The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40)
April 29 - May 2, 2019, Leuven, Belgium

# Robot fault detection and remaining life estimation for predictive maintenance

Riccardo Pinto[a], Tania Cerquitelli[b]

[a]Comau, Via Rivalta 30, Grugliasco, Italy
[b]Politecnico di Torino, Department of Control and Computer Engineering, Corso Duca degli Abruzzi 24, Turin, Italy

## Abstract

In this work some possible solutions to implement a Robotics-oriented predictive maintenance approach are discussed. The data-driven methodology is described from the data collection to the design of an appropriate dataset and finally to the use of some of the most promising algorithms in the field of machine learning. The whole process is composed by several building blocks that can be combined to realize a data analysis on industrial robots. Some of the most promising techniques in Predictive Maintenance for Industrial machines were included in the proposed methodology, together with a Survival Analysis study, and then evaluated with proper performance metrics. Experimenting this methodology on a real use-case with Comau industrial robots showed the validity of the approach and opened to the inclusion of such a process in a service-oriented solution.

*Keywords:* Industry 4.0; machine learning; data analytics; robot fault forecasting

## 1. Introduction

The definition for Industry 4.0 was first introduced in 2011 at the Hannover Messe trade fair by the Germany Trade and Invest (GTAI) as: *"A paradigm shift made possible by technological advances which constitute a reversal of conventional production process logic. Simply put, this means that industrial production machinery no longer simply processes the product, but that the product communicates with the machinery to tell it exactly what to do. [...] (Industry 4.0) represents the technological evolution from embedded systems to cyberphysical systems, an approach that connects embedded production technologies and smart production processes."* In the Industry 4.0 era, artificial intelligence is revolutioning the whole manufacturing process. The name Industry 4.0 refers to the forth Industrial

---

* Riccardo Pinto
 *E-mail address:* riccardo.pinto@comau.com

Revolution, with the first three coming about through mechanization, electricity, and IT. Specifically, with the advent of the Internet of Things (IoT) concept, together with new application of machine learning methods, the monitoring of physical processes can be simplified and can easily lead to data-driven decision making. The real-time communication between humans, machines and sensors can reduce downtime, spot production defects and improve the supply chain, allowing the self improvements of all these steps by learning from experience. Companies are beginning to use all these new technologies to increase their understanding in terms of production process efficiency, turning a traditional manufacturing facility into a *smart factory*. To this end several studies have already shown an efficient integration in the context of Industry 4.0. [1] has exploited the role of Big Data Analytics in the production process using the *Cross-industry Standard Process for Data Mining* methodology, while some predictive maintenance approaches regarding the connection between machines and Internet applications has been treated in [2]. The use of relatively common machine learning algorithms like K-NN, Random Forest or even Deep networks has been explored in the industrial field in works like [13], where a probability of failure when the useful life of the part is close to its end has been predicted. In [14] an application of K-NN was proposed to predict the replacing of tungsten filaments in a ion implantation, a very important step in the semiconductor manufacturing process that is often a bottleneck in production lines due to the high cost of the tool [15]. Last, in [16] anomaly detection in thermal images were treated using infrared thermography, training a Deep Neural Network model to learn the shape of a normal thermal image and detecting anomalies with a pixel-based comparison. All this increasing interest and effectiveness of predictive maintenance techniques has convinced Comau to investigate in some effective techniques [9] [10] [11] not only to evaluate components lifetime, but also to predict specific faults that could arise with robots and similar industrial machines. The first problem with such a goal is the ability to collect data, especially for older robots already in production lines, mainly for business reasons. The main interest of Comau clients is in fact to reduce production stops to the lowest, so any intervention has to be planned in advance and it can be done only by specialized technicians. In this work we present a general description of the kind of data in the dataset, moreover we discuss a building-block methodology to attack similar problems and finally the result of the application of four different techniques (Survival Analysis, Convolutional Neural Networks, Random Forest and k-NN) will be presented and compared in terms of accuracy (i.e. the capability to correctly predict a broken robot).

The paper is organized as follows. Section 2 and 3 introduce the case study of robot fault detection, while section 4 presents the proposed architecture. In Section 5 the dataset is described, while Section 6 describes the preprocessing steps needed to run the algorithms detailed in Section 7. Finally the Validation procedure and the results are described in Section 8, before presenting the conclusions in Section 9.

## 2. Case study: NJ robot family cables in spot welding operations

The NJ-*X* robot family is a medium-to-high payload robot, where *X* represents the maximum payload of a particular model (e.g. the maximum payload of an NJ-170 is 170kg). This family of robots is widely used for applications like body assembly, cosmetic sealing, packaging, laser welding cutting, press to press application and more. In automotive applications NJ-*X* robot family is widely used for spot-welding operations in which the robot is equipped with a welding gun to perform the operation in an industrial production line. Such configuration, named *Spot Welding Machine*, can be floor mounted or installed on the ceiling. The welding gun (equipped with a lighter transformer) has the flange connected directly to the robot without any adapter kit.

To better explain the specific problem arising in such context, a short description of a spot welding operation is given. The equipment used in the spot welding process consists of tool holders and electrodes. The tool holders function as a mechanism to hold the electrodes firmly in place and also support optional water hoses that cool the electrodes during welding. The tool holder in this case study is the robot itself, with all the mechanism to provide the required materials. Electrodes instead are made of a low resistance alloy (like copper), and their design varies in shape and size depending on the specific application. A spot welding task is divided in three step. During the first one the electrodes of the welding gun are put in contact with the surface of the metal area, applying a light pressure. Then, the current from the electrodes is applied and after the metal is cooled via the coolant holes in the center of the electrodes. Typical coolants used for this operation are water and a brine solutions. Therefore, since such a configuration requires the installation of the welding gun as a tool for the NJ robot family, all the cables needed to provide the welding gun
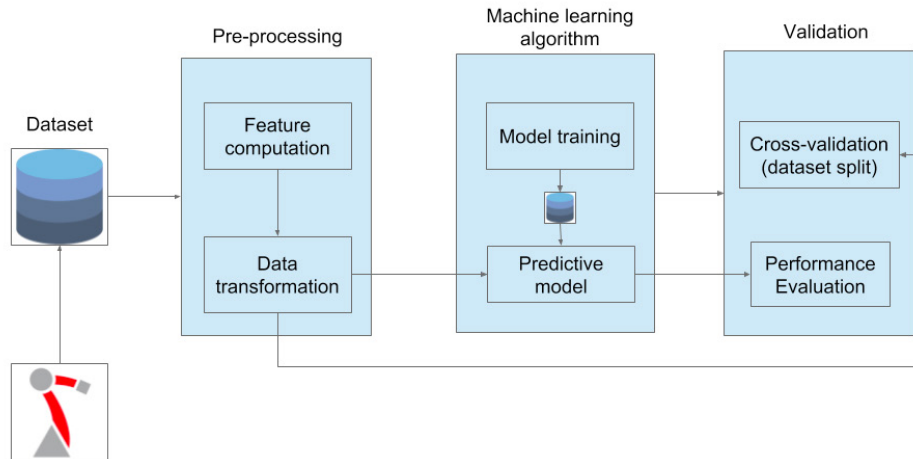
Fig. 1. The data analytics architecture

with power supply must be hosted inside the robot links, in order to avoid a dangerous exposition to fire and also to mechanical damages.

The problem evaluated as case study in this work is related to this kind of configuration. The considered fault (if not treated) could lead to a manual intervention so to restart the robot job properly; such an intervention has to be done in a stressful situation for the client that, in the most critical scenarios, can also experience an high cost due to a stop of his production line. The ability to correctly predict such a fault in advance and plan a fix to the problem outside the production schedule (e.g. on Sundays), could therefore lead to an important cost saving for the client and to a better maintenance policy of Comau.

## 3. Problem definition and performance evaluation

Given a dataset of NJ robots for Spot application the task is to correctly predict robots that will present the choking cables problem. Such robots will be called here *broken robots*. Therefore given $\mathbf{X}$ the matrix of input data (either featured data or raw data) the goal is to correctly predict a function $f : \Re^n \longrightarrow 1, 0$ that given entries not already in the dataset it is able to identify broken robots. The Survival Analysis approach, instead, is slightly different and it is properly detailed later. To have a fair performance measurement between all the approaches the results will be measured in terms of accuracy and recall (i.e. the capability of correctly recognizing a broken robot). Due to the impossibility of having a direct feedback in the plant (mainly for client security and business reasons) some of the data were left out of the training set, so to have a test set to evaluate the different approaches. To do this the dataset has been splitted following a Stratified K-Fold approach and a Leave-one-out one, to obtain a balanced train set and test set (to overcome the limitation of a very unbalanced dataset described above)

## 4. Data analytics architecture

The proposed data analytics architecture includes 3 main steps: (i) *Data preparation*, (ii) *Machine learning algorithm*, and (iii) *Validation*.

First data are prepared to be processed by the data preprocessing block by computing a set of relevant features to capture the key aspects of robot activities to correctly predict faults, then data are normalized to feed the machine learning algorithms in a proper way. Different machine learning algorithms have been integrated in the proposed architecture to train a prediction model to forecast robot faults. The machine learning algorithm is then tuned with

the proper parameters and, according to the nature of the algorithm itself, it is trained on the prepared data. After the tuning/training step a prediction is done on test data and finally evaluated according to the metrics (accuracy, precision and recall). All the steps are detailed in the following sections. Real-Data was manually collected by Comau technicians from two industrial plants under the supervision of the Head of Quality. For the purpose of this study we categorize data in two types: *raw data* and *featured data*. Raw data are axes coordinates of the robot during the execution of a specific production program running during the acquisition , while featured data are extracted from raw data and classified using some important features related to the problem. When speaking about raw data, every robot move (i.e. a row of the dataset) is characterized by the following information: (i) Ax4 (real): robot coordinate of axis 4, (ii) Ax5 (real): robot coordinate of axis 5, (iii) Ax6 (real): robot coordinate of axis 6, and (iv) secMove (real): the time (in milliseconds) in between two adjacent moves.

These kind of data was collected directly from the execution of a robot program in the production plant, but Comau technicians cut the information about the first three axis since (on their experience) their movements are not related to the problem root cause. On the other hand, according to the experience of Comau technicians having prior knowledge of the use case, the following features were included in the dataset:

- S/N C5G (string) : ID number of a control box
- Type (string) : robot family
- Floor (bool): if the robot is position on the ground or not
- Tool change (bool): the presence of the tool change option
- Spot (bool): an information about the robot performing a spot wielding or not. Only entries with these values set to 1 are useful for our specific analysis.
- Broken (bool): if the robot was affected by problem or not
- Cars (long int): the number of cars produced by the robot (a time reference)
- Delta (int): the number of times the robot joint angles for a specific joints greater than 200 deg.
- Axis (int): the number of times the robot axis was greater than 100 cm

From the whole dataset of 859 records, only 554 robots satisfy the requirement of being spot robots, and therefore being involved in the case study. Both raw data and features were used to train the prediction model to forecast robot faults.

### 4.1. Data preparation

Both raw and featured data need some preprocessing in order to be used by the machine learning algorithm. The success of a machine learning algorithm is strongly affected by proper data quality. If the data contain outliers and irrelevant information, a preprocessing step is required to correctly exploit any machine learning algorithm and obtain good performance performance in terms of accuracy.[3]. Correctly in this work the feature computation step elaborates the data applying three different techniques to feed the algorithms: data normalization, Principal Component Analysis and conversion to images (just for the CNN algorithm). For each algorithm the impact of each technique is evaluated and compared with the others.

### 4.2. Feature computation and Data transformation

In this module the features are cleaned or processed to feed the machine learning algorithms. The first step is the normalization of the feature, used to scale down the transformation of the features to avoid an excessive difference between the maximum and minimum values, (e.g. 0.001 and 10000). This is important for many algorithms such as the neural networks and k-Nearest Neighbourhood algorithms [17], here we always execute a min-max normalization [17]. Moreover a Principal component analysis step is done on the set of features to perform a data reduction in terms of number of features. Each algorithm can be fed with a different number of components varying the number of components from 0 to 10. In the sensitivity analysis of each algorithm this number has been varied in order to understand if the application of PCA on the dataset can affect the accuracy of the algorithm.

Last, this module converts also the raw data coming from the moves to greyscale images. This is done only for the convolutional neural networks machine learning algorithm which has already achieved great success in extracting
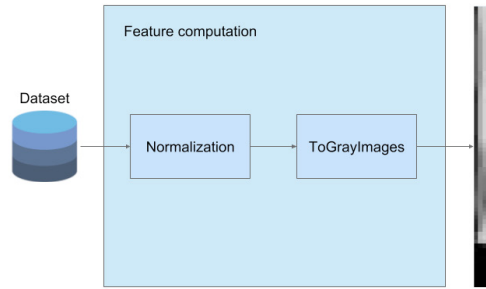
Fig. 2. The conversion of the raw data information of axis coordinates, first normalizes the real numbers between 0 and 1 and then writes the data in an 8-bit black and white image. Zeros are associated to black pixels, while ones are associated to white pixels. Intermediate values are therefore represented by greyscale pixel.

features from images [12] [4]. Therefore the conversion of the dataset entries to properly formatted images is expected to enhance the algorithm performance. The conversion is described in figure 2. Since the input representation must have the same dimensionality a zero-padding technique is used to uniform values [5]

### 4.3. Machine learning algorithm

The machine learning algorithm module integrates different machine learning algorithms to address the robot faults prediction. Specifically it includes: Survival analysis, Extremely Randomized Trees, K-Nearest Neighbor and Convolutional Neural Network, briefly described below.

***Survival analysis.***. Survival analysis was considered to fit the problem of the robot case study, due to the low rate of failures (i.e. broken robots). To apply such an algorithm, a time component is needed, in order to estimate as output a regression of covariates (i.e. features) to estimate lifetimes. Since the only time information that is present in the database is the number of cars produced by a single robot, that quantity is taken as $t$ variable to apply Cox Model and Aalen's additive model in order to estimate the cumulative hazard function of a robot fault. Due to the nonlinear relationship between joints position and robot position, a Cox Model approach was chosen and measured in performance using the corcondance index (CI, i.e. the probability that for a pair of randomly chosen samples, the sample with the higher risk prediction will experience an event before the other sample).

***Extremely Randomized Trees***. Differently from a conventional Random Forest approach, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate and the best of these randomly-generated thresholds is picked as the splitting rule. In contrast to the original work presented in [8], the scikit-learn implementation combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class. The specific input parameter of the algorithm has been set based on the results of a sensitivity analysis described in Section 5.2.

***K-Nearest Neighbors***. k-Nearest Neighbors (k-NN) is a simple approach to classification in which a distance (e.g. Euclidean norm, Mahalanobis) between samples is computed. Each point of the input space is labeled according to this distance from its k closest neighboring samples. K is given as a parameter in input and represents the number of samples to be considered to determine the label (i.e. the neighborhood). The specific parameter of the algorithm has been set according to a sensitivity analysis described in Section 5.2 and the final results has been obtained by using a Minkowsi distance, a leaf size of 30 and 5 number of neighbors.

***Convolutional neural network (CNN)***. CNN is a neural network approach used to train a model useful to some kind of predictions [17]. The network layers has been built with the following architecture: the input layer is defined as a Convolutional one to filter windows of two-dimensional inputs (the move image produced as described in Figure 2. The output of this layer is a linear/matrix multiplication of the input, data are passed to a Polling layer and then an Activation layer with a Rectified Linear Unit activation function that apply a non-linear operation. After that the data is flattened through a Flatten layer and then given in input to a Dense layer in which every input is connected to every output by a weight. Finally, after another Activation layer the output is returned.

### 4.4. Validation

***Cross-validation techniques and data splitting*** . This building block evaluates the quality of the performed predictions by exploiting different quality metrics: recall, precision and accuracy. In the cross-validation step the database is splitted in order to create the necessary sets for both training and testing the algorithms. We followed two approaches: a Stratified K-Fold approach and a Leave One Out one. In the *Stratified K-Fold* a model is trained using *k*-1 folds (i.e. a smaller set of data) and it is validated on the remaining part of the data that it is used as a test set to compute performance measure such accuracy. This approach can be computationally expensive, but it is a good approach when the number of samples is small. A *Leave one out* approach it is equivalent to a *Stratified K-Fold* where k is equal to the number of samples. This method can be very costly when there is an high number of samples, but it can produce more accurate results when there is a small ratio between positive and negative sample.[6]. For all the algorithms a stratified k-fold of 5 folds was confirmed to be a sufficient number of splits, except for the CNN case in which a Leave One Out approach outperformed the 5 folds split. Finally, the performance evaluation module analyzes the performance of one of the machine learning algorithm described in Section 4.3 according to three different metrics: accuracy, precision and recall. This building block aims at improving the quality of the prediction models in terms of accuracy, to this aim it includes two main blocks: the performance evaluation module and the cross-validation module. Accuracy is the fraction of predictions the model gets right. Precision, instead is the fraction of positive identifications that are actually correct, while recall is the proportion of actual positives that are correctly identified from the prediction model. The computation of these 3 measures can give a clear metrics of an algorithm performance.

***Sensitivity analysis***. To test a broad variety of configurations for the algorithms described in Section 4.3 a Grid Search is performed to do an exhaustive search over specified parameter values. The parameter list to give in input to the grid search is different for every algorithm and specified in the relative section. For example in the Extremely Randomized Trees the search is started on the number of estimators, criterion and max features parameters. Final parameters are selected to maximize the f-measure (i.e. the armonic average of prediction and recall) and identify the best configuration for each algorithm, $F_1 = (\frac{recall^{-}1+precision^{-}1}{2})$.

## 5. Experimental evaluation

Here we discussed a set of experiments performed to evaluate the effectiveness of the proposed approach in correctly performing the robot fault predictions. Furthermore we discussed the algorithm performance by varying the input parameters. This sensitivity analysis has been performed to set the best input parameters for each exploited analytics algorithms. For the Survival Analysis algorithms the package lifelines [7] was used, while scikit[6] was used for the Extremely Randomized Trees and k-Nearest Neighbor and finally Keras over Tensorflow provided the implementation for the Convolutional Neural Network architecture.

***Survival Analysis***. Here we discuss the effectiveness of the Survival Analysis in estimating the remaining life of the robot under analysis. The Cox model algorithm can be parametrized according to the following parameters: (i) weights *w* denotes the weight per subject. This column is expelled and not used as a covariate, but as a weight in the final regression. Default weight is 1. (ii) Beta initialize the starting point of the iterative algorithm. Default is the zero vector. (iii) step size *s* set an initial step size for the fitting algorithm.

***Extremely Randomized Trees***. The Extremely Randomized algorithm can be be parametrized according to the following parameters: (i) number of estimators *n*: denotes the number of trees in the forest; (ii) criterion denotes the function to measure the quality of a split. Possible inputs to this parameters are *gini* for Gini impurity and *entropy* for the information gain. (iii) Max feaures represents the number of features to consider when looking for the best split. Possible options are *sqrt* and *log2*, where $50 \leq n \leq 1000$, $sqrt = \sqrt{n}$ and $log2 = \log_2 n$.

***k-Nearest Neighbor***. The k-Nearest Neighbor algorithm can be be parametrized according to the following parameters: (i) number of neighbors *n*: denotes the number of neighbors, where $1 \leq n \leq 10$; (ii) weights denotes the weight function to be used in prediction. Possible inputs to this parameters are *uniform* where all the points are weighted equally, *distance* where the points are weighted by the inverse of their distance. (iii) Algorithm: The kind of algorithm used to compute the nearest neighbors. Possible options are *ball tree*, *kd tree*, or *brute*.

***Convolutional Neural Network***. The Convolutional Neural Network Activation layer has been parametrized according to the activation function where the possible options include: (i) softmax: softmax activation function; (ii) relu: rectified linear unit function; (iii) tanh: hyperbolic activation function.

## 5.1. Survival Analysis

The concordance index obtained by training the model on the featured dataset using a PCA analysis is *0.76*, while using a PCA on the featured dataset gives a concordance rate of *0.8*. Based on this result it was possible to generate an output list of possible robots candidate to become broken robots in the future, ordered by probability of having the need to plan an intervention. The impact of the PCA analysis is present as shown in Figure 3, but not so relevant for the chosen metric of evaluation, thus it is possible to estimate the remaining life of the robot with or without PCA.

| Algorithm | Extremely Randomized Trees (Accuracy) | k-NN (Accuracy) |
|---|---|---|
| Without PCA | 0.95 | 0.95 |
| With PCA | 0.98 | 0.98 |
| Algorithm | Extremely Randomized Trees (Precision) | k-NN (Precision) |
| Without PCA | 0.98 | 0.98 |
| With PCA | 0.98 | 0.98 |
| Algorithm | Extremely Randomized Trees (Recall) | k-NN (Recall) |
| Without PCA | 0.96 | 0.98 |
| With PCA | 0.96 | 0.98 |

| Algorithm | Survival Analysis (cross-index) |
|---|---|
| Without PCA | 0.76 |
| With PCA | 0.80 |

Fig. 3. Cross-index results of the Survival Analysis algorithm with and without the contribution of the Principal Component Analysis

Fig. 4. Accuracy, Precision and Recall results of both the Extremely Randomized Trees and the k-Nearest Neighbor algorithm with and without the contribution of the Principal Component Analysis

## 5.2. Extremely Randomized Trees and k-NN

Figure 4 shows the accuracy, precision and recall scores of the algorithms proposed in our work with and without PCA. The impact of the PCA analysis is almost negligible for all the algorithms both in terms of accuracy and precision, thus it is possible to configure the analytics process without exploiting the PCA.

### 5.2.1. CNN

The impact of the Data to Image transformation in terms of correctly classified robots can be evaluated in the confusion matrices in Figures 5 and 6. Figure 5 show the confusion matrix of the convolutional neural network trained with data transformed using the Data to image block, while Figure 6 represents a confusion matrix without this transformation.

|    | TP  | TN |
|----|-----|----|
| PP | 250 | 2  |
| PN | 1   | 41 |

Fig. 5. The confusion matrix of the best CNN approach with the Data to Image block. As it can be seen also from Figure 6 results are slightly better with respect to the untransformed raw data

|    | TP  | TN |
|----|-----|----|
| PP | 241 | 4  |
| PN | 12  | 0  |

Fig. 6. The confusion matrix of CNN approach without the Data to Image block, so with data image, where PN = Predicted Negative, PP = Predicted Positive, TN = True Negative and TP = True Positive. The main diagonal of this matrix represents correctly predicted entry

## 6. Conclusions

In this work a process to support predictive maintenance using Machine Learning algorithms was presented. The process was structured as a framework composed by three modules. In the pre-processing the impact of the feature computation step as well as the impact of the data transformation step (i.e. the Principal Component Analysis process and the Data to Image step on the featured and raw dataset) was evaluated. In the case of the k-NN and extremely randomized trees algorithms the impact of the PCA was quite negligible, while in the Convolutional Neural Network the transformation to image produced an increment of 9% in terms of accuracy, and of precision and recall of 4%. Therefore in Section 4.3 the main algorithms used in literature for predictive maintenance problems were implemented and evaluated on the use case presented in Section 2. The algorithms, evaluated in terms of accuracy, precision and recall, showed to be quite effective in predicting the broken status of the robot on the evaluated data. The best configuration for each specific input parameter of the algorithm was identified through a Grid Search algorithm and in terms of *f1-score*. The process showed to be effective in predicting the use case while having a good generality to be adapted on different use cases.

## References

[1] Nio, Mikel, Jos Miguel Blanco, and Arantza Illarramendi. "Business understanding, challenges and issues of Big Data Analytics for the servitization of a capital equipment manufacturer." Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015.
[2] Babiceanu, Radu F., and Remzi Seker. "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook." Computers in Industry 81 (2016): 128-137.
[3] Kotsiantis, S. B., D. Kanellopoulos, and P. E. Pintelas. "Data preprocessing for supervised leaning." International Journal of Computer Science 1.2 (2006): 111-117.
[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[5] Simard, Patrice Y., Dave Steinkraus, and John C. Platt. "Best practices for convolutional neural networks applied to visual document analysis." null. IEEE, 2003.
[6] Fabian, Pedregosa, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12 (2011): 2825-2830.
[7] Davidson-Pilon, C. "Lifelines." Github repository: https://github. com/CamDavidsonPilon/lifelines (2016).
[8] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[9] Borgi, Tawfik, et al. "Data analytics for predictive maintenance of industrial robots." Advanced Systems and Electric Technologies (IC_ASET), 2017 International Conference on. IEEE, 2017.
[10] Hoppenstedt, Burkhard, et al. "Techniques and Emerging Trends for State of the Art Equipment Maintenance SystemsA Bibliometric Analysis." Applied Sciences 8 (2018): 1-29.
[11] Noman, Mohammed A., et al. "Overview of predictive condition based maintenance research using bibliometric indicators." Journal of King Saud University-Engineering Sciences (2018).
[12] Zhang, Jiajia, Kun Shao, and Xing Luo. "Small sample image recognition using improved Convolutional Neural Network." Journal of Visual Communication and Image Representation 55 (2018): 640-647.
[13] Butte, Sujata, A. R. Prashanth, and Sainath Patil. "Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks." Microelectronics and Electron Devices (WMED), 2018 IEEE Workshop on. IEEE, 2018.
[14] Susto, Gian Antonio, et al. "Machine learning for predictive maintenance: A multiple classifier approach." IEEE Transactions on Industrial Informatics 11.3
[15] Lin, Shin-Yeu, and Shih-Cheng Horng. "A classification-based fault detection and isolation scheme for the ion implanter." IEEE Transactions on Semiconductor
[16] Lile, Cai, and Li Yiqun. "Anomaly detection in thermal images using deep neural networks." Image Processing (ICIP), 2017 IEEE International Conference on. IEEE,
[17] Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. "Introduction to data mining. 1st." (2005).