2.2 Random forest

Introduced by Breiman (2001), Random forests are an integration of tree predictors where every tree depends on the values of a random vector separately (Breiman, 2001). A similar distribution applies for all the trees in the forest. The tree classifier of a forest has a generalization error which relies on the strong correlation between all trees in the forest. Classification accuracy increases significantly when the group of trees is enlarged. A primary example is bagging, where to raise every tree, an arbitrary selection (without replacement) is done from the set examples. Another example is random split selection where arbitrarily, the split is selected from among the K best splits at every single node. A random forest algorithm consists of rotating many decision trees that are randomly constructed and then generating them. Bootstrap sampling (OOB: Out-Of-Bag sampling) is used in RF to have a better estimate of the distribution of the original data set. Indeed, bootstrapping means randomly selecting a subset of the data for each tree rather than using all the data to build the trees. In statistical terms, if the trees are uncorrelated, this reduces the forecast variance. The main advantage of random forests is their resistance to variances and biases.
The random forest algorithm is used in the regression case to predict a continuous dependence and classification variable in order to predict a categorical dependent variable. For the regression type, a random forest consists of a set of simple prediction trees; each is capable of producing a numerical response when presented with a subset of explanatory variables or predictors. The error in this forecast is called Out Of Bag (OOB).
For the classification type, a categorical variable with N modalities is broken down into a disjunctive array (with N-1 variables) according to a 0-1 coding scheme. Thus, a categorical variable with N modalities can be considered as a set of N-1 variables, of which only one will assume the value 1 for a given observation. In fact, the ability to make predictions on a random subset of predictive variables is one of the strengths of the Random Forest module, which makes it particularly well suited to processing data sets with extremely high predictive variables. This random feature selection encourages systems diversity, and by the end, it enhances classification performance. The random forest is constructed by sampling arbitrarily the features subset as well as the training subset with regard to every system. The majority vote combines the final prediction. Finally, the random forest attains a favorable and vigorous performance with various applications (Cheng, Chan, & Qiu, 2012; Friedman, Hastie, & Tibshirani, 2001).

2.3 Particle swarm optimization

PSO called swarm intelligence or collective intelligence is developed by Eberhart and Kennedy in 1995 (Kennedy & Eberhart, 1995; Shi, 2001). The overall behavior of the PSO is not programmed in advance but emerges from the sequence of elementary interactions between individuals. In this context, many researchers have applied the PSO in several machine learning for optimization (Lin, Ying, Chen, & Lee, 2008; Xiaodan, 2017).
The Optimization method of PSO can be iterative; each particle consists of changing the velocity toward its fitness value and global version of PSO. For the movement, the particle must decide on its next movement (its new speed) by linearly combining three pieces of information: its current speed V_{ij}^{n} (velocity), its best performance already found P_{ij}^{n} and which is known as the personal best position (*pbest*), and the best performance of its neighbors or informants P_{gj}^{n} known as the global best position (*gbest*) (See equation below). The iteration of the velocity and position of the particles are getting by following equations:

$$V_{ij}^{k+1} = w * V_{ij}^{k} + c_1 * r_1\left(P_{ij}^{k} - x_{ij}^{k}\right) + c_2 * r_2(P_{gj}^{k} - x_{ij}^{k})$$

$$x_{ij}^{k+1} = x_{ij}^{k} + V_{ij}^{k+1}$$

Where $P_{ij} = (p_{i1}, p_{i2}, \dots p_{im})$ and $P_{gj} = (p_{g1}, p_{g2}, \dots p_{gm})$. k is the number of iteration, x_{ij}^{k} present the particle position. Positive coefficient r_1 and r_2 are random number, generated uniformly in the range [0 1]. w presents the inertia coefficient of PSO algorithm. This weight is updated according to the following equation (Chen et al., 2011):

$$w = w_{min} + (w_{max} - w_{min})\,\frac{(t_{max} - t)}{t_{max}}$$

Where t_{max} presents the maximum number of iterations. Usually, the inertia coefficient is generated in the range [0.4 0.9] (Chen et al., 2011). The coefficients c_1 and c_2 are calculated as follows:

$$c_1 = (c_{1f} - c_{1i})\frac{t}{t_{max}} + c_{1i}$$

$$c_2 = (c_{2f} - c_{2i})\frac{t}{t_{max}} + c_{2i}$$

Withc_{1f},c_{1i}, c_{2f} , and c_{2i} are a positive constants.
The position of the particles as well as their initial velocity must be initialized randomly according to a uniform law.
The original process for implementing the local version of PSO is as follows:
Step 1: Initialize randomly a population. Step 2: Measure the fitness of each particle in the population (calculate fitness score for each particle using selected features). Step 3: Update the velocity and position of each particle by looking for the best performance for each particle (local optimum). If the current fitness is better than the previous fitness, the previous *pbest* is replaced with the current *pbest*. Finally, in step 4: continues until the process converges. Stop the algorithm if the termination criterion is satisfied; otherwise, return to step 2.

3 Proposed PSO-RF optimization approach

In this paper, the PSO-RF model is developed and applies for forecasting of obsolescence risk. This main objective of this paper is to apply particle swarm optimization to enhance the classification performance of random forest algorithm by searching for the optimal parameters for RF and discovering the best subset of features as well.
The proposed PSO-RF optimization approach consists of 4 steps as follows:
Step 1: The initialization of the data processing. The dataset is initialized to construct the RF model based on supervised learning (known data), also the data is split into two groups (training and testing) randomly.
Step 2: The initialization of PSO and RF parameters. For the PSO, the number of generations, population size (number of particles) and so one, are initialized. The position and velocity set to x^0_i and v^0_i respectively, are determined as well. Each particle has (1) dimension (d) which is the length of features and the number of parameters to be optimized. (2) Position (x): position of the i^{th} particle. The initialization the parameters of RF (number of trees and the number of variables to split on at each node (mtry)) are included into the algorithm as well.
Step 3: PSO is adopted to construct PSO-RF model. In fact, PSO is looking for the optimal solution of particles by evaluation of fitness based on the update particle velocity and its position. However, if the current fitness is better than the previous fitness, the previous *pbest* is replaced with the current *pbest* until to find the optimal solution (if current *gbest* is better than previous *gbest*, then replace *gbest* score and *gbest* particle. If the particle is already created and evaluated, then generate a new one. As introduced by Kennedy, the PSO is searching in a discrete space (0 or 1). Each feature in the PSO represents a binary bit (0 or 1), where 1 represents a selected feature while 0 represents a feature that is not selected. The features selected are based on the particle's position.
Step 4: presents the training subset of features selection and parameters optimization is introduced in the RF model; therefore the PSO-RF is obtained. For each terminal node of the tree, these steps are repeated until the specified number of trees is reached and the minimum node size is obtained. Next, the Out of Bag (OOB) error for the model is estimated. For classification, OOB error is estimated as the proportion of times that the categorical variable is not equal to the true class prediction. Finally, the output is represented as an ensemble of trees$\{T_b\}$. To make prediction at a new point x, let $\widehat{c_b}(x)$ be the class predicting of the b^{th} random forest tree. The equation is given as follows:

$$\widehat{C_{rf}^{b}} = majority\ vote\{\widehat{c_b}(x)\}.$$

Finally, the precision of the model is calculated by comparing the current state with the state predicted by the model using a confusion matrix. The fitness function is calculated as follows:

$$f = accuracy\ (\text{AUC})$$

The architecture of the proposed method is illustrated in Figure 1.

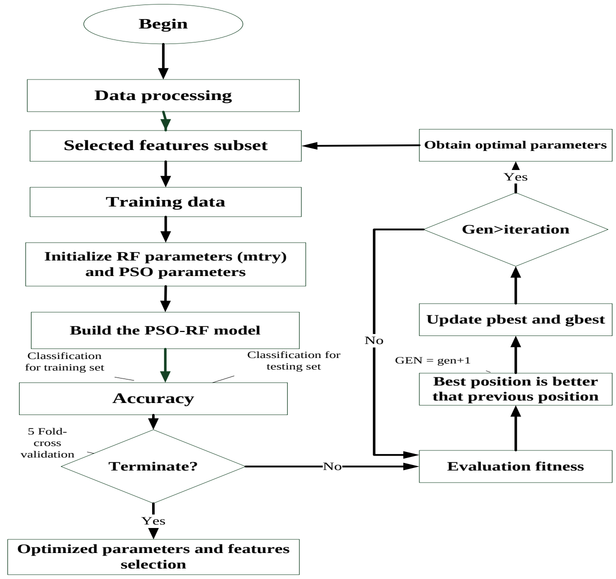

Figure 1. Architecture of the PSO-RF Model

4. Numerical case study

The R programming is adopted to develop the PSO-RF. To measure the performance of PSO-RF approach, a dataset is used, taken from (Jennings et al., 2016). A total of 999 instances provided information about cellphones and thirteen attributes were used as predictive variables. The output present two class (in production/end-of-life). About 70% of the data are randomly selected as the training set for constructing the model while 30% of the data are used as the test set to validate the model accuracy. To obtain a better estimation of classification accuracy, the K-Folds cross-validation method presented by (Salzberg, 1997) was applied and set to 5.
For features selection, we have introduced an assumption for the model: selected features should be greater or equal to 2. The parameters setting for PSO is obtained as follows: number of iteration and number of populations (particles) are set to 10 and 50 respectively. In fact, the iteration tries to generate new particle, however when there is so many particles which were already used, it will try to find other ones. If the solution is already near to the optimal, the result will not be change. Based on the experimental results, PSO is faster to find the solution with small iteration. As suggested by (Ratnaweera, Halgamuge, & Watson, 2004): $c_{1i} \leftarrow 2.5$, $c_{1f} \leftarrow 0.5$, $c_{2i} \leftarrow 0.5$, $c_{2f} \leftarrow 2.5$, $w_{max} \leftarrow 0.9$, $w_{min} \leftarrow 0.4$.
To examine the effectiveness of this approach, the PSO-RF is benchmarked with GA-RF (random forest based on genetic algorithm). The characteristic of GA is as follows: maximum generations, population, crossover and mutation are set to 20, 50, 0.8 and 0.1 respectively.

4.1 Experimental results and discussion

The results in this paper are described in terms of accuracy (AC), error rate, sensitivity (SE), specificity (SP), and Cohen's KAPPA, which are calculated by the following equations (Woods & Bowyer, 1997; Woods, Kegelmeyer, & Bowyer, 1997):

$$AC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$error\ rate = 1 - AC$$

$$SE = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

TP, *TN*, *FP* and *FN* are defined as true positive, true negative, false positive and false negative. For KAPPA equation, Pr(a) presents the probability of success of classification (accuracy) and Pr(e) presents the probability of success due to chance. In order to validate the accuracy of the proposed PSO-SVM algorithm, the results obtained by PSO-RF is compared with the GA-RF (RF with genetic algorithm) developed by (Grichi, Beauregard, & Dao, 2018). The classification accuracy rates of GA-RF are cited from their original papers that was achieved a good predictive performance. The accuracy of PSO-RF, GA-RF and RF was initially presents by a confusion matrix (See Table 1) for testing set. The comparisons of the algorithms are shown in Table 2 and 3.

Table 1. Confusion matrix of GA-RF, PSO-RF and RF algorithms

		Predict	
	Actual	Available	Discontinued
GA-RF	Available	**166**	12
	Discontinued	8	**113**
PSO-RF	Available	**170**	9
	Discontinued	4	**116**
RF	Available	**163**	15
	Discontinued	11	**110**

Table 2. The accuracy measures comparison across algorithms (testing sample)

Accuracy measure	RF (%)	GA-RF (%)	PSO-RF (%)
Accuracy	91.3	93.3	**96**
No Information Rate	58.2	58.2	58.2
Kappa	82.1	86.2	**91**
Sensitivity	88	90.4	**97.7**
Specificity	93.7	**95.4**	92.8
Error rate	8.7	6.7	**4**
Balanced Accuracy ((Sensitivity+ Specificity)/2)	90.8	92.9	**95.25**

Table 3. The accuracy measures comparison across algorithms (training sample)

Accuracy measure	RF	GA-RF	PSO-RF
Accuracy	94.7	**98.6**	**98.6**
No Information Rate	58.3	58.3	0.583
Kappa	89.1	97.1	97
Sensitivity	91.4	98	**98.8**
Specificity	97.1	**99**	98.3
Error rate	5.3	**1.4**	**1.4**
Balanced Accuracy	94.3	98.5	**98.55**

The result obtained from the PSO-RF model proves that the prediction through parameters optimization and choosing the best small features can improve significantly the classification accuracy of the RF. The experiments results were compared with RF-PSO and RF. PSO-RF approach yielded a higher classification accuracy rate compared to the other approaches. Thus, PSO-RF yielded more appropriate subset with few iteration.
Figure 2 shows the Receiver Operator Characteristic curve, that presents the false positive rate vs. true positive rate. These curves present as follows: green, blue, orange for PSO-RF, RF and GA-RF respectively.

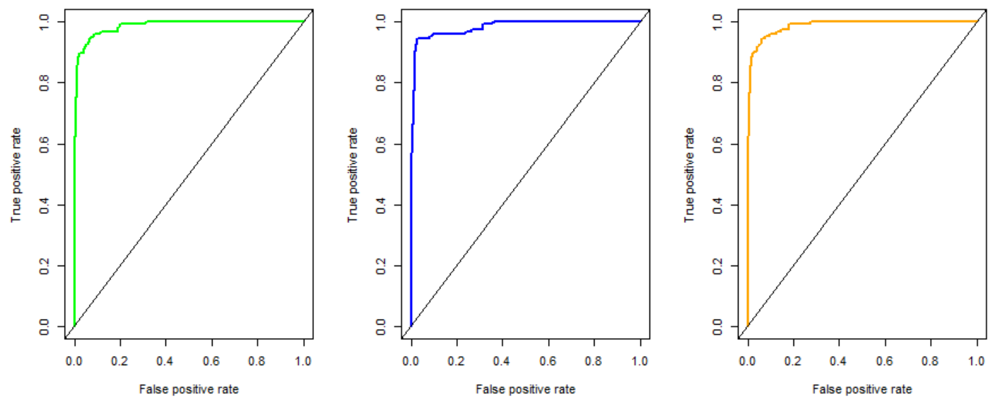

Figure 2. ROC curve

The use of feature selection and parameter optimization were found to improve the classification accuracy rate for random forest to improve the forecasting of obsolescence risk. Experimental results show that PSO-RF has better performance than that of GA-RF.

5. Conclusion

This paper presents an improved approach for obsolescence forecasting risk with a high degree of accuracy based on machine learning and meta-heuristic PSO. PSO search for the optimal parameter value for RF to obtain a subset of beneficial features. The optimal set features were adopted for the training and testing of RF model to improve the classification accuracy of the model. In order to validate this approach, PSO-RF was compared to RF and GA-RF. Experimental results show that PSO-RF outperformed GA-RF with 96% of accuracy.
For future work, larger data with more features can give more accurate results. Other datasets and real-world problems for obsolescence forecasting can be tested using this approach.
Other optimization algorithms can also be used, such as ant colony which is widely used for optimization, and compare it with the existing approach.

References

Bartels, B., Ermel, U., Sandborn, P., & Pecht, M. G. (2012). *Strategies to the prediction, mitigation and management of product obsolescence* (Vol. 87): John Wiley & Sons.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Chen, H.-L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.-J., & Liu, D.-Y. (2011). A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. *Knowledge-Based Systems, 24*(8), 1348-1359.

Cheng, Y.-Y., Chan, P. P., & Qiu, Z.-W. (2012). *Random forest based ensemble system for short term load forecasting.* Paper presented at the Machine Learning and Cybernetics (ICMLC), 2012 International Conference on.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1): Springer series in statistics Springer, Berlin.

Gao, C., Liu, X., & Wang, X. (2011). *A model for predicting the obsolescence trend of FPGA.* Paper presented at the 2011 9th International Conference on Reliability, Maintainability and Safety: Safety First, Reliability Primary, ICRMS'2011, June 12, 2011 - June 15, 2011, Guiyang, China.

Grichi, Y., Beauregard, Y., & Dao, T.-M. (2018). Optimization of Obsolescence Forecasting Using New Hybrid Approach Based on the RF Method and the Meta-heuristic Genetic Algorithm. *American Journal of Management*(18(2)). (Forthcoming)

Grichi, Y., Beauregard, Y., & Dao, T. (2017). *A random forest method for obsolescence forecasting.* Paper presented at the Industrial Engineering and Engineering Management (IEEM), 2017 IEEE International Conference on (pp. 1602-1606). IEEE.

Jennings, C., Wu, D., & Terpenny, J. (2016). Forecasting obsolescence risk and product life cycle with machine learning. *IEEE Transactions on Components, Packaging and Manufacturing Technology, 6*(9), 1428-1439. doi: 10.1109/TCPMT.2016.2589206

Josias, C., Terpenny, J. P., & McLean, K. J. (2004). *Component obsolescence risk assessment.* Paper presented at the IIE Annual Conference and Exhibition 2004, May 15, 2004 - May 19, 2004, Houston, TX, United states.

Jungmok, M., & Namhun, K. (2017). Electronic part obsolescence forecasting based on time series modeling. *International Journal of Precision Engineering and Manufacturing, 18*(5), 771-777. doi: 10.1007/s12541-017-0092-6

Kennedy, J., & Eberhart, R. (1995). *PSO optimization.* Paper presented at the Proc. IEEE Int. Conf. Neural Networks.

Lin, S.-W., Ying, K.-C., Chen, S.-C., & Lee, Z.-J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications, 35*(4), 1817-1824.

Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on evolutionary computation, 8*(3), 240-255.

Rojo, F. R., Roy, R., & Kelly, S. (2012). *Obsolescence risk assessment process best practice.* Paper presented at the Journal of Physics: Conference Series.

Rojo, F. R., Roy, R., & Shehab, E. (2010). Obsolescence management for long-life contracts: state of the art and future trends. *International Journal of Advanced Manufacturing Technology, 49*(9-12), 1235-1250. doi: 10.1007/s00170-009-2471-3

Salzberg, S. L. (1997). On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery, 1*(3), 317-328. doi: 10.1023/a:1009752403260

Sandborn, P. (2007). Software obsolescence-Complicating the part and technology obsolescence management problem. *IEEE Transactions on Components and Packaging Technologies, 30*(4), 886-888.

Sandborn, P. (2013). Design for Obsolescence Risk Management. *Procedia CIRP, 11*(0), 15-22. doi: http://dx.doi.org/10.1016/j.procir.2013.07.073

Sandborn, P., Prabhakar, V., & Ahmad, O. (2011). Forecasting electronic part procurement lifetimes to enable the management of DMSMS obsolescence. *Microelectronics Reliability, 51*(2), 392-399. doi: 10.1016/j.microrel.2010.08.005

Shi, Y. (2001). *Particle swarm optimization: developments, applications and resources.* Paper presented at the evolutionary computation, 2001. Proceedings of the 2001 Congress on.

Solomon, R., Sandborn, P. A., & Pecht, M. G. (2000). Electronic part life cycle concepts and obsolescence forecasting. *IEEE Transactions on Components and Packaging Technologies, 23*(4), 707-717. doi: 10.1109/6144.888857

van Jaarsveld, W., & Dekker, R. (2011). Estimating obsolescence risk from demand data to enhance inventory control—A case study. *International Journal of Production Economics, 133*(1), 423-431. doi: https://doi.org/10.1016/j.ijpe.2010.06.014

Voller, V. R., & Porté-Agel, F. (2002). Moore's Law and Numerical Modeling. *Journal of Computational Physics, 179*(2), 698-703. doi: http://dx.doi.org/10.1006/jcph.2002.7083

Woods, K., & Bowyer, K. W. (1997). Generating ROC curves for artificial neural networks. *IEEE Transactions on medical imaging, 16*(3), 329-337.

Woods, K., Kegelmeyer, W. P., & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE transactions on pattern analysis and machine intelligence, 19*(4), 405-410.

Wu, D., Jennings, C., Terpenny, J., Gao, R., & Kumara, S. (2017). *Data-Driven Prognostics Using Random Forests: Prediction of Tool Wear.* Paper presented at the ASME 2017 12th International Manufacturing Science and Engineering Conference collocated with the JSME/ASME 2017 6th International Conference on Materials and Processing.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., . . . Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1-37. doi: 10.1007/s10115-007-0114-2

Xiaodan, W. (2017). *Forecasting short-term wind speed using support vector machine with particle swarm optimization.* Paper presented at the 2017 International Conference on Sensing, Diagnostics, Prognostics and Control (SDPC), 16-18 Aug. 2017, Los Alamitos, CA, USA.

Yun, C., Ping, L., & Li, Y. (2010). *Aftermarket demands forecasting with a Regression-Bayesian-BPNN model.* Paper presented at the 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2010), 15-16 Nov. 2010, Piscataway, NJ, USA.

Biographies

Yosra Grichi is currently pursuing the Ph.D. degree at the department of mechanical engineering, École de technologie supérieure (ÉTS), University of Québec (Canada). She received a bachelor’s degree (Licence degree) and Master’s degree in mechanical engineering from University of Tunisia (ENSIT) in 2012 and 2015 respectively. Her current research interests include predictive modeling, management of manufacturing systems, data mining, machine learning and statistics.

Thien-My Dao is a Professor in the Department of Mechanical Engineering, École de technologie supérieure (ÉTS), University of Québec (Canada). He received a Bachelor’s Degree, Master’s Degree and PhD in Mechanical Engineering (Design option) from Sherbrooke University in 1969, 1971 and 1974 respectively. After four (4) years in industry sector, he joined the Department of Mechanical Engineering of École de technologie supérieure, University of Québec, in 1976 where he is teaching particularly production and operations management, quality management and design of manufacturing systems. His research interests are in optimization of the design, the reliability and the management of manufacturing systems.

Yvan Beauregard is a professor in the Department of Mechanical Engineering at École de Technologie Supérieure. Mr. Beauregard holds a Bachelor's degree in Industrial Engineering from École Polytechnique de Montréal, a Master's Degree in Administration from McGill University and a PhD in Mechanical Engineering from Concordia University. He has more than thirty years of industrial experience at Pratt & Whitney and IBM Canada. His research interests include among others operations and risk management, as well as product development performance improvement