2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Active learning using one class SVM for fault detection and diagnosis of process systems

Piyush Dasrao Hambarde, Ankit Ajay Mandpe, Santhosh Kumar Varanasi

Abstract— **Today's process industries are investing heavily in the field of process safety and efficiency so as to ensure that the plant is running at the optimal conditions and the overall profit is maximized. Early and fast detection and diagnosis of faults is one of the way of ensuring this objective. Owing to better and reliable data acquiring systems and/ or due to boom of Artificial Intelligence in the era of Industry 4.0, researchers are looking at various ways of incorporating different Machine Learning techniques for development of efficient fault detection and diagnosis methods. As it is not feasible to train a model to detect all possible faults that may arise in a plant, it is required to have a methodology, that is capable enough to learn about any new faults that might occur in the due course of operating a process and is the main objective of this paper. To achieve this objective, a two stage approach wherein, in the first stage, an unsupervised learning strategy using one-class SVM is considered in this paper to detect whether a fault is existing or a new one. In the second stage, a multi-class classifier using Neural networks like Extreme learning machine (ELM) has been employed to detect the nature of fault. The efficacy of the proposed method is demonstrated by considering a Tennessee Eastman process and the results are compared with existing methods.**

I. Introduction

The modern day industries desire to create maximum quality products out of the available resources while also maintaining the safety of the plant. One way to ensure this is early detection and diagnosis of possible faults in a process plant. Fault detection refers to the process of identifying the possibility of deviation from the normal behavior of the process plant and fault diagnosis refers to the process of identifying the root cause of the fault. Fault detection and diagnosis can be performed through various techniques like, model based ([1], [2]) , rule based ([3], [4]) and process history or data based ([5], [6]). In the model based approach,a mathematical model is developed to mimic the normal behavior of the plant and any deviation from this will lead to a possible fault [7]. On the other hand, rule based approaches rely on defining thresholds on certain process variables and once a variable exceeds these thresholds, a fault warning is expected to be raised.

Owing to the presence of large and reliable datasets due to better data acquisition systems resulting a significant amount of growth in data based fault detection approaches over model based approaches ([8],[9],[10],[11]). On a broad sense, the data-driven approaches for FDD can be classified into traditional statistics based approaches and novel machine learning based approaches [12]. In the traditional statistical approaches, methods like Principal Component Analysis (PCA) [13] and Dynamic Principal Component Analysis (DPCA) [14] have been used widely for fault detection and diagnosis. Apart from the traditional statistics based approaches , various machine learning approaches include use of Support Vector Machines (SVMs) ([15], [16]) and Neural networks (NNs) ([17], [18]).

Upon closer inspection it was further observed that most of the existing data driven approaches for fault detection are supervised learning based, which require a significantly amount of the possible different faults that occurred or may occur in the process plant. However, acquiring the dataset with all possible faults might be quite expensive and impractical for the process industries. Further, as the range of fault categories are wide and different, training a model apriori incorporating all the categories would be challenging [19]. Hence, it is required to have a fault detection and diagnosis methodology, that is capable enough to learn about any new faults that might occur in the due course of operating a process and is the main objective of this paper.

Motivated by the work of [19], a two stage approach wherein, in the first stage, an unsupervised learning strategy using one-class SVM is considered in this paper to detect whether a fault is existing or a new one. In the second stage, a multi-class classifier using Neural networks like Extreme learning machine (ELM) has been employed to detect the nature of fault. The main difference between the proposed work and the work of [19] lies in the way in which the classifier is defined. The authors in [19], considered a shallow neural network as a classifier. Although a shallow neural network is efficient, the computational cost required for training the model is significant and in real-time implementation, this cost has to be as minimal as possible. Hence, in this paper, owing to its simple and fast yet efficient learning capabilities, an Extreme learning machine (ELM) is used as a classifier. A comparison between the two classifiers (NN and ELM) is also made to demonstrate the efficacy of the proposed approach both in terms of accuracy and computational cost.

The rest of paper is organized as follows. Section II explains about the proposed approach starting with the problem statement in Section II-A and Section II-B briefly explains about the preliminaries required for the paper. Section II-C explains the details of the proposed methodology for Data-based adaptive fault detection and diagnosis followed by a case study on Tennessee Eastman Process in Section III and finally ends with Section IV as Conclusions.

The authors are with the Department of Chemical Engineering, Department of Chemical Engineering, Indian Institute of Technology Jodhpur, Nagaur Road, Karwar, Jodhpur, Rajasthan - 342037, India. email:hambarde.1@iitj.ac.in, mandpe.1@iitj.ac.in, skvaranasi@iitj.ac.in

II. Data-based fault detection and diagnosis

A. Problem Statement

It is essential to have a fault detection and diagnosis methodology, that is capable enough to learn about any new faults that might occur in the due course of operating a process, and is the main objective of this paper. Similar to the work of [19], a two stage approach wherein, in the first stage, an unsupervised learning strategy using one-class SVM is considered to detect whether a fault is existing or a new one. In the second stage, a multi-class classifier using Neural networks like Extreme learning machine (ELM) has been employed to detect the nature of fault.

The main difference between the proposed work and the work of [19] lies in the way in which the classifier is defined. The authors in [19], considered a shallow neural network as a classifier and in this paper, owing to its simple yet efficient learning capabilities, an Extreme learning machine (ELM) is used as a classifier. A brief introduction to One-class SVM and ELMs and details of the proposed methodology will be provided in the rest of the section.

B. Preliminaries

1) One-class SVM: Support vector machines (SVM) is a supervised machine learning algorithm widely used for the task of regression and classification. SVM work on the principle of establishing a line or a hyperplane that classifies the features of a dataset such that the features between different classes have maximum margin. The minimum distance of data points from the line or hyperplane is referred to as margin. SVMs are in general efficient methods for both linear and non-linear classification tasks [20], [21]. Researchers looked for the ways of using the standard SVM in the area of Fault detection, but faced with the challenges owing to its supervised learning nature as discussed in Section I.

One class SVM (OCSVM) is a special type of Support Vector Machine (SVM) majorly designed for the purpose of adaptive fault detection or abnormality detection. Unlike the standard SVM , OCSVM is an unsupervised learning algorithm that works best for tasks of anomaly detection as only the normal dataset is required to training. One-class SVM tries to learn a boundary such that it has a maximum margin between the non-faulty data points and origin [22]. Whenever, a new data is located outside this boundary the One-class SVM classifies it as faulty data point.

Consider a training dataset $\{x_i \in \mathbb{R}^m, i = 1,2,\cdots,N\}$ with a corresponding single class label as $y_i = 1, \forall\ i$, the optimization problem of One-class SVM can be written as follows:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + b \tag{1}$$

subject to

$$w\phi(x_i) + b \geq 0, \quad i = 1,2,\cdots,N$$

where, $\phi(.)$ denotes a mapping function through which a linear hyperplane is constructed for separation of sample points from the origin. Since it is not always feasible to perfectly separate the samples from origin [9], the optimality constraints can be relaxed with the help of slack variable ς_i as

$$\min_{w,\varsigma_i,b} \frac{1}{2}\|w\|_2^2 + \frac{1}{vN}\sum_{i=1}^{N}\varsigma_i^2 + b \tag{2}$$

subject to

$$w\phi(x_i) + b + \varsigma_i \geq 0, \quad \varsigma_i \geq 0 \quad i = 1,2,\cdots,N$$

where, $v \in (0,1)$. Similar to the case of binary SVMs [20], the optimization problem is converted into unconstrained form through the formulation of Lagrangian. Considering $\alpha_i \geq 0$ and $\eta_i \geq 0$ as Lagrange variables, the unconstrained formulation is given as

$$L(w,\varsigma,b) = \frac{1}{2}\|w\|_2^2 + \frac{1}{vN}\sum_{i=1}^{N}\varsigma_i + b - \sum_{i=1}^{N}\alpha_i\left(w\phi(x_i) + b + \varsigma_i\right) - \sum_{i=1}^{N}\eta_i\varsigma_i$$

By defining a Kernel function as $K(x_i,x_j) = \phi(x_i).\phi(x_j)$ and considering all the partial derivatives of unconstrained formulation to zero, a dual form of the unconstrained formulation be defined as

$$\min_{\alpha_1,\alpha_2,\cdots,\alpha_N} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(x_i,x_j) \tag{3}$$

subject to

$$0 \leq \alpha_i \leq 1, \quad \sum_{i=1}^{N}\alpha_i = vN$$

The formulated dual problem can be solved using standard quadratic programming techniques ([9], [23]).

Various researchers utilized the idea of One-class SVM for Fault detection. For example, the authors in [24] used the idea of one class SVM to detect anomalies and combined the results with output to calibrate the posterior probability to look for false alarms. A robust one-class SVM which is insensitive towards outliers was also proposed by the authors in [22] for fault detection.

2) Extreme learning machine: ELM is special class of single hidden layer feed-forward neural network where in, the weights and biases in the hidden layer of the network are arbitrarily generated. Due to this, unlike feed-forward neural network, ELM doesn't require iterative training and hence usually faster.

Consider a training data-set of m input variables x_i and an output variable y i.e.,.. $\{(x_i^j, y^j),\ i = 1,2,\cdots,M,\ j = 1,2,\cdots,N\}$ be the dataset, the architecture of ELM consists of an input layer, one hidden layer, and an output layer as shown in Fig. 1. Input layer receives sets of feature vectors and hidden layer contains neurons where in data transformation through a non-linear functions (such as sigmoid or relu) called activation function takes place. The number of nodes in hidden layer determines the complexity of model and precision of predictions. The output layer gives the response of the model.

For a network with q-activation functions in the hidden layer, let $\{(W_{i,j}, b_j),\ i = 1,2,\cdots,m,\ j = 1,2,\cdots,q\}$ denote

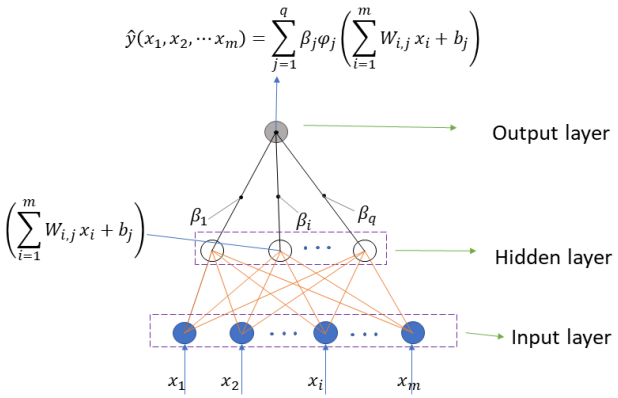


Fig. 1: Single hidden layer feed-forward neural network representation

the weights and biases of the hidden layer and let $\{(\beta_i),\ i = 1,2,\cdots,q\}$ denote the weights of the output layer. Let $\varphi(.)$ denote the activation functions in the hidden layer, the response of the output layer for l^{th} data point is defined as

$$\hat{y}(x_1^l, x_2^l, \cdots, x_m^l) = \sum_{j=1}^{q} \beta_j \varphi \left(\sum_{i=1}^{m} W_{i,j} x_i^l + b_j \right) \tag{4}$$

Unlike the traditional feed-forward neural network where in, all the weights and biases of the hidden and output layer are trained using methods like backpropogation, in ELM, the weights and biases of the hidden layer are generated randomly [25] and only the parameters at the output layer (β_i) are trained for. Since these parameters appear linearly, the procedure of training an ELM is straightforward and highly efficient.

The procedure for training an ELM is as follows: Given the number of neurons in hidden layer, the weights and biases in the hidden layer are generated randomly. The non-linear mapping is then executed using appropriate activation function. Since the problem appear linear in weights, after the execution of non-linear mapping, a linear regression method is employed for estimation of weights in the output layer. This can be performed through the employment of least squares method for non-linear activation functions in output layer or through the calculation or Moore-Penrose psuedo inverse in the case only when the activation function at output layer is linear [25], [26].

Owing to its fast training speed and efficiency in handling high dimensional data, ELMs has been applied to a wide range of machine-learning problems, including classification, regression, and clustering [25], [26]. In particular, ELMs play a major role for the task of fault detection. For example, the authors in [27], [28], [29] utilized ELM as a classifier for fault detection in various applications.

C. Proposed Methodology for FDD

In this section, inspired from the work of [19], an adaptive and active learning based fault detection and diagnosis framework is proposed. The overall idea of the proposed work is to utilize the one-class SVM in an unsupervised way to detect the presence of known/new fault and then utilize ELM as a supervised learner to classify the fault type. In the proposed methodology, it is assumed that only one fault exists in the process at any given point in time and the time horizon of the presence of fault is significant enough for training/updating the model. The overall workflow of the proposed methodology is given in Fig. 2 and the details are as follows:

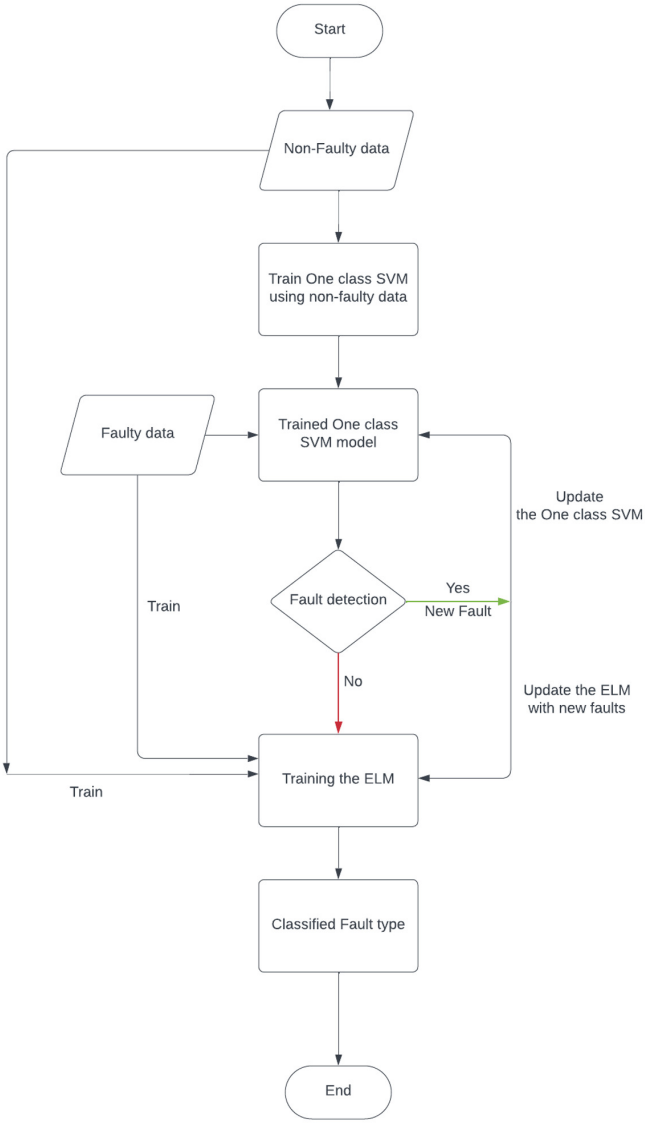


Fig. 2: Workflow of the proposed methodology

In the first step of training, a non-faulty dataset is utilized for training of a one-class SVM such that it attempts to learn a boundary that classifies faulty data from the Non-faulty dataset. After this step whenever a dataset with fault is introduced, the one-class SVM will detect the faulty data and will assigned a label (for example, as fault 1). This labelled faulty data will be passed to the ELM alongside the non-faulty data for training in a supervised manner. In the

second step of training, the non-faulty dataset is combined with the fault 1 and the one-class SVM is re-trained using this combined dataset. In doing so, we have successfully trained a one-class SVM which can pass the data which is either non-faulty or fault 1 and raises a flag whenever it detects any other fault and label as fault 2. ELM will then detect whether the data point is a non-faulty or fault 1. Once after obtaining sufficient number of fault 2 dataset from the one-class SVM, one-class SVM and ELM will be re-trained for, to detect the presence of non-faulty data or fault 1 or fault 2. Since it is assumed that the time horizon of the fault is significant, it can be noted that whenever a new fault is detected by the one-class SVM, is labelled as fault i and the re-training of one-class SVM and ELM will take place. Due to this active re-training of the models, the approach is capable of detecting and diagnosing any new faults that might arise in the due course of operation of a process.

III. Tennessee Eastman process - Fault detection and diagnosis

The efficacy of the proposed method for adaptive fault detection is demonstrated by considering a Tennessee Eastman Process (TEP), which is a benchmark case study in the field of FDD. This section explains a brief description of the process followed by the results of fault detection on the TEP process.

A. Process Description

Tennessee Eastman Process (TEP) is a well-known benchmark problem in chemical engineering, for the development of plant-wide control and monitoring strategies [30]. As shown in Fig. 2, TEP process consists of five primary units which are reactor, condenser, compressor, separator and stripper.

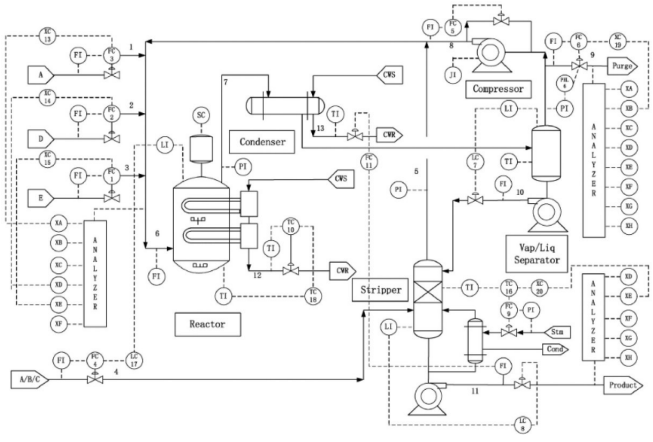


Fig. 3: TEP flow diagram ([30])

The process consists of 51 variables in total of which 41 are measured variables (of which 22 input variables are continuous process measurements and the rest 19 are the composition measurements) and the remaining 12 are manipulated variables (11-valves and 1 speed measurement). The process contains 8 various feed components labelled as A-H to produce two products from four reactants. The gaseous reactants A,C,D and E along with the inert component B are introduced in the reactor to form liquid products G and H, apart from this F is obtained as a by-product of the reaction.

In operation of such a process there are possibilities of more than 21 different process faults to occur [30], [31]. Among the possible faults in the process, *IDV* 1, *IDV* 4, *IDV* 9 and *IDV* 11 as detailed in Table I are an accurate representations of overlapping information and are considered to be challenging to detect in general [31]. Due to these reasons, the faults of *IDV*1 and *IDV*4 were considered in this paper to test the efficacy of the proposed methodology. Extension of the work for the remaining faults will be pursued in future.

TABLE I: Tennessee Eastman process faults ([30])

Variable	Description	Type	Fault ID
IDV 0	Non-faulty data	-	0
IDV 1	A/C feed ratio, B composition constant (Stream 4)	Step	1
IDV 4	Reactor cooling water inlet temperature	Step	2
IDV 9	D feed temperature (Stream 2)	Random Variable	3
IDV 11	Reactor cooling water inlet temperature	Random Variable	4

B. FDD results

In this section, the results acquired after applying the proposed methodology on the process will be presented. For this study, the datasets of Non-faulty data (Fault ID: 0), IDV1 (Fault ID: 1) and IDV4 (Fault ID: 2) with each of size 7201×53 were considered. Of this 7201 samples, 60% is used for training, and the remaining 40% is used for validation purposes. The dataset used in this paper is taken from [32]. The training is performed in python using scikit-learn package on a laptop with Intel i5 11th Gen processor running at 3.1GHz using 8GB of RAM. The kernel used is **RBF** (Radial Basis Function) and the choice of hyper-parameters for training are: gamma as **Auto** (This is a parameter for non-linear hyperplanes. It highlights the extent of influence exerted by an individual training example. gamma parameter is specific to the RBF kernel and is typically set to 'auto,' which defaults to 1 / no. of features). and nu to be $1e-6$ (the nu value is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors).

To train the first step of the one-class SVM, 60% of the Non-faulty data (Samples from 1 to 4321) was considered. Once when the training of one-class SVM is completed, 60% *IDV* 1 faulty data (4321 samples) were introduced to observe the accuracy of Fault detection by the One-class SVM, the accuracy of fault detection turned out to be 100%. Now the fault data detected by One-class SVM is labelled as fault 1 and One-class SVM is retrained using the combined data of non-faulty and fault 1. Alongside this, an ELM with corresponding hyper-parameters as the number of neurons in hidden layer as **32**, activation function as **ReLU** was

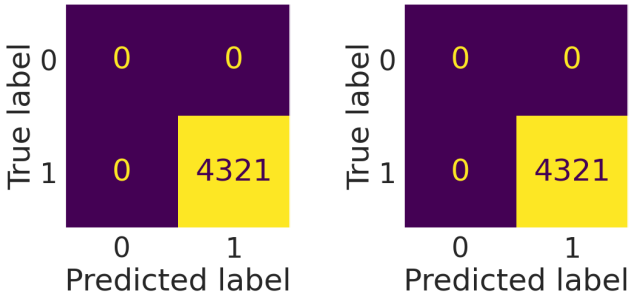
trained for detection of non-faulty and fault 1 datasets. The training accuracy of ELM is 99.27%. Now a dataset of 2878 samples (20% of validation data from non faulty and fault 1) is considered for testing purpose and the accuracy of the proposed method with test data set is 100%. Now 60% of *IDV* 4 dataset is introduced and the fault detection accuracy of one-class SVM turned out to be 100% . The fault dataset detected by one-class SVM is now labelled as fault 2 and the one-class SVM and ELM are re-trained using the 60% datasets of non-faulty, fault 1 and fault 2. A dataset of 4320 samples (20% of validation data from non faulty, fault 1 and fault 2) is considered for testing purpose and the accuracy of the proposed method with test data set is 100 %. Although a sequential fault dataset is considered for demonstration, the same approach will hold true for non-sequential faulty dataset. The only requirement is that only one-fault occurs at a time. To further demonstrate the significance of the proposed method, the ELM part for classification is now replaced with a Neural network and the same steps above are repeated this time by considering a SLFN with the number of neurons in hidden layer as **32** and the activation function as **ReLU**. The overall results with the proposed method and the method where in SLFN is considered are reported in Table II and the corresponding confusion matrices are shown in Figs. 4, 5 and 6.

TABLE II: Framework and Results

Step	# of Samples	Fault ID	Model update and Results
1	4321	0	Trained One-class SVM
2	4321	1	Unknown Fault detected with 100% accuracy 4(a)
3	4321	0 & 1	Updated One-class SVM with new Fault
4	4321	0 & 1	Trained both ELM and NN. Training accuracies are 99.24% and 97.20% respectively.
5	2878	0 & 1	Testing accuracy of fault 1 classification with ELM and NN are 100% 5(a) & 98.81% respectively 6(a) Fault 1 classified
6	4321	2	Unknown Fault detected with 100% accuracy 4(b)
7	4321	0 & 1 & 2	Updated One-class SVM & ELM
8	4320	0 & 1 & 2	Testing accuracy of fault 4 classification with ELM and NN are 100% 5(b) & 73.05% respectively 6(b) Fault 4 classified

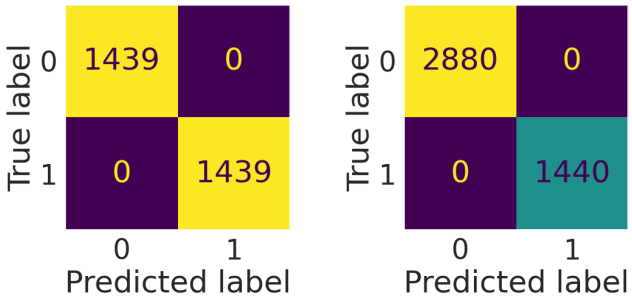
From the results mentioned in the Table II and the corresponding confusion matrices, it can be observed that One-class SVM was highly accurate with an accuracy of 100% in both the cases when Fault-1 and Fault-4 were introduced. ELM proved to be more efficient than NN with the same architecture with an accuracy of 100% in both cases (Steps 5 and 8) over NN which has a test accuracy of 98.81 % and 73.05% for detection of faults 1 and 2 respectively.

It was further observed that when the # of neurons in the neural network were increased to 150 with the choice



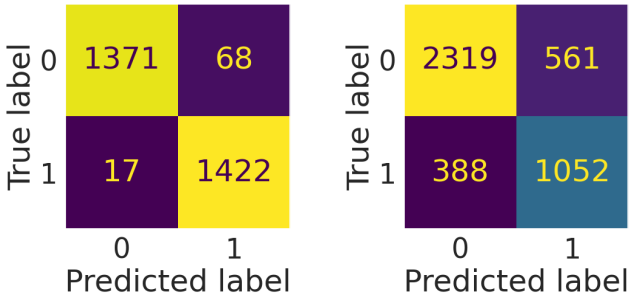
(a) Step 2 (b) Step 6

Fig. 4: Confusion matrix of classification with One-class SVM i.e., Steps 2 and 6 of Table II.



(a) Step 5 - ELM (b) Step 8 - ELM

Fig. 5: Confusion matrix of classification with ELM i.e., Steps 5 and 8 of Table II



(a) Step 5 - NN (b) Step 8 - NN

Fig. 6: Confusion matrix of classification with NN i.e., Steps 5 and 8 of Table II

of activation function being tanh, an improvement in the accuracy (99.79%) was observed. Apart from the accuracy, ELM surpassed NN with a very low computation time as ELM took only **0.0253** seconds while the NN took **3.234** seconds for the first training (Step 4). For the second training (Step 7) ELM took approximately **0.0214** seconds while NN required **4.501** seconds. Thus ELM is proved to be efficient both in terms of accuracy and computational cost over a neural network which utilizes backpropogation for training.

IV. Conclusions

The main objective of this paper was to provide an active learning based FDD model which is capable enough to detect

known/unknown faults and also update itself to classify the faults. The proposed framework uses a two stage strategy where in the first stage, a One-class SVM is utilized for detecting the presence of fault in an unsupervised learning manner and in the second stage, ELM is employed for classification of the detected fault. The proposed framework was tested on the benchmark case study of Tennessee Eastman Process and the results authenticate the benefit of the proposed methodology. The proposed framework is best suited for datasets with limited to no knowledge of future faults. Further, the study also proved ELM to be more efficient and computationally fast over NN of the same architecture. The possibility of relaxing the assumption of having only one fault in the process at any given point in time will be pursued in future.

REFERENCES

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[2] N. Gupta, R. De, H. Kodamana, and S. Bhartiya, "Batch-to-batch adaptive iterative learning control - explicit model predictive control two-tier framework for the control of batch transesterification process," *ACS omega*, vol. 7, no. 45, pp. 41 001–41 012, 2022.

[3] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies," *Computers & chemical engineering*, vol. 27, no. 3, pp. 313–326, 2003.

[4] Z. Xing and Y. He, "Multi-modal information analysis for fault diagnosis with time-series data from power transformer," *International Journal of Electrical Power & Energy Systems*, vol. 144, p. 108567, 2023.

[5] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[6] A. Yadav, T. K. Gaurav, H. J. Pant, and S. Roy, "Machine learning based position-rendering algorithms for radioactive particle tracking experimentation," *AIChE Journal*, vol. 66, no. 6, p. e16954, 2020.

[7] R. Raveendran, H. Kodamana, and B. Huang, "Process monitoring using a generalized probabilistic linear latent variable model," *Automatica*, vol. 96, pp. 73–83, 2018.

[8] L. Jack and A. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," *Mechanical systems and signal processing*, vol. 16, no. 2-3, pp. 373–390, 2002.

[9] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *Journal of process control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[10] M. T. Amin, S. Imtiaz, and F. Khan, "Process system fault detection and diagnosis using a hybrid technique," *Chemical Engineering Science*, vol. 189, pp. 191–211, 2018.

[11] M. G. Don and F. Khan, "Dynamic process fault detection and diagnosis based on a combined approach of hidden markov and bayesian network model," *Chemical Engineering Science*, vol. 201, pp. 82–96, 2019.

[12] J. Goetz, A. Brenning, H. Petschko, and P. Leopold, "Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling," *Computers & geosciences*, vol. 81, pp. 1–11, 2015.

[13] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual reviews in control*, vol. 36, no. 2, pp. 220–234, 2012.

[14] H. Kodamana, R. Raveendran, and B. Huang, "Mixtures of probabilistic pca with common structure latent bases for process monitoring," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 838–846, 2017.

[15] M. Ge, R. Du, G. Zhang, and Y. Xu, "Fault diagnosis using support vector machine with an application in sheet metal stamping operations," *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 143–159, 2004.

[16] Z. Yin and J. Hou, "Recent advances on svm based fault diagnosis and process monitoring in complicated industrial processes," *Neurocomputing*, vol. 174, pp. 643–650, 2016.

[17] T. Sorsa and H. N. Koivo, "Application of artificial neural networks in process fault diagnosis," *Automatica*, vol. 29, no. 4, pp. 843–849, 1993.

[18] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.

[19] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique," *Computers & Chemical Engineering*, vol. 134, p. 106697, 2020.

[20] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[21] R. Gholami and N. Fakhari, "Support vector machine: principles, parameters, and applications," in *Handbook of neural computation*. Elsevier, 2017, pp. 515–535.

[22] S. Yin, X. Zhu, and C. Jing, "Fault detection based on a robust one class support vector machine," *Neurocomputing*, vol. 145, pp. 263–268, 2014.

[23] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[24] V. A. Sotiris, W. T. Peter, and M. G. Pecht, "Anomaly detection through a bayesian support vector machine," *IEEE Transactions on Reliability*, vol. 59, no. 2, pp. 277–286, 2010.

[25] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[26] Z. Zhao, J. Zhang, Y. Sun, and H. Tian, "Fault detection and diagnosis method for batch process based on elm-based fault feature phase identification," *Neural Computing and Applications*, vol. 27, pp. 167–173, 2016.

[27] F. de Assis Boldt, T. W. Rauber, and F. M. Varejão, "Cascade feature selection and elm for automatic fault diagnosis of the tennessee eastman process," *Neurocomputing*, vol. 239, pp. 238–248, 2017.

[28] V. M. Janakiraman and D. Nielsen, "Anomaly detection in aviation data using extreme learning machines," in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 1993–2000.

[29] X. Peng, H. Li, F. Yuan, S. G. Razul, Z. Chen, and Z. Lin, "An extreme learning machine for unsupervised online anomaly detection in multivariate time series," *Neurocomputing*, vol. 501, pp. 596–608, 2022.

[30] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & chemical engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[31] K. Zhang, K. Qian, Y. Chai, Y. Li, and J. Liu, "Research on fault diagnosis of tennessee eastman process based on kpca and svm," in *2014 Seventh international symposium on computational intelligence and design*, vol. 1. IEEE, 2014, pp. 490–495.

[32] Z. Liu, C. Li, and X. He, "Evidential ensemble preference-guided learning approach for real-time multimode fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 5495–5504, 2024.