Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Knowledge elicitation for fault diagnostics in plastic injection moulding: a case for machine-to-machine communication

Rok Vrabič [a], Dominik Kozjek [a], Peter Butala (1) [a,*]

[a] *University of Ljubljana, Ljubljana, Slovenia*

In most manufacturing processes the defect rate is very low. Sometimes, only a few parts per million are defective because of a faulty process. For this reason, fault diagnostics is faced with extremely imbalanced data sets and requires large volumes of data to achieve a reasonable performance. This paper explores whether a machine-to-machine approach can be used, in which several work systems share the process data to improve the accuracy of the fault-detection model. The model is based on machine learning and is applied to industrial data from approximately two million process cycles performed on several injection moulding work systems.

Manufacturing system; Predictive model; Machine-to-machine

1. Introduction

Large amounts of process-related data are gathered on the manufacturing shop floor by machine controllers, monitoring systems, data-input terminals, etc. However, this data is rarely analysed using methods that go beyond elementary statistics and standard quality control, such as control charts. With new technologies and concepts, such as the Industrial Internet-of-Things (IIoT) [1] and Cyber-Physical Production Systems (CPPS) [2], the amount of digital data will continue to increase in the future. Elementary statistics and simple visualisations will not be sufficient to discern the intricate relationships between the observed parameters due to the size, the dimensionality, and the complexity of the data [3]. Uncovering these relationships is the key to system and process improvement.

New approaches to knowledge elicitation are offered by computational tools and methods based on new paradigms, such as Big Data, and with methods for data mining, machine learning, and other techniques of artificial intelligence. This is especially relevant to the domain of quality control, where the goal is to uncover the relationship between the parameters of the process and the quality of the products. Powerful, computer-generated models can serve to predict faults before they occur and, together with expert interpretation, can provide an insight into the root causes of faults, leading to further process improvements.

The challenge, however, is that quality rates are already very high for most types of manufactured parts, with defect rates often as low as only a few parts per million (ppm). Modelling such rare events presents a challenge for computer-generated models. Consequently, large amounts of data are needed before the performance of the model can become useful in practice.

An approach that addresses this is to establish communication channels between the controllers of identical or similar work systems (e.g., machining systems) that execute similar processes. By sharing and merging the process data, the performance of the learned model can be improved due to the increase in the size of the dataset. Another aspect of this is that when a work system learns a relation between the process parameters and the parameters related to quality, the knowledge can be shared with others to prevent the occurrence of similar defects and faults.

The aim of the paper is (1) to develop a procedure for modelling faults in manufacturing processes based on extremely imbalanced, large datasets and, on this basis, (2) to investigate how the sharing and merging of datasets from several similar work systems via machine-to-machine (M2M) communication influences the performance of the model, and (3) to study whether the fault model is applicable to a work system that is not contributing data to the merged dataset, i.e., without any prior knowledge.

The focus of the research is on cyclic manufacturing processes, in which batches of products are produced by repeating process steps. A case study of plastic injection moulding is presented. Real industrial data from approximately two million process cycles performed on five injection moulding work systems within a period of 6 months served as the basis for the research.

2. Machine-to-machine (M2M) communication

M2M is understood as any form of communication between devices, wired or wireless, supporting collaborative decisions, without any human intervention [4]. In the manufacturing domain, the term is often used in the context of supervisory control and data acquisition (SCADA) systems in which the devices are interconnected to monitor and control the manufacturing processes [5]. In this context, M2M represents the backbone providing the means for real-time communication and data transfer. Several authors discuss the importance of M2M communication in the context of Ubiquitous [6] and Cloud Manufacturing, where they emphasise the importance of scalability [7], and that the acquired data can be used together with advanced data analytics methods to support prognostics [8].

Fig. 1 shows the role of M2M communication within a broader context of knowledge elicitation. The data is exchanged between the work systems using a shop floor information system that is a part of the company's information system. In turn, the data can also be shared with the work system manufacturer. This supports two learning loops: (1) a M2M Learning Loop, the purpose of which is to extract knowledge from the process data by sharing the data and the knowledge amongst the work systems, and (2) a Manufacturer's Learning Loop, which the work system manufacturer uses to learn from the data aggregated from all their work systems across different companies.

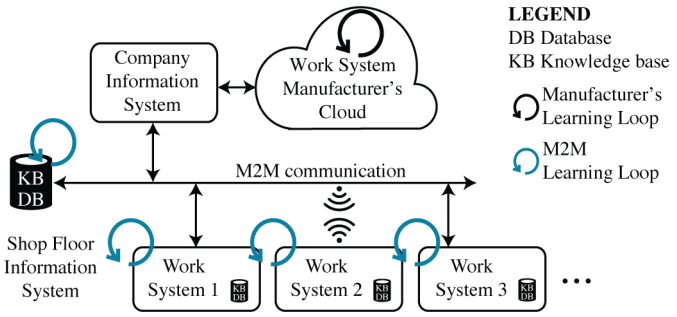

Figure 1. Role of M2M in broader context of knowledge elicitation.

The paper explores how the M2M Learning Loop can be employed in the context of fault diagnostics for plastic injection moulding.

3. Fault diagnostics employing M2M communication

Fault diagnostics affects the quality of the products and the availability of the system through fault detection and root-cause analysis. Conventionally, model-based approaches that define the normal and faulty states are used. However, for well-studied processes, it is very difficult to improve the fault detection performance beyond a certain point. Besides, the model-based approaches are both process-specific and difficult to generalise [9]. A promising approach is to use learning techniques [10] in which the relationship between the process and the quality parameters is learned based on real data. In many cases this can be challenging due to the imbalanced nature of the data. Since there are usually only a handful of defects per million pieces, there is much more information about the normal process regime than there is about the faulty one [11, 12]. M2M communication can contribute to improving the performance of the learned model through sharing the data among similar work systems to increase the overall volume.

3.1. The machine learning workflow for fault diagnostics

The proposed workflow integrates the M2M analysis objectives and the classification techniques for imbalanced datasets [13]. The workflow and the M2M cross-evaluation are shown in Fig. 2. The raw collected data is pre-processed and efficiently stored, e.g., in a NoSQL database. To ensure unbiased validation and evaluation, the partitioning into validation groups (folds) and the under-sampling are performed. For the training of the model, the data is filtered to include only a subset of the work systems. Feature vectors are calculated, faulty cycles are oversampled, and features are selected separately for each combination of training folds. The model is cross-validated using the test data filtered to another subset of work systems, but not under-sampled and over-sampled, to test the performance of the model on imbalanced data. Confusion matrices for different discrimination thresholds of the binary classifier (normal/faulty) are obtained. The result is the summed confusion matrix at the best performing thresholds, which are, in turn, used to calculate the classifier performance (e.g., the f-measure).

The fault diagnostics model is derived from data describing a set of C cycles of the process, denoted as $cycles$ (Eq. 1).

$$cycles = \{c_1, c_2, c_3, \dots, c_i, \dots, c_C\} \quad (1)$$

For each cycle, the data consists of the cycle id, the name of the $machine$, the date and time ($dateTime$), P process-related, O other parameters, and a target class y_i, as shown by Eq. 2. The value of the target class denotes whether the cycle is normal (0) or faulty (1).

$$c_i = \{c_i^{id}, c_i^{machine}, c_i^{dateTime}, c_i^{P,1}, c_i^{P,2}, \dots, c_i^{O,1}, c_i^{O,2}, \dots\} \rightarrow y_i \quad (2)$$

The group of normal cycles' id values is denoted as $\boldsymbol{IDs}^{normal}$ and the group of faulty cycles as $\boldsymbol{IDs}^{faulty}$. The cycle ids are set consecutively, so that the cycles performed on the same work system ($machine$) are grouped together and ordered by $dateTime$ in ascending order (Eqs. 3, 4, and 5).

$$\forall i \in [1, C]: c_i^{machine} \in \{m_1, m_2, m_3, \dots m_M\} \quad (3)$$

$$\forall i, j \in [1, C] \wedge \forall k, l \in [1, M]: \left(c_i^{machine} = m_k \wedge c_j^{machine} = m_l \wedge k < l\right) \Rightarrow c_i^{id} < c_j^{id} \quad (4)$$

$$\forall i, j \in [1, C]: \left(c_i^{dateTime} < c_j^{dateTime} \wedge c_i^{machine} = c_j^{machine}\right) \Rightarrow c_i^{id} < c_j^{id} \quad (5)$$

Because of the imbalanced nature of the dataset the data for the training and the testing of the learning algorithm must be prepared with care. If the training data was randomly sampled, the ratio of faulty to normal cycles would vary significantly with each sampling, which would result in unreliable models. To address this, the data is grouped into K groups of normal (consisting an ordered set $\boldsymbol{G}^{normal}$) and faulty $\boldsymbol{G}^{faulty}$ cycles. Each group inside the sets $\boldsymbol{G}^{normal}$ and $\boldsymbol{G}^{faulty}$ contains approximately the same number of normal and faulty cycles, respectively (Eqs. 6 and 7).

$$\boldsymbol{G}^{normal} = \left[G_1^{normal}, G_2^{normal}, G_3^{normal}, \dots, G_K^{normal}\right] \quad (6)$$

$$\boldsymbol{G}^{faulty} = \left[G_1^{faulty}, G_2^{faulty}, G_3^{faulty}, \dots, G_K^{faulty}\right] \quad (7)$$

To balance the ratio of faulty to normal cycles, the groups containing the normal cycles are under-sampled to a chosen amount (obtaining the groups S_k^{normal}, Eq. 8).

$$\boldsymbol{S}^{normal} = \left[S_1^{normal}, S_2^{normal}, S_3^{normal}, \dots, S_K^{normal}\right]; \quad \forall k \in [1, K]: S_k^{normal} \subseteq G_k^{normal} \quad (8)$$

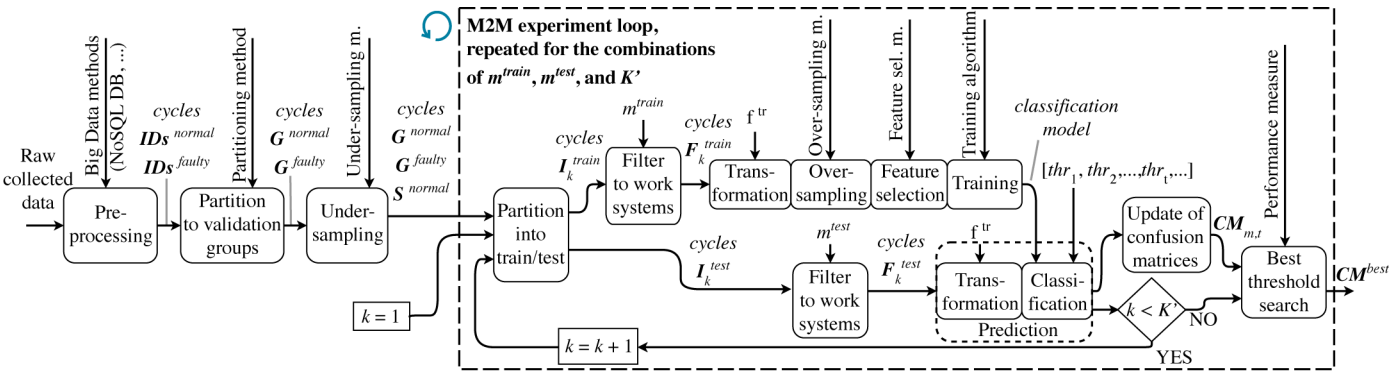

Figure 2. Learning and M2M cross-evaluation of the models.

The data is then transformed into feature vectors $\boldsymbol{x}_n$ by applying a transformation function f^{tr}, as shown in Eq. 9.

$$\boldsymbol{x}_n = \mathrm{f}^{\mathrm{tr}}\left(c_n^{id}\right) = \left[x_{n,1}, x_{n,2}, x_{n,3}, \dots\right] \tag{9}$$

The faulty cycles that are used for the training are over-sampled to further reduce the imbalance. The rest of the workflow follows an established machine learning methodology and consists of the feature selection and training of the model using well-known algorithms.

The model is validated with K'-fold cross validation using a chosen amount K' out of the K predetermined folds (Eqs. 6, 7, and 8). For each fold k, the model is trained using the under-sampled normal and over-sampled faulty cycle data, except for the data belonging to the k-th fold, and tested on all the normal and faulty data of the k-th fold (Eqs. 10 and 11).

$$\begin{aligned} &I_k^{train} = \left\{S^{normal} \cup G^{faulty}\right\} \backslash \left\{S_k^{normal} \cup G_k^{faulty}\right\}; \\ &S^{normal} = S_1^{normal} \cup S_2^{normal} \cup S_3^{normal} \cup \dots \cup S_{K'}^{normal}, \\ &G^{faulty} = G_1^{faulty} \cup G_2^{faulty} \cup G_3^{faulty} \cup \dots \cup G_{K'}^{faulty} \end{aligned} \tag{10}$$

$$I_k^{test} = \left\{G_k^{normal} \cup G_k^{faulty}\right\} \tag{11}$$

3.2. The M2M cross-evaluation

The goal of the cross-evaluation is to assess the performance of the models trained on the data of one subset of work systems and then tested on the same subset or another one. The methodology consists of filtering the training and the testing data to include only samples from a subset of work systems (m^{train} and m^{test}, respectively). The filtered sets of cycle ids are denoted as F_k^{train} and F_k^{test}, respectively. The result of each test are confusion matrices at each discrimination threshold candidate thr_t of the binary classifier (classifies as normal/faulty). The confusion matrix for work system includes the number of true positives, false positives, false negatives, and true negatives, as shown in Eq. 12. For each configuration of m^{train}, m^{test} and the chosen number of folds for cross-validation K', the confusion matrices of individual test folds are summed together.

$$\boldsymbol{CM}_{m,t} = \begin{bmatrix} TP_{m,t} & FP_{m,t} \\ FN_{m,t} & TN_{m,t} \end{bmatrix} \tag{12}$$

The best discrimination threshold thr_m^{best} is determined for each work system m using the maximum value of the chosen performance measure, e.g., the f-measure (i.e., the F1 score). The result is the chosen performance measure of the overall confusion matrix at best discrimination thresholds for individual work systems.

4. Application to plastic injection moulding

A case of fault diagnostics on five plastic injection moulding work systems is presented. The data includes information about tools, dates and times, process parameters, and alarms for approximately two million process cycles performed over a period of 6 months. Unplanned work system stops are considered as faults. The ratio of faulty to normal cycles is 1:1484. An excerpt from the process parameters data is shown in Table 1.

The feature vectors (Eq. 9) are built to describe the dynamics of the process and strive to capture the relative changes in the process parameter values. They include several changes to the average values for each process parameter. Let the average value of the parameters of the starting k cycles and the ending j cycles before the n-th cycle be denoted as $\langle c_n \rangle_{j-k}$. The feature vectors then include the changes to the process parameter averages with respect to the average k=30 and j=15 cycles before the n-th cycle (Eq. 13).

Table 1
Excerpt of the process parameter data.

Date	Time	Cycle no.	Cycle time [s]	Inject. Time [s]	Max. inject. pressure [bar]	...
...	...	...	...	...	...	...
12/08/15	11:01:48	3922	21.84	1.35	2553	...
12/08/15	11:02:10	3923	21.83	1.35	2553	...
12/08/15	11:02:32	3924	21.81	1.34	2553	...
12/08/15	11:02:54	3925	21.84	1.35	2553	...
...	...	...	...	...	...	...

The rationale behind the selection of these features is that the fault diagnostics system should also be able to predict the faults before they occur. Therefore, the relative changes to the process parameters describe the dynamics of past cycles, but not those of the current cycle. The feature vectors also include the features that are of categorical type and give information on cycles about (1) the day of the week day, (2) the part of the day $dayPart$, (3) the corresponding work system $machine$ and (4) the $product$.

$$\begin{aligned} \boldsymbol{x}_n = \big[&\dots, \langle c_n \rangle_{1-1} - \langle c_n \rangle_{15-30}, \langle c_n \rangle_{2-2} - \langle c_n \rangle_{15-30}, \langle c_n \rangle_{3-3} - \\ &\langle c_n \rangle_{15-30}, \langle c_n \rangle_{4-4} - \langle c_n \rangle_{15-30}, \langle c_n \rangle_{5-5} - \langle c_n \rangle_{15-30}, \langle c_n \rangle_{5-10} - \\ &\langle c_n \rangle_{15-30}, \langle c_n \rangle_{10-15} - \langle c_n \rangle_{15-30}, \dots, \\ &x_n^{day}, x_n^{dayPart}, x_n^{machine}, x_n^{product}\big] \end{aligned} \tag{13}$$

The value of y_n is determined from the table of alarms, which specifies the cycle at which the alarm occurred together with the severity of the alarm – whether it is only a warning (e.g., a non-critical deviation of a process parameter) or an actual fault, i.e., an unplanned work system stop.

The algorithms for under- and over-sampling, feature selection, and training were selected based on their performance during an initial testing of several algorithms. The under- and over-sampling are set to produce approximately 2,000 normal and 2,000 faulty cycles per fold. The under-sampling of the normal cycles is semi-random to ensure that all the nominal values (work system ids, days of the week, etc.) are app. equally represented. For over-sampling, the SMOTE algorithm is used [14, 15]. Five nearest neighbours are considered when constructing synthetic samples. For feature selection, select-from-model method [16] is used. The feature selection threshold is determined by the mode mean. The AdaBoost-SAMME.R meta-algorithm, with decision-stumps as the base estimators (optimised CART alg. with Gini impurity as the splitting metric), is chosen as the training algorithm [16, 17]. The maximum number of estimators and the learning rate are set to 300 and 1.0 respectively to avoid over-fitting.

4.1. Results

The M2M cross-evaluation compares the performance of the model trained on all the data and tested on all the data (ALL), the model trained on the data of a single work system and tested on the data of the same work system (SM), and the model trained on the data of all-except-one work systems and tested on the data of the remaining one (M-1). The goals of the analysis are (1) to evaluate the procedure for modelling faults based on extremely imbalanced datasets, (2) to investigate how the M2M communication influences the performance of the model (ALL and SM models) and (3) to study whether anything can be learned about a work system without considering its data (M-1). Fig. 3 compares the models for different numbers of folds K'.

The number of chosen folds directly influences the amount of the used data. This ranges from 174 faulty and 228,150 normal cycles for $K' = 2$ to 1042 faulty and 1,546,456 normal cycles for $K' = 10$. The models trained on small amounts of data tend to have poorer performance. The performance improves when there are approximately 100 faults considered per work system.

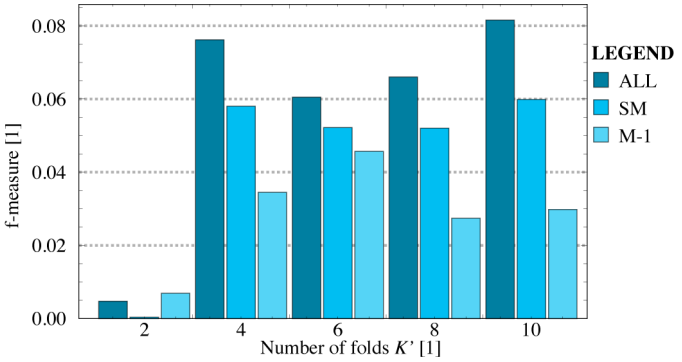

Figure 3. M2M cross-evaluation for different numbers of chosen folds K'.

(1) For $K' = 2$ there is not enough data to train a useful model using the proposed workflow configuration. The performance of the model does not depend significantly beyond $K' = 2$. The joint f-measure for the ALL model with K' = 10 is 0.082, with a corresponding precision of 0.117 and a recall of 0.063. The model is therefore able to predict 6.3% of the faults with 11.7% probability. This is significantly better than random guessing, because of the extremely low ratio of faulty to normal cycles (1:1484) for which the probability of a fault occurring is 0.07%.

The performance of the method heavily depends on the degree of the imbalance of the dataset. The method was tested for a tool that was used in 23,762 cycles and had a ratio of faulty to normal cycles of 1:263. In this case, the f-measure was found to be 0.230, with corresponding precision of 0.565 and a recall of 0.144. This is approximately 3 times better than the performance of the ALL model. Better performance would also be achieved by considering larger amounts of data as it would include more information about the conditions that result in faults. Another consideration is that the set of acquired parameters is not enough to describe some of the fault types. To improve this, additional parameters would have to be measured.

(2) The models are compared using the proposed workflow and trained with the same parameters. The ALL model performs significantly better than the other two, regardless of the number of folds used and, consequently, the amount of data used. The single work system model SM performs 78.6% as well as the ALL model, on average, for the cases where $K' > 2$. This means that the fault diagnostics model is improved when the data from other work systems is considered along with the data from the single work system, which speaks in favour of the case for M2M communication.

(3) The prediction is worse when the data of the tested work system is not used for the training (M-1), being only 49.7% as good as that of the ALL model (on average for $K' > 2$). Even so, this suggests that there is potential to build-in the generic knowledge through the Manufacturer's Learning Loop, which could then be improved to capture the specifics of the products through the M2M Learning Loop.

The results were cross-checked using several other algorithms, some of them substantially different from the presented workflow (e.g., a neural network based approach). The ratios between the f-measures of the three models remain app. the same across the algorithms. This implies that the observed patterns can be attributed to the data and suggests that the findings can be generalised.

5. Conclusion

The paper presents the concept of the M2M Learning Loop and a case study demonstrating its successful implementation for fault diagnostics in plastic injection moulding. The results are promising and show (1) that it is possible to create a machine-learning-based fault diagnostics model that is trained on process data, (2) that the model can be improved by sharing the data among work systems (M2M), and (3) that it is possible to generalise the process knowledge and apply it to a different work system (without prior knowledge), to some extent.

The approach uses the data that is commonly acquired by existing machine controllers and is therefore relatively easy and inexpensive to implement.

Learning from the data by considering the data from all the machines of a manufacturer is not in the scope of the analysis. However, as it is observed that the amount of data improves the performance of the fault diagnostics models, manufacturers could use a similar approach to build knowledge models into the controllers of their work systems, thus improving their performance. Future work includes improving the machine learning workflow towards a system for root-cause identification and the prognostics of faults.

Acknowledgement

This work was partially supported by the Ministry of Higher Education, Science and Technology of the Republic of Slovenia, grants no. 1000-15-0510 and C3330-16-529000, and by the Slovenian Research Agency, grant no. P2-0270.

References

[1] Xu LD, He E, Li S (2014) Internet of things in industries: A survey, IEEE Transactions on Industrial Informatics 10(4):2233–2243.
[2] Monostori L, Kádár B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, Sauer O, Schuh G, Sihn W, Ueda K (2016) Cyber-physical systems in manufacturing, CIRP Annals – Manufacturing Technology 65(2):621–641.
[3] Precup RE, Angelov P, Costa BSJ, Sayed-Mouchaweh M (2015) An overview on fault diagnosis and nature-inspired optimal control of industrial process applications, Computers in Industry 74:75–94.
[4] Wan J, Chen M, Xia F, Li D, Zhou K (2013) From machine-to-machine communications towards cyber-physical systems, Computer Science and Information Systems 10(3):1105–1128.
[5] Verma PK, Verma R, Prakash A, Agrawal A, Naik K, Tripathi R, Alsabaan M, Khalifa T, Abdelkader T, Abogharaf A (2016) Machine-to-machine (M2M) communications: A survey, Journal of Network and Computer Applications 66:83–105.
[6] Kim DH, Song JY, Lee JH, Cha SK (2009) Development and evaluation of intelligent machine tools based on knowledge evolution in M2M environment, Journal of Mechanical Science and Technology 23(10):2807–2813.
[7] Putnik G, Sluga A, Elmaraghy H, Teti R, Koren Y, Tolio T, Hon B (2013) Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap, CIRP Annals – Manufacturing Technology 62(2):751–774.
[8] Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, Helu M (2015) Cloud-enabled prognosis for manufacturing, CIRP Annals – Manufacturing Technology 64(2):749–772.
[9] Teti R, Jemielniak K, O'Donnell G, Dornfeld D (2010) Advanced monitoring of work operations, CIRP Annals – Manufacturing Technology 59(2):717–739.
[10] Colledani M, Tolio T, Fischer A, Iung B, Lanza G, Schmitt R, Váncza J (2014) Design and management of manufacturing systems for production quality, CIRP Annals – Manufacturing Technology 63(2):773–796.
[11] Hu Y, Baraldi P, Di Maio F, Zio E (2016) A systematic semi-supervised self-adaptable fault diagnostics approach in an evolving environment, Mechanical Systems and Signal Processing 88(1):413–427.
[12] Kumar A, Shankar R, Choudhary A, Thakur LS (2016) A big data MapReduce framework for fault diagnosis in cloud-based manufacturing, International Journal of Production Research 54(23):7060–7073.
[13] Ganganwar V (2012) An overview of classification algorithms for imbalanced datasets, International Journal of Emerging Technology and Advanced Engineering 2(4):42–47.
[14] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16:321–357.
[15] Lemaître G, Nogueira F, Aridas CK (2016). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, arXiv:1609.06570.
[16] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011) Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12:2825–2830.
[17] Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55(1):119-139.