International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7190-7196

Performance Assessment of Ensemble Decision Trees-based Fault Detection System in a Chemical Process

Mr. Rahul Shrivastava [1,*], Dr. K N Gupta[2], Prof. NN Dutta[3]

[1,2,3] Department of Chemical Engineering, Jaypee University of Engineering and Technology, AB-road, Raghogarh, Guna – 473226, Madhya Pradesh, India.

*(*Corresponding Author)*

Abstract

This paper presents an ensemble decision trees-based approach for fault detection and isolation in a chemical process. This is one of the well-known machine learning techniques, which is also known as random forest (RF) classifier. RF has some advantages over other classification techniques like It can deal with substantial number of variables effectively without over-fitting, It is quick and simple to execute, provides good results even with large number of classes, parameter tuning is not a problem and identification of the important variables in classification. By considering every one of these characteristics of RF was tested on benchmark Tennessee Eastman (TE) process. Results conclude that it can be proved one of the potential tools for fault detection and isolation in industrial systems.

Keywords: Fault diagnosis; Tennessee Eastman Process; Random forest;

INTRODUCTION

Although catastrophes and disasters may be irregular in chemical plants, but minor accidents are very common causes personal injury, illness, and material loses. Minor accidents are generally occurred due to faulty operating conditions. Fault is defined as a departure of an observed variable or a calculated parameter from an acceptable range. This departure may be occurred due to several reasons like sensor malfunctioning, control failure, valve choking etc. A good fault detection and diagnosis (FDD) system should be able to arrest these departures precisely and activate the necessary control actions. FDD works in four steps i.e., fault detection, isolation, estimation and reconstruction. Fault detection means to identify that something is going wrong with the process. Fault isolation detects the location of fault. Fault estimation will determine the fault magnitude and overall impact on the system. Finally, fault reconstruction takes the control action to bring the system back to the normal behavior. In this work, fault detection and isolation (FDI) were focused. Numerous methods have been proposed in last three decades to deal with FDI related problems in industrial systems. These methods can be divided into three categories: Quantitative model-based methods, qualitative model-based methods, and process history-based methods. A quantitative model-based method uses mathematical models (based on first principles) of physical system, whereas qualitative model-based method uses available information and knowledge of a physical system, process history-based methods require neither first principles-based modeling approach nor qualitative knowledge; instead, they need a large amount of historical data that contain the typical trends and fault information enabling the development of effective FDI system. In general, chemical processes are complex and non-linear in nature, so, it is hard to develop mathematical models or extract exact information and knowledge of such type of systems to build an effective FDI system. For these systems, researcher resorted to historical data-based methods, which proved to be more helpful than any other techniques. But these methods demand lot of historical data in faulty and normal operating conditions. These days, plants are vigorously instrumented, so historical data is easily accessible. All the machine learning techniques, for example artificial neural networks (ANN), support vector machines (SVM), RF *etc*, belong to the class of historical data-based methods. Indeed a considerable amount of work has been done in the past to create effective FDI system utilizing such methods. SVM and ANN have been assessed extensively by the FDD community *e.g.,* [4-16]. However each technique has its own advantages and disadvantages.

This paper focuses on the application and evaluation of RF classifier for FDI in TE process. Though random forest classifier has been used for FDI for other systems like [17-21], even in [18], researcher used RF on TE process but they just considered only two faults and mainly focused on to utilize RF variable importance measures to understand system dynamics, whereas in our study we considered 15 faults and focused on all the properties of RF like from the literature it is evident that RF has some advantages over other classification techniques like It can deal with substantial number of variables effectively without over-fitting, It is quick and simple to execute, provides good results even with large number of classes, parameter tuning is not a problem, identification of the important variables in classification and produces an internal unbiased estimate of the generalization error during the forest building process. By considering every one of these characteristics of RF, it was tested on TE process.

RANDOM FOREST

RF classifier is actually an ensemble of decision trees, which was suggested by Breiman [22-24]. The rationale behind using an ensemble of decision trees is that a set of classifiers

do perform better classifications than an individual classifier does. Each tree in the ensemble is grown at the behest of a randomly generated subset of observations and variables from the training data and the final predictions are made by utilizing some combining strategy (e.g., averaging in case of regression and majority of votes in classification) over the ensemble. Since each of these trees is grown using an injection of randomness, these procedures are called random forests. RF increases the diversity of the trees by making them grow from different training data subsets created through bagging or bootstrap aggregating. When we use a single decision tree for the classification, it faces the problem of pruning and over-fitting, whereas RF does not face the problem of over-fitting and trees are grown to full extent. Decision tree is grown by using a set of binary rules *i.e.,* recursive binary splitting to calculate a target value. But unlike of decision tree, in RF at each node, a given number (denoted by m) of input variables are randomly chosen and the best split is calculated only within this subset. To find out the best split at each node, one of the two criteria i.e., Gini index or Cross-entropy is used [25]. The Gini index was used as a best split criterion in this study. Class prediction for a given instance will be based on combined result of all the trees involved in the prediction model. In case of classification, class with majority of votes will be a winner.

According to Breiman [24], each tree in the ensemble is grown as follows: [*e.g.*, Fig.1] –

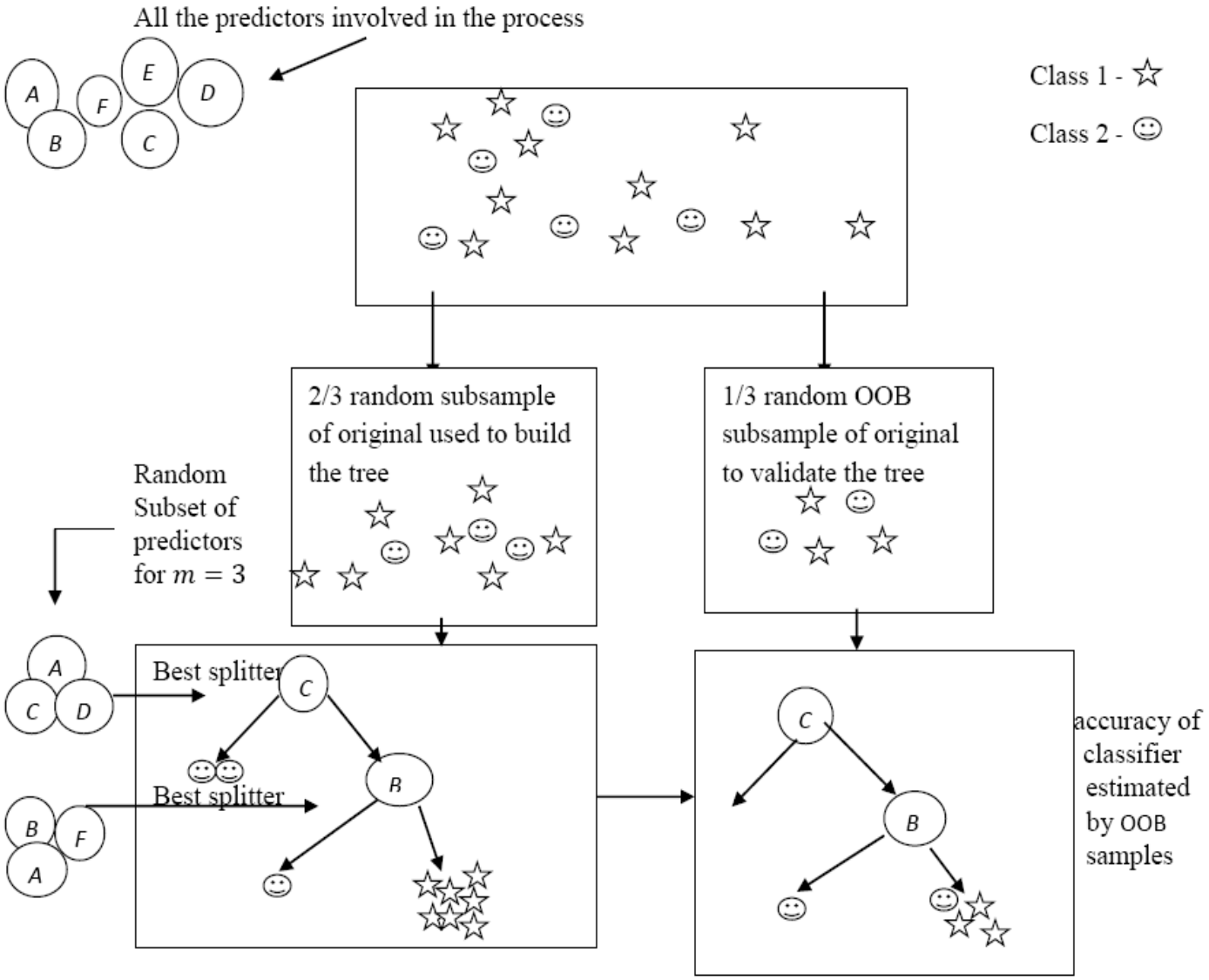

Figure 1. Decision tree building process in RF for classification [21]

From the data set, a bootstrap sample is drawn by random sampling with replacement. Each subset selected using bagging to make each individual i^{th} tree grow usually contains 2/3 of the dataset and 1/3 samples are kept aside which are

known as out-of-bag (OBB) samples. Note that a different OOB subset is formed for every i[th]-tree. These OOB elements, which are not considered for the training of the i[th] -tree, can be classified by the i[th]-tree to evaluate performance. Furthermore, when the RF makes a tree grow, it uses the best split of a random subset of input features or predictive variables in the division of every node. Therefore, this can decrease the strength of every single tree, but it reduces the correlation between the trees, which reduces the generalization error [24]. Here m is essentially the only one parameter needed to be adjusted in the algorithm. If there are a total of M input variables, m $(m << M)$ variables out of M are randomly selected at each node and the best split on these m is employed to split the node. The value of m is held constant during the forest building process. The tree is grown to the maximum size (i.e., until no further splits are possible) and not pruned back. Repeat the above steps until ntree (a sufficiently large number) such trees are grown. In other words, each tree is grown using a particular bootstrap sample, so ntree bootstrap samples will be drawn from the training data set. For estimating the importance of each input variable in prediction, firstly, the error e is calculated for OOB data, using the original training data set. Then for each input variable, x_p, where $p \in \{1, \ldots, k\}$, the p^{th} variable is randomly permuted to generate a new set of samples. The OOB estimate of error e_p with the new set of samples is computed and the difference of these two errors $(e\text{-}e_p)$ will be a measure of the p^{th} variable importance.

TE BENCHMARK PROCESS

This section gives a brief introduction of industrial benchmark TE process. This process was developed by Downs and Fogel [26] for the purpose of process monitoring, process control and FDD studies. This is widely accepted as a benchmark for control and FDD studies. RF method was tested on this process to demonstrate its characteristics. Fig.2 illustrates the process flow diagram of this process. This process has five major units, i.e. reactor, condenser, compressor, separator and stripper. Four reactants and two products are there in the process. Additionally, an inert and a by-product are also present making a total of 8 components. This process measures total 53 variables (41 are process variables and 12 are manipulated variables) (see [27] for variables description). This simulator has the facility to generate data for 21 faults (see Table 1 for type of faults). Several simulation schemes were developed over the years; we followed Ricker [28]. We have considered only initial 15 known faults in our analysis. Data-set and code are available on http://depts.washington.edu/control/LARRY/TE/download.html.Total 4500 samples were generated, out of which 3375 were used for training and 1125 for testing. Simulator was run for 51 hours and sampling time was 10 minutes (300 samples for each class/fault), because the disturbance was introduced after 1 hour and lasts for remaining 50 hours.

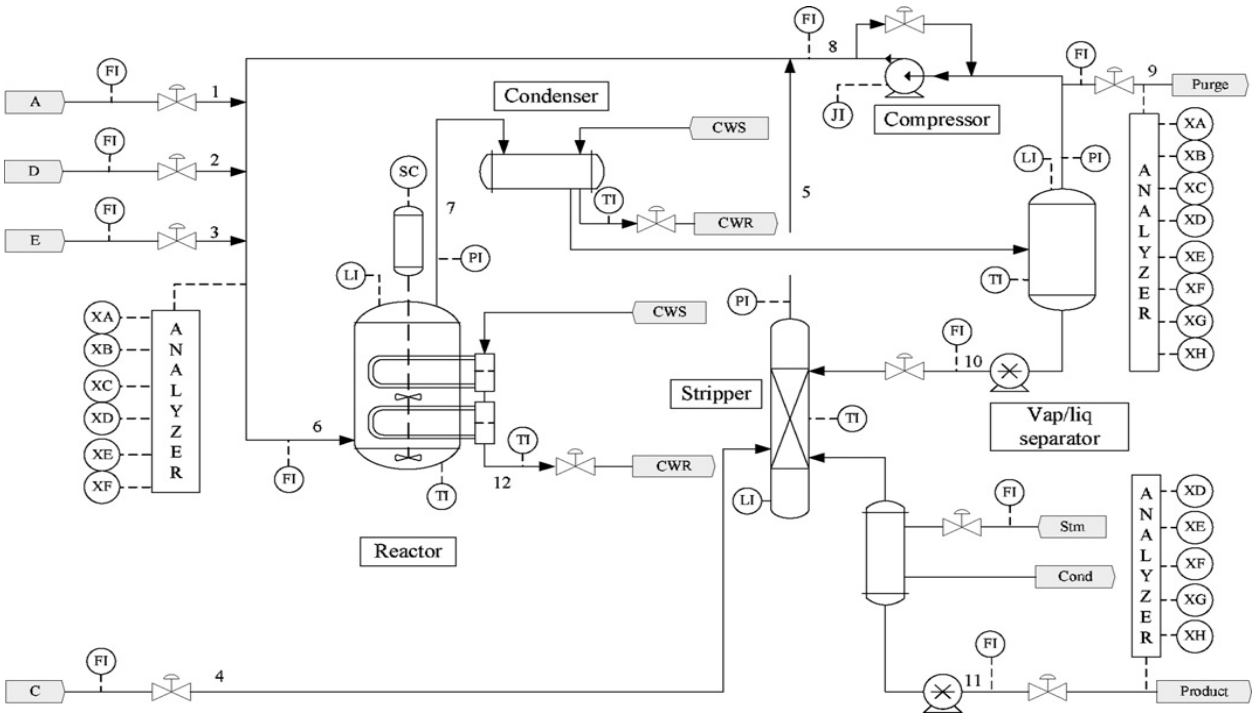

Figure 2. The Tennessee Eastman process

Table 1. Descriptions of process faults in TE process

Fault No.	Process variable	Type
IDV(1)	A/C feed ratio, B composition constant	step
IDV(2)	B composition, A/C feed ratio constant	step
IDV(3)	D feed temperature	step
IDV(4)	Reactor cooling water inlet temperature	step
IDV(5)	Condenser cooling water inlet temperature	step
IDV(6)	A feed loss	step
IDV(7)	C header pressure loss-reduced availability	step
IDV(8)	A, B, and C feed composition	Random variation
IDV(9)	D feed temperature	Random variation
IDV(10)	C feed temperature	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	Unknown	Unknown
IDV(17)	Unknown	Unknown
IDV(18)	Unknown	Unknown
IDV(19)	Unknown	Unknown
IDV(20)	Unknown	Unknown
IDV(21)	The valve fixed at steady state position	Constant position

RESULTS AND DISCUSSION

Effect of the number of trees and predictive variables (m) on the classifier's accuracy

MATLAB ® (The MathWorks, Inc., Natick, MA, USA) software was used for result generation. Initially, 1000 trees were trained with 3375 training samples for different values of m. Total 53 variables are there in the process, though, the default value of m is approximately 7, however the range of m was taken from 4-18 with the increment of 2. Fig.3 illustrates OOB error at different values of m.

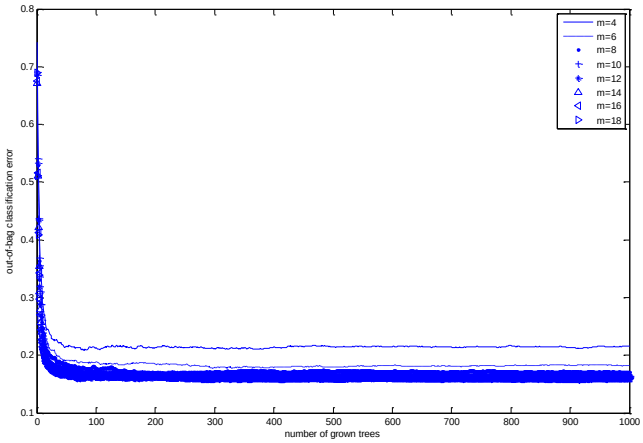

Figure 3. OOB error at different values of m

The ensemble of 1000 trees provided OOB error less than 20% at each value of m (except at $m = 4$). After 100 trees the change in OOB error was insignificant, however adding more trees in the forest, does not create a problem, because over-fitting is not a problem in RF, but training time will increase. So, it was decided to fix the number of trees at 250 (because we can take any number of trees above 100, after 100 trees there is not any significant variation in OOB error). Fig.4 illustrates the overall classification accuracy of ensemble of 250 trees on 1125 testing samples at different values of m. Maximum classification accuracy was achieved at $m = 12$.

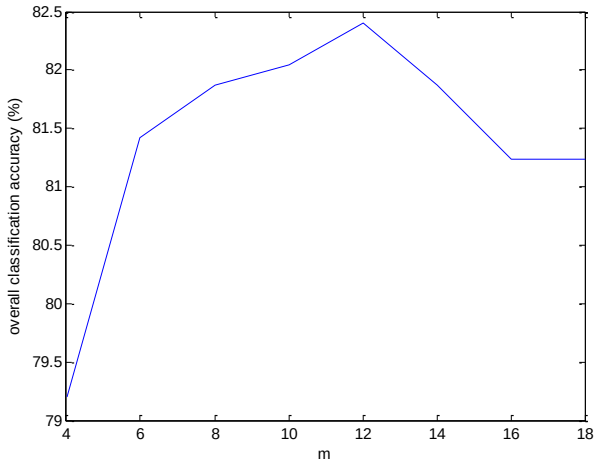

Figure 4. Overall classification accuracy with variation in m

So, the value of m was fixed at 12.

Effect of the number of classes on the classifier's accuracy

After obtaining the best values of number of trees and m, number of classes were varied and observed the effect on the classification accuracy of the ensemble. Number of samples were the same i.e., 3375 while increasing gradually the

number of considered classes. The results obtained are shown in Fig.5.

Fig.5 illustrates the variation in classification accuracy with increasing number of classes. Generally, the tendency is that with increasing number of classes the classification accuracy should continuously be decreasing, but classification accuracy reaches to a maximum with seven classes in Fig.5. This abnormal behavior can be more clearly observed taking into account Fig.6, which illustrates classification accuracy achieved for each fault/class. The ensemble classifies faults 2,4,5,6 and 7 with very good accuracy, which improves the overall classification accuracy. But, with 9 classes, the overall accuracy suddenly drops, because fault 3 & 9 were very badly identified, infect classifier has confusion between these two faults, which is very much clear from Table 2 also. Table 2 shows the classification accuracy for individual class. Each class was tested for 75 samples. Third and ninth row of the Table 2 clearly indicates that 32 samples of fault 3 were misclassified as fault 9, whereas 44 samples of fault 9 were misclassified as fault 3. If there was not an overlapping between class 3 and 9, the ensemble would provide classification accuracy of more than 82.5%.

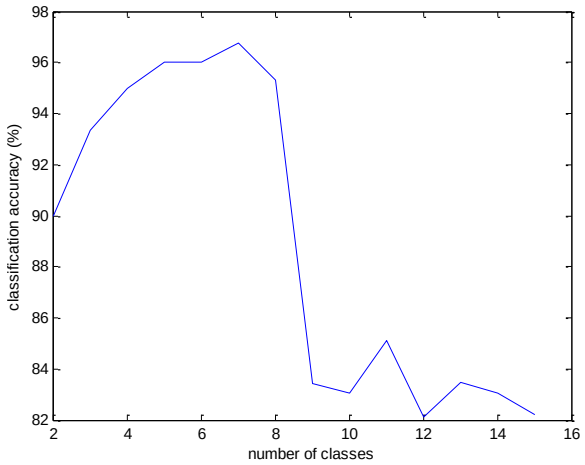

Figure 5. Classification accuracy with increase in number of classes.

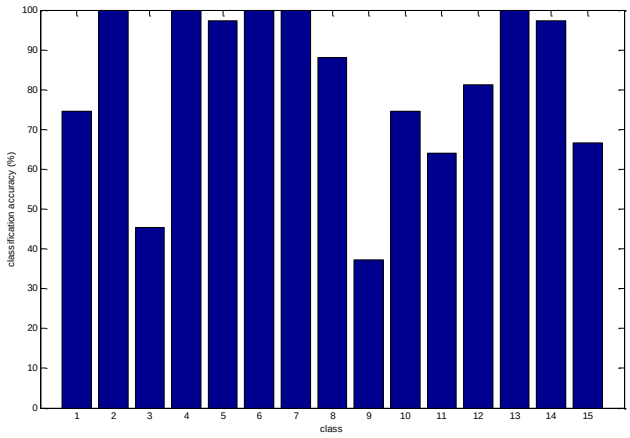

Figure 6. Classification accuracy for individual classes

Table 2. Confusion matrix

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	% Accuracy
F1	56	2	0	0	0	0	1	12	0	2	0	0	0	1	1	74.67
F2	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	100
F3	0	0	35	0		0	0	0	32	1	0	0	4	0	3	45.33
F4	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	100
F5	0	0	0	0	73	0	0	0	0	0	0	2	0	0	0	97.33
F6	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	100
F7	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	100
F8	9	0	0	0	0	0	0	66	0	0	0	0	0	0	0	88
F9	0	0	44	0	0	0	0	0	26	0	0	0	0	0	5	37.33
F10	0	0	9	0	0	0	0	0	6	56	0	2	0	0	2	74.67
F11	0	2	0	0	0	0	1	1	1	2	49	1	0	17	1	64
F12	0	0	1	0	4	0	0	0	2	0	3	60	0	0	5	81.33
F13	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	100
F14	0	0	0	0	0	0	0	0	0	0	1	1	0	73	0	97.33
F15	0	0	6	0	0	0	0	0	11	1	0	6	0	0	51	66.67

Variable importance in prediction

As a part of the algorithm building exercise, RF also gives variable importance as its natural outcome. This information can be utilized to understand the system dynamics. By using this quality of RF we identified the important variables in TE process, which are important in fault/class prediction.

Fig.7 clearly illustrates that Variables 2, 8, 17, 26, 32, 36, 37, 40, 46, 50, and 53 have no importance in prediction. For each variable, you can permute the values of this variable across all of the observations in the data set and compute how much worse the misclassification error becomes after the permutation. This can be repeated for each variable. Plot the increase in misclassification error for each input variable, the bigger this value, the more important the variable. By removing those 11 unimportant variables, once again the ensemble was trained with 250 trees and default value of m. Fig.8 shows that ensemble provided almost same OOB error and classification accuracy for the testing samples in both the cases (see Table 3). Training time also reduced substantially with same classification accuracy due to less number of variables.

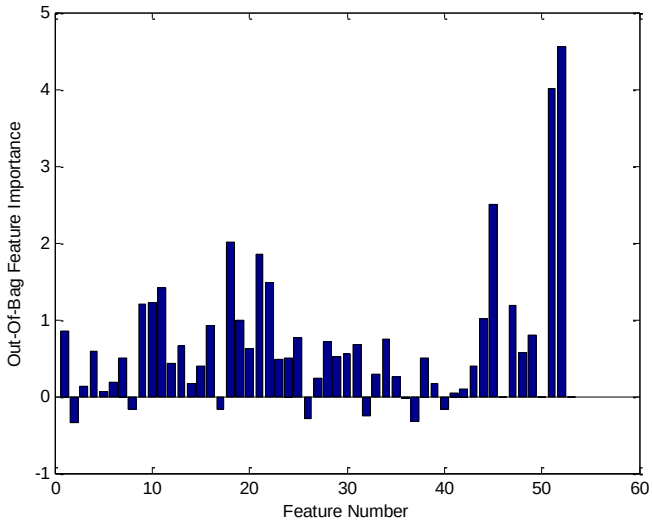

Figure 7. Variable importance in prediction

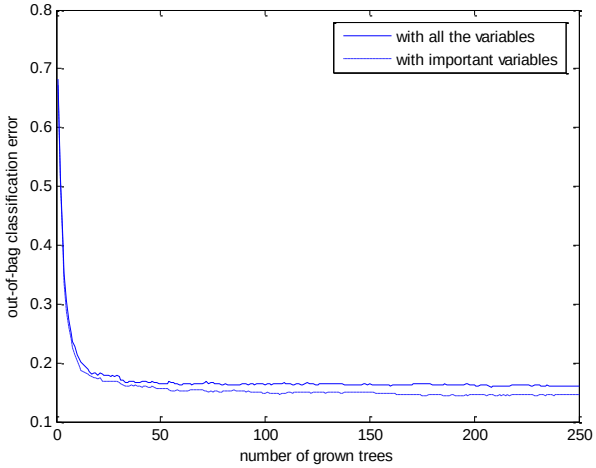

Figure 8. Comparision of OOB error with all and important variables

Table 3. Classification accuracy with all and important variables

	Classification accuracy with testing samples
With all the variables	81.78
With important variables	81.79

CONCLUSIONS

It is evident from the results that RF is easy to understand and implement. It is easy to set parameters in RF and parameter search is limited to a defined range. It does not face the problem of over-fitting. RF has the ability to handle large number of variables and classes efficiently and identify the unimportant variables not participating in prediction. As far as classification accuracy is concerned, it has provided approximately 82% with 15 fault classes, which is very much satisfactory in the condition of overlapping between class 3 and 9.

REFERENCES

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, A review of process fault detection and diagnosis Part I: Quantitative model-based methods, Comput. Chem. Eng. 27 (2003a) 293-311.

[2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, A review of process fault detection and diagnosis Part-II: Qualitative models and search strategies, Comput. Chem. Eng. 27 (2003b) 313-326.

[3] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, A review of process fault detection and diagnosis Part III: Process history based methods, Comput. Chem. Eng. 27 (2003c) 327-346.

[4] J. C. Hoskins, D. M. Himmelblau, Artificial neural network models of knowledge representation in chemical engineering, Comput. Chem. Eng. 12 (1988) 881-890.

[5] S. R Naidu, E. Zafiriou, T. J. McAvoy, Use of neural networks for sensor failure detection in control systems, IEEE Control. Syst. Mag. 10 (1990) 49-55.

[6] V. Venkatasubramanian, R. Vaidyanathan, Y. Yamamoto, Process fault detection and diagnosis

using neural networks-I: Steady-state processes, Comput. Chem. Eng. 14 (1990) 699-712.

[7] T. Marcu, L. Mirea, Robust detection and isolation of process faults using neural networks, IEEE Control. Syst. 17 (1997) 72–79.

[8] Y. Zhou, J. Hahn, M. S. Mannan, Fault detection and classification in chemical processes based on neural networks with feature extraction, ISA Trans. 42 (2003) 651–664.

[9] S. Rajakarunakaran, P. Venkumar, D. Devaraj, K. S. P. Rao, Artificial neural network approach for fault detection in rotary system, Appl. Soft Comput. 8 (2008) 740–748.

[10] W. L. Tan, N. M. Nor, M. Z. Abu Bakar, Z. Ahmad, S. A. Sata, Optimum parameters for fault detection and diagnosis system of batch reaction using multiple neural networks, J. Loss Prev. Process Ind. 25 (2012) 138-141.

[11] A. Kulkarni, V. K. Jayaraman, B. D. Kulkarni, Knowledge incorporated support vector machines to detect faults in Tennessee Eastman Process, Comput. Chem. Eng. 29 (2005) 2128-2133.

[12] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, Fault diagnosis based on support vector machines and systematic comparison to existing approaches, Comput. Chem. Eng. 21 (2006) 1209-1214.

[13] Y. Li, Z. Wang, J.Yuan, On-line fault detection using svm-based dynamic MPLS for batch processes, Chinese J. Chem. Eng. 14 (6) (2006) 754–758.

[14] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, Simultaneous fault diagnosis in chemical plants using support Vector Machines, Comput. Chem. Eng. 24 (2007) 1253-1258.

[15] Y. Mao, Z. Xia, Z. Yin, Y.Sun, Z. Wan, Fault diagnosis based on fuzzy support vector machine with parameter tuning and feature selection, Chinese J. Chem. Eng. 15 (2) (2007) 233-239.

[16] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, Performance assessment of a novel fault diagnosis system based on support vector machines, Comput. Chem. Eng. 33 (2009) 244–255.

[17] Yang, B-S., Di, X., and Han, T., (2008), "Random forests classifier for machine fault diagnosis"**,** *Journal of Mechanical Science and Technology,* vol. 22, no. 9, pp. 1716-1725.

[18] Aldrich, C., and Auret, L., (2010), "Fault detection and diagnosis with random forest feature extraction and variable importance methods", *IFAC Proceedings Volumes*, vol. 43, no. 9, pp. 79-86.

[19] Samantaray, S. R., (2012), "Ensemble decision trees for high impedance fault detection in power distribution network", *International Journal of Electrical Power & Energy Systems,* vol. 43, no. 1, pp. 1048-1055.

[20] Yao, Q., Wang, J., Yang, L., Su, H., and Zhang, G., (2016), "A fault diagnosis method of engine rotor based on Random Forests", *IEEE* ***Conference Publications,*** **pp. 1-4.**

[21] Shrivastava, R., Mahalingam, H., and Dutta, N. N., (2017), Application and Evaluation of Random Forest Classifier Technique for Fault Detection in Bioreactor Operation, Chemical Engineering Communications, 204:5, 591-598.

[22] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.

[23] L. Breiman, Some Infinity Theory for Predictor Ensembles, Technical Report 577 (2000) UC Berkeley.

[24] L. Breiman, Random forests, Mach. Learn.45 (2001) 5–32.

[25] G. James, D. Witten, T. Hestie, R. Tibshirani, An introduction to statistical learning, Springer, 2013, pp 312.

[26] J. Downs, E. Fogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (1993) 245–255.

[27] L. Chiang, E. Russell, R. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag, London, 2001.

[28] N. Ricker, Decentralized control of the Tennessee Eastman challenge process, J. Process Control 6 (4) (1996) 205–221.