Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Fault prediction as a service in the smart factory: addressing common challenges for an effective implementation

Anis Assad Neto*. Elias Ribeiro da Silva **. André Souza*
Fernando Deschamps*. Edson Pinheiro de Lima*. Sérgio Eduardo Gouvêa da Costa*

**Pontifícia Universidade Católica do Paraná, Curitiba, PR 80215-901*
Brazil (Tel: 55-41-3271-2579; e-mail: Fernando.deschamos@pucpr.br).
*** University of Southern Denmark, Mads Clausen Institute,*
Sønderborg, 6400 Denmark (e-mail: elias@mci.sdu.dk)

Abstract: Fault prediction in manufacturing systems has consistently been an important theme in engineering research. Data-driven methods to deliver this service are gaining momentum due to developments regarding information and communication technologies. Particularly, fault prediction may be interpreted as a supervised learning classification problem, in which algorithms trained by operational data gathered from the shop-floor are capable of informing managers whether a machine might enter in a failure state or not. Despite the relevance of this approach, implementations are hindered by several challenges. In this work, we review approaches aimed to deal with four of these challenges, namely: limited amount of training data, unbalanced training data sets, uncertainty regarding which variables should be monitored, and uncertainty regarding how exactly historical data should be employed in the algorithm's training. To deal with training sets with limited size, learning procedures observed to perform well with a lower volume of training data can be used, such as the Random Forests technique. Alternatively, transfer learning techniques can be utilized to adapt models trained in a virtual domain with abundant synthetic data to the real manufacturing system domain. To deal with unbalance among classification classes, cost-sensitive learning methods can be employed to alter the penalties incurred when misclassifications occurs in the minority class. Alternatively, resampling methods can be applied before learning occurs. Lastly, both the decisions regarding which variables to track, and to what extent historical data should be included in the training process, can be addressed through the use of specific feature selection methods.

Keywords: Production activity control, intelligent maintenance systems, maintenance models and services.

1. INTRODUCTION

We are living a period in which novel technologies promise to drastically change the way manufacturing operations are performed. This has been commonly referred to as a fourth industrial revolution (Liao et al., 2017). Upcoming technologies allow shop-floor resources to gain both enhanced communication capabilities (e.g. near real-time gathering of production data); as well as smart decision support capabilities (e.g., capacity to predict potential faults in production equipment). It is the beginning of the era of smart factories (Burke et al., 2017).

In an effort to organize and better exploit these technologies, a number of structured architectures are being proposed in the literature, such as the manufacturing digital twin framework (Zhuang et al., 2018) or the cloud manufacturing system (Li et al., 2010). In these applications, data is usually gathered from the physical world through the use of a network of cutting-edge sensors, software and embedded objects, currently referred to as the 'internet of things' (IoT) (Kang et al., 2016); furthermore, all these data is generally stored in big-data enabled platforms, following the model of the cloud. Lastly, with the objective of offering a decision support service, analytic tools are typically used to provide descriptive analyses and predictions, which are prepared according to the specific service demands of stakeholders.

A plethora of services may be offered through the employment of these steps. Specifically in manufacturing systems, services related to machine fault prediction are commonly mentioned as key applications (Qiao et al., 2019; Xu et al., 2019; Stojanovic and Milenovic, 2019). One way to deliver this type of service consists in acquiring operational data from machines in the shop-floor and using these data to train and deploy supervised learning classification methods, capable of predicting and informing managers weather a machine might enter in a failure state. This can be done through the classification of an analysed instance in classes that represent both normal behaviour as well as an expectancy of imminent failure.

This data-driven approach to fault prediction is gaining more popularity as smart technologies are rapidly evolving (Xu et al., 2019); however, despite its practical relevance, a few challenges may hinder the success of implementation efforts.

In particular, we discuss four specific challenges. Firstly, companies may only have a limited amount of training data available to develop the classifier, due to the young age of many smart factory implementation projects. Secondly, due to the nature of fault occurrences in the shop-floor, these training data sets may be highly unbalanced in regard to the distribution of classification classes. Thirdly, despite the ubiquitous presence of sensors in manufacturing systems, many companies may still be unsure about which operational variables should be monitored and fed to the fault prediction classifier. And lastly, as logs of past data related to each monitored variable are saved in the cloud, another challenge arises in trying to determine how exactly this historical data can be best used as input to the classifier.

In this work, we review strategies designed to overcome all these four obstacles to effective fault prediction. At first, we focus on characterizing each challenge in more detail. Subsequently, we investigate a series of approaches specifically intended to deal with these challenges and assess how these approaches fit together in the context of fault prediction in manufacturing systems. The importance of this work lies in the mapping of common challenges faced by companies to credible and up-to-date solutions being produced in the machine learning literature.

The remainder of this paper is structured as follows: Section 2 presents a theoretical background regarding strategies commonly used to address fault prediction in manufacturing systems. Section 3 describes the identified four challenges of classification-based fault prediction in more detail. Section 4 reviews several solutions aimed at dealing with the stated challenges. Section 5 presents our concluding remarks.

2. MANUFACTURING FAULT PREDICTION STRATEGIES

Methods to predict the occurrences of failure in production represent an important theme in engineering research. Different strategies for this have been proposed from distinct literatures. Nguyen and Medjaher (2019) establish an important classification of these strategies in two main groups: model-based techniques and data-driven techniques.

To achieve fault prediction capability, the model-based group is mainly focused in the development of an effective stochastic model to represent the system degradation over time and guide predictive maintenance interventions. Examples of applications involve the use of the Markov decision process with dynamic programming techniques, as can be observed in the work of Huynh et al. (2019). Issues associated with this model-based view mostly revolve around the fact that it requires users to possess a highly refined knowledge of how equipment degradation occurs. It is also difficult to formalize or model the deterioration mechanisms, and even theoretically modelled systems may be not applicable in practice due to the amount of variables in real production systems that can affect the validity of a model (Huynh et al., 2019; Nguyen and Medjaher, 2019). To top it off, simplifications of the theoretical models may lead to non-optimized decisions, which one can argue that makes modelling lose its main purpose.

As opposed to the modelling-based perspective, the data-driven approach to fault prediction is valuable exactly because in its objective of making predictions about future behaviour it neither relies on prior knowledge of the complex deterioration mechanisms of real machines nor faces difficulties making a transition from theory to practice (Nguyen and Medjaher, 2019). Instead, these methods are capable of getting better at the task of predicting failures through learning from experience contained in data sets.

These methods can be divided in (i) *supervised learning techniques*, when training data is labelled and algorithms are aware of the correct responses that correspond to a given input (Gupta et al., 2019); (ii) *unsupervised learning techniques*, when unlabelled data is provided and the algorithms must learn from patterns on their own (Janjua et al., 2019), and; (iii) *semi-supervised learning*, when there are both sections of labelled and unlabelled data (Yin et al., 2019). Unsupervised and semi-supervised approaches mostly treat fault prediction as an anomaly detection problem, as can be seen in Strauß et al. (2018). In these cases, techniques such as the Isolation Forests proposed by Liu et al. (2008) can be used. Supervised learning fault prediction, differently, can be categorized between regression problems (mapping input variables into continuous functions), and classification problems (mapping input variables into discrete classes). Regression-based fault prediction focuses on attempting to predict the exact residual useful life (RUL) of an analysed instance, based on prior training with a data set labelled with RUL information (Nguyen and Medjaher, 2019). Classification-based fault prediction, on the other hand, attempts to classify an instance into categorical classes such as “not going to fail” and “imminent failure” (or even more categories representing different warning levels), without explicitly predicting the RUL; an example can be seen in Xu et al. (2019). Methods such as Support Vector Machines (SVM), Logistic Regression and Random Forests can be employed in these cases (Strauß et al., 2018). Notably, it is specifically in the context of classification-based techniques that we delimitate the analysis which is further conducted in this work. An illustration of how all the reviewed fault prediction strategies are hierarchically classified can be seen in Fig. 1.

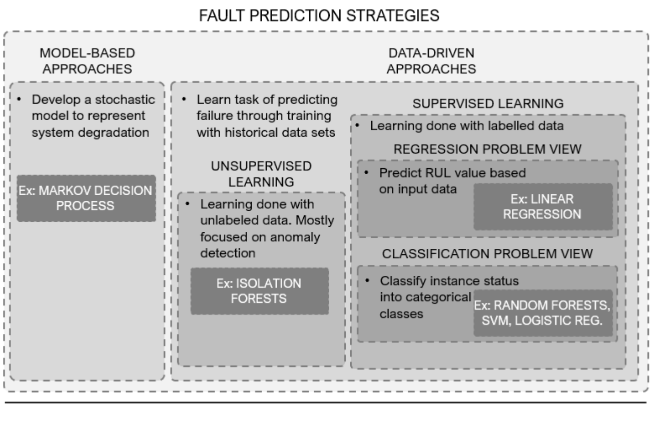

Fig. 1. Classification of fault prediction strategies.

3. CHALLENGES OF CLASSIFICATION-BASED FAULT PREDICTION

Despite not requiring extensive prior knowledge of degradation mechanisms, classification-based fault prediction has other pitfalls, which we have summarized in four main challenges that are addressed in this work.

- C#1: limited amount of training data.
- C#2: unbalanced training data sets (in regard to the distribution of classification classes).
- C#3: uncertainty regarding which variables should be monitored.
- C#4: uncertainty regarding how exactly historical data should be employed in training.

We begin by C#1, the challenge of having a limited amount of training data to develop an effective fault prediction service. The link between a big amount of training data and the performance of data-driven methods has been documented in a series of important works. Banko and Brill (2001), for instance, published one of the first key works demonstrating how data volume may benefit performance, regardless of specific choices of algorithms. Particularly, the authors observed that, for natural language processing tasks, a series of learning-based techniques had a significant improvement of performance when trained with larger data sets. Similarly, Halevy et al. (2009) discussed applications where a similar learning algorithm would achieve an improved performance when exposed to a much greater volume of training instances. As a last example, when discussing the potential of deep learning techniques, Najafabadi et al. (2015) argued that these sophisticated techniques are inherently suited to exploit massive amounts of data, through which it is possible to explore and understand highly complex data patterns. That being said, many organizations just recently started to systematically gather operational data, as IoT technologies have risen in popularity and smart factory projects began to gain traction. As a consequence, it is not uncommon for companies to have only a limited volume of operational data available to train a classifier, which is employed to deliver the fault prediction service.

To better assess the second challenge, it is adequate to assume that in most of the mature manufacturing systems, failures related to production equipment can be rather unusual events (Janjua et al., 2019). Consequently, operational data used as input to train the fault prediction classifier might have an unbalance regarding the number of instances labelled in each particular class, e.g. the number of instances labelled as 'not going to fail' is much greater than the number of instances labelled as 'imminent failure' (considering a setting where only these two classes are used to categorize instances). The main issue derived from using unbalanced training sets comes from the fact that, in these cases, standard classification approaches will generally be biased towards the majority class (Nanni et al., 2015), resulting in a decreased capacity to correctly deal with the minority class.

Since the gathered data used as input to train the classifier may include any number of variables of interest (e.g. temperature, vibration), the third challenge is related to the uncertainty organizations face when choosing which of these variables should be tracked by sensors. In this regard, according to Turabieh et al. (2019), one of the major issues that affect the performance of the learning algorithms is data dimensionality. Furthermore, monitoring every possible variable is not only unpractical, but may also harm the classifier's performance (Xue et al., 2016). From the perspective of the classification problem at hand, adequately choosing which operational variables are going to be monitored is fundamentally what is typically addressed as feature selection, i.e., a method employed to select an optimum subset of relevant features which leads to the least error for learning a classification model (Zorarpacı and Özel, 2016).

As sensors are deployed in the shop-floor and gathered measures are saved in the cloud, every variable of interest eventually becomes associated with a vector of historical measures. Hence, the fourth challenge addressed in this work deals with deciding how exactly this historical data should be fed to the fault prediction classifier, i.e., how far back should the classifier reach to understand the patterns that lead to particular classification classes, and also what type of data should be input (only the raw historical measures, summarizing statistics, or both).

All these addressed challenges are related to the process of training the fault prediction classifier. Therefore, a visual depiction of the relationship between the challenges and the different dimensions of a training dataset is provided in Fig. 2. Every data instance is portrayed as a square 'block' of information with two dimensions (the selected features or variables of interest – C#3, as well as the utilized historical data regarding each feature – C#4). The data blocks are stacked together to build the training set, providing the shape with a third dimension (quantified by the number of training instances – C#1). Lastly, instances with distinct labels are colored differently – C#2.

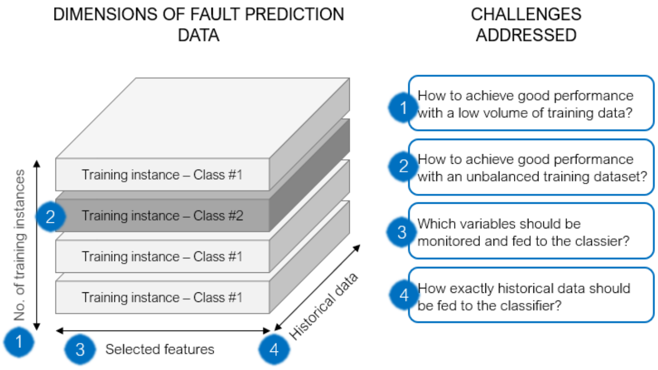

Fig. 2. Dimensions of data and the challenges of classification-based fault prediction.

4. APPROACHES TO HANDLE FAULT PREDICTION CHALLENGES

In this section, we review available approaches aimed at dealing with the challenges which commonly hinder effective fault prediction.

Firstly, we address classification-based machine learning methods, focusing specially in approaches that have been deployed in similar problems and have been found to perform well with a low amount of training data (C#1). Cerrada et al., (2016) employed the Random Forest (RF) classification technique to deal with a fault diagnosis problem in spur gears; the authors argue that the RF technique has shown adequacy to handle applications in the field of engineering where there is a low number of available learning data. RF classifiers consist in the combination of multiple independently sampled tree predictors and are presented in detail in Breiman (2001).

Fault prediction classifiers built on deep learning models, such as the application of a Long Short-Term Memory (LSTM) network as can be observed in Nguyen and Medjaher (2019) typically require a large volume of input data, which turns these approaches away from the scope of our work. Nonetheless, Xu et al. (2019) offered a strategy to somehow take advantage of the benefits of deep learning while handling the low data volume issue, through the concept of transfer learning (i.e. train a model in a source domain and apply the model in a different, but analogous, target domain). The authors proposed a digital twin assisted fault diagnosis approach in which a deep learning model is fully trained in the virtual domain, using abundant simulation data as input and bypassing the real-world data availability constraints; Thereafter, the model is transferred and minimally adjusted to the physical space to perform fault prediction.

Next, we review strategies designed to deal with unbalance among classification classes (C#2). Classification class unbalance has been studied by a number of authors. The works of Haixiang et al. (2017) and Nanni et al. (2015) cover a plethora of techniques to deal with the matter. Two main approaches can be verified: the first approach consists in the use of preprocessing techniques, which are executed before the development of the machine learning model; the second approach consists in the use of cost-sensitive learning methods, in which a higher cost is incurred when the machine learning model misclassifies an instance from the minority class, providing more robustness against class unbalance during the learning process. Cost-sensitive methods have been shown by Haixiang et al. (2017) to be less popular than resampling, but more computationally efficient. Furthermore, the authors identified the most popular preprocessing technique to be resampling, which is operationalized by either an over-sampling (adding more instances) of the minority class to reduce or eliminate the unbalance of the dataset, or an under-sampling (reducing instances) of the majority class to reach the same goal. Of particular interest to our analysis (due to the objective of also dealing with a lack of training data) is the over-sampling approach, which is typically implemented through random duplication or the more refined Synthetic Minority Oversampling Technique (SMOTE), proposed by Chawla et al. (2002).

For the next topic, we assess feature selection strategies for classification problems that can be encompassed in the context of this work (C#3). Cai et al. (2018) argue that for these particular cases, the concept of relevance or correlation between a specific feature and the class label is central to selecting features. The authors also state that heuristic methods are commonly employed to determine an optimal subset of features. In this regard, Xue et al. (2016) state that evolutionary heuristic techniques based on the genetic algorithms (GA) have been widely applied to feature selection problems. The authors note that GA's typical binary codification naturally fits with indicating whether a feature is selected or not. Furthermore, as illustrated by Cerrada et al. (2016) use of GA to enable feature selection for a fault prediction RF classifier, it is possible to optimize the chosen features in relation to multiple performance metrics, which are embedded in the heuristic's fitness function. Focusing on different types of techniques, Speiser et al. (2019) compared the performance of several feature selection methods specifically in the context of the RF classifier (which was previously addressed in this section). The authors found out that the strategies which granted the lowest out-of-bag (OOB) error metrics came from the VSURF package (Genuer et al., 2015) and the Boruta package (Kursa and Rudnicki, 2010). On the other hand, if the area under the curve (AUC) is used as the performance metric, Jiang's approach (Jiang et al., 2004) was found to be the most effective method. Particularly in the tests carried out on datasets with more than 50 features, both varSelRF (Díaz-Uriarte and Alvarez de Andrés, 2006) and Boruta packages had low OOB errors and low computational times (despite having the lowest OOB error, VSURF exhibited a high computational time).

Hereafter, we address how historical data may be fed to supervised classification algorithms (C#4). Brownlee (2016) presents two particular ways. The first consists of adding lag features containing raw features' values of different ages. The second consists of adding descriptive statistics summarizing a time window of the features' historical values. As can be observed in both these approaches, historical data of a particular feature end up being represented somehow as additional features, which means that a strategy of how to optimally employ historical data can also be determined by the feature selection methods discussed in the previous paragraph. As an example, Donate and Cortez (2014) utilized evolutionary heuristics to determine if time lags of a specific size were going to be used as input for a time series forecaster application.

A synthesis of all the approaches reviewed in this section to deal with fault prediction challenges can be seen in Fig. 3.

Lastly, we show in Fig. 4 how a subset of the aforementioned techniques can be integrated into a unified architecture to enable classification-based fault prediction, specifically when the four addressed challenges are in play. This framework is by no means the only possible way in which the techniques reviewed in this section can be arranged, as a plethora of distinct strategies may be developed.

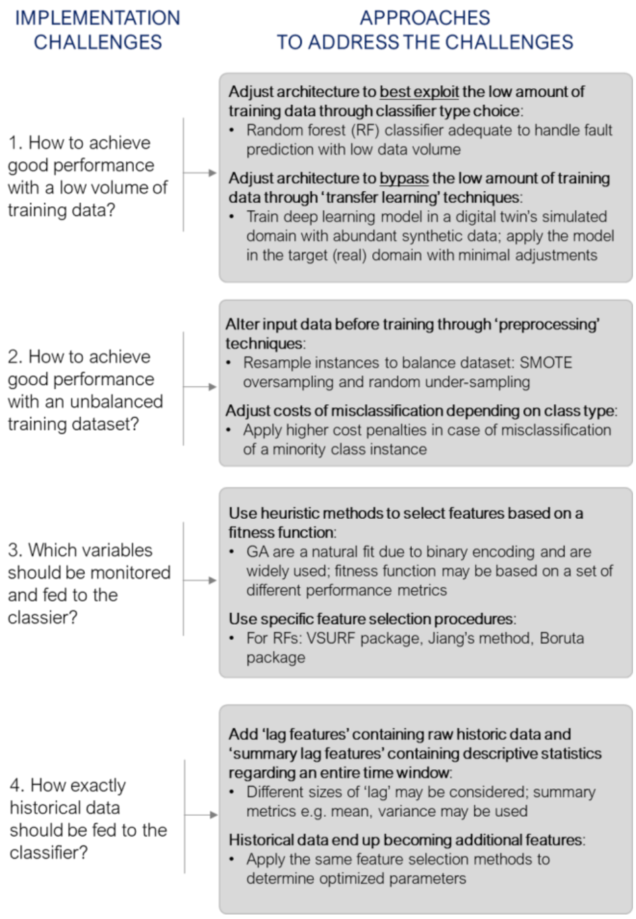

Fig. 3. Synthesis of the approaches aimed to address fault prediction challenges.

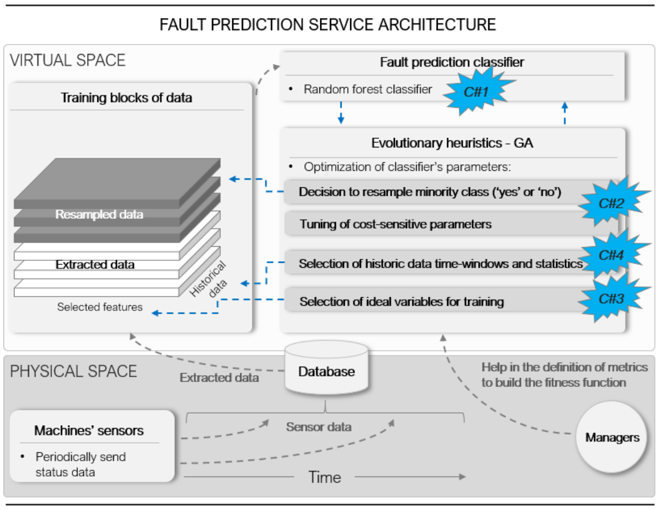

Fig. 4. Fault prediction service architecture designed to overcome four common challenges of implementation.

In the architecture presented in Fig. 4, we conceptually illustrate how an evolutionary heuristic based on GA can optimize several parameters of a fault prediction classifier, in order to overcome the challenges addressed in this work. The RF technique is chosen due to its potential compatibility with smaller training data sets (in the context of C#1). The GA fitness function should be developed based on metrics (and weights of importance among metrics) determined with the support of managers, so that the concept of performance sought by the GA is clear and aligned with the management's view. Several parameters of the classifier are subject to the GA optimization. The parameters are organized as follows: two parameters, represented by two bits of binary data, are optimized to mitigate the effect of unbalanced datasets in the context of C#2: firstly, a decision is made to define whether resampling should occur or not; secondly, a decision is made to define whether the values of the weights associated with the classes of the RF should be tuned or not (if tuning is activated, the weights are adjusted inversely proportional to the class frequencies). This way, both the reviewed techniques of preprocessing and cost-sensitiveness can be deployed in the search of better performance. Next, the other parameters are used to perform the feature selection. All potential variables of interests (in the context of C#3), as well as their respective vectors of historical data and summarizing statistics (in the context of C#4) are also coded as single bits of binary data: when a bit is set to 1, the 'feature' it represents is activated and serves as an input to train the classifier; when a bit is set to 0, the 'feature' is not used to train the classifier. The GA looks for the subset of features that generates the greatest level of fitness.

5. CONCLUSIONS

Our work comes with limitations. Firstly, we delimitate our study to the challenges that specifically affect the service providing capabilities of fault prediction systems, and not the development of their whole structure. Therefore, the pitfalls regarding the implementation of real-time data acquisition and storage technologies lie outside the boundaries of our analysis. Secondly, we specifically address challenges that besiege fault prediction when viewed as a supervised learning classification problem; this view, admittedly, is not comprehensive (see Section 2). Thirdly, the challenges addressed in this work are presented in a broad perspective, as we do not look at the specific issues and characteristics of any particular domain. In this regard, an analysis capable of providing tailored suggestions to deal with a diverse set of application domains represents an important direction for future work.

ACKNOWLEDGEMENTS

This research was conducted through a grant funded by project AGENC/04312/2019 registered at the Associação Paranaense de Cultura – APC.

REFERENCES

Banko, M. and Brill, E. (2001). Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 26–33.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32.

Brownlee, J. (2016). Basic Feature Engineering With Time Series Data in Python. *Time series*.

Burke, R., Mussomeli, A., Laaper, S., Hartigan, M. and Sniderman, B. (2017). The smart factory. *Deloitte University Press*.

Cai, J., Luo, J., Wang, S. and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79.

Cerrada, M., Zurita, G., Cabrera, D., Sánchez, R. V., Artés, M. and Li, C. (2016). Fault diagnosis in spur gears based on genetic algorithm and random forest. *Mechanical Systems and Signal Processing*, 70–71, 87–103.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357.

Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(3), 1-13.

Donate, J.P. and Cortez., P. (2014). Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting. *Applied Soft Computing Journal*, 23, 432-443.

Genuer, R., Poggi, J.M. and Tuleau-Malot, C. (2015). VSURF: An R package for variable selection using random forests. *R Journal*, 7(2), 19–33.

Gupta, L., Salman, T., Zolanvari, M., Erbad, A. and Jain, R. (2019). Fault and performance management in multi-cloud virtual network services using AI: A tutorial and a case study. *Computer Networks*, 165, 1-6.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H. and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73, 220–239.

Halevy, A., Norvig, P. and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2), 8–12.

Huynh, K. T., Grall, A., & Bérenguer, C. (2019). A parametric predictive maintenance decision-making framework considering improved system health prognosis precision. *IEEE Transactions on Reliability*, 68(1), 375–396.

Janjua, Z.H., Vecchio, M., Antonini, M. and Antonelli, F. (2019). IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge. *Engineering Applications of Artificial Intelligence*, 84, 41–50.

Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., … Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5(81).

Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., … Noh, S.D. (2016). Smart Manufacturing: Past Research, Present Findings, and Future Directions. *International Journal of Precision Engineering and Manufacturing-green Technology*, 3(1), 111–128.

Kursa, M.B. and Rudnicki, W.R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36(11), 1–13.

Li, B., Zhang, L. and Chai, X. (2010). Introduction to Cloud Manufacturing. *ZTE Communications Technology*, 16, 5-8.

Liao, Y., Deschamps, F., Loures, E.F.R. and Ramos, L.F.P. (2017). Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. *International Journal of Production Research*, 55(12), 3609–3629.

Liu, F.T., Ting, K.M., Ting and Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R. and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1), 1–21.

Nanni, L., Fantozzi, C. and Lazzarini, N. (2015). Coupling different methods for overcoming the class imbalance problem. *Neurocomputing*, 158, 48–61.

Nguyen, K.T.P. and Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, 188(February), 251–262.

Qiao, Q., Wang, J., Ye, L. and Gao, R.X. (2019). Digital twin for machining tool condition prediction. *Procedia CIRP*, 81, 1388–1393.

Speiser, J.L., Miller, M.E., Tooze, J. and Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, 134, 93–101.

Stojanovic, N. and Milenovic, D. (2019). Data-driven Digital Twin approach for process optimization: An industry use case. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 4202–4211.

Strauß, P., Schmitz, M., Wöstmann, R. and Deuse, J. (2018). Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 1474–1483.

Turabieh, H., Mafarja, M. and Li, X. (2019), Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. *Expert Systems with Applications*, 122, 27–42.

Xu, Y., Sun, Y., Liu, X. and Zheng, Y. (2019). A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning. *IEEE Access*, 7, 19990–19999.

Xue, B., Zhang, M., Browne, W.N. and Yao, X. (2016). A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, 20(4), 606–626.

Yin, L., Wang, H., Fan, W., Kou, L., Lin, T. and Xiao, Y. (2019). Incorporate active learning to semi-supervised industrial fault classification, *Journal of Process Control*, 78, 88–97.

Zhuang, C., Liu, J. and Xiong, H. (2018). Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *International Journal of Advanced Manufacturing Technology*, 96(1–4), 1149–1163.

Zorarpacı, E. and Özel, S.A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, 62, 91–103.