A Comparative Study of Autonomous Learning Outlier Detection Methods Applied to Fault Detection

Clauber Gomes Bezerra*, Bruno Sielly Jales Costa†, Luiz Affonso Guedes‡ and Plamen Parvanov Angelov§

*Campus EaD
Federal Institute of Rio Grande do Norte - IFRN
Natal, Brazil
Email: clauber.bezerra@ifrn.edu.br
†Campus Natal - Zona Norte
Federal Institute of Rio Grande do Norte - IFRN
Natal, Brazil
Email: bruno.costa@ifrn.edu.br
‡Department of Computer Engineering and Automation
Federal University of Rio Grande do Norte - UFRN
Email: affonso@dca.ufrn.br
§Lancaster University, Data Science Group, School of Computing and Communications,
Lancaster LA1 4WA, United Kingdom
Chair of Excellence, Carlos III University, Madrid, Spain
Email: p.angelov@lancaster.ac.uk

Abstract—Outlier detection is a problem that has been largely studied in the past few years due to its great applicability in real world problems (e.g. financial, social, climate, security). Fault detection in industrial processes is one of these problems. In that context, several methods have been proposed in literature to address fault detection. In this paper we propose a comparative analysis of three recently introduced outlier detection methods: RDE, RDE with Forgetting and TEDA. Such methods were applied to the data set provided in DAMADICS benchmark, a very well-known real data tool for fault detection applications. The results, however, can be extended to similar problems of the area. Therewith, in this work we compare the main features of each method as well as the results obtained with them.

Fig. 1. Outlier in a dataset.

I. INTRODUCTION

In a data set, there may exist elements which significantly differ from the others in respect to one of more of its features. Such element is called outlier. Hawkins [1] defines outlier as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". To exemplify, consider the chart shown in Fig. 1, where a signal is sampled over time. Analyzing this set of data samples, one can observe the existence of a sample which possesses an amplitude value that is considerably larger than the others. This sample, thus, may be classified as an outlier.

Outlier detection is an area of research that has gained considerable attention in the past few years due to its large applicability. Among them, one can mention military surveillance for enemy activities, intrusion detection in cyber security, fraud detection for credit cards, insurance or health care and fault detection in safety critical systems [2], [3], [4].

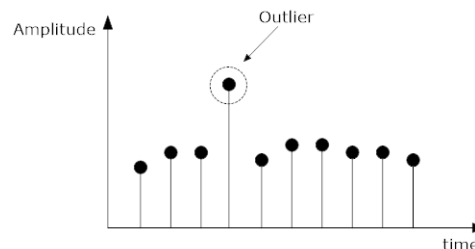In the specific case of fault detection, an outlier may indicate the start of a fault in one or more equipments of an industrial process. A fault, thus, may cause a series of problems in the process, such as unscheduled stoppages, production losses, decrease of equipment lifespan and accidents involving human life and the environment [5].

For that and other reasons, applications in the area of outlier detection have gained extreme importance. Therefore, several researches have been developed aiming to detect, as soon as possible, faults in industrial equipment and, thereby, help to prevent problems in the future. Among the main strategies one can mention monitoring, using statistically defined thresholds, of a set of calculated or measured process variables. An outlier, then, is detected if the value of a data sample exceeds a threshold. Very often, that type of strategy is used jointly with a signal processing method to determine the beginning and end of a fault.

For such applications, large amounts of data need to be processed in real-time, online, as soon as the data samples are available from the sensors. Therefore, fault detection

techniques in industrial processes need to be computationally efficient in order to handle large amounts of data and obtain real-time responses. Moreover, such techniques needs to be reliable and robust, providing accurate outputs even in the presence of faults or uncertainties [6].

In this work we propose a study and application of a set of outlier detection methods to fault detection problems. Therewith, we present a comparative analyses of the results obtained with RDE, RDE with forgetting and TEDA outlier detection techniques. For comparison purposes, we use a set of multivariate data provided by the very well-known fault detection benchmark DAMADICS.

The remainder of this paper is organized as follows: In Section II, we present a succint definition of the chosen outlier detection methods. In Section III, we describe DAMADICS benchmark, which will be used as basis of comparison for the above mentioned methods. The results obtained from the experimental study are detailed in Section IV. Finally, the Section V presents the conclusions of this work.

II. OUTLIER DETECTION METHODS

Several approaches to outlier detection in industrial processes have been recently introduced. In this section we describe four of these methods, which will be later used in a real industrial fault detection application.

A. RDE

Recursive Density Estimation (RDE) was first introduced by Angelov [7], is part of a patent [8] and can be used, among other applications, to outlier detection. It was recently applied with success to fault detection problems [9], [10], [11]. RDE is based on the calculation of the density of the analyzed data set and this density determines how close on the n-dimensional feature space a data sample is from the other at the time instant k. Through the analysis of the density one can identify if a particular data sample differs from the concept of 'normality' within the data read so far.

To exemplify that idea, consider a set of 2-dimensional data. At a given time instant k, the data are distributed according to Fig. 2(a) and its density is D_k. Let us have a new point P at the time instant $k + 1$. Supposing that P is close to the points read so far, as in Fig. 2(b), the value of density at the time instant $k + 1$, D_{k+1}, will be close to the D_k. On the other hand, if P is far from the previously read data samples, as in Fig. 2(c), the density will drop at D_{k+1}. Therefore, one can observe that low values of density will often indicate outliers in the data stream.
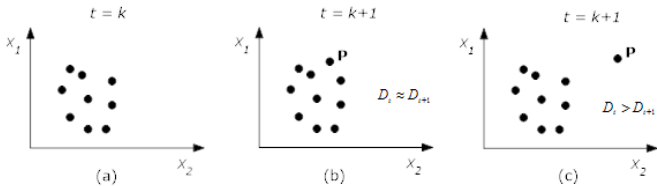

Fig. 2. Behavior of density in a example data set.

Considering a cluster Λ consisting of a set of data samples, where each sample is a vector x, the density function d, calculated from x to cluster Λ, is based on a Cauchy kernel and defined by the equation [7]:

$$d_\Lambda(x_k) = \frac{1}{1 + \frac{1}{N_\Lambda} \sum_{i=1}^{N_\Lambda} \|x_k - x_i\|^2} \tag{1}$$

where d_Λ, denotes the local density of cluster Λ; N_Λ denotes the number of data samples associated with cluster Λ. In the case of FD applications, x_k represents the feature vector with values for the instant k.

The distance is calculated between the given vector (measured at the time instant k) and other vectors that belong to the same cluster as x (measured at previous time instances). It can be shown, that this formula can be derived as an exact (not approximated or learned) quantity as [7]:

$$D(x_k) = \frac{1}{1 + \|x_k - \mu_k\|^2 + X_k - \|\mu^2\|} \tag{2}$$

where μ_k represents the mean and X_k is the scalar product of all data samples. They can be recursively updated by:

$$\mu_k = \frac{k-1}{k}\mu_{k-1} + \frac{1}{k}x_k, \quad \mu_1 = x_1 \tag{3}$$

$$X_k = \frac{k-1}{k}X_{k-1} + \frac{1}{k}\|x_k\|^2, \quad X_1 = \|x_1\|^2 \tag{4}$$

Thereby, using equations (2), (3) and (4) it is possible to calculate the density for a given data sample at the instant k with respect to all data previously read by storing and updating only the values of μ and X in memory, resulting in a fast calculation algorithm with very low computational effort and memory use. Moreover, RDE does not need any user-defined parameters or thresholds nor offline training.

B. RDE with Forgetting

A new method for outlier detection based on traditional RDE was recently introduced in [12]. It is called RDE with forgetting and was proposed as an alternative to RDE, presenting equivalent results for most cases of fault detection and improved results for specific data sets.

The formulation of RDE with forgetting replaces the terms $(k-1)/k$ and $1/k$ from equations (3) and (4) of traditional RDE by user-defined learning parameter or forgetting factor. By analyzing these equations, one can observe that the mentioned terms are used as weights of a weighted mean between the mean accumulated value of the input and its current value. For online applications, the weights are updated at each iteration of the algorithm, resulting in two problems that, often, need to be considered: 1) the finite numerical representation in computers limit the number of samples, forcing periodic flushings of data, and 2) the weights $(k-1)/k$ and $1/k$ tend to be considerably discrepant over time.

Table I shows the evolution of weights on traditional RDE for a generic application with 1 data point/second sampling rate. It is easy to observe that after 100 data samples the weight $1/k$ reaches 1%, making that the new input values have little

k	Time	RDE		RDE with Forgetting	
		$(k-1)/k$	$1/k$	α	$1-\alpha$
1	00h00m01s	0.00%	100%	95%	5%
100	00h01m40s	99.00%	1.00%	95%	5%
1000	00h16m40s	99.90%	0.10%	95%	5%
10000	02h46m40s	99.99%	0.01%	95%	5%
86400	24h00m00s	99.99%	0.01%	95%	5%

influence on the current mean. In practice, RDE becomes less susceptible to changes in its calculated mean μ.

RDE with forgetting, on the other hand, works with the complementary concepts of forgetting factor and learning parameter $1-\alpha$. Therefore, the following equations are used on density calculation:

$$D(x_k) = \frac{1}{1 + \|x_k - \mu_k\|^2 + X_k - \|\mu^2\|} \quad (5)$$

$$\mu_k = \alpha\mu_{k-1} + (1-\alpha)x_k, \quad \mu_1 = x_1 \quad (6)$$

$$X_k = \alpha X_{k-1} + (1-\alpha)\|x_k\|^2, \quad X_1 = \|x_1\|^2 \quad (7)$$

In this approach, the weights are, *a priori* user-defined and static over time. The value of α can be adequately adjusted in order to be more or less sensitive to the behavior of the system.

As an example, consider the input signal presented in Fig. 3. For $t < t_1$, the signal is stable in a normal state of operation. Then, a faulty state is triggered from instants t_1 to t_2. From t_2 and onwards, the signal is stable again in a normal state of operation, however, as can be easily observed, this new concept of 'normality' is different from the previously reached. For this specific behavior, which is often reproduced in fault detection problems, RDE would take a considerably longer period to detect the end of the fault than RDE with forgetting, since the latter is able to quickly follow the new operation scenario.
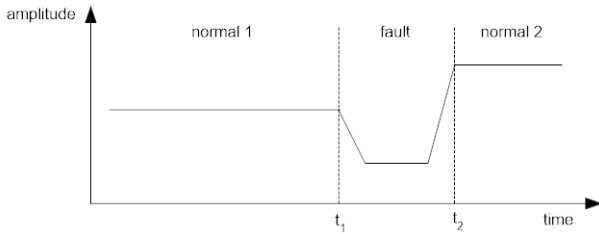

Fig. 3.   Signal with 3 states.

C. TEDA

Typicality and Eccentricity Data Analitics (TEDA) approach builds upon the well-known Recursive Density Estimation (RDE) algorithm framework, however, with major differences in its formulation. TEDA was first introduced in [13] and, since then, has been applied to different classification problems [14].

One of the main advantages of TEDA is that its formulation aims to generalize and avoid restrictive assumptions often mandatory in traditional statistical approaches and probability theory, e.g. independence of individual data samples from each other, large amounts of data and *a priori* assumptions of data distribution (e.g. normal/Gaussian) [13]. Traditional statistical methods are often very suitable for pure random processes, however might ignore the dependence of data in real processes, such as industrial, social, economic, climate, physical, and so on.

Consider a data space $X \in \mathbb{R}$, consisting of n-dimensional data samples. For this space, one can define a distance $d(x, y)$ (e.g. Euclidean, Mahalanobis, cosine or any other). Then, let us consider the data samples as an ordered sequence $\{x_1, x_2, \ldots, x_k, \ldots\}$ where the index k represents the time instant of the observation.

For the whole data set/stream, we can define sum distance to some particular point $x \in X$, for each element up to the kth one:

$$\pi^k(x) = \sum_{i=1}^{k} d(x, x_i) \quad (8)$$

The eccentricity at the time instant k, when using the Euclidean distance measure, can be defined as [14]

$$\xi^k(x) = \frac{2\pi^k(x)}{\sum_{i=1}^{k} \pi^k(x_i)} = 2\frac{\sum_{i=1}^{k} d(x, x_i)}{\sum_{i=1}^{k} \sum_{j=1}^{k} d(x_i, x_j)}, \quad (9)$$

$$k \geq 2, \qquad \sum_{i=1}^{k} \pi^k(x) > 0$$

As a complement to the eccentricity, the typicality is also defined as [15]:

$$\tau(x_k) = 1 - \xi^k(x) \quad (10)$$

Typicality and eccentricity can be, similarly to RDE, calculated recursively. It can be shown, that equation (9) can be derived as an exact quantity as [13]

$$\xi^k(x) = \frac{1}{k} + \frac{(\mu_x^k - x)^T(\mu_x^k - x)}{k[\sigma_x^k]^2} \quad (11)$$

The values of mean μ_x^k and variance σ_x^k can be updated recursively by

$$\mu_x^k = \frac{(k-1)\mu_x^{k-1}}{k} + \frac{x_k}{k}, \quad k \geq 1, \quad \mu_x^0 = 0 \quad (12)$$

$$\mu_{x^T x}^k = \frac{(k-1)\mu_{x^T x}^{k-1}}{k} + \frac{x_k^T x_k}{k}, \quad k \geq 1, \quad \mu_{x^T x}^0 = 0 \quad (13)$$

$$[\sigma_x^k]^2 = \mu_{x^T x}^k - [\mu_x^k]^T \mu_x \quad (14)$$

III.   CASE OF STUDY

As basis of comparison for the methods described in Section II, they are here applied to a fault detection problem using real data from an industrial actuator. Data from a well-known fault detection and diagnosis benchmark were used. The benchmark is called DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) and it was first introduced in [16].

DAMADICS has been largely used in literature for comparison of many different fault detection and diagnosis approaches [17], [18], [19]. The benchmark provides data from a real water evaporation process in a boiler of a sugar factory in Poland. The plant consists of three actuators, where each one of them is used for flow control of a part of the process. The actuator used in DAMADICS is illustrated in Fig. 4 and is composed by the following parts:

- Control Valve: controls the water flow on the pipe.

- Pneumatic Servo System: controls a rod connected to the control valve, allowing opening variations.

- Positioner: used to internally correct rod incorrect positioning, caused by friction, pressure variations and so on.
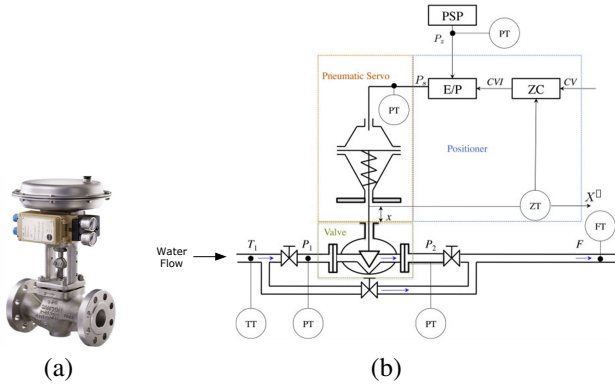

Fig. 4. Actuator model of DAMADICS benchmark [20]: (a) external view e (b) internal schema.

DAMADICS provides a library for MATLAB/SIMULINK, allowing 19 different abrupt and incipient fault simulations, but also provides the data files from real plant acquisitions on different days. The faults may occur in each of the parts of the actuator and also externally to it.

IV. EXPERIMENTS AND RESULTS

In this section we present a series of experiments using the three methods described in Section II and the data set provided by DAMADICS benchmark. The data are organized in several files, each one corresponding to the plant operation of a full working day and providing data of 32 different process variables. The sampling rate is 1 data point/second, therefore, each file contains 86400 samples of each one of the 32 variables per day.

DAMADICS provides data corresponding to 25 full working days, however, only 4 of them present faulty behaviors. A total of 19 artificial faults were manually added to the process in all of its three different actuators. The faults can be classified in 4 different groups, named f16, f17, f18 and f19, and described in Table II.

TABLE II. DESCRIPTION OF THE ARTIFICIALLY GENERATED FAULTS.

Fault code	Description
f16	Positioner supply pressure drop
f17	Unexpected pressure drop across the valve
f18	Partly opened bypass valve
f19	Flow rate sensor fault

Since the faults are located in different parts of the plant, different variables need to be selected as input to the algorithm. The feature selection is an important stage of the task. The feature selection methods can be classified as either quantitative, such as principal component analysis (PCA) [21], partial least squares (PLS) [22], linear discriminant analysis (LDA) [23], or qualitative, for example expert systems [24] or trend modeling methods [25]. For more information on the use of feature extraction methods, the reader is referred to the work of [26], [27] and [28].

However, in this work, we chose to use the same variables as defined in DAMADICS descriptive manual [20]. For each fault, the behavior of 2 input variables is analyzed. Likewise, the interval to be analyzed in each fault is also determined based on DAMADICS manual. In this paper, this interval is called fault stream. Each fault stream consists of a period of time where the signals start in a normal state of operation, followed by a faulty state and, again, normal operation. The only exception is the fault stream #13, which never leaves the faulty state once it begins.

For each of the analyzed methods, we chose to use the following criteria for outlier detection:

- **RDE:** we consider an outlier the data sample at the time instant k where its density value D_k is less than the difference between the mean density of the data set and its standard deviation σ_k^D, i.e.

$$D_k < \overline{D}_k - \sigma_k^D \qquad (15)$$

where, $\overline{D}_k$ and σ_k^D are updated by the equations

$$\overline{D}_k = \frac{k-1}{k}\overline{D}_{k-1} + \frac{1}{k}D_k, \quad \overline{D}_1 = D_1 \qquad (16)$$

$$\left(\sigma_k^D\right)^2 = \frac{k-1}{k}\left(\sigma_{k-1}^D\right)^2 + \frac{1}{k}(D_k - \overline{D}_k)^2,$$
$$\left(\sigma_1^D\right)^2 = 0 \qquad (17)$$

- **RDE with forgetting:** we use the same criteria as in RDE (equation (15)). However, D_k and σ_k^D are calculated by equations

$$\overline{D}_k = \alpha\overline{D}_{k-1} + (1-\alpha)D_k, \quad \overline{D}_1 = D_1 \qquad (18)$$

$$\left(\sigma_k^D\right)^2 = \alpha\left(\sigma_{k-1}^D\right)^2 + (1-\alpha)(D_k - \overline{D}_k)^2,$$
$$\left(\sigma_1^D\right)^2 = 0 \qquad (19)$$

where α is the user-defined forgetting factor.

- **TEDA:** we consider an outlier the data sample at the time instant k where its normalized eccentricity value ζ is higher than the threshold $5/k$, i.e.

$$\zeta(k) > \frac{5}{k} \qquad (20)$$

The analysis of the obtained results considers three different measures: true positive rate (TPR), false positive rate (FPR) and total hit rate (THR) [29]. TPR is defined by

$$TPR = \frac{n_f}{N_f}100 \qquad (21)$$

where n_f is the number of *correctly* detected faulty samples and N_f is the total of faulty samples.

FPR, on the other hand, is defined as

$$FPR = \frac{n_n}{N_n}100 \qquad (22)$$

where n_n is the number of normal samples *incorrectly* classified as faulty samples and N_n is the total of normal samples within the designated interval.

Finally, THR is defined as

$$THR = \frac{n_t}{N_t}100 \qquad (23)$$

where n_t is the number of correctly classified samples, both normal and faulty, and N_t is the total of samples of the data stream.

Therefore, each of the three methods was applied to all 19 fault streams provided by DAMADICS. The obtained values of TPR, FPR and THR are shown in Tables III, IV and V. For the specific case of RDE with Forgetting, an $\alpha = 0.99$ was used.

TABLE III.   RESULTS OBTAINED WITH RDE.

| Item | Actuator | RDE | | |
		TPR	FPR	THR
1		94.91%	71.80%	61.57%
2		78.99%	1.52%	93.11%
3		0.00%	0.00%	66.45%
4	Actuator 1	0.00%	0.00%	47.26%
5		84.16%	15.50%	84.48%
6		86.14%	22.50%	79.24%
7		100.00%	61.58%	76.62%
8		0.00%	0.00%	85.15%
9		0.00%	0.00%	88.56%
10	Actuator 2	94.44%	0.00%	98.02%
11		92.31%	0.00%	97.03%
12		95.35%	0.00%	98.02%
13		0.00%	0.00%	6.67%
14		94.37%	33.52%	74.16%
15		78.43%	4.00%	91.54%
16	Actuator 3	91.21%	36.36%	76.12%
17		98.21%	62.07%	54.73%
18		97.62%	86.67%	34.53%
19		98.30%	90.67%	48.38%
Mean		**67,60%**	**25,59%**	**71,67%**

TABLE IV.   RESULTS OBTAINED WITH RDE WITH FORGETTING.

| Item | Actuator | RDE with Forgetting | | |
		TPR	FPR	THR
1		9.19%	12.80%	48.18%
2		27.54%	4.28%	76.92%
3		0.00%	0.00%	66.45%
4	Actuator 1	61.32%	0.00%	79.60%
5		68.32%	8.14%	90.27%
6		58.42%	6.75%	86.23%
7		8.70%	16.05%	37.26%
8		100.00%	63.95%	45.54%
9		0.00%	0.00%	88.56%
10	Actuator 2	100.00%	58.46%	62.38%
11		94.87%	0.00%	98.02%
12		100.00%	43.10%	75.25%
13		18.64%	30.00%	22.06%
14		11.80%	11.03%	67.72%
15		88.24%	9.33%	90.05%
16	Actuator 3	0.00%	0.00%	54.73%
17		92.86%	15.17%	87.06%
18		0.00%	0.00%	74.85%
19		2.27%	5.78%	53.87%
Mean		**44,32%**	**14,99%**	**68,68%**

After the experiments, one can observe that TEDA provided the highest average TPR among the selected methods,

TABLE V.   RESULTS OBTAINED WITH TEDA.

| Item | Actuator | TEDA | | |
		TPR	FPR	THR
1		92.01%	6.70%	92.65%
2		83.33%	2.90%	93.31%
3		36.63%	4.50%	75.75%
4	Actuator 1	0.00%	0.00%	47.26%
5		72.28%	10.43%	88.41%
6		73.27%	4.75%	90.82%
7		100.00%	3.16%	98.80%
8		93.33%	10.47%	90.10%
9		91.30%	19.66%	81.59%
10	Actuator 2	91.67%	0.00%	97.03%
11		9.74%	0.00%	96.04%
12		93.02%	0.00%	97.03%
13		0.09%	0.00%	6.75%
14		80.76%	8.28%	88.71%
15		68.63%	0.00%	92.04%
16	Actuator 3	83.52%	34.55%	73.63%
17		83.93%	48.28%	60.70%
18		93.65%	59.20%	54.09%
19		97.16%	77.78%	55.11%
Mean		**74,96%**	**15,30%**	**77,89%**

with 74.96%, followed by RDE, with 67.60% and, finally, RDE with Forgetting, with a considerably lower value of 44.32%. Moreover, RDE was not able to detect 5 of the 19 faults, while TEDA did not succeed in only 2 of them.

Although RDE with Forgetting obtained the lower TPR value, it is more responsive to input changes than the other methods, since it managed to partially detect faulty samples in all faults, including those where RDE and TEDA did not succeed. This feature of RDE with Forgetting is also noticed when analyzing the good results obtained for short fault streams (where the fault occurs for less than 60s, e.g. fault items #8 to #12).

Regarding to the average FPR, RDE with Forgetting and TEDA obtained similar results (about 15%). RDE, on the other hand, resulted in a higher FPF of 25.59%, since the RDE algorithm takes a little longer to detect the end of a fault. Therefore, even when the plant has already left the faulty state of operation, RDE is not able to promptly detect such changes for a few fault items. This is particularly true for the cases where the normal state of operation after the fault is spatially far from the normal state of operation before the fault, e.g. fault items #15 to #19.

Finally, TEDA also obtained the highest average THR, with 77.89%, followed again by RDE, with 71.67% and, then, RDE with Forgetting with 68.68%. In Figure 5, the charts corresponding to the results obtained for fault item #1 during the analyzed time interval is shown. The red dotted lines define the actual beginning and end of the fault. Fig. 5(a) shows the two input signals used in all three methods. Fig. 5(b), 5(c) and 5(d) show the detection results for RDE, RDE with Forgetting and TEDA, respectively. Similarly, Fig. 6 shows the charts of input signals and results for fault item #7.

It should be noticed that, in both fault items, RDE is able to promptly detect the beginning of the fault, however, takes a little while to detect its end, resulting in a large number of false positive, i.e. a high FPR. RDE with Forgetting also detects the beginning of the fault, however, after that, the input signals remain without any significant changes, resulting in a low TPR. TEDA, on the other hand, obtained the highest THR for those fault items, being able to correctly detect the
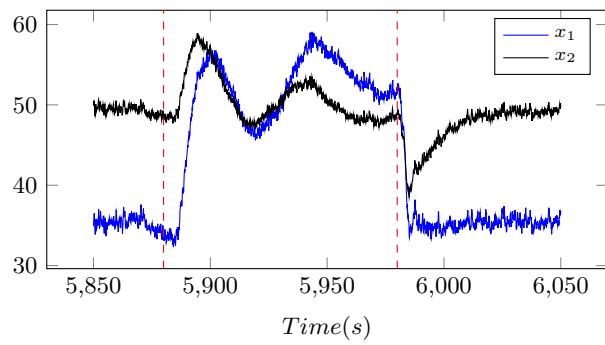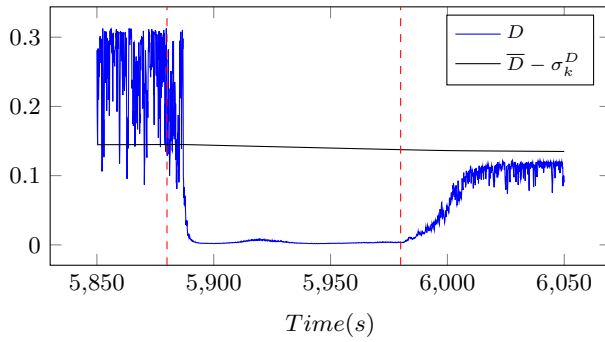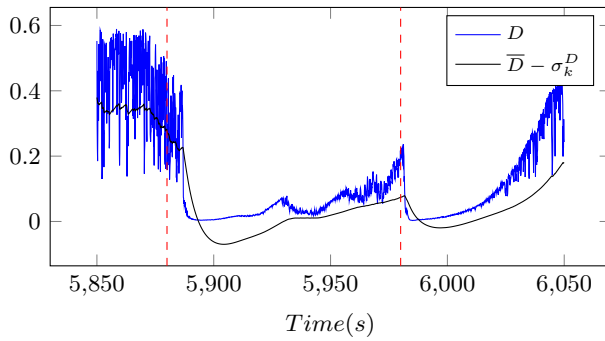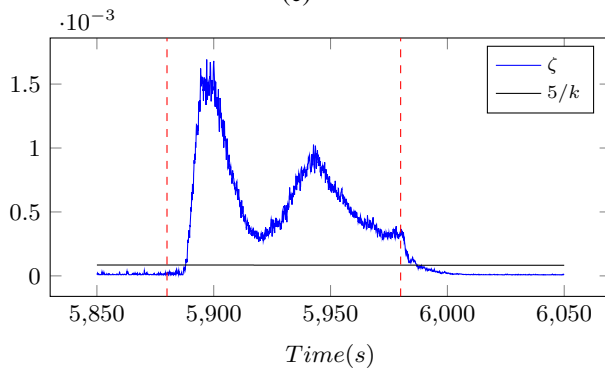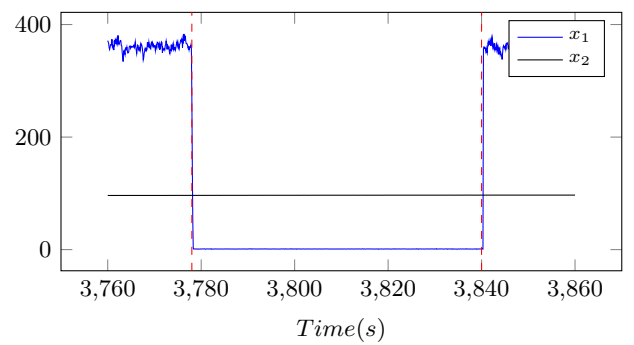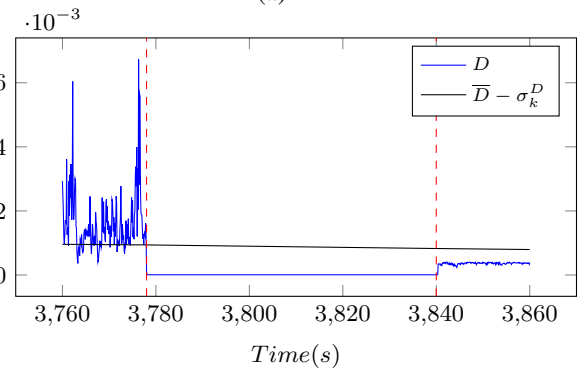
Fig. 5. Charts obtained in fault item #1: (a) original signal, (b) RDE, (c) RDE with forgetting e (d) TEDA.

Fig. 6. Charts obtained in fault item #7: (a) original signal, (b) RDE, (c) RDE with forgetting e (d) TEDA.
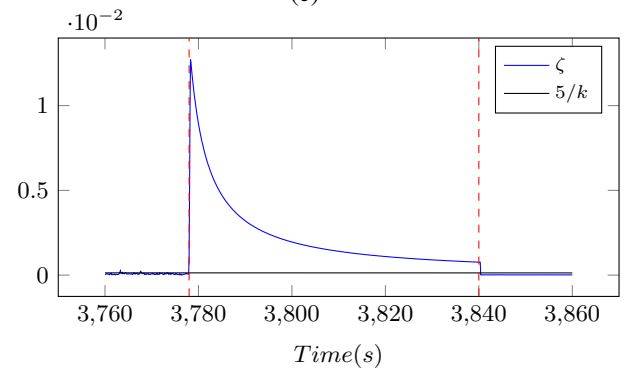
beginning and end of the faults.

V. CONCLUSION

In this work, a comparative study of three recently introduced methods for outlier detection was proposed. RDE, RDE

with Forgetting and TEDA were applied to a well-known real data fault detection application and the results were analyzed. It should be highlighted that, the detection of the beginning and end of the faults is not in the scope of this work. To do so, additional use of signal analysis is often necessary.

The detection of an outlier, using each of the three methods, is straightforward; the distance from each read point to all previous data samples is calculated and used in the particular equation for each method. A calculated measure is, then, compared to a statistically defined threshold. All mentioned methods are based on recursive equations, resulting in quick calculations and very low computational effort, speed- and memory-wise, allowing their use in online and real-time applications.

RDE presented satisfactory results for outlier detection, however, obtained the less impressive values for false positive detection. Moreover, for a few short length fault items, RDE was not able to detect one single outlier due to the low sensibility to input changes intrinsic to RDE method.

RDE with Forgetting, on the other hand, is considerably more responsive to such changes, being able to partially detect the faults that were not detected by the other methods. Moreover, RDE with Forgetting resulted in a low false positive rate. However, due to this high responsive structure, the true positive rate was significantly low. Different values of α can be used in order to achieve better results.

Finally, TEDA presented the best results among all analyzed methods, obtaining the highest outlier detection rate and the lowest false positive rate. Moreover, TEDA succeeded in detecting 17 of 19 evaluated fault streams.

REFERENCES

[1] D. M. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[2] K. Singh and D. S. Upadhyaya, "Outlier detection: Applications and techniques," *International Journal of Computer Applications*, vol. 89, no. 6, pp. 307–323, 2014.

[3] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligent Review*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[4] C. C. Aggarwal, *Outlier Analysis*. Springer, 2013.

[5] V. Venkatasubramanian, "Abnormal events management in complex process plants: Challenges and opportunities in intelligent supervisory control," in *Foundations of Computer-Aided Process Operations - FOCAPO2003*, January 2003, pp. 117–132.

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

[7] P. Angelov, *Autonomous Learning Systems: From Data to Knowledge in Real Time*. John Willey and Sons, 2012.

[8] ——, "Anomalous system state identification," 2012, gB1208542.9, priority date 15 May 2012.

[9] B. S. J. Costa, P. Angelov, and L. A. Guedes, "Real-time fault detection using recursive density estimation," *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 4, pp. 428–437, 2014.

[10] ——, "Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier," *Neurocomputing*, vol. 150, no. A, pp. 289–303, 2015.

[11] B. Costa, P. Angelov, and L. A. Guedes, "A new unsupervised approach to fault detection and identification," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 1557–1564.

[12] ——, "Rde with forgetting: an approximate solution for large values of k with an application to fault detection problems," in *SLDS2015, THE THIRD INTERNATIONAL SYMPOSIUM ON STATISTICAL LEARNING AND DATA SCIENCES*, 2015, to appear.

[13] P. Angelov, "Anomaly detection based on eccentricity analysis," in *Proc. IEEE Symposium Series in Computational Intelligence (SSCI 2014)*, Orlando, Florida, U.S.A., Dec. 2014.

[14] D. Kangin and P. Angelov, "Recursive svm based on teda," in *THE THIRD INTERNATIONAL SYMPOSIUM ON STATISTICAL LEARNING AND DATA SCIENCES (SLDS 2015)*, London, UK, April 2015, pp. 117–132, to appear, 2015.

[15] P. Angelov, "Evolving takagi-sugeno fuzzy systems from data streams (ets+)." in *Evolving intelligent systems : methodology and applications*, ser. IEEE Press series in Computational Intelligence, P. Angelov, D. Filev, and N. Kasabov, Eds. John Wiley and Sons and IEEE Press, 2010, pp. 21–50.

[16] M. Bartys, R. Patton, M. Syfert, S. de las Heras, and J. Quevedo, "Introduction to the damadics actuator fdi benchmark study," *Control Engineering Practice*, vol. 14, no. 6, pp. 577–596, 2006.

[17] M. D'Angelo, "Fuzzy/bayesian change point detection approach to incipient fault detection," *IET Control Theory & Applications*, vol. 5, pp. 539–551(12), March 2011. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-cta.2009.0033

[18] Y. Kourd, N. Guersi, and D. Lefebvre, "Neuro-fuzzy approach for default Diagnosis: Application to the DAMADICS," in *IEEE International Conference on Digital Ecosystems and Technologies*, 2010, pp. 107–111.

[19] V. Puig, J. Quevedo, T. Escobet, F. Nejjari, and S. de las Heras, "Passive Robust Fault Detection of Dynamic Processes Using Interval Models," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 1083–1089, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1109/tcst.2007.906339

[20] Damadics information web site. [Online]. Available: http://diag.mchtr.pw.edu.pl/damadics/

[21] A. Malhi and R. X. Gao, "Pca-based feature selection scheme for machine defect classification." *IEEE T. Instrumentation and Measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[22] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, 2011.

[23] F. Song, D. Mei, and H. Li, "Feature selection based on linear discriminant analysis," in *Proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application - Volume 01*, ser. ISDEA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 746–749. [Online]. Available: http://dx.doi.org/10.1109/ISDEA.2010.311

[24] S. S. Pande and B. S. Prabhu, "An expert system for automatic extraction of machining features and tooling selection for automats," *Comput.-Aided Eng. J.*, vol. 7, no. 4, pp. 99–103, Aug. 1990. [Online]. Available: http://dx.doi.org/10.1049/cae.1990.0024

[25] S. Dash, R. Rengaswamy, and V. Venkatasubramanian, "Fuzzy-logic based trend classification for fault diagnosis of chemical processes," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 347 – 362, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135402002144

[26] M. J. Anzanello, *Feature Extraction and Feature Selection: A Survey of Methods in Industrial Applications*. John Wiley & Sons, Inc., 2010. [Online]. Available: http://dx.doi.org/10.1002/9780470400531.eorms0321

[27] M. D. Levine, "Feature extraction: A survey," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1391–1407, Jun. 2005.

[28] H. Liu, G. Chen, S. Jiang, and G. Song, "A survey of feature extraction approaches in analog circuit fault diagnosis." in *PACIIA (2)*. IEEE Computer Society, 2008, pp. 676–680.

[29] M. Grbovic, S. Vucetic, W. Li, P. Xu, and A. K. Usadi, "A boosting method for process fault detection with detection delay reduction and label denoising," in *Proceedings of the First International Workshop on Data Mining for Service and Maintenance*, ser. KDD4Service '11. New York, NY, USA: ACM, 2011, pp. 7–11. [Online]. Available: http://doi.acm.org/10.1145/2018673.2018675