reliability engeeneering, solution for remaining useful life prediction and cost analysis
- Exploration: validate a risk based approach for cost reduction, validate possible following improvements (of aspects not taken in account)
- Data gathering: Various testing was performed 24h a day with some typical defect simulation. The defects simulated and tested was: bearing defect, magnet losses. In this last particular case the tests were performed first removing just 30% of one magnet, then 1 single magnet.

Week 9

- Documentation: report writing and literature checking (important for following development and research of the product)
- Exploration: final validation of work done by all team
- Code development: check code quality and for eventual bugs, create plots of performances for evaluate architecture from experts
- Data gathering: test performed simulate the typical defect of lack of grounding connection from the control panel parallel to test of removal of 2 magnets on rotor.

7 Formal description of the settings

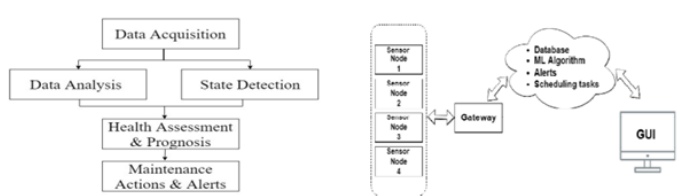


Figure 8: System Architecture (left) and block diagram (right)

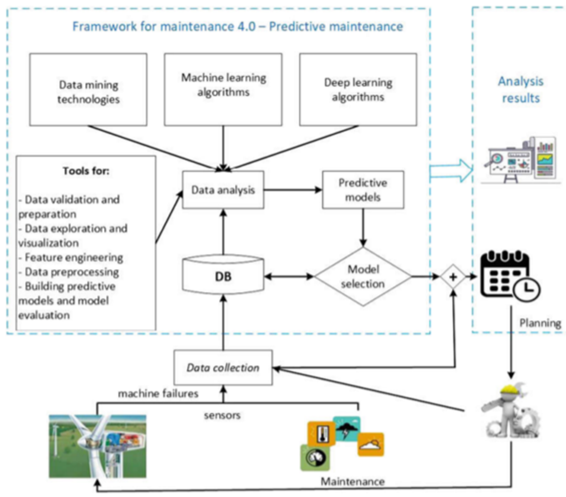


Figure 9

In figure 8[112] and in figure 9[164] is depicted typical predictive AI assisted maintenance model.

That is typically a four stage control loop

- Data collection
- Storing in database
- Data analysis and prediction
- Data presentation [15]

Our solution is located at third stage of this control loop. The assumption that were agreed with ITG in the development were in fact

- Not taking in account database issues
- Not taking in account data presentation
- Not taking in account hardware constraints

For last point in particular

- we suppose a perfect data transmission to a central database
- we suppose a device having hardware constraint capable of doing predictions in acceptable time for planning interventions

Given that, the aim is to learn an unknown degradation function [156] Following the general problem description in [156] in a preventive maintenance model we have

- A set of m devices
- A set of behavioural data of this devices

Training/test data that are collected during a continuous period to predict target. Test and train data are defined over a time horizons that naturally splits training and test data [156]

We should underline that this operative configuration and data characteristics are different from what we had in input. Our data come from same machine with separate faults added. Instead in a real problem formulation and in production we would have as defined (and as also confirmed by ITG)

- multiple machines
- data series that could cover possible multiple problems in a machine
- data fed progressively

We will call this two “setting”

- Actual setting, described by data and label we originally have
- Business setting

In this report we will mainly refer to business setting(unless specified) because it allow us to extend objectives and to provide actual value to enterprise (a simple cassification of faults in actual setting was straightforward with a very simple classifier as showed in performance section).

Given our three objectives in chapter 3 we face different but highly correlated problems, for every one we produced a type of autonomous agent that deals with this problem

- Fault detection and isolation for magnet, bearings, current
- Remaining useful life prediction for magnet
- Optimizing maintenance intervention with a cost model

This corresponded to our work iteration and these agents will be described in the following sections.

7.1 General architecture developed

Facing the need for finding at the same time a coded implementation with limited resources but also a proposal for future development it was decided to connect implementation to a new devised framework to program general prediction task in enterprises. For data analysis tasks is difficult to propose architecture that are suitable for a vast range of data. An history of proposed architecture is [97]) [112] [6] and must necessary take in account

- OSA-CBM of MIMOSA composed by a set of insulated layer

- TATEM architecture, introduce risk inside their prediction intended as sum of factor to decide if travel[6]
- gadha, based on OSA-CBM, latest general architecture for prediction [97]) [112]

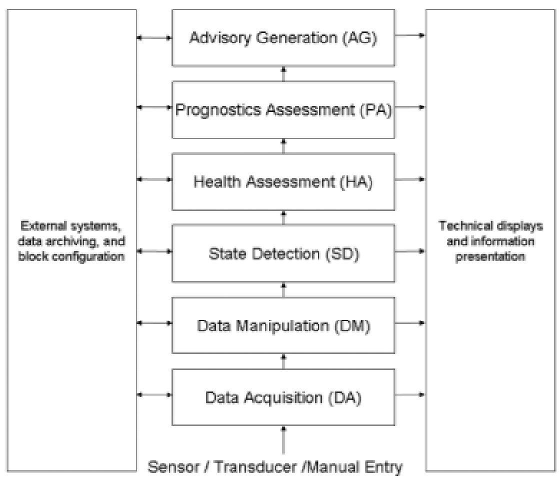


Figure 10: OSA - CMB architecture

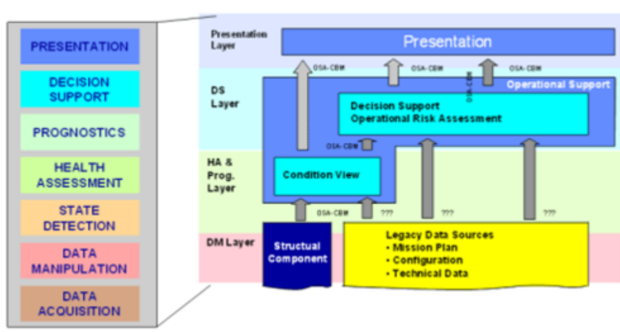


Figure 11

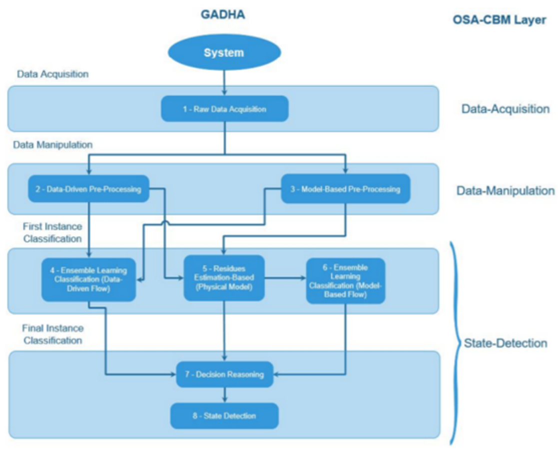


Figure 12

The latter model was taken as reference to lead to our new proposal. Analyzing details of this model blocks we have

- 1: Data acquisition (involve transforming analogical into digital signal and some operation like filtering noise or enhance signal)
- 2, 3: Data manipulation (e.g. LDA dimensionality reduction)
- 4: Classification based on an ensemble model groups, see later
- 5, 6: Operation done to integrate physical and mathematical degradation model in prediction
- 7: Given vector of all classification by all ensemble make a prediction of the current system state. If anfis is used to aggregate prediction vector as suggest 7 and 8 are aggregated

In gadha case there is an anfis approach to a degradation prediction that is in fact an (exotic) example of stacked ensemble at layer fourth of this architecture (even if the author does not underline that for this layer in the paper). [170][107] In the gadha paper was not underlined importance for enterprise of using an ensemble approach on a layer 4 that make able to

- use more prediction from below models to enhance last prediction (dealing with the free lunch theorem problem [164])
- help transitioning an enterprise to AI solution when data are collected gradually

An interesting and fundamental improvement that an enhanced architecture should provide is in a 5th layer (called decision making layer) whose objective will be to weight if intervene based on a cost based machine learning ensemble model (collecting challenge exposed by TATEM) constructed over the prevision of faults of 3rd layer. As underlined in other inspection works [46] if we are dealing with prospected cost (from now on we will talk about classic concept of risk as probability multiplied by impact) using an ensemble model could indirectly weight factor of risk using prediction from other models.

Using a risk based layer also has advantage of connecting a layer directly to costs of enterprise offering help advice for decisions to management and overcome typical reticence of management in adopting AI solutions [1] It's worth noting that our architecture allow for hybrid approach using both physics-model-based/mathematical and data drive approaches (practices that enhances a health monitoring systems prediction [97]) [112]) inside each task simply aggregate them with other classifier prediction.

There are typically three approaches for ensemble that are exposed in [97][112][129] [106][3][130].

We opted for a stacked generalization that allows to obtain

- independent classifier, miximize substitution capabilities of third layer models
- capability to train them in parallel for future performance concern

Finally we underline the importance of generality of the architecture. gadha does not explain how organize layer in a multitask situation as our case. We proposed to replicate 3rd layer architecture for each related task following a decomposition heuristic that

1. split task if provide highly correlated output (if the task could be seen as multi label or multitask probably are suited to stay together)
2. split task if will gain advantages from specific architectural choice (specific classifier used in the layer replication)
3. split task if need different input (this would be last concern to achieve high uniformity of input inside a layer replication)

In order to reach a desired value of third layer replication that would be good in our case for developers but could be balanced between comprehensibility, fast coding and other issues.

In our case our task were divided in third layer as

- multi label prediction of faults regarding bearings and electricity
- remaining useful life estimation

While decision making layer is represented by our maintenance scheduling agent. Our architecture is depicted in figure 13 with the agent implemented in the code.

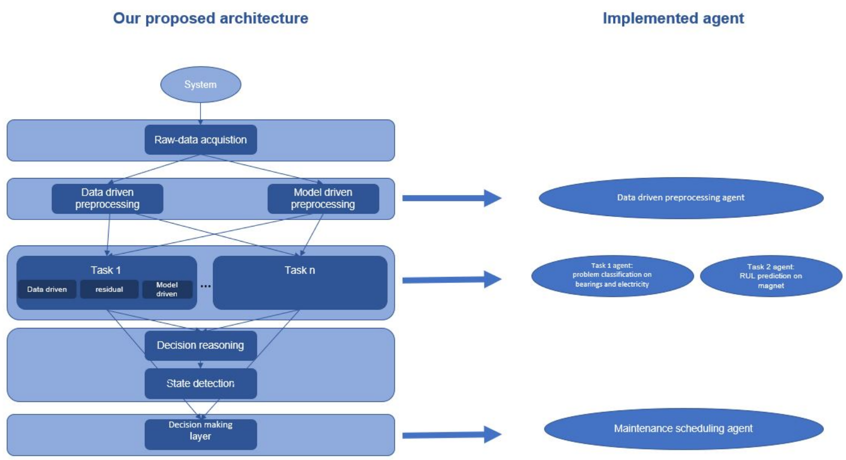


Figure 13

For completeness sake it is important to point out the existence in our code of a data driven preprocessing agent with capability to deal with data acquisition, cleaning, balancing, feature extraction to give data to other agents.

7.2 Fault detection and isolation

Fault detection and isolation the process of identify a fault and its cause.[54]. This work is done by our first task.

Our agent takes as input

1. A set of machines, each ones has a set of measures (and features extracted) and failure labels in different periods (50 periods corresponding to 50 elevator movement, see later) in different time horizon (7 days in our case, see later)

2. given a time horizon t in which the agent is in this moment, train is done on all t-1 horizon data to predict all t label . so each operative agent predict a list of values that represent failures in each single periods, that's mean that each operative agent try predict all failure in all single engine activations for a specific machine in the next time horizons

Given we have 2 fault situation we have to predict for that agent (bearings fault and electricity fault) that could happen at the same time this is a multi label task. The agent is able to ask for data, to select best scalers and hyperparameters, to train on previous data, to do the prediction as described above, to evaluate his performance.

7.3 Predict RUL of magnet

To argue for choices also to enterprise engineer it suitable have to give a mathematical formulation of the problem that could bridge gaps between reliability engineer typical process and our situation.

Consider a single-unit deteriorating system consisting of 1 component or 1 group of associated components. The system failure behaviour might be described by a damage accumulation model. The system state at time t can be summarized by a random aging variable X_t [114]. In absence of reparation it increase from $X_0 = 0$. The system fails when the aging variable is greater than L (or similarly if some attributes exceed a particular level of performance set a priori[22][36][35]. Typically this process is considered in continuous time as a positive increasing process with s-independent stationary increments (jumps[30]) like the used in maintenance gamma process [188][1][153][154][169][155][115][179]

Typically to measure this process two metrics are used

- TTF, time of total failure
- RUL, residual time of operativity in appropriate condition [100]

SOH, state of health, is an alternative to this typical metrics

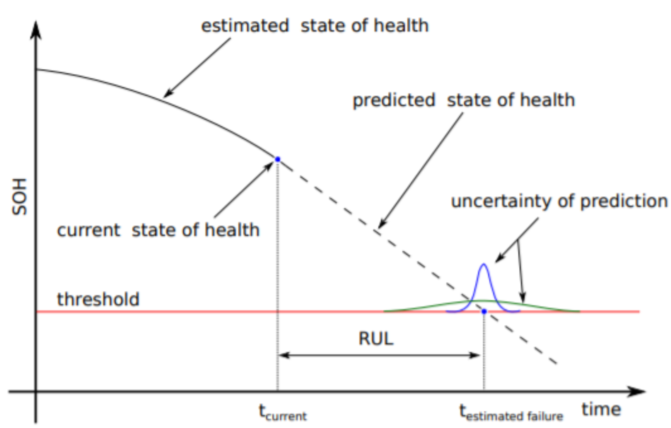


Figure 14

The figure shows relations between SOH and RUL. progressive failure of the systems is hardly monitored and translated as label. it could be useful start from a SOH index to calculate RUL. The SOH could easily be translated as values it assumes the random variable Xt we talked at start of this chapter. In a simplified fashion suitable for most predictive algorithm we discretize this Xt [17][109]. Given the fact that Xt could be discretized and at the same time approximated by SOH we could use this definition to make a direct correspondence with the number of magnet lacking in a engine and the SOH. Going further would be interesting having the possibility to estimate a lifetime function defined as expected value of RUL [164].

This could possibly be done in absence of continuous degradation simply having observed some machine faults (the ones we do not know nothing about except for time we stopped monitor that are called censored observation) using the concepts of survival analysis [83].

Our agent takes as input

1.

2. A set of machines, each ones has a set of measures and SOH labels in different period in different time horizon

3. Given a time horizon t in which the agent is in this moment, train is done on all t-1 horizon data to predict all t label. Each agent predicts a list of values that represent SOH in each single periods,

Given we have 4 SOH incompatible categories this is a multiclassification problem. This agent has also another machine learning function that

- Taken A set of prevision of SOH of the agents, given labels defined as a couple (survival time, see/censored) for each SOH prevision, in different time horizon

- Given a time horizon t in which the agent is in this moment, train is done on all t-1 horizon data to predict all t degradation function .

The agent is able to ask for data, to select best scalers and hyperparameters, to train on previous data, to do the prediction as described above, to evaluate his performance

7.4 Scheduling maintenance

Ein concordance with ITG this intervention criteria were selected

- bearings are damaged

- at least one magnet is lost

This agent simulate a delfi process receiving as features predictions of all classifier of RUL prediction and fault detection. The final step is predicting if a particular engine should be changed before next time horizon to minimize risk or equivalently costs.

So for the ensemble agents

1. given a time horizon t in which the agent is in this moment, train is done on all t-1 horizon data rappresented by failure predictions of each operative agents to predict all t-1 labels of the time horizon

2. so it predict a single value for each machine representing needing of engine substitution in the next time horizon.

3. we suppose that time to substitute an engine is less than next time horizon

4. we suppose that in case of total failure the substitution is done during the period but if it is not predicted we face a cost

The agent is able to ask for data, to select best scalers and hyperparameters, to train on previous data, to do the prediction as described above, to evaluate his performance.

The most important aspect of this ensemble agent model is its loss that was injected directly inside the classifier (modifying library code). Typically for evaluation prediction model is used f1 score [156][16][168]. The main problem of this metrics are the facts that they do not take in account costs of existing differences between maintenance cases (for ouself near start day cases cost more than near end day cases) [156]. Previously cost function metrics were founded in literature [178][44][43][22] but not integrated directly inside the model in place of optimization function until Spiegel (2018). In particular as reported by [156]. A casually permuation of true and false negative give same f1 score, that lead to corresponding same F1 isoline that intercept different cost level if there is difference between inspection cases. In respect to the work reported [156] we insert the cost function in a precise hierarchical level of an architecture incorporating both f1 and cost based prediction in an ensemble classifier and in the possible following work we will extend it to different distribution function estimated by a RUL predictor. A mathemathical exposition of the cost function would be too much extensive for this report and insert in appendix 4.

8 Technical process adopted

This process was followed with every iteration (agent production)

1. problem definition
2. instrument definition
3. Preprocessing (modelling indicator phase)
4. Model selection
5. Forecasting (with parameter tuning)

This process is a fusion of the core machine learning process intertwined with a classical predictive maintenance processes detailed in [5]. We give formal problem definition beforehand for clarity. Now we describe accurately everyone of each other step.

8.1 Instrument definition

The programming languages of our choice is Python(3.0). Advantages of this choices are reported here [122][123][124][125][126][165]

Core libraries used were

- Pandas [173]
- NumPy [172][116]
- Scikit-learn[175][132]

- math[52]
- matplotlib [62]
- sksurv[82]
- imblearn [57]
- seaborn[86]
- plotly[68]
- scipy[85]
- statistics[53]
- json[51]
- itertools[50]

During development Conda was used locally as package management system [20] included in anaconda individual edition [4], while python virtual env was used in remote.

Ide used where spider locally [157][158] and VsCode to connect to the Azure virtual machine via SSH.

For version control GitHub were used [37][38][39]

For handling evolution of development Louis Dorad canvas was adopted [27][25][26][21]

8.2 Preprocessing

This procedure during preprocessing was followed

1. dataset individuation and exploration
2. handling of structure and data format
3. dataset balancing
4. anomaly detection and handling
5. features rescaling
6. feature extraction
7. feature selection

8.3 Handling data structure

As we previously said the natural way to split over a train and test set is a time horizon fashion[156] [168][41]. is an approach used for historical data strictly correlated with cross validation concept. When number of datapoints is low to not degrading performances [9] cross validation is useful. But an inspection system obtain data highly correlated historically and so in a concrete context should be treated accordingly. Classical cross validation techniques suppose independent data instead [134]. The correct cross validation should be time series cross validation [146] unfortunately this function do not consent to split given a particular column value and need to be reimplemented manually. Using a time horizon split in prediction will lead naturally to this implementation choice.

In our business setting we have modified the original dataset labelling measurements with an identifier of a fake machine number and a fake day from which the measurement comes. In total simulate a process of predicting maintenance over 50 machines for 7 days in which each days is composed of 50 measurements for each machine. This modification do not change in any way the prediction of type of fault:

- even if we supposed some observations come from same machines at the end every measurements could be considered independent given time frames of measurements and modality of data acquisition (as stated from ITG engeeneer)
- the class label of original process remain the same
- the number of days or number of time windows reflecting only our desired and a priori number of time series crossvalidation split (seven)
- the number of measurements per day is representative of average small condominium number of lift travel of enterprise

Different measurements in typical setting have slight data variations due to noise in the signal, for this reason we added gaussian additive noise. This would lead to increasing generalization given nearly perfect separability of our data. In figure 15, on the right we can see how 1000 randomly sampled data of 2 different runs of bearings and magnet failure had a variation of the distribution and on the left the noise allows the distributions to overlap.

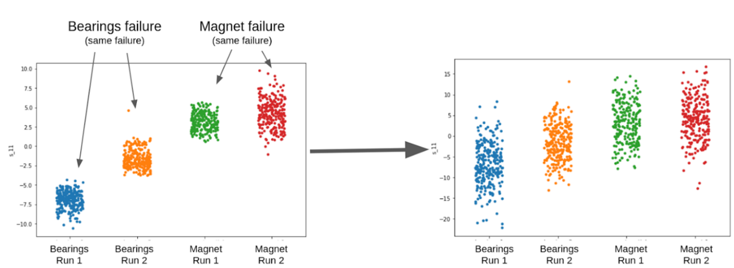


Figure 15

To increase further noise and gaining more realism of prediction

- we added erroneous label (we randomly change label using a poisson with lambda=1/1000, parameter concorded with enterprise) to simulate erroneous insertion of label in the system by an operator [66]
- to break uniform consecutive set of same label we insert a divergent label. This create situation of a [65]

And using pair plot method we could visualize how we achieved more superposition of distribution and generally acceptable low correlation within variables.

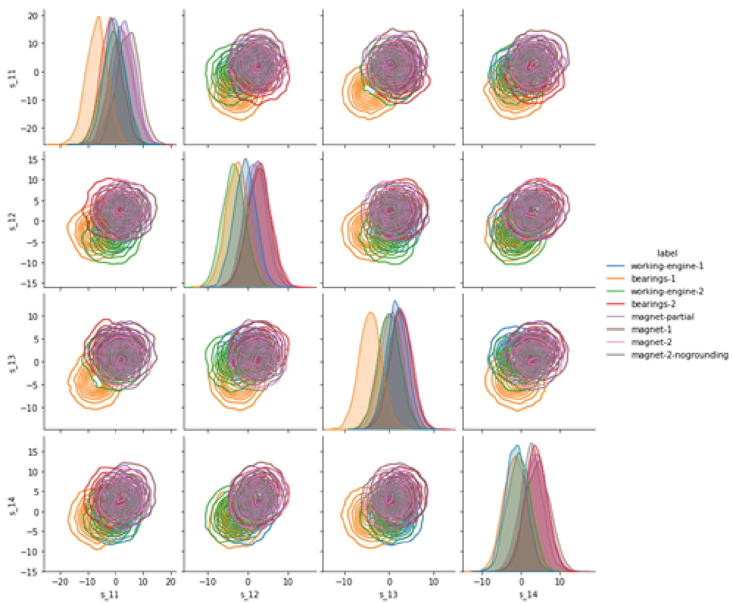


Figure 16

8.4 Dataset balancing

Lacking balancing in dataset could lead model to prefer overrepresented classes [150][107].

Two tecnhiques are commonly used

- under sampling [150]
- over sampling[107]. This methods could lead to overfitting and is not guaranteed to lead to better classification.

Both methods revealed useful to find anomalous events [101]. Given our small dataset oversampling was only viable choice to efficiently train classifier afterward, using SMOTE [58] a reliable technique [160]

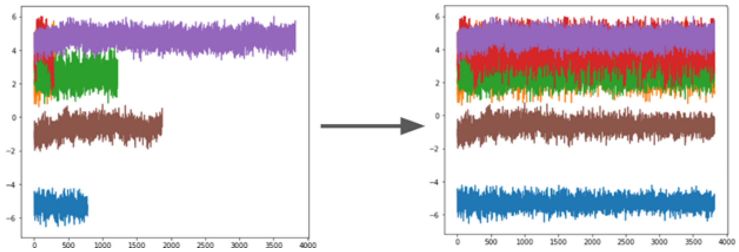


Figure 17

Using PCA we verified that SMOTE extends previously distributions of data.

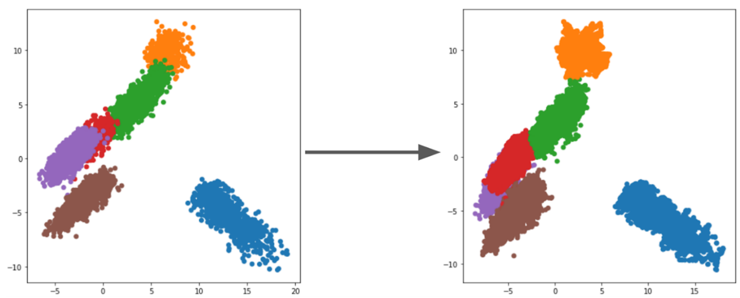


Figure 18

Testing lead to better performances for data augmentation.

8.5 Outlier removal

An outlier could be signal or noise. In our case we test the models with and without outlier leading to increase performances after outlier removal. [107] This lead us to identify it as noise. After identification we done [160] a direct removal (data synthesis to substitute outlier was not considered because it will be based only on a small datasets lead to other approximation)

Two methodologies were tested, and the second approach was the most useful

- Interquartile range (IQR) [107]
- Z-score o standard score. Tipically using 3 sigma [152][176]

both were not computed in all the features but instead in a preselected subset that was visually containing them.

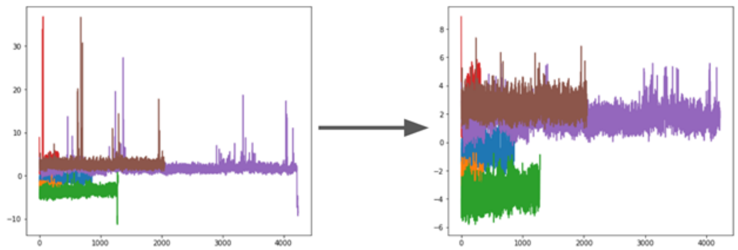


Figure 19

8.6 Feature extraction

Following previous literature we were able to extract following features [177]

- RMS : root mean square
- Variance
- skewness : normalised third central moment
- kurtosis : normalised fourth central moment
- PV : peak value
- 50th percentile
- Interquartile range
- Mean absolute deviation
- Impulse factor
- Trimmed mean

These are all the features able to identify increasing of vibration parameters [2].

Exploring descriptive task could benefit inspections[10]. Clustering is frequently used to identify nonconforming in data leading to identification of hidden trend in unsupervised way[110] (clustering showed also capability of identify errors in data like our process of erroneous label generation [46]).

The cluster label of which a data come from could be directly used as features as we did (figure 20).

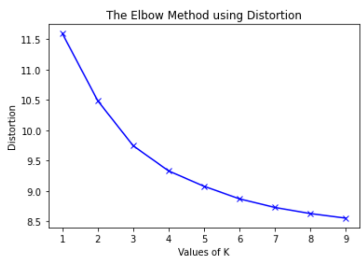


Figure 21

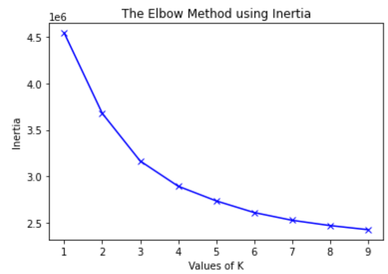


Figure 22

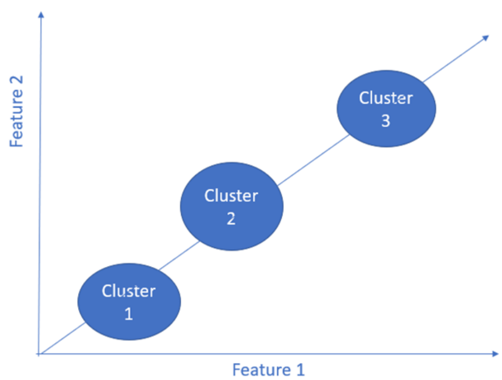


Figure 20

To apply it we use the elbow method on inertia and distortion to find optimal number (4 in our case).

Our clusters (figure 23) seem in fact show a trend and capability to discriminate.

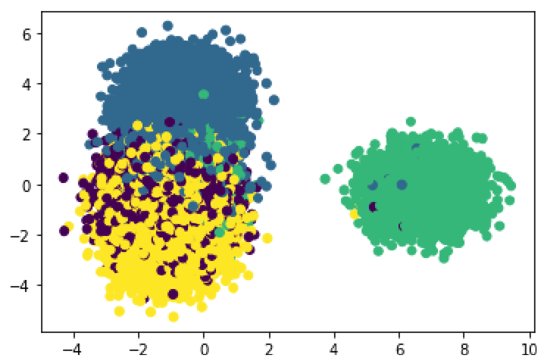


Figure 23

8.7 Feature scaling

Rescaling is especially useful for convergence and assure discrimination capability of some models we will use [107].

Given the fact that we could not know a priori which scaler enhance performance we selected the scaler dynamically in the agents via a time series cross validation approach. Different scaler are tested for each task crossvalidating performances each time an agent has to do a prevision. Scaler associated to best performances are selected. We tested

- Standard Scaler. [133]
- MinMax Scaler. Used typically for low variability data (adopted for example [160][78])
- MaxAbs Scaler [77][107]

8.8 Feature selection

Typically number of samples increase exponentially in number of features [18] [171] in a phenomena described by Hughes [95] [107]. Too much dimensions lead data to be difficult to be grouped [180], at same time too low parameters make model biased [9]. We resort to a custom wrapper method coupled with Filter methods [9]. The first one bottom up constructs increasing subsets of features selecting subsets which lead to best cross validated performances. Before reaching our wrapper bottom up method we reduce the features [59] with filter methods

1. using trees to find usefule feature, it make no assumptions so its our first choice to find at least n features [73][69]
2. if not n features are found try use linear SVC to obtain coefficient of features in number at least n [73][70]
3. if no more than n features were extracted before, n features are extracted using select kbest with ANOVA [72][74]

This n features are used to limit wrapper bottom up maximum feature set number. The number n is selected in this way:

- For uncorrelated features, the optimal feature size is $N - 1$ (where N is sample size)
- As feature correlation increases, and the optimal feature size becomes proportional to $\sqrt{N}$ for highly correlated features [94].

To test correlation the corr function of pandas [67] with kendall non parametric test was used (this because make no assumption about underlined distribution and is preferred when samples are small [49]).

This procedure is repeated every time an agent acquire new data for prediction.

8.9 Model selection

After that we should select suitable models for our prevision. Our core criteria are robustness and interpretability. Interpretability is a core feature for a prediction model to asses ratio of a prevision while robustness is useful because sensors data are not cross-checked with other datasets. Another key factor is the possibility to extract probability to measure confidence[9].

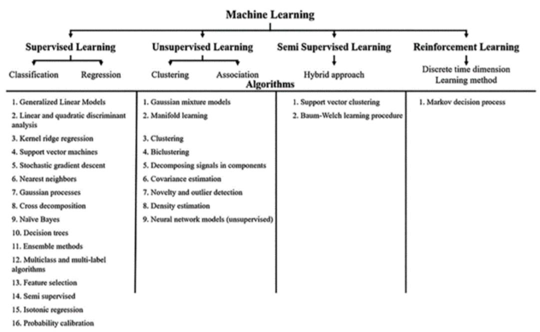


Figure 24

8.9.1 Fault prediction

For multilabel fault prediction and classification we opted for one vs one approach that is more efficient for a small dataset like ours [55][60].

The classifier used for this type of agent are:

- K-Nearest Neighbors,It offers interpretability and non linearity.
- Support vector machine. It could represent complex nonlinear function, there is a closed form solution, is scalable, is efficient and has an error bound. Support vector machines were used previously for prevision of rare events like for example fiscal frauds [103] and more recently in predictive maintenance [111]
- LDA, was used from years by national inspection offices (like U.S. Internal Reveneu Service) for more than 40 years to spot anomalous situation [29][166]. LDA do hard assumption like same covariance of classes but has a linear decision function that is easy to interpret [136].
- Logistic regression, has an interpretability advantage [107]. It was extensively used in prediction of non conformities [11] and financial frauds [103].

For multiclass RUL classification in a OVO process Ridge regression was tested. It is considered more efficient for solving multiclass problems than logistic regression [137] Another approach suitable with one vs one multiclass method is SVM Another fast nonparametric model used is KNN After extracting this labels we used a survival forest to predict degradation function [84] because of the interpretability of random forests and high performance also without optimization and tuning.

For maintenance scheduling random forest classifier was used. Its adoption was due to

- Nonlinear model able to spot nonlinear correlation between inspection periods[156]
- It is one of the most adopted model in predictive maintenance [13][16][41][168]
- Requires small dataset, reduce overfitting[19]
- It's capable of extract automatically decision rules fundamental for a decision making level [135][107]

8.9.2 Hyperparameters tuning

Typically a grid search is used for hyperparameter optimization [145][151] but given the fact we desire an agent approach with multiple classifiers in which parameter tuning is done fast in a crossvalidated fashion every time arrive data. to do that it selects best crossvalidation combinations of parameters tuned on dataset labels avaiable in a certain day. We have first determine what feature to tune (using concept of tunability [121]) and approximating intervals or suitable values to be evaluated during agent execution using literature [96] For limitation in hardware (and because most research is devoted to single parameter values analysis [117]) we will tune only single parameter at once.The default scaler was applied before tuning based on literature best scalers references for different classifier used.

Tree based models[32][149][140]

- `max_features` It should depend on irrelevant features number [128][37][42][117]. More irrelevant feature need more bigger value. the value $\sqrt{m}$, where m is the number of features, is most stable value between dataset so is selected as default value. To that a list of $\sqrt{m} * 2, \sqrt{m} * 4, \sqrt{m} * 6$ were added if high correlation (high number of irrelevant features) while $\sqrt{m}/2, \sqrt{m}/4, \sqrt{m}/6$ was used as hyperparameter list if correlation of features was low
- `min_samples_split` Default values are typically 1 equal to N [96] but if lower than N increase prediction variability [108] and using a value greater than 1 risk to introduce bias [14] given that a list values of [1.0, 0.5, 0.25, 0.1, 0.01] was selected as proportion of min samples split of total sample. `min_samples_leaf` [104] [117] underline runtime performance increasing if it is increased, without degrading accuracy. A list of [1, 5, 10, 20, 40, 60] was selected `n_estimators` In [117] is showed how increasing of performances is typically more accentuated until 100 classifier. So we opt for multiple of 10 until 100.
- `bootstrap` a tunable parameter in [163][96]
- `criterion` tested for both gini and entropy

Forest for features selection could be done more straightforward [96]

- number of trees[34] more trees than default are required for stable importance estimates. A value of 100 is used to compromise between high estimators numbers and performances [105].
- `max_features`/ node size could be not considered and putted to a low value (to make stable importance prediction [159]) in our case of continuous values and tree completely unfolded[33]

tree based model are not influenced by scaling [87]
SVM [80][79]

- C hyperparameter and Gamma were tested with exponentially growing sequences as efficient method [47] in range [0.0001, 0.001, 0.1, 1.0, 100]
- Kernel function rbf, linear and sigmoid (Polynomial kernel was not tested for technical constraints)

Optimization parameters:

- Tol setted as minimum value for convergence (1e-3)
- Shrinking parameters could accelerate elaboration [31]

scaler used is standard scaler
KNN [148]

- k number of neighbors . A too smal value like 1 create to coars boundaries while greater than 50 neighbors make the model of difficoult interpretation [93] all values between 5 and 60 were tested.
- we tested metrics avaiable for real values [147]
- weights, uniform or inversely weighted in distance

Standard scaler is a suitable starting choice because of distance being calculated by the algorithm [63].

Logistic regression [75]

- hyperparameter C. following SVM approach an exponential increasing search was adopted with values [0.0001, 0.001, 0.1, 1.0, 100]
- penalty, with l2, useful for generalization, l1 for a sparse solution [113][145].
- class_weight to values none or balanced

Not influencing model performance parameters choice

- solver as liblinear given efficiency for small dataset and support for both method of regularization [137]
- tol setted to smallest found convergence value as 1e-3

Scaler used in literature is standard scaler[161].

Linear discriminant analysis[71].
Being a classifier that solve a closed form solution it has not "real" hyperparameters:

- Shrinkage typically could help if there are less data
- solver with lsqr that could use shrinkage and has a fast closed form solution

Scaler used in literature is standard scaler [161].

9 Results and discussion

9.1 Metric used

9.1.1 Confidence

In machine learning there is no a clear cut definition of confidence [48][64] and typically is used a synonym of Probability. In this context instead we want to recover the original meaning of confidence, using the specific notion of metrics for the second definition. In particular we are talking about [56]

Confidence interval can be used to estimate the true error of the model as a function of the sampling error [91][89].

In our case we are estimating samples of independent processes. Internally time series of machines there is correlation but not between machines so we could apply this approximation (Binomial proportion confidence interva).

$$\text{interval} = z * \sqrt{\text{error} * (1 - \text{error})}/n$$

with classic 95% confidence [90] In our case T-distribution is used instead of z because information about the distribution is not known [91]

We used confidence because there is an accountability purpose in justify the prevision to the customers and for model improvements, main concern of ITG [92]

Secondly there are some interesting opportunities

- Learning ensemble model on data is prone to overfitting. Using also confidences as a criteria for feature selection is a way to balance unreliable performance metrics
- Confidence could be another metric to furthermore improve models when f1 score could not be furthermore improved
- Confidence could actively use to evaluate model with increasing data point in a following experimentation phase

9.1.2 Proper machine learning metrics

f1 score is the firstline metric used in inspection [61]. We used this metrics coupled with confidence for evaluate performance of third layer classifier as well to choose parameters, scaler and features. From confusion matrix [143][107] we included in each agent also other cross validated metrics to allow different comparisons:

- precision or strike rate in inspection context [45].
- Recall: called also efficiency of inspection [45]

An interesting element is necessary trade off between maximise inspection benefits (precision) and minimize cost (recall) [12]. Selecting inspection case based on a probability threshold has impact on this metrics [45] to evaluate that each agent is provided with precision-recall curve [138][139][144][88].

The setting specific metrics included for completeness the percentage of real examination effort needed ((real effort-actual effort)/ real effort)*100 where real effort is $TP + TN + FP + FN$ and actual effort is $TP + FP$ [101][45].

More exotic metrics included:

- balanced accuracy score [141]

In case the agent classifier calculate also probabilities:

- Brier score [138][142]
- receiver operating characteristic (ROC) [174][76][138][102].

We opt for metric comprises between 0 and 1. This has advantage of being agile used by technical community (being the standard) like future developer and at same time easy using any bijective function transforming the interval $[0, \inf[$ to make comparisons between models. In the future qualitative or more easy interpretable outputs for non technical members could be more useful. An example is savings cumulation (called savings) included in the fifth layer agent able to calculate savings obtained by the agent until now.

9.2 Code produced: brief explanation

The code is divided in

- notebook presenting results and elaboration to prove our efforts
- code of the business setting implementation

The latter is composed by 4 files

- `ensemble_agent_var.py` is the code for the scheduling agent
- `opearative_agent_var.py` is the code for RUL and fault detection and isolation agent
- `data_manipulation.py` is the code for data extraction and manipulation agent
- `forest_variation.py` contains the new type of three with modified cost function

The code Is executable easily from `ensemble_agent_var.py`. When compiled it creates automatically a scheduling ensemble agents (referenced by ensemble name) which contains 4 fault isolation agents (with classifier respectively knn, logistic, svm, lda) and 3 RUL prevision agents (with classifier knn, svm, ridge + random survival forest for rul prediction). Using `agent.operative_cycle()` methods all agent below scheduling agent receive a day of data (starting from third day to have a base for training and crossvalidation in the two previous days). Their previsions are fed to scheduling agents automatically. Metrics of performances presented before are easily visualized accessing to proper attributes (for example `agent.savings` obtain savings predicted with scheduled intervention until that day).

```
In [10]: ensemble.savings
Out[10]: 3439.795918367347
```

Figure 25

The prevision of each agent could be accessed with `*name_of_agent*.last_pred.` . For the scheduling agent for example using `ensemble.last_pred`.

```
In [11]: ensemble.last_pred
Out[11]:
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Figure 26

An interesting possibility is to obtain the RUL prediction in plotted form for a specific datapoint via `*RUL_agent_name*.auxiliary.depict_survival(*indice_previsione*)`. It shows a graph (figure 27) depicting how survival probability evolve with number of trips of elevator.

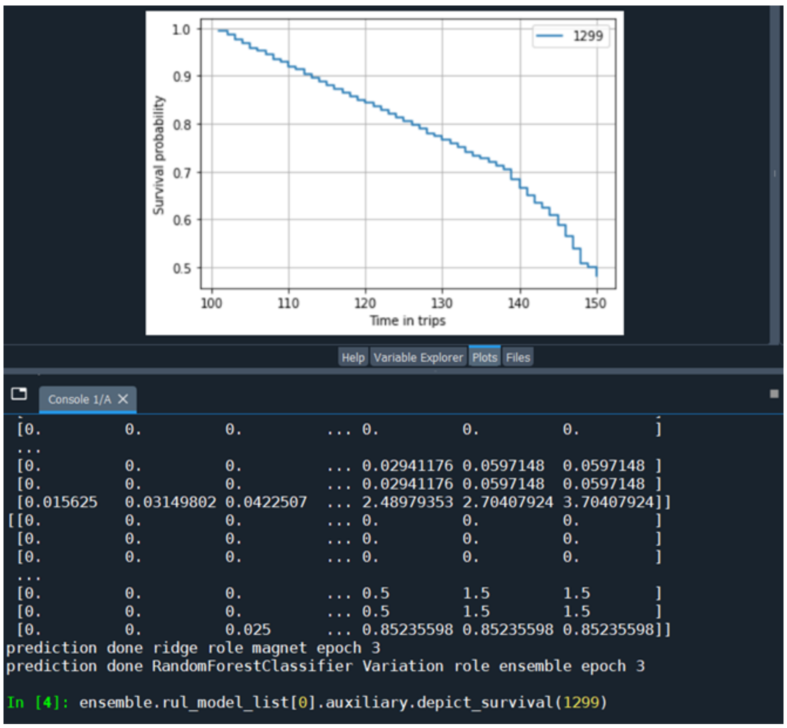


Figure 27

The code of this agent is written to be easily modified once someone comprehend class contained in agents and functioning make able to pass different classifier to each agents, different number and type of agents for each architectural layer, different features, different data manipulation.

9.3 Performances

A depiction of performance of a simple KNN classifier is in the notebook 7 showing how the dataset could already be fitted perfectly matched with agents performances calculated on average f1+confidence (notebook 8).

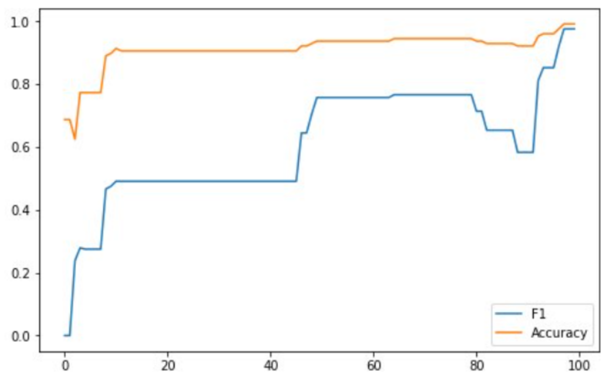


Figure 28

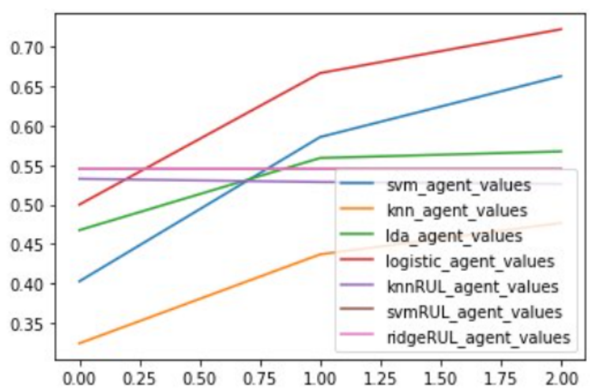


Figure 29

We could observe that the total score we used taking in account confidence is weighting the prediction low confidence negatively make able to select more properly a good model and good number of machine. More interesting is the performance of cost model. Given that the performance of cost model increase with data , during our trial the cost model overcome the old standard model (notebook 9).

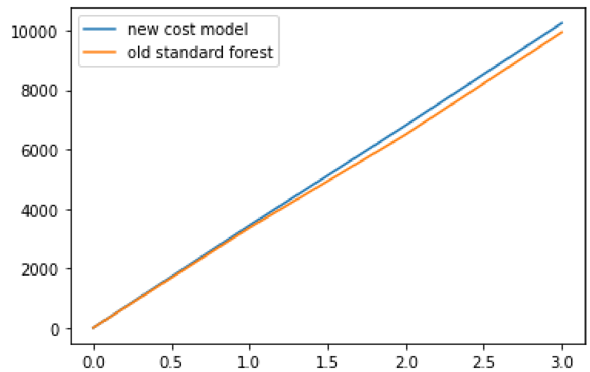


Figure 30

Figure 30 shows the ability of schedule agent to increase savings in long run weighting better the prediction that lead to best savings.

9.4 Expected benefits for ITG Tecnologie

ITG would now be able to start with a predefined architecture to guide the development of an actual product. Our new devised architecture is a conceptual scheme that can also cope with other industrial application of enterprises in case wanted to expand its AI use. The architecture proposed has clear advantages for

- Actuality, it is an extension of a 2021 architecture
- Cost savings and reporting for management this could lead to an easy approval by management of new developed model and fasten the research of solution comparing economic benefits.
- Flexibility, This architecture make also able to leverage learning and integration in data collecting being easily modlurazied and make able to substitute an agent with another more efficient with more datas in a smooth way (in fact the ensemble will give weights to model previsions based on performances so no need to modify code for remove old model)

The architecture is proven feasible by our direct agent implementation and we espect it to be taken along with implementation also for future research.

The business setting implementation offer

- different metric evaluations for testing for future development
- a measure of confidence that will make able researcher and enterprise to evaluate evolution of prevision with more data
- an independent agent structure that automatise all goals of ITG challenge in a smooth way and deals also with concrete case of monitoring (multiple faults in a

single machine, correlation in time frames, need of different level of prediction etc.)

The Colab notebooks analyze issues and characteristics of data that were collected and offer insights on advantages and disadvantages of data collected with this approach. We hope that this and teams interaction has and will enhanced data maturity of ITG we strive the report to offer also a vast bibliography to guide further development of the solution based on this architecture.

10 Further work

10.1 Suggestions and guidelines for the company on how to increase probability to industrialize results

10.1.1 Modify cost function

The cost function improve benefits increasing gap and reducing time window but with the risk of more imbalance until a peak of performance is reached. Initially the model should forecast near term event, when data are collected in large number is possible to reduce imbalance with downsample leading to possibility to increase prediction time window. Time gap should instead evaluated in a more trial and error way . Other extensions that could come naturally to this cost function should be

- extension to multitasking problem of other prediction that could benefit of this cost function like managing also other type of useful interventions
- extension to multiclass problems trying predict more articulated scheduling instead of simple check or non check an elevator
- introduce the RUL extimation in the cost function
- introduce in the risk calculation also a risk depndente on risk profile of specific user that use this elevator
- introduce directly SOH extimation in the residual life of the component in the cost function obviously the cost function should be tuned to real parameters of costs, time gap etc before starting exercise that depend on assesses
- estimating risk via acoustic model to model discomfort or via standards for comfort like ISO 2372

10.1.2 Using cloud platform

The usage of json format for storing data could be useful for integrating in cloud storage options like amazon web services [112][15] cloud computing is an increasing industrial 4.0 paradigm [199] and heavily use in predictive manufacturing [200]. A cloud approach make it easy data fusions (see later) and could be adopted gradually, for example do long term prevision with all data in cloud and short term in local databases

[201] the cloud possibility appear more important if the sensor will increase sampling rate or at a certain number of data collected.

10.1.3 Integration of IoT

A possible implementation could be the implementation of a multiplatform approach that after user login hast he possibility to see ist purchased machines automatically stored and retrieved for a database. Via its account he could monitor the status [187].

Integration of this multiplatform approach on mobiles would lead a lot of benefits. For example if integrating this application on mobiles could open possibility of superimposing machine information over a picture of the same machine (or a real time video if we want to go to augmented reality) [111].

We think that a maintenance assisted approach could be a long term objective to be integrated in the predictive maintenance module given the servitization interest, and so given the fact that maintainer are external and could do faster maintenance if they are guided, as depicted in figure 31.

This approach would gain strength with coupling with cloud [15].

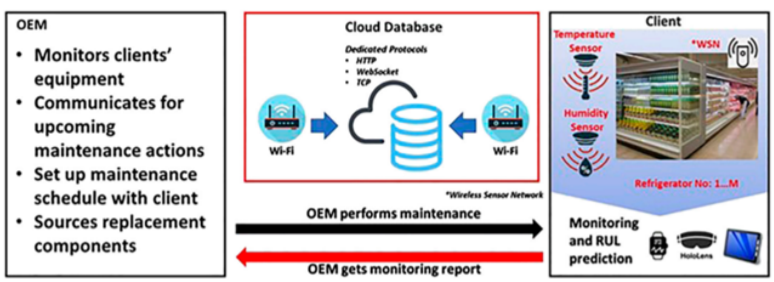


Figure 31

10.1.4 Nice Interfaces

It is important and useful having nice interface for user (for that we suggest image of machine or 3d model of machine).

Given the desire of servitization this tool of predictive maintenance when directed to manager could also include view that contains more economical data. This could be:list report, scorecard, dashboard with kpi.

This would help remember users the added value of the predictive maintenance solutions and could be easily coupled with our fifth layer architecture agents.

10.1.5 Convince stakeholders

To leverage maximum efficiency inside ITG it is useful to have economical prospect like previous mentioned for customer also for internal management. Being able seeing savings that the tool consents will convince management to invest more in the tool and that would lead developing via new module or experimentation [184]. It is obviously important that this model is comprehended in at least its basic functioning and reliability basis to not make it a black box model. This leads to more confidence in its capabilities of prediction.

10.2 How to enhance the data collection and management in the future

10.2.1 Create continouslike data

It has been suggested that in real world application an event-trigger paradigm, in which the sensor would measure the motor only when the motor starts with a suitable sample rate, would be more suited for this specific problem make able to analyse entirely the start, stabilization and stop of the engine at the specific moment and increase possible features included

[178] suggests defect harmonic analysis (until six) to examine defects in engine moving average for pruning data.

10.2.2 Treat data

Many methods can be used to handle the classes imbalance that can improve accuracy, for example testing more undersample in order to minimize the error of random sampling from the larger class. Moreover is possible to combine oversampling and undersamlpling by repeating undersampling to train the model and then combining all the dataset for a model training (i.e. oversampling) [101].

10.2.3 Increase source data

Also maitaning less but random inspections could be beneficial to obtain random samples from engines enriching the model [202] this casual inspection could also be programmed via mathemathical model that take in account costs like [43] evolution of [203] More advanced data synthesis approach could be tested like autoencoder for lack of label in RUL prediction [204].

We have used until now event data. But it is important to obtain also similar knowledge from areas not strictly related to maintenance. This is called data integration. Integration with data from erp e shop floor level could be a viable possibility to gain more data and more efficient services [5][15]. It is possible to add data via environmental conditions and log data. Also economical data like warranty information could be very useful[185]. Case base reasoning and fleet wide approach are viable options [186].

The insertion of new information fusion layer is proposed[186].

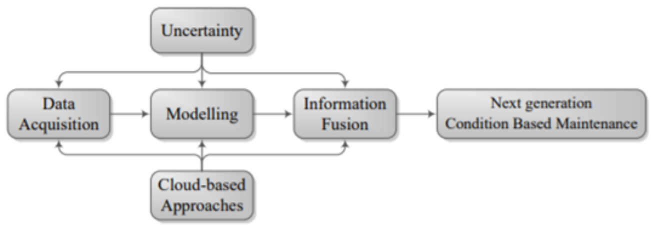


Figure 32

10.2.4 Create a physical simulation

A vast array of literature underline importance of couple physical or mathematical model of degradation with machine learning model. Our architecture consent naturally this. The usage of this simulation models could help constructing what is called a digital twin of each engine meaning a reproduction of the elevator in a 3d model[187].

10.3 How to better tackle the challenge

10.3.1 Other confidence calculation method

Confidence estimation is very important. Assumption of independence between machines could be avoided with technique like presented here (a technique that use Montecarlo simulation) [188][189].

We have to underline that this calculation of confidence is viable only if we are not supposing a distribution a priori (and is the classical statistical notion of confidence). In the cases we integrated a statistical or physical model could be better used Bayesian Confidence Intervals. [190]

10.3.2 Parameter tuning

Hyperparameter testing and selection should be done in respect of actual technical constraints. Other hyperparameters metrics are in [191]. For a review of tuning methods [192]. For tuning neural network hyperparameters see [193] Specific methods for trees applicable also to forest that are a very hard tuning algorithms (and so worth be mentioned) are for example in [194][195].

10.3.3 Test new models

A simple model not tested for time constraints that could be used is Bayesian network. Usage of Bayesian network was also coupled with logic based methods like ANFIS to estimate a single prediction from many predictions[196] To predict RUL other possibilities of white box models are

- Bayesian linear regression

- Boosted decision tree regression
- Neural network regression[160]

Using forest also for fault classification could be done when more data are available. In this case other type of used forest are: decision jungle, boosted decision tree [160].

If data fusion were used sparse Bayes learning and relevance vector machines could be used [186]. Doing RUL and failure prediction could be probably done better with a single neural network but this should be tested only with more data. In general deep Weibull model and long short term memory network were vastly used for predictive maintenance [198].

10.4 Consideration for signal in actual environments

The signals in actual environment should be polished by noise using for example low pass filters or Kallmann filters. Generally it is suggested (i) high-pass or band-pass filtration, (ii) envelope detection or demodulation of the band-pass filtered signal [197].

References

[1] Authors, ecc