

# Feature Extraction and Failure Detection Pipeline Applied to Log-based and Production Data

Rosaria Rossini<sup>1</sup><sup>a</sup>, Nicolò Bertozzi<sup>1</sup><sup>b</sup>, Eliseu Pereira<sup>2</sup><sup>c</sup>, Claudio Pastrone<sup>1</sup><sup>d</sup>  
and Gil Gonçalves<sup>2</sup><sup>e</sup>

<sup>1</sup>LINKS Foundation, Turin, Italy

<sup>2</sup>SYSTEC, Research Center for Systems and Technologies, Faculty of Engineering, University of Porto, Porto, Portugal

**Keywords:** Predictive Maintenance, Machine Learning, Feature Engineering, Manufacturing, Log Data, Drilling.

**Abstract:** Machines can generate an enormous amount of data, complemented with production, alerts, failures, and maintenance data, enabling through a feature engineering process the generation of solid datasets. Modern machines incorporate sensors and data processing modules from factories, but in older equipment, these devices must be installed with the machine already in production, or in some cases, it is not possible to install all required sensors. In order to overcome this issue, and quickly start to analyze the machine behavior, in this paper, a two-step log & production-based approach is described and applied to log and production data with the aim of exploiting feature engineering applied to an industrial dataset. In particular, by aggregating production and log data, the proposed two-steps analysis can be applied to predict if, in the near future, I) an error will occur in such machine, and II) the gravity of such error, i.e. have a general evaluation if such issue is a candidate failure or a scheduled stop. The proposed approach has been tested on a real scenario with data collected from a woodworking drilling machine.


## 1 INTRODUCTION


Industry 4.0 paradigm aims to improve the plant level of a factory by the means of different technology assets, such as Internet of Things (IoT) sensors, Artificial Intelligence (AI), data integration and aggregation, and so on and so forth. Industry 4.0 brings the advantage of knowing better and in detail both processes and machines involved in the production. This advantage creates the possibility to not only knowing and monitoring the plant but also to improve the process as well as the life and the work of the machine. In this context, RECLAIM<sup>1</sup> positions itself as a project that has the goal to increase productivity, extending the lifetime of the machines and reducing the time and cost of machinery refurbishment and/or re-manufacturing. This objective will be achieved designing and developing a set of tools supporting sev-


eral activities: from the monitor of machines' health status, to the implementation of adequate recovery strategy (e.g., refurbishment, re-manufacturing, upgrade, maintenance, repair, recycle, and so on and so forth). As RECLAIM shows, most of the time, it is possible to improve the life of the machine and its performance by improving the maintenance schedule and/or manage the working time.


In order to do that, the authors present a two-steps approach for machine diagnostics, based on log and production data, that can predict and analyze the failures of the monitored machine. The goal is to apply a pipeline with steps that include data cleaning, feature extraction, and predictive tasks to an industrial dataset without using sensors data. After a preprocessing step used to prepare the data and create the input features, the classification algorithm predicts a failure will happen in the next prediction windows (PW), using the features present in the observation windows (OW). The next component of the pipeline is a severity estimation model that computes the level of gravity of the predicted failure.


The strength, novel, innovative and convenient aspect of this work is the possibility to do not install sensor data for the failure prediction, but using

<sup>a</sup> <https://orcid.org/0000-0003-2029-7086>

<sup>b</sup> <https://orcid.org/0000-0003-2049-9876>

<sup>c</sup> <https://orcid.org/0000-0003-3893-3845>

<sup>d</sup> <https://orcid.org/0000-0003-0471-8434>

<sup>e</sup> <https://orcid.org/0000-0001-7757-7308>

<sup>1</sup><https://www.reclaim-project.eu>

only historical machine logs and production indicators. In particular, the approach enables health monitoring and the prediction of failures on older equipment without the need of install new sensors. Those attributes permit companies to reduce costs buying new monitoring components and speeds the process of analyzing the machine behavior and deploying the predictive solution, because the monitoring system (of log and production data) was on the machine since the beginning of its operation generating historical data.

The paper is organized as follows: in Section 2, the authors introduce a literature review about predictive maintenance and fault diagnosis. Section 3 describes briefly the scenario in which the application is described as well as the data available for it. Section 4 presents the core solution presented in this paper. Finally, Section 5 shows the results and Section 6 concludes the paper by summarizing and discussing the work.

## 2 BACKGROUND

Nowadays modern machines are able to monitor a large set of parameters, variables or indicators. The production data is useful to build analytical solutions, such as decision support systems or predictive maintenance solutions (Rosaria et al., 2021).

Among the operational data, machine failures and alarms are some of the data sources most common in the shop-floor. In fact, the PLCs continuously produce this log information about the machine, including also internal events, warnings, alarms, errors, machine or components status or cycles. Logs are generated automatically at a very high rate, daily, hourly, and contains timestamps about the information that is reported. These log data can be stored into databases or files, providing valuable information for machine diagnostics (Xiang et al., 2018). Those diagnostics algorithms can include degradation models or log-based predictive maintenance (Gutschel et al., 2019), (Wang et al., 2017). Despite of the structure of the log file, managing these information can be an important for extracting information about different aspect of the machine production. As it is possible to see in Section 3 log files can be also be involved in the failure prediction.

## 3 DATASET

The data used for this work is from a woodworking drilling machine described in detail in the subsection

3.1. That machine generates two different types of data, 1) event log data, and 2) production data, which are described in the subsection 3.2.

### 3.1 Scenario

The machine of interest is a woodworking drilling machine (Brema VEKTOR15), composed of a set of drill bits, divided into two spindles. The total number of different drills is about 40/50. The life, in hours, of a drill bit depends on multiple factors, such as the hardness and wood quality. The quality of the material depends on the suppliers and on what is indicated in the specifications of the purchased wood. For instance, the percentage of presence of metal residues in the chipboards.

The shape of the drilled hole and the noise emitted by the saw in case of cutting are good indicators about the health of these tools. Due to the difficulty in getting these measurements from the machinery, normally the drill bits are substituted or at regular intervals or thanks to the operator's experience.

### 3.2 Exploratory Data Analysis

The dataset used to design the pipeline is composed of two parts: 1) the production data and 2) the log data. The first one contains all the articles produced, and the second one all the events occurred in the machine. The extensions of those documents are .ter and .btk, which are a particular type of text files, exported/generated by the machine.

#### 3.2.1 Production Data

The production dataset, in Figure 1a, contains all the pieces of wood worked in a particular time interval. The description of the columns is the following: 1) "Programma", file that contains all the drilling operations that must be made on the piece, 2) "Commento", details about the drilling, 3) dimensions of the board, L for length, H for height, and S for width, and 4) starting and ending time of the two working phases (Start1, End1, Start2, End2).

One of the goals of this preliminary statistical analysis is to evaluate to what degree of the working time is influenced by the material (type of wood such as poplar, ebony, walnut, etc.), the dimensions of the board and the number of drills. A plausible starting point is represented by the computation of derivative variables like the volume, which integrates together the length, the height and the width, and the time intervals T1, T2 and INT. Instead, the value of T1 is the difference, in seconds, between End1 and Start1, T2

	Programma	Commento	L	H	S	Start1	End1	Start2	End2
0	CUCINE\ARMADI\H00_C TRAVETTI.TCN	R. Schenker: (05.02.2019)Programma (per Spalla...	2032	600	16	05/02/2021 07:24:08	05/02/2021 07:25:06	05/02/2021 07:25:08	05/02/2021 07:25:58
1	CUCINE\ARMADI\H00_C TRAVETTI.TCN	R. Schenker: (05.02.2019)Programma (per Spalla...	2032	600	16	05/02/2021 07:25:10	05/02/2021 07:26:56	05/02/2021 07:26:58	05/02/2021 07:27:51

(a) Production data.

	Time	Description	Type	Code	Task	Status	ModAddr	Module	GroupCode
0	2021-01-11 06:28:00	Albatros inizia l'esecuzione (3.1.10 Service P...	Done	1	NaN	NaN	NaN	NaN	NaN
1	2021-01-11 07:15:00	Attuale livello di accesso: Service	Done	6	NaN	NaN	NaN	NaN	NaN

(b) Log data.

Figure 1: Dataset preview including machine production (a) and log data (b).

the difference between End2 and Start2 and, finally, INT the difference between Start2 and End1.

The VEKTOR15 machine executes the drilling of wood boards. Thus, the time required by the machine to perform those holes is linked to the number of holes. In this view, variables T1 and T2 could be directly connected to the number of operations performed by the VEKTOR15, and consequently to the product categories. Additionally, variables T1 and T2 are completely uncorrelated, which means that the first production phase does not give any information about the time required by the subsequent stage.

### 3.2.2 Log Data

The log dataset, in Figure 1b, contains all the errors emitted by the machine in a particular time interval. The description of the columns is the following: 1) timestamp of the error (time), 2) details about the error (description), 3) type, which includes three possible categories of error (Cycle, Done and System), and 4) additional details about the error (Code, Task, Status, ModAddr, Module, GroupCode).

In this case, the analysis is oriented to extract the most serious stops, to retrieve a general pattern that is specific-independent and to design a machine learning model that can predict future possible errors. The only type of error that causes a stop of the machine is the "System". Then, all other logs can be discarded.

The available information about the specifications of the error are in the field "Description". Summarizing the description into a sentence with a reduced number of words and without the redundant indication of the device numbers makes the analysis easier and more interpretable. For instance, the error "XX: Il servoazionamento YY non è collegato" has a unique error code, independently of the value of XX and YY. After, the computation the final list of errors includes 19 types of errors.

Figure 2a shows the entire production and the entire generated log during 10 months of operation. It is possible to notice that each production block is periodic, inline with the definition of a working week.

The space between each block represents a weekend. Inside each block there are five smaller rectangles that indicates a working day as well as in the first two weeks of August, due to the summer holidays, and in some days over the year, due to the festivities. Last, there is an absence of errors in some periods, e.g., February and May.

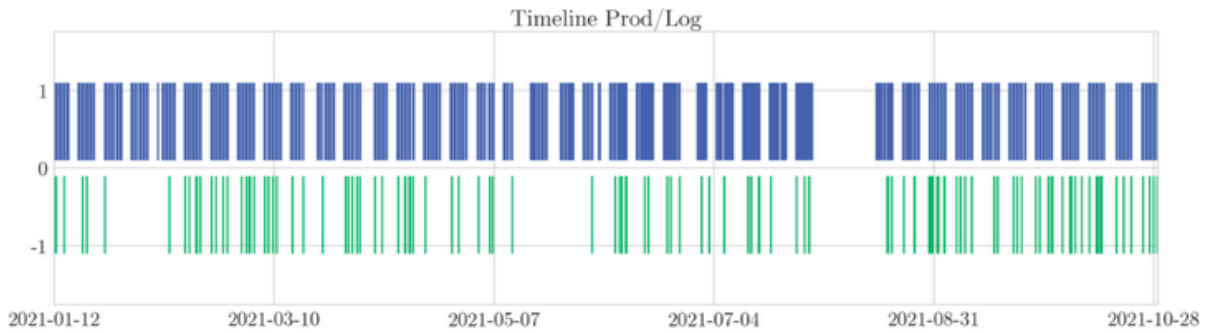
The distribution and the occurring time of the error is import and by observing the data, it is possible to see that, multiple kind of errors (log samples with different description/code) occur at the same time. For our purpose, it is more important the chance of the error to occur more than the type of error and, in a second instance, the severity of it, i.e., its duration. The machine learning models that will reach this goal will be based on a set of production and error indicators, trying to find a causality between the time-production and the insurgence of errors.

## 4 APPROACH

The analytical pipeline presented in this work, pre-processes the data, extracting essential features from both data sources (log events and production data), computes observation, and prediction windows, and feeds a binary classification algorithm, that will predict if the class label of the current input sample is equal to 1 (stop) or 0 (normal operation). In case of a predicted stop, the model is triggered for the computation of the severity (time to repair), that has four levels of gravity.

### 4.1 Feature Extraction

Some of the features used to train the models are the production indicators. Figure 3 shows the list of the most produced categories in 9 months. The column "Count" indicates the number of drilled boards; the "Ratio" reflects the percentage of production assigned to each category. From Figure 3, it is possible to cover the 95% of production by summing only a



(a) Production and logs in the temporal line.



(b) Production and system errors in an extremely limited period of time, about 30 minutes.

Figure 2: Production and logs temporal line, with the entire period (a) and a period of 30 minutes (b).

small subset of categories, like “PG”, “BASI”, “PENSILI”, “ARMADI”, and “BASI RINDERKNECHT [...]”. This percentage is the threshold used to decide which categories to include as features. After setting these variables, the production indicators for each Observation Window (OW) include the cumulative number of drilled boards, together with the starting time of the production.

Programma	Count	Ratio
PG	2640	0.310661
BASI	2533	0.298070
PENSILI	2500	0.294187
ARMADI	215	0.025300
BASI RINDERKNECHT AG H=782	129	0.015180
ARMADI CON CASSETTI INTERNI	74	0.008708
FONDI	62	0.007296
BASI BORA	50	0.005884
MOBILI CEPPI 100 MM	46	0.005413
BASI PER FORNO	23	0.002707
ARMADI SPECIALI	23	0.002707
BASI DIVISIONE CASSETTI SPECIALE	23	0.002707
BOILER	22	0.002589

Figure 3: Number of drilled boards for a subset of categories.

Figure 4 shows a typical subset of possible production indicators with an OW with size 24, i.e., a temporal window of 30 minutes. Each row of the production dataset represents 12 working hours (the sum of

the preceding 24 row).

The log data pass through the same procedure, using a temporal region defined precisely as the production one. Given the number of errors in each window (30 minutes), the cumulative sum is computed to determine a historical characterization of the errors over time. This approach allows the model to predict, considering the MTTF of each log category. The error features are appended to the production ones as part of the input of the classification model. Those features are essential for the classification model because stops and failures occur periodically or after a certain number of produced articles.

The next step is to incorporate the working hours and days into the prediction pipeline, allowing the operational interpretation of the model results. For instance, if at the 17:40 of a working day, the model predicts a failure in the next three temporal windows (1.5 hours) corresponds to a failure prediction for the first 1.5 hours of production of the next day. Removing no working hours and days is also essential to guarantee a balancing between the classes labels in the dataset. Usually, the working day begins at 6, included, and ends at 18, excluded. Additionally, the intervals between 6-7, 12-13, and 17-18 are characterized by low values of productivity and failures.

The number of errors and produced items allows the computation of two metrics, that are essential to evaluate the importance of each time interval: 1) the sum of produced items and errors (PE indicator), and 2) the ratio of errors in the production (the number of errors divided by the sum of errors and production) (ER indicator). A low value of both indicators is a

	BASI BORA	PG	BASI	ARMADI	ARMADI SPECIALI	PENSILI	FONDI	ALTRO
2021-01-13 06:00:00	2.0	29.0	90.0	12.0	0.0	109.0	0.0	3.0
2021-01-13 06:30:00	2.0	29.0	90.0	12.0	0.0	109.0	0.0	3.0
2021-01-13 07:00:00	2.0	29.0	90.0	12.0	0.0	109.0	0.0	3.0

Figure 4: Example of some production indicators.

synonym of low interest in such time intervals.

Table 1: List of the working indicators for each hour.

Interval	PE	ER	Risk
5-6	0	0	No
6-7	461	4.77	Yes
7-8	8308	32.64	Yes
8-9	8045	35.11	No
9-10	6401	37.45	No
10-11	9737	37.99	No
11-12	7912	32.42	No
12-13	276	0	Yes
13-14	6543	30.92	No
14-15	9519	47.31	No
15-16	8252	40.52	No
16-17	6608	44.87	No
17-18	816	57.72	Yes
18-19	0	0	No

Table 1 presents the values of the PE and ER indicators. From the table, it's possible to infer to 1) drop the interval 12-13 due to the low value of PE and ER, 2) maintain the interval 17-18 due to an extremely high value of ER, and 3) consider if it is more convenient to use also the samples contained in the interval 6-7, because of the lower value of PE and ER. The removal of the interval 6-7 causes a drop in the model's precision mainly because the samples present in that interval represent the initial base in predicting the failures of the next time intervals. The low values in the interval 12-13 are due to the lunch break, which is not zero because there are multiple lunch shifts; this also supports the lower value of production in the interval 13-14.

The filtering process also contemplates the not working days. Those days include the summer holidays and some festivities that cause the closure of the plant and the consequent stop of production. In addition, if some of these days fall on Tuesday or Thursday, for instance, usually also the working days near to the weekend, respectively Monday and Friday, could be candidates for closure. Additionally, this filtering process permits lightening the unbalance of the dataset.

## 4.2 Binary Classification Model

The binary classification model must predict if there will be an error in the future time frame, given the production, the actual time, and the historical list of errors. The prediction model will implement a classification algorithm, where the class label equal to 1 corresponds to the prediction of failure in the future time frame, and a class label of 0 to the normal operation of the machine (no failure).

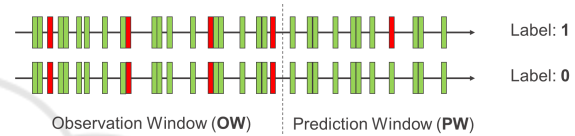


Figure 5: Observation and prediction windows, with different values of class label.

The prediction refers to a limited time slot, which introduces the concepts of observation window and prediction window. Figure 5 represents the two different typical scenarios (prediction of failure or normal behavior), where a temporal window is a fixed-sized period. The set of windows used to make a prediction are the observation window (OW); the ones associated with the prediction form are the prediction window (PW). Then, given a series of OWs, the model predicts if an error will occur between the actual time and the end of the PW. The optimal configuration is characterized by a low value of PW because gives a final precision very confined in time. However, this introduces a trade-off between the precision (number of prediction windows) and the model's performance. So, to reach high performance, it is necessary to forecast the error with low precision. The model results show that trade-off and compare different models with different values of OW and PW.

The classification algorithms tested are the Random Forest (RF) and the k-Nearest Neighbour (k-NN). The RF has a good performance in classification problems. The k-NN performs well in clustered samples, where the distance metric easily separates the data samples. Looking at the samples distribution, considering the high level of isolation of label items equal to 1, the suitable algorithm is the k-NN. The hyperparameters are tuned for the entire pipeline, including the preprocessing and the algorithm compo-

nents. Regarding the preprocessing, the hyperparameters to tune are the number of observing and prediction windows, OW and PW, and the test size. For the classification model, the hyperparameters tested are the number of estimators of the RF and the number of neighbors of the k-NN. The approach used to the hyperparameter tuning was the k-fold cross-validation. In datasets unbalanced, like this one, the adoption of the accuracy as the evaluation metric is not ideal due to the weight of the majority class. In this manner, due to the high number of 0-labeled samples, the accuracy can reach the same percentage of these samples concerning the total number of rows of the dataset.

### 4.3 Stop Severity Estimation

The prediction of the error severity consists of estimating the gravity of the VEKTOR15 stops and failures. The model presented provide information about the gravity of the failure. The computed information is useful to understand if the predicted stop will be a short stop or a failure that causes the stoppage of the machine for several days. Figure 6 illustrates a typical log cluster of a drilling machine. The production samples have both the starting and the ending timestamp, while the log samples reported only the indication of the timestamp when the error occurred. For this reason, it is not directly extracted from each cluster of events an accurate value of duration or time to repair (TTR) in case of failure. The log cluster includes the events (second row) between the production interruption and its restart. The computation of each cluster duration considers the first event as starting timestamp and the last one (restart of machine) as the ending timestamp. In this way, it's possible to approximate the stop duration and associate a severity to each stop. There are four levels of failure severity, 1) no failure (label 0), and 2) three incremental values of failure severity (labels 1, 2, and 3). The association between the failure severity and the event cluster duration is performed using different thresholds. Those thresholds come from the 25th, 50th, and 75th percentiles of the cluster duration distribution. The subdivision in the percentage of labels is 30% label 1, 45% label 2, and 25% label 3, presenting all labels a good value of balancing (close to 33%).



Figure 6: Definition of cluster duration. The faded orange box indicates the uncertainty generated by the process.

Once each cluster has associated one severity

class, each temporal window has to be labeled depending on the severity of the clusters it contains. Windows containing more than one cluster, the label associated is the one of the cluster with higher severity. After the computation of the new labels, the classification model passes through the same procedure as the one done in binary classification, being fed by a series of OW and predicting the severity label of the PW. The algorithms tested were the RF and the k-NN. The binary classification model and the stop severity failure are complementary because they provide different indicators and are specialized in different tasks, as show the results further ahead.

## 5 EXPERIMENTS & RESULTS

As mentioned before, the entire pipeline is tested using the k-fold cross-validation. Different hyperparameters are experimented for both classification models and OW and PW.

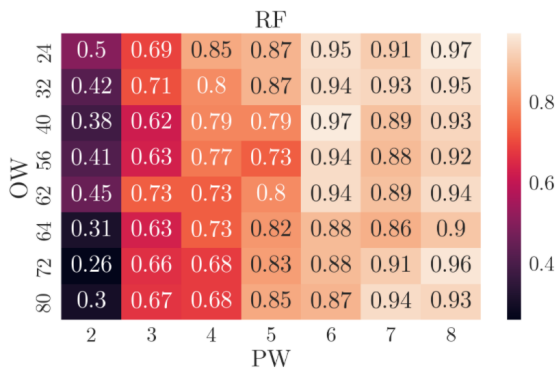
### 5.1 Stop Detection Results

The f1-score is the used metric because it is the harmonic mean of precision and recall and can be used as a general indicator.

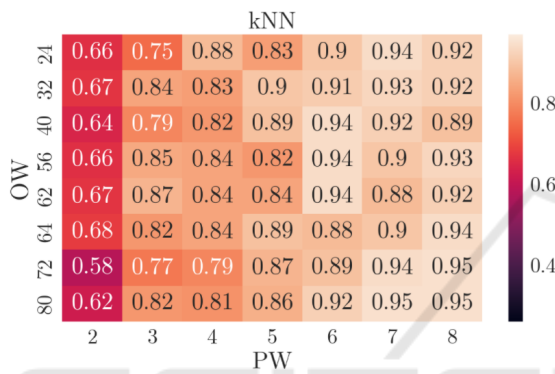
Figure 7 reports the results obtained with the Random Forest and the k-Nearest Neighbour classifier with different observation and prediction windows. The optimal classification model will be the one that reaches the best performance in terms of f1-score, recall, and with the lowest number of prediction windows. Besides the number of PW, the decision criteria between the different models instances will be the arithmetic mean between the f1-score and the recall of class 1. After analyzing Figure 7, it's noticeable that the pairs of (OW, PW) which satisfy the previous objective function are (24, 4) for the RF and (62, 3) for the k-NN.

Table 2 compares the two classifiers' best results, allowing the selection of the best one to be used in production and on-site. Due to the high level of unbalance, the indicators related to class 0 are not significant for that analysis. On the other hand, the recall, precision, and the f1-score of class 1 are more meaningful. As mentioned before, the usage of those metrics reduce false positives (FP) and false negatives (FN).

It is observable that the k-NN results are slightly better than the ones obtained by the RF because the recall is higher for the minority class (1-labelled), which reduces the number of unpredicted failures (type II error). The level of f1-score is slightly lower in the k-



(a) Results for the Random Forest (RF).



(b) Results for the k-Nearest Neighbour (k-NN).

Figure 7: F1-score results obtained by the Random Forest (a) and k-Nearest Neighbour (b) in the binary case.

NN when compared to the RF; however, this decrease is due to the higher precision of the RF. Since the type I errors (false alarms) have less influence on the shop-floor than the type II errors, the selected algorithm should be the one with higher recall; even with a slight decrease of f1-score. Additionally, using a higher observation window (OW) allows the model to consider more historical information, including more failure patterns. Those considerations imply that the k-NN should be the selected classification model to execute on production because of the performance metrics and the selected OW and PW.

### 5.2 Severity Prediction Evaluation

As in the binary task, the accuracy is not the ideal performance metric for evaluating the severity model due to the dataset imbalance. For this reason, the corrective coefficient used is the goodness ( $g$ ) defined in Equation 1, where  $\hat{y}$  is the predicted class label,  $y$  the real class label, and  $w$  the weights assigned to each class. The goodness is adopted as corrective coefficient because the difference between the predicted severity labels is crucial, i.e., it is not the equivalent

Table 2: Classification reports obtained with the Random Forest (a) and with the k-Nearest Neighbour (b).

Class	Precision	Recall	F1	Support
0	0.98	0.99	0.99	625
1	0.90	0.84	0.87	67
macro avg	0.90	0.84	0.87	692
weight avg	0.97	0.98	0.98	692

(a) Random Forest results with OW=24 and PW=4.

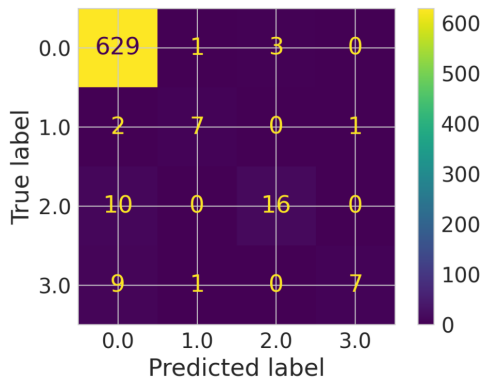
Class	Precision	Recall	F1	Support
0	0.99	0.98	0.99	633
1	0.81	0.89	0.85	53
macro avg	0.90	0.93	0.92	686
weight avg	0.98	0.98	0.98	686

(b) k-Nearest Neighbour results with OW=62 and PW=3.

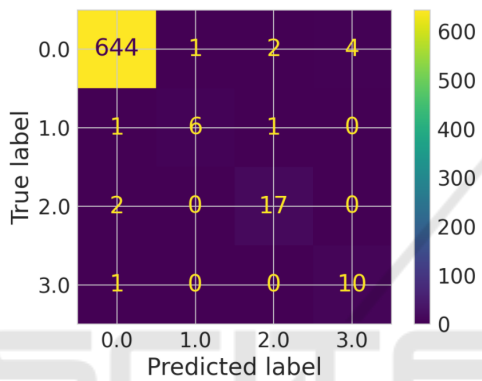
predict a 0 or a 2 when the real class is equal to 3. This situation reflects in the staff being prepared for a small maintenance of the machine (predicted label 2) or not being prepared at all (predicted label 2) when a failure of high severity will happen (real label 1.)

$$g(y, \hat{y}, w) = \frac{1}{\sum w_i} \sum_{i=1}^m w_i \frac{\sum_{j=1}^i P(\hat{y} = i | y = j)}{\sum_{j=1}^n P(\hat{y} = i | y = j)} \quad (1)$$

A low value of goodness means a high level of misclassifications; instead, a high value of goodness indicates a minimal presence of critical situations like the one described above. The classification algorithms experimented for this task were the k-NN and RF, where the k-NN obtained better results than the RF, practically for each pair of windows. Figure 8 reports the confusion matrices for the optimal configurations of OW and PW after applying the goodness as the corrective coefficient. As a correction coefficient, the goodness minimizes the distance between the predicted severity and the real class. That effect is noticeable in the confusion matrices, particularly in the k-NN one, where the algorithm accurately predicted the severity of the failure almost every time. So, the severity prediction model has an excellent performance estimating the gravity of the stop; however, it has lower performance when it comes to detecting if it's a failure or not (high number of FN and FP in the label 0). That issue is addressed by using the binary classification model to predict if there is a stop or not, and after, execute the severity model to estimate the gravity of the expected stop.



(a) Confusion matrix for the Random Forest (RF) with OW=24 and PW=4.



(b) Confusion matrix for the k-Nearest Neighbour (k-NN) with OW=62 and PW=3.

Figure 8: Confusion matrices obtained with the Random Forest (a) and with the k-Nearest Neighbour (b).

## 6 CONCLUSIONS & FUTURE WORK

The work in this paper developed provides a valuable pipeline for Prognostic and Health Management (PHM). The pipeline was applied to a dataset generated by a woodworking drilling machine (Brema VEKTOR15). That data includes machine log events (alarms, stops, failures) and production data (produced pieces, including product type or working time). The analytical pipeline preprocessed that data, extracting essential features from both data sources, computed observation, and prediction windows, and feeding the binary classification algorithm, which if predicts a stop triggers the model for the computation of the severity (time to repair). Those indicators provide essential information for the maintenance team, mainly operational insights about when a failure will occur and its impact. The usage of the severity model provides essential insights to the operators because

it informs them if the predicted stop has a higher or lower impact, which traduces in having a short stop or the failure that could stop the machine for days.

The evaluation of the models results in the selection of the k-NN algorithm for the binary classifier (with OW=62 and PW=3) and severity predictor (with OW=62 and PW=3). The excellent performance of the k-NN for the two different tasks results from the same input data that feeds each one of the models.

As future work, the goals pass through applying the failure predictions (including severity) to decision support systems for the machine life and product quality optimization. Another goal will be to validate the algorithms using a simulation environment that emulates the industrial shop floor. Finally, within the project, it's planned to apply this pipeline to other industries like textile and white goods manufacturers.

## ACKNOWLEDGMENT

The work presented here was part of the project "RECLAIM- RE-manufaCturing and Refurbishment LARge Industrial equipMent" and received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869884. The authors thank PODIUM SWISS SA for providing the data used in this paper and Asia Savino for the support on data validation.

## REFERENCES

- Gutsch, C. et al. (2019). Log-based predictive maintenance in discrete parts manufacturing. *Procedia CIRP*, 79:528–533. 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.
- Rosaria, R. et al. (2021). Ai environment for predictive maintenance in a manufacturing scenario. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8.
- Wang, J. et al. (2017). Predictive maintenance based on event-log analysis: A case study. *IBM Journal of Research and Development*, 61(1):11:121–11:132.
- Xiang, S., Huang, D., and Li, X. (2018). A generalized predictive framework for data driven prognostics and diagnostics using machine logs. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, pages 0695–0700.