




Predictive Maintenance Experiences on Imbalanced Data with Bayesian Optimization Approach

Nicola Ronzoni¹, Andrea De Marco², and Elisabetta Ronchieri^{1,3} 

¹ Department of Statistical Sciences, University of Bologna, Bologna, Italy

`nicola.ronzoni@studio.unibo.it`

² BitBang s.r.l., Bologna, Italy

`ademarco@bitbang.com`

³ INFN CNAF, Bologna, Italy

`elisabetta.ronchieri@cnaif.infn.it`

Abstract. Predictive maintenance solutions have been recently applied in industries for various problems, such as handling the machine status and maintaining the transmission lines. Industrial digital transformation promotes the collection of operational and conditional data generated from different parts of equipment (or power plant) for automatically detecting failures and seeking solutions. Predictive maintenance aims at e.g., minimizing downtime and increasing the whole productivity of manufacturing processes. In this context machine learning techniques have emerged as promising approaches, however it is challenging to select proper methods when data contain imbalanced class labels.

In this paper, we propose a pipeline for constructing machine learning models based on Bayesian optimization approach for imbalanced datasets, in order to improve the classification performance of this model in manufacturing and transmission line applications. In this pipeline, the Bayesian optimization solution is used to suggest the best combination of hyperparameters for model variables. We analyze four multi-output models, such as Adaptive Boosting, Gradient Boosting, Random Forest and MultiLayer Perceptron, to design and develop multi-class and binary imbalanced classifiers.

We have trained each model on two different imbalanced datasets, i.e., AI4I 2020 and electrical power system transmission lines, aiming at constructing a versatile pipeline able to deal with two tasks: failure type and machine (or electrical) status. In the AI4I 2020 case, Random Forest model has performed better than other models for both tasks. In the electrical power system transmission lines case, the MultiLayer Perceptron model has performed better than the others for the failure type task.

Keywords: Failure types · Machine status · Predictive maintenance · Bayesian optimization · Imbalance data · Random Forest · Multilayer Perceptron

1 Introduction

Predictive maintenance (PdM) is a specialization of condition-based maintenance that requires data from sensors (e.g., used to monitor the machine) as well as other operational data, aiming at detecting and solving performance equipment issues before they take place. PdM focuses on utilizing predictive info in order to schedule the future maintenance operations [1]. Through the data collected with the machine under operation, descriptive, statistical or probabilistic approaches can be used to drive prediction and identify potential problems. With PdM various goals can be achieved, such as reducing the operational risk of missing-critical equipment; controlling cost of maintenance by enabling just-in-time maintenance operations; discovering patterns connected to various maintenance problems; providing key performance indicators. Machine learning is successfully applied in industrial systems for predictive maintenance [8, 28]. Selecting the most appropriate machine learning methods can be very challenging for the requirements of the predictive maintenance problem. Their performance can be influenced by the characteristics of the datasets, therefore it is important to apply machine learning on as many datasets as possible.

In this paper the field of predictive maintenance in manufacturing and transmission line is considered. The aim is to classify failure modes in class-imbalanced tabular data according to a supervised fashion. These tabular data contain data points as rows, and regressors and targets as columns. Moreover machine learning models and neural networks, combined with Bayesian optimization approach [26], may open new perspectives in detecting fault configurations. With this in mind, we have developed a pipeline that satisfies the following requirements on tabular data: supporting versatile applicability in different contexts; tackling classification problems with imbalanced classes; discriminating among different failure modes (primary dependent variable); being able to recognize faulty and healthy status (secondary dependent variable) with the same settings (i.e., hyperparameters) used for the primary dependent variable. In order to be versatile, the pipeline supports multiple models under different settings. The best set of hyperparameters for each model is the result of an informed search method based on Bayesian optimization solution. Once the best set of hyperparameters has been selected, the model is fitted twice with different tasks. The multi-output approach fits the same model for each dependent variable: first to solve a multi-class problem (primary dependent variable), then to figure out a binary problem (secondary dependent variable). Furthermore, we have considered two different use cases, such as AI4I 2020 predictive maintenance dataset [9] and electrical power system transmission lines [16] dataset.

The model has been built by considering four machine learning methods: three techniques belong to the ensemble machine learning methods (where the model makes predictions based on different decision trees) such as Adaptive Boosting, Gradient Boosting, Random Forest; and one technique is an artificial neural network, such as Multilayer Perceptron. Our contribution, beside applying for a predictive maintenance solution based on Bayesian optimization technique, aims at replying to the following research question: can a model, whose settings

are tuned on a more complex task (primary dependent variable), perform well also on a relative simple task (secondary dependent variable) connected to the complex one?

2 Background and Related Works

With regards to the related works, our starting point has been focused on maintenance modelling approaches on imbalanced classes. A short literature review has been performed to highlight previous research in the predictive maintenance field. The machine learning-based PdM strategies are usually modeled as a problem of failure prediction. With imbalanced classes, ensemble learning methods as well as neural networks, combined with Bayesian optimization method, can be used to improve model performances. In the following we have summarized some papers related to the machine learning methods, such as Adaptive Boosting, Gradient Boosting, Random Forest and Multilayer Perceptron, adopted in this study showing interesting results in PdM.

Adaptive Boosting is one of the most popular ensemble learning algorithms [30]. I. Martin-Diaz et al. [17] presented a supervised classification approach for induction motors based on the adaptive boosting algorithm with SMOTE - i.e., a method that combined together the oversampling of the unusual minority class with the undersampling of the normal majority class - to deal with imbalanced data. The combined use of SMOTE and Adaptive Boosting presents stable results. Another ensemble method is the Gradient Boosting method. Calabrese et al. [6] used a data-driven approach based on machine learning in woodworking industrial machine predictive maintenance. They achieved an excellent accuracy 98.9%, recall 99.6% and precision 99.1% with the Gradient Boosting model.

Some other studies use the Random Forest classifier [21]. M. Paolanti et al. [22] described a machine learning architecture for predictive maintenance of electric motors and other equipments based on Random Forest approach. They developed their methodology in an industrial group focused on cutting machine, predicting different machine states with high accuracy (95%) on a data set collected in real time from the tested cutting machine. Qin et al. [24] used the Random Forest method to predict the malfunction of wind turbines. The wind turbine data is obtained through the use of the supervisory control and data acquisition (SCADA) approach [24].

V. Ghate et al. [10] evaluated the performance of the developed MultiLayer Perceptron and self-organizing map neural network based classifier to detect four conditions of the three-phase induction motor. In this work, the authors computed statistical parameters to specify the feature space and performed Principal Component analysis to reduce the input dimension. M. Jamil et al. [16] applied artificial neural networks for the detection and classification of faults on a three-phase transmission lines system. All the artificial neural networks adopted the back-propagation architecture, providing satisfactory performances once the most appropriate configuration has been chosen.

There are many methods for optimizing over hyperparameter settings, ranging from simplistic procedures, like grid or random search [2], to more sophisticated model-based approaches using Random Forests [15] or Gaussian processes [26].

3 Dataset Description

3.1 AI4I 2020 Predictive Maintenance Dataset

The AI4I 2020 predictive maintenance dataset [9] comprises 10000 data points, representing synthetic multivariate time series. It represents a manufacturing machine where variables reflect (simulate) real signals registered from the equipment. The dataset indicates at each timestamp (row) a failure type and the machine status. The failure type consists of six independent classes including a *no failure* class. If at least one of the five failure modes (one of the 5 classes that differ from *no failure* class) is true, the machine stops and the machine status is set to *failure*; while if the failure type belongs to *no failure* class the machine status is set to *working*. At each timestamp we have a set of regressors and two connected targets: the failure type (multi-class) and the machine status (binary class). The set of regressors includes the original variables and the estimation of variables of which there are no measurement, by employing measures on other variables that have been recorded.

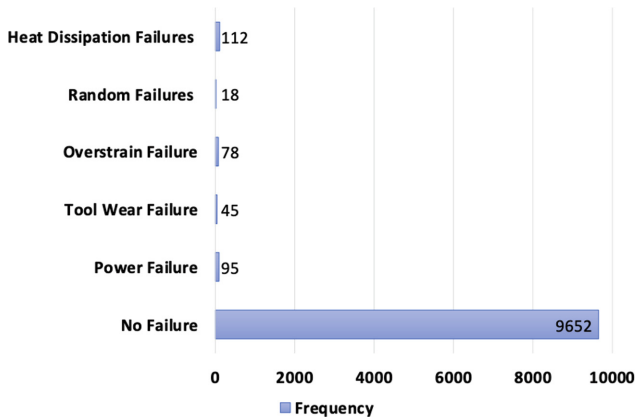


Fig. 1. AI4I 2020 dataset: failure types frequencies

The original variables that represent directly real signals are: type, process temperature in Kelvin [K], rotational speed in revolutions per minute [rpm] and tool wear in minutes [min]. Type is a categorical variable that is composed of three classes, representing the quality variants of the process (i.e., low 50%,

medium 30% and high 20%). We have mapped this variable with a label encoding. Process temperature symbolizes the temperature of the machine that is generated by using a random walk process, normalized to a standard deviation of 1 K, and added to the air temperature plus 10 K. The temperature of the environment is generated by using a random walk process, normalized to a standard deviation of 2 K around an average of 300 K. Rotational speed reproduces the rate of rotation in revolution per minute of the cutting tool that is computed from a power of 2860 W with a normal noise distribution. Tool wear returns the minutes of regular operation of the cutting tool in the process.

The estimated variables are defined based on specific failure mode that may occur. Heat dissipation in Kelvin [K] depicts the absolute difference between the air temperature and process temperature. Power in Watt [W] acts as a sample of the power, obtained by the multiplication of the torque (in Newton meter) applied to the cutting tool and the rotational speed (in radians per seconds): the torque is generated from a Gaussian distribution around 40 N meter with a standard deviation of 10 N meter and no negative values. Overstrain in Newton meter per minute [Nm \times min] outlines the demand of resilience by the cutting tool, expressed by the product of Tool wear (in minutes) and Torque (in Newton meter). The set of seven regressors reproduces physical conditions responsible of the failure modes. Figure 1 shows the six failure types on the manufacturing machine.

3.2 Electrical Power System Transmission Lines Dataset

The electrical power system transmission lines [16] dataset collects 7861 points of a three-phase power system. The electrical system consists of 4 generators of 11×10^3 Volts [V], each pair located at each end of the transmission line. Data simulates signals from the circuit under no faulty condition as well as different fault conditions.

The measures of line voltages in Volts [V] and line currents in Amperes [A] for each of the three phases (i.e., A, B, C) are collected at the output side of the power system. Each point in the dataset takes values for: the six regressors (line currents and line voltages for each of the three phases), the failure types and the electrical status (connected tasks). The failure type consists of six independent classes including a *no failure* class. As in the AI4I 2020 dataset, if at least one of the five failure modes is true (one of the 5 class that differ from the *no failure* class), the electrical transmission goes down and the electrical status is set to *failure*; while if the failure type belongs to *no failure* class the electrical status is set to *working*. Figure 2 shows the failure types on the electrical system.

4 Exploratory Analysis

The AI4I 2020 and the electrical power system transmission lines datasets share some similarities. They have a small number of regressors and data points. Both

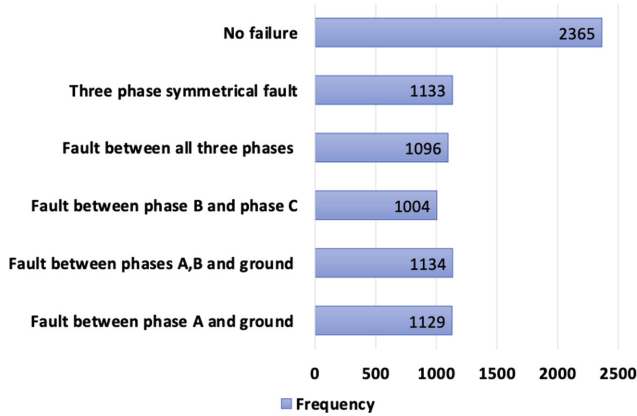


Fig. 2. Electrical power system transmission lines dataset: failure types frequencies

datasets contain 6 classes for the failure type task. The distribution of points across the known classes is biased with different degree of skewness.

Figure 1 demonstrates the severe imbalance of AI4I 2020 dataset. Even when the five failure modes are aggregated together at the machine status level, the distribution of *failure* and *working* labels is uneven by a large amount of data points. The *random failure* class (standing for the chance of 0.2% to fail regardless the value of its regressors) is included in the *no failure* class due to its low frequency and its randomness.

Figure 2 represents the slight imbalance of the electrical power system transmission lines dataset. Failure modes are in the same order of magnitude of the *no failure* class. Furthermore, when these five modes are aggregated together in the electrical status, the *failure* class becomes more frequent than the *working* one.

According to the way machine (or electrical) status is defined in Sect. 3, it is reasonable to consider it connected and correlated to the failure type. The machine (or electrical) status is a binary variable in which one class aggregates together the five failure modes of the failure type. Specifically the *failure* class, in the machine (or electrical) status dependent variable, includes the five failure modes, while the *working* class corresponds to the *no failure* class of the failure type dependent variable.

In the two datasets, faults are defined by a single condition or multiple conditions: for AI4I 2020 dataset the *tool wear failure* consists of randomly selected tool wear time between 200 and 240 min, as showed in Fig. 3; the *heat dissipation failure* occurs when the heat dissipation is below 8.6 Kelvin [K] and the rotational speed is below 1380 revolution per minute [rpm]. For the electrical power system transmission lines dataset, the *fault between phase A and ground* only involves one phase that is shorted to ground; the *three phase symmetrical fault* considers all phases that are shorted to ground, as displayed in Figs. 4, 5.

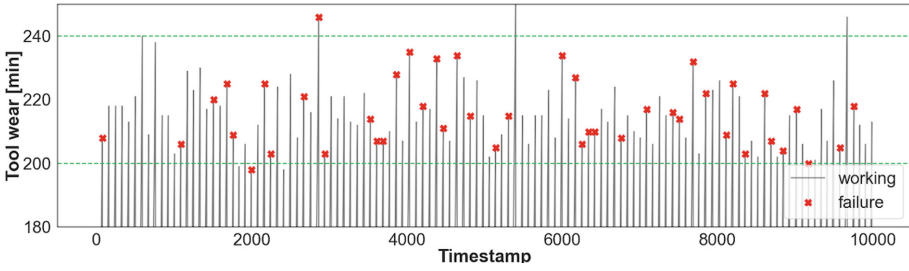


Fig. 3. AI4I 2020 dataset: tool wear failures

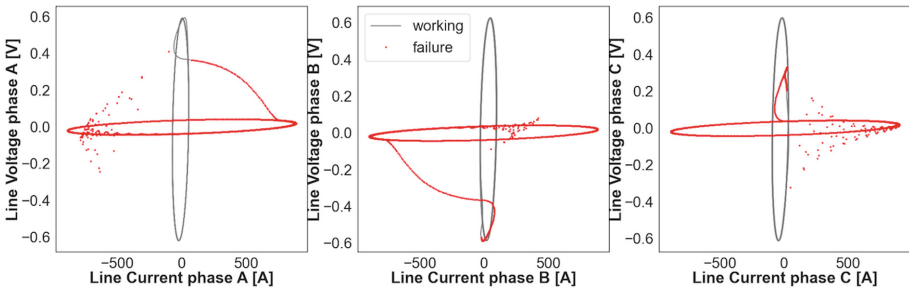


Fig. 4. Electrical power system transmission lines dataset: three phases symmetrical faults.

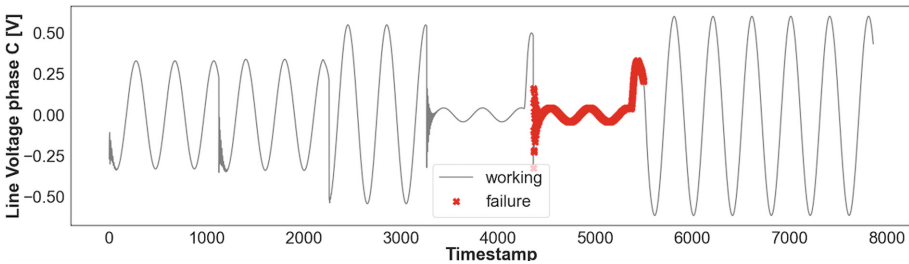


Fig. 5. Line voltage phase C: three phases symmetrical faults.

5 Methodology for Predictive Maintenance

The main objective of this study has been the development and validation of a predictive maintenance model for machine status and fault type detection by using different use cases. A generalized workflow is shown in Fig. 6 that schematizes the actions that have been undertaken to implement predictive maintenance model.

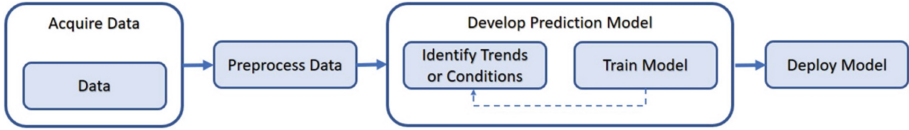


Fig. 6. Methodology overview

Our work has started with the datasets selection. We have continued processing data to prepare them for machine learning (ML) techniques, then we have built our ML-based prediction models and assessed them with respect to machine status and failure type tasks. The acquired data (already introduced in Sect. 3) are from AI4I 2020 predictive maintenance and electrical power system transmission lines. Considering the characteristics of data (such as the imbalance of data and single or multiple-failure conditions), we have decided to build multi-class and binary imbalanced classifiers. Four multi-output models have been analyzed and compared: two boosting ensemble classifiers - i.e., Adaptive Boosting and Gradient Boosting that differ in the way they learn from previous mistakes; one bagging ensemble classifier - i.e., Random Forest that trains a bunch of individual models in a parallel way without depending from previous model; and one neural network method - i.e., Multilayer Perceptron.

For all the classifiers an informed search method based on a Bayesian optimization algorithm has been applied to choose the best set of hyperparameters. Figure 7 shows details to build our models. For each model the pipeline is characterized by two phases: firstly we have found a combination of hyperparameters that best describes train data; and secondly we have deployed the model on unseen data by using the best settings found on the train set.

The first phase takes advantage of: an objective function, a range of values for each model's hyperparameter (i.e. hyperparameters spaces), a conjugate prior distribution for the hyperparameters (prior Knowledge) and an acquisition function, allowing to determine the next set of hyperparameters to be evaluated. At each step the search computes a posterior distribution of functions by fitting a Gaussian process model on the set of hyperparameters candidate and the objective function whose output have tried to maximize. The next set of hyperparameters to be evaluated is determined by the upper confidence bound acquisition function [27]. This function tries to explore regions in the hyperparameter spaces where we have been getting good performance based on previous search, rather than trying all possible combinations of hyperparameters. The optimization process [26] is repeated for a defined number of iterations, then the model is fitted on unseen data with the values of the hyperparameters leading to the global optimum of the objective function.

Despite the multi-output setting, the objective to be maximized in the Bayesian search [19] is only computed on the failure type. This choice wants to test the reliability of the hyperparameters selection computed on multi-class problem, even into binary problem. In extending a binary metric to multi-class problem, the data is covered as a collection of binary problems, one for each

class. We have considered the macro average of the metric. The macro average computes the metric independently for each class and then takes the average. The macro average of recall scores per class (macro-recall) [18] is taken into account for the maximization of the objective function. It is a satisfactory metrics able to deal with imbalance at different levels [29], as in our cases of the severe imbalanced AI4I 2020 dataset and the slight imbalanced electrical power system transmission lines dataset.

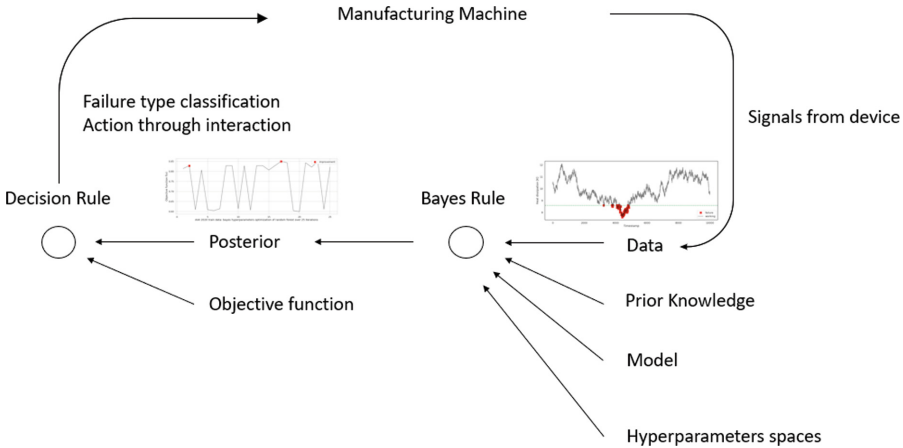


Fig. 7. Model pipeline

5.1 Algorithm Description

Let us summarize the used machine learning techniques together with the already introduced optimization strategy.

The Bayesian optimization approach fits probabilistic models for finding the maximum of the objective function. At each iteration, it exploits the probabilistic model, through the acquisition function, in order to decide where in the hyperparameters spaces next evaluates the objective function. It takes into account uncertainty by balancing exploration (not taking the optimal decision since the sets of hyperparameters explored are not sufficient to correctly identify the best set) and exploitation (taking the best set of hyperparameters with respect to the sets observed so far) during the search. The most commonly used probabilistic model for the Bayesian optimization solution is the Gaussian process due to its simplicity and flexibility in terms of conditions and inference [5].

Adaptive Boosting [11] is an algorithm that combines several classifiers into one superior classifier. This classifier begins by fitting a tree classifier on the original dataset and then fits additional copies of the classifier on the same dataset, where the weights of incorrectly classified instances are adjusted, so that subsequent classifiers focus more on difficult cases.

Gradient Boosting [7] is a family of powerful machine learning techniques that has shown success in a wide range of practical applications. It builds an additive model based on decision trees. The model fits a new decision tree on the residual errors that are made by the previous predictor. For this reason it learns from the residual errors directly, rather than update the weights of data points.

Random Forest (RF) classifier [12] is an ensemble learning algorithm based on bagging method of trees. It fits a number of decision tree classifiers using samples from the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Random Forest uses bootstrap sampling and feature sampling, and it considers different sets of regressors for different decision trees. Furthermore, decision trees, used in all ensemble models, are immune to multicollinearity and perform feature selection by selecting appropriate split points.

Multilayer Perceptron [20] is used as an artificial neural network classification algorithm. It trains using back propagation and supports multi-outputs by changing the output layer activator. The softmax activator is used with multi-class while the logistic activator is used with binary output. In the Bayesian optimization we have considered just one hidden layer. This option prevents overfitting since the small number of data points in the two datasets [14]. In addition, a simple architecture of the neural network can manage changes in the dependent variable from which it learns. We have specified the range for the number of neurons (hidden layer size) used in the Bayes optimization based on the number of regressors of the dataset. Ensemble learning algorithms are not sensible to scaling. However, the MultiLayer Perceptron is sensitive to feature scaling. In order to compare the performances with the same input data, we have applied a scaling on the regressors to obtain all values within the range $[0,1]$.

5.2 Training Phase

Despite data from manufacturing machine and transmission lines, registered at regular cadence, creates multivariate time series, we have not considered the temporal order of observations (data points) in the analysis. Our aim has been to discover configurations of regressors values connected to various maintenance problems. Data points (rows) are randomly located in the train or test set, generating two independent subsets. We have performed the hyperparameter selection, with the Bayesian optimization solution, in the train set by a cross-validation approach as shown in Fig. 8.

The training set is split into five smaller sets, preserving the percentage of samples for each failure type class. A model, with a given set of hyperparameters, is trained using four of the folds as training data and it is validated on the remaining fold. This fold is used to compute the performance measure: the macro-recall of the failure type (primary dependent variable). The performance measure reported by the five-fold cross validation is the average of the macro-recall values computed in the loop. Furthermore, it is the objective function to be maximized in the Bayesian optimization solution. The cross-validation approach

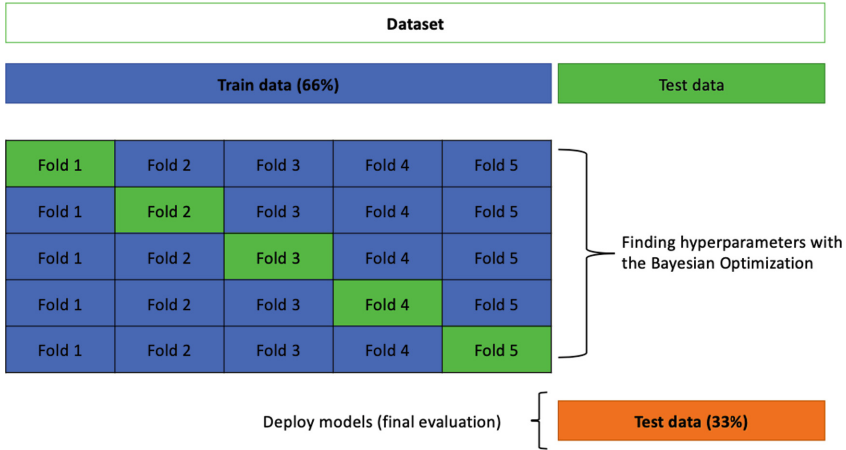


Fig. 8. Training phase description for both datasets.

can be computationally expensive: however, it is used along the Bayesian optimization method that is trustworthy for finding the maximum of an objective function (that is expensive to evaluate).

The 33% of the datasets have been used as test set to deploy our predictive maintenance models on failure type and machine (or electrical) status. Test data have been scaled by using minimum and maximum values of each regressor of the train set to simulate unseen data. In order to do a fair comparison among the models, we have used the Bayesian optimization approach with 20 iterations and 5 random explorations for each model to tune hyperparameters. The hyperparameters for each model are listed in Table 1, while the hyperparameters spaces are available online along with our code [25].

6 Results and Discussions

To evaluate the performances of the models we have taken into account the known threshold metrics for imbalanced datasets [13]. We extend binary metrics to multi-class problem, by taking the macro average of the metrics, as mentioned in Sect. 5. We have also reported confusion matrices for the models that perform better with respect to the failure type and machine (or electrical) status classification tasks in the two datasets.

Table 1. Best hyperparameters suggested by Bayesian optimization

Hyperparameter	Adaptive Boosting	Gradient Boosting	Random Forest	MultiLayer Perceptron
learning_rate	×	×		×
max_depth		×	×	
max_features		×	×	
n_estimators	×	×	×	
subsample		×		
activation				×
momentum				×
validation_fraction				×
batch_size				×
solver				×
early_stopping				×
max_iter				×
alpha				×
hidden_layer_size				×

In Figs. 9, 11 Adaptive Boosting shows poorest quality with respect to Gradient Boosting in the failure type classification. Among boosting ensemble classifier, both Adaptive Boosting and Gradient Boosting use an additive structure of trees. However the lack of previous tree, identified by high-weight data points (Adaptive Boosting), behaves worse than the residuals of each tree that step-by step reduce the error by identifying the negative gradient and moving in the opposite direction (Gradient Boosting).

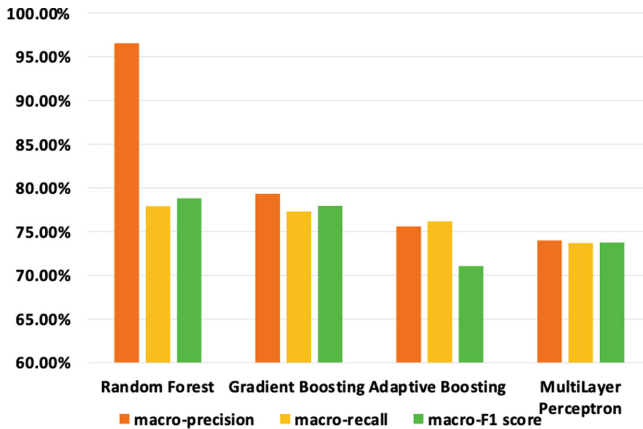


Fig. 9. Models performances on AI4I 2020 test data: failure type task

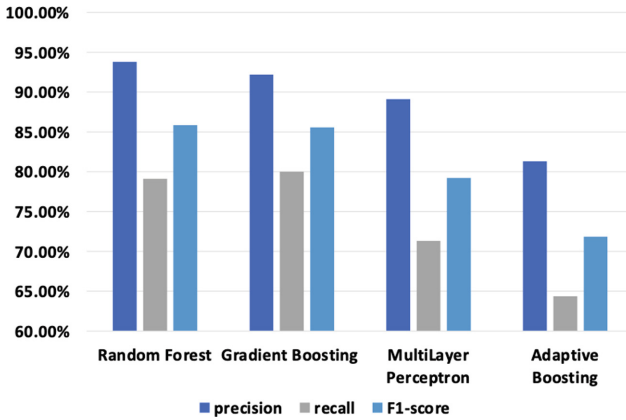


Fig. 10. Models performances on AI4I 2020 test data: machine status task

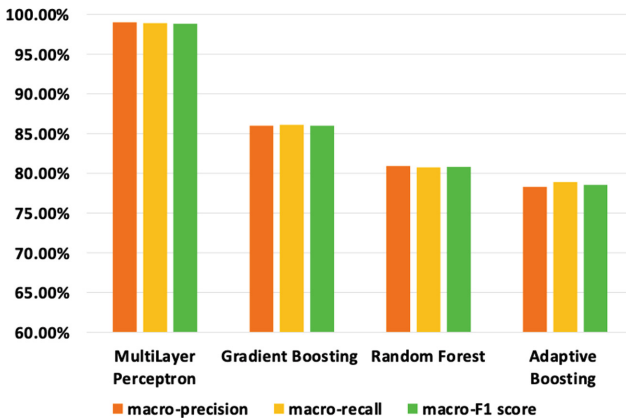


Fig. 11. Models performances on electrical power system transmission lines test data: failure type task

For both tasks, the Random Forest model performs better than other models in the AI4I 2020 case (severe imbalanced dataset), as shown in Figs. 9, 10. In our experimental setting, we have made a choice between two different ways of averaging results from trees. Which one of the two methods performs better is still an open research question based on bias-variance trade off. In our application, Random Forest combines each decision tree by computing the average of their probabilistic prediction [23]: for each data point, the ensemble method computes the average among trees of the probabilistic prediction of belonging in each class, and selects the class with the highest average probabilistic prediction. Some errors can be removed by taking the average of the probabilistic predictions. The model can achieve a reduction in variance by combining diverse trees at the cost of increasing the bias.

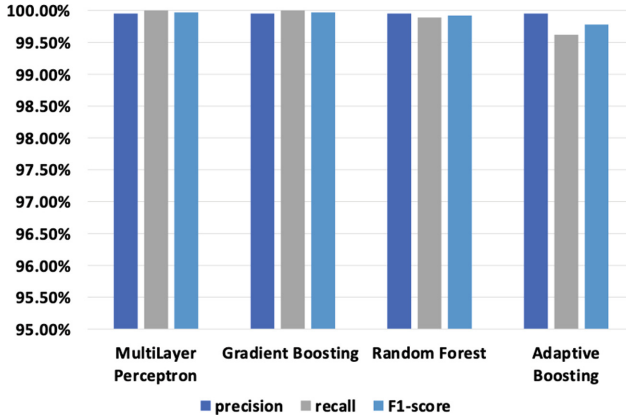


Fig. 12. Models performances on electrical power system transmission lines: electrical status task

Alternatively, each decision tree votes for a single class [4]: for a single data point each tree returns a class, the selected class will be the one with the highest absolute frequency among the trees. Notice that this combination can lead to a reduction in bias at the cost of greater variance.

In the failure type task, the MultiLayer Perceptron model performs better than other models in the electrical power system transmission lines case (slight imbalanced dataset), as shown in Figs. 11, 13. In this work MultiLayer Perceptron model is not superior to tree-based ensemble learning methods in the AI4I 2020 case (severe imbalanced dataset). It is not understood fully why neural network cannot reveal its superiority of predictive quality in presence of severe class-imbalanced tabular data. In this direction a recent work [3] has shown that tree ensemble still outperforms the deep learning models on tabular data. In addition, the MultiLayer Perceptron model has a large amount of hyperparameters, while tree ensemble models have fewer hyperparameters as displayed in Table 1. Therefore, the MultiLayer Perceptron model may need a higher number of iterations in the Bayesian optimization solution to produce predictive quality. Generally, a small number of hyperparameters is more desirable in order to improve the versatility and robustness of the model.

In the AI4I 2020 case, the *tool wear failure* is undetected in the four models with diverse level of evidence. It is unclear why the models is not able to properly recognize this class. We have identified two possible reasons: the number of observed failures in the dataset is quite small (see Fig. 1); this failures are defined as random failures within a large range of tool wear minutes as shown in Fig. 3.

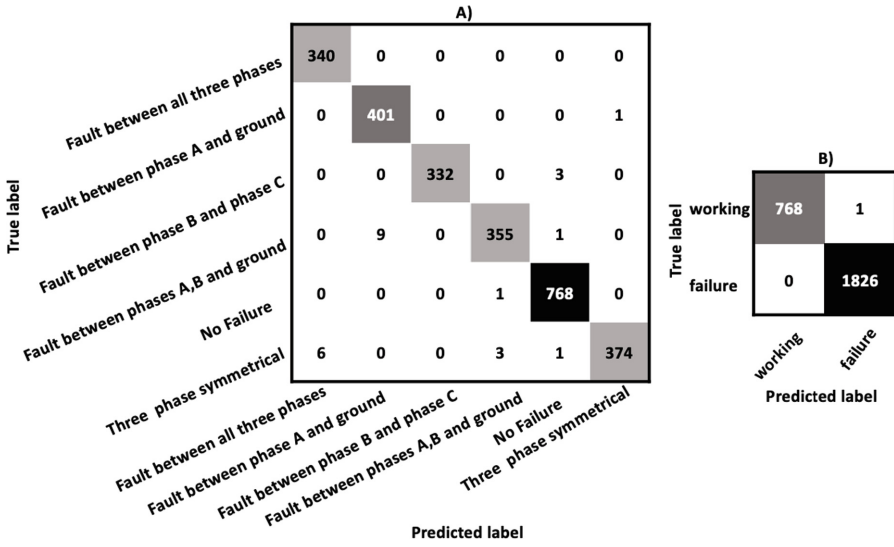


Fig. 13. Electrical power system transmission lines dataset: MultiLayer Perceptron performances, A) failure type task, B) electrical status task.

The proposed models present encouraging results not only for the multi-class problem (primary dependent variable), but also for the binary problem (secondary dependent variable) on imbalanced data, showing good generalization properties. The choice of selecting the best set of hyperparameters with the Bayesian optimization method on the multi-class problem, and fitting the model in the multi-output configuration, has proved to be reliable; firstly solving the failure type problem, secondly recognizing faulty and healthy status related to the first one. The electrical status is detected with outstanding quality in the electrical power system transmission lines case, where all models have excellent performances (see Fig. 12). Random Forest model identifies better the machine status than the other models in the AI4I 2020 case (Fig. 10), however its performance is biased by the *tool wear failures* that remain mainly not perceived (see Fig. 14).

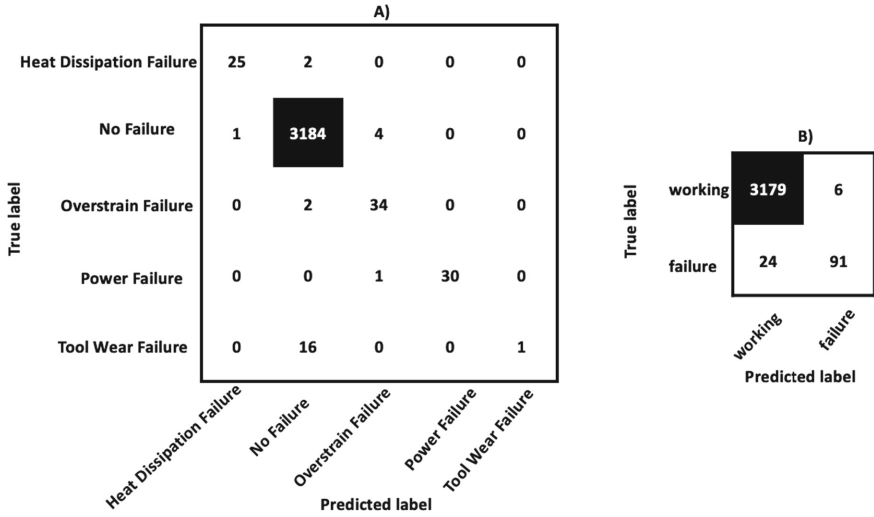


Fig. 14. AI4I 2020 dataset: Random Forest performances, A) failure type task, B) machine status task.

7 Conclusions

In this paper we present a pipeline based on the Bayesian optimization solution for imbalanced datasets in order to design and develop multi-class and binary imbalanced classifiers. We have considered four models, such as Adaptive Boosting, Gradient Boosting, Random Forest and MultiLayer Perceptron. The Bayesian optimization approach has been used to determine the best combination of hyperparameters for model variables to improve machine learning performances. This pipeline has been applied to two different imbalanced datasets, such as AI4I 2020 and electrical power system transmission lines, to classify two connected tasks: failure type and machine (or electrical) status.

Our work includes promising evidence on the ability of models (whose hyperparameters are tuned in the Bayesian optimization method on the failure type variable) to perform well not only on the failure type classification but also on the machine (or electrical) status by using a multi-output configuration.

In the AI4I 2020 case, the Random Forest model performs better than other models for both tasks. In the electrical power system transmission lines case, the MultiLayer Perceptron model performs better than others for the failure type task. Despite the different domain of the datasets, we have observed the tendency of tree ensemble models to succeed on the severe imbalanced dataset, the liability of the MultiLayer Perceptron model to succeed on the slight imbalanced dataset. This work can serve as a valuable starting point for data scientists and practitioners interested in predictive maintenance solutions with imbalanced tabular data.

Acknowledgements. We would like to thank [BitBang S.r.l](#) that funded this research; in particular Matteo Casadei, Luca Guerra and the colleagues of Data Science team.

References

1. Amruthnath, N., Gupta, T.: A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), pp. 355–361 (2018)
2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**, 281–305 (2012)
3. Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep neural networks and tabular data: a survey. arXiv preprint [arXiv:2110.01889](#) (2021)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Brochu, E., Cora, V.M., De Freitas, N.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint [arXiv:1012.2599](#) (2010)
6. Calabrese, M., et al.: SOPHIA: an event-based IoT and machine learning architecture for predictive maintenance in Industry 4.0. *Information* **11**(4), 202 (2020). <https://www.mdpi.com/2078-2489/11/4/202>
7. Cao, H., Nguyen, M., Phua, C., Krishnaswamy, S., Li, X.: An integrated framework for human activity recognition. In: 2012 ACM Conference on Ubiquitous Computing, pp. 621–622 (2012)
8. Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., P. Francisco, R., Basto, J.P., Alcalá, S.G.S.: A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **137**, 106024 (2019)
9. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
10. Ghate, V.N., Dudul, S.V.: Optimal MLP neural network classifier for fault detection of three phase induction motor. *Expert Syst. Appl.* **37**(4), 3468–3481 (2010)
11. Hastie, T., Rosset, S., Zhu, J., Zou, H.: Multi-class AdaBoost. *Stat. Interface* **2**(3), 349–360 (2009)
12. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. SSS, Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
13. He, H., Ma, Y.: *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, Hoboken, 216 pages (2013)
14. Heaton, J.: *Introduction to Neural Networks with Java*. Heaton Research, Inc. Chesterfield (2008)
15. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A.C. (ed.) *LION 2011*. LNCS, vol. 6683, pp. 507–523. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25566-3_40
16. Jamil, M., Sharma, S.K., Singh, R.: Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus* **4**(1), 1–13 (2015). <https://doi.org/10.1186/s40064-015-1080-x>
17. Martin-Diaz, I., Morinigo-Sotelo, D., Duque-Perez, O., de J. Romero-Troncoso, R.: Early fault detection in induction motors using AdaBoost with imbalanced small data and optimized sampling. *IEEE Trans. Ind. Appl.* **53**(3), 3066–3075 (2017)
18. Mosley, L.: A balanced approach to the multi-class imbalance problem. Ph.D. thesis, Iowa State University (2013)

19. Nogueira, F.: Bayesian Optimization: open source constrained global optimization tool for Python (2014). <https://github.com/fmfn/BayesianOptimization>
20. Orrù, P.F., Zoccheddu, A., Sassu, L., Mattia, C., Cozza, R., Arena, S.: Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. *Sustainability* **12**, 4776 (2020)
21. Ouadah, A., Leila, Z.-G., Salhi, N.: Selecting an appropriate supervised machine learning algorithm for predictive maintenance. *Int. J. Adv. Manufact. Tech.* **119**, 4277–4301 (2022)
22. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., Loncarski, J.: Machine learning approach for predictive maintenance in Industry 4.0. In: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6. IEEE (2018)
23. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
24. Qin, S., Wang, K., Ma, X., Wang, W., Li, M.: Chapter 9: step standard in design and manufacturing ensemble learning-based wind turbine fault prediction method with adaptive feature selection. *Communications in Computer and Information Science Data Science*, pp. 572–582 (2017)
25. Ronzoni, N.: Predictive maintenance experiences on imbalance data with Bayesian optimization. https://gitlab.com/system_anomaly_detection/predictivemeintenance
26. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inform. Proc. Syst.* **25**, 2951–2959 (2012)
27. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.: Gaussian process optimization in the bandit setting: no regret and experimental design. *arXiv preprint arXiv:0912.3995* (2009)
28. Susto, G., Schirru, A., Pampuri, S., Mcloone, S., Beghi, A.: Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Trans. Ind. Inf.* **11**, 812–820 (2015)
29. Urbanowicz, R.J., Moore, J.H.: ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. *Evol. Intell.* **8**(2), 89–116 (2015)
30. Vasilić, P., Vujnović, S., Popović, N., Marjanović, A., Željko Durović: Adaboost algorithm in the frame of predictive maintenance tasks. In: 23rd International Scientific-Professional Conference on Information Technology (IT), IEEE, pp. 1–4 (2018)