FaultBuster: data driven fault detection and diagnosis for industrial systems

Bergantino, Nicola; Caponetti, Fabio; Longhi, Sauro

Published in:
7th Workshop on Advanced Control and Diagnosis

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

FaultBuster: data driven fault detection and diagnosis for industrial systems

Nicola Bergantino * **Fabio Caponetti** **,*,*** **Sauro Longhi** **

* *Integra Software, Jesi, Italy (e-mail:n.bergantino@integrasoftware.it)*
** *Università Politecnica delle Marche, DIIGA, Ancona, Italy (e-mail: f.caponetti@diiga.univpm.it, sauro.longhi@univpm.it)*
*** *Technical University of Denmark, DTU-Electrical Engineering, Kgs.Lyngby, Denmark*

AbstractEfficient and reliable monitoring systems are mandatory to assure the required security standards in industrial complexes. This paper describes the recent developments of FaultBuster, a purely data-driven diagnostic system. It is designed so to be easily scalable to different monitor tasks. Multivariate statistical models based on principal components are used to detect abnormal situations. Tailored to alarms, a probabilistic inference engine process the fault evidences to output the most probable diagnosis. Results from the DX 09 Diagnostic Challenge shown strong detection properties, whereas the need of further investigations in the diagnostic system.

Keywords: Statistical Process Control, Fault detection, Artificial intelligence, Diagnostic inference

1. INTRODUCTION

Alarms are essential in every process, system and industrial complex. Unplanned shut-downs are one of the most serious sources of production loss. Even if triggered by minor problems, intervents may require the temporary process stop. Detect and diagnose quickly and precisely the root cause of a developing abnormal situation is a key to keep systems working as smoothly as possible.

Plant monitoring represents also a way to fullfill environmental regulations on emissions. Governments impose fines on violations of environmental protection regulations, which erode profits and the social image.

The other and probably the most important reason for plant monitoring is related to safety. According to the Abnormal Situation Management Consortium, petrochemical plants on average suffer a major incident every three years, which usually cause human casualties. These incidents occur not usually because of major design flaws or equipment malfunctions, but rather simple mistakes. In all these cases a prevention system (comprising means of detection and diagnosis, logic/control equipment and independent means of control) would probably have prevented the deploying of these situations.

This paper presents the first developments of FaultBuster, an industrial fault detection and diagnosis system. It is built to extract as much information as possible from the data flowing from and to the monitored plant without embedding specific process knowledge.

FaultBuster is composed by co-operating modules organised in the Integra Agent Framework (IAF), an agent architecture discussed by Caponetti et al. (2009). Modules can be exchanged or modified independently to let the system be easily scalable. FaultBuster may tune on-line a statistical model to detect deviant situations. Tailored to detection is the diagnosis. Based on the knowledge learnt from past observations and diagnosis examples it gives the set of the probable faults that may be occurring.

FaultBuster has been participant of the DX 09 Diagnostic Challenge Competition (`http://dx-competition.org`), demonstrating low rates of missed/false alarms whereas some problems in the diagnostic part. The competition winner, ProDiagnose used a probabilistic approach, accomplishing the diagnostic task with Bayesian Network models compiled to Arithmetic Circuits (Ricks and Mengshoel, 2009).

In this paper it is described the fault detection and diagnosis algorithm implemented in FaultBuster and the DX 09 Competition results are presented as benchmark.

2. ADAPT TESTBED

Researchers at NASA Ames Research Centre have developed the Advanced Diagnostics and Prognostics Testbed (ADAPT). It allows performance assessment of diagnostic algorithms in a standardised testbed and repeatable failure scenarios. The hardware of the testbed is an Electrical Power System (EPS) of a space exploration vehicle and consists of three major modules: a power generation unit, a power storage unit and a power distribution unit (See Fig.1). The system has hybrid dynamics where mode transitions are commanded or triggered by events. the installed sensors provide data sample only at the rate of $2Hz$, which cannot capture the dynamics of some ADAPT subsystems that operate at much higher frequencies. ADAPT has been used as basis for the Diagnostic Challenge 2009 to which all the data used to produce the results discussed refers to.

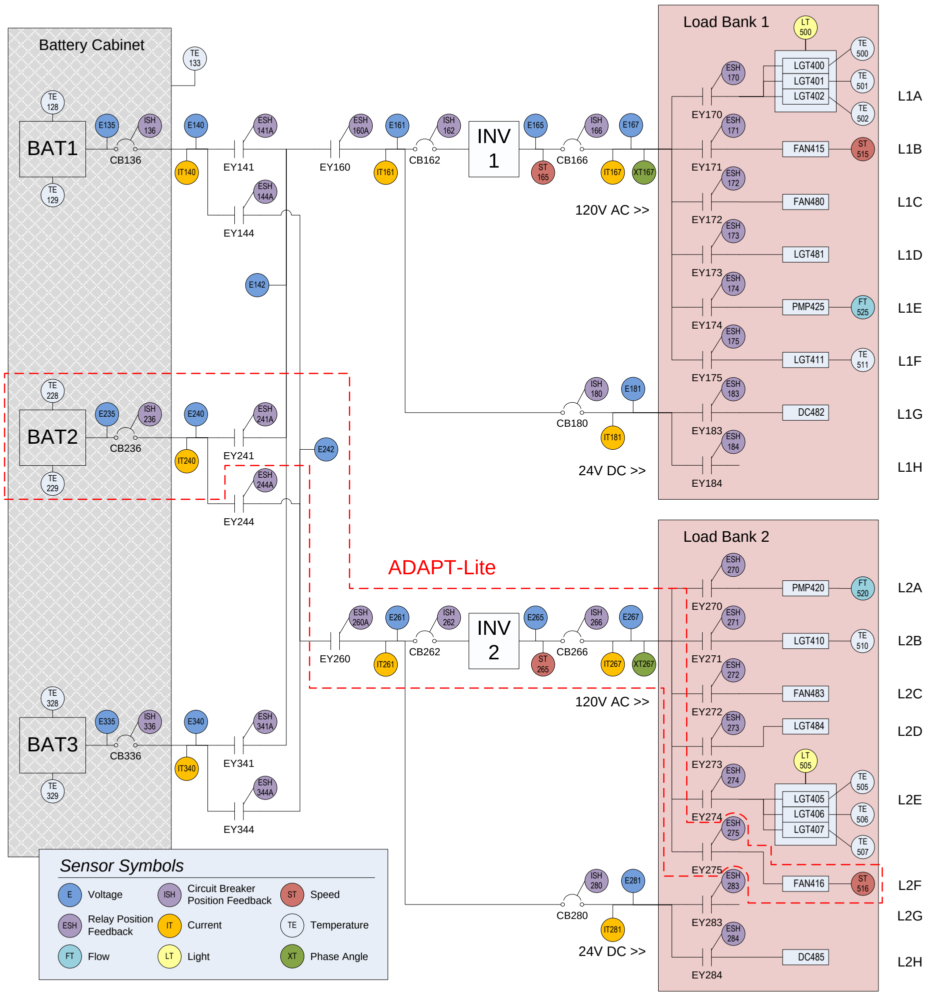

Figure 1. The ADAPT Tier 2 EPS. ADAPT Tier 1 (Lite) is a subset of Tier 2. (Courtesy of Tolga Kurtoglu)

3. FAULTBUSTER

FaultBuster is engineered to fullfill the requirements of a fast, accurate, reliable and reconfigurable diagnostic system. Opposite and tight constraints lead to implement a tradeoff solution.

The system was born to supervise tightly coupled, complex industrial systems. Quite often poor process knowledge is available, whereas huge archives of data may be accessible. This is the case when small-medium enterprises wants to improve their throughput and quality by monitoring already running machinery.

Statistical process monitoring techniques have been heavily researched in the last few years. Multivariate statistical methods based on Principal Component Analysis (PCA), partial Least Squares, and Independent Component Analysis have been used and extended with success in various applications (Qin, 2009; Liu et al., 2009; Du et al., 2007; Al Ghazzawi and Lennox, 2008; Liu et al., 2005; Bakshi, 1998).

FaultBuster processes all the observations using a statistical reference model and an adaptive detection scheme. Detection is based on residuals built from multivariate statistics of the data projected through the model. Once an alarm is issued the contribution of each sensor to the abnormal situation is determined. In this way, the vector of contribution rates represents the fault pattern that an inference engine based on Markov logic networks has to interpret. The inference output is the set of most probable faults. Fig. 2 shows the concept scheme of the system.

3.1 Statistical model

FaultBuster can be bootstrapped on a pre built model or configured to fit on-line a PCA model. For industrial systems, with slow dynamics, a pre-built model adapted on-line would be the best option. Due to the high number of working modes which ADAPT may present and the limited amount of available training examples available, a pre-built model resulted to be too conservative.

ADAPT works fault-less for $30s$ after a boot. The first observations collected on-line compose a training dataset constructed in a way that columns represent the monitored variables (m) and each row an observation (n).

$$X \in A_{(n \times m)}$$

Because of the different magnitude of variables, the dataset is standardised to null mean and unit variance. Boolean

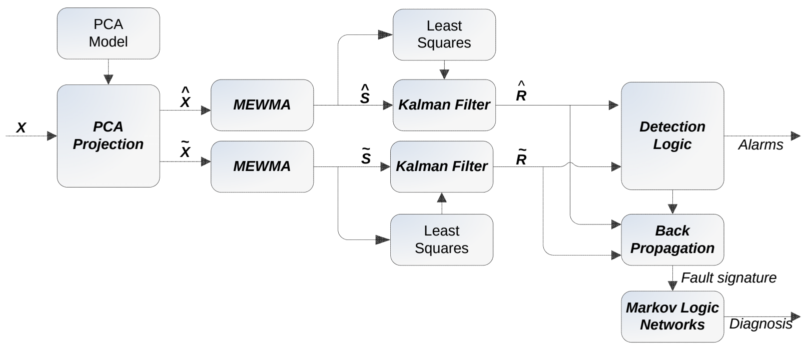

Figure 2. Concept scheme of FaultBuster.

observations from ADAPT are fed to the model by adding a Gaussian noise with small standard deviation to the true value.

The dataset can be decomposed as:

$$X = \hat{X} + E, \tag{1}$$

The matrices $\hat{X}$ and E represent the modeled and the residual variation of X.

$$\hat{X} = TP^T,$$
$$E = \hat{T}\hat{P}^T,$$

where $T \in \Re^{n\times d}$ and $P \in \Re^{m\times 1}$ are the score and loading matrices.

The decomposition of X is such that the matrix $[PP^T]$ is orthonormal and $[TT^T]$ is orthogonal. The columns of P are eigenvectors of the correlation matrix R associated to the largest l eigenvalues. The columns of $\hat{P}$ are the remaining eigenvectors of R. The correlation matrix is evaluated on the scaled dataset as

$$R = \frac{1}{n-1} XX^T$$

The PCA model partitions the measurement space (m dimensional) into two orthogonal subspaces. One spanned by the first l principal components, in which the normal data variations should occur, and a residual space where abnormal situations and noise fall. The interested reader may refer to Jackson (1991).

Once a command is issued, ADAPT is supposed to change working mode. Several solution may be appliable to continue the monitoring, e.g. a new PCA model may be trained. To minimise the work load and supervise fast dynamical systems like ADAPT, FaultBuster implements an adaptive detection solution.

To detect faults in discrete dynamic, FaultBuster has to be extended with a discrete observer by embedding knowledge on the expected state after commands or triggers.

3.2 Fault detection

Each observation x is decomposed by the PCA model into:

$$\hat{x} = PP^T x, \tag{2}$$

the projection on the principal component subspace (PCS), and

$$\tilde{x} = (I - PP^T)x, \tag{3}$$

the projection on the residual subspace.

Two statistical distance measures are commonly computed to generate residual signals, the Hotelling's T^2 and the squared prediction error (SPE)

$$T^2(x) = ||D_{\lambda_k}^{-1/2} P^T x||^2, \tag{4}$$
$$SPE(x) = ||\tilde{x}||^2 = x^T (I - PP^T)x. \tag{5}$$

Where $D_{\lambda_k}^{-1/2} = diag(\lambda_i^{-1/2})$ with $\lambda_{i=1,\dots,l}$ equal to the first l eigenvalues of the correlation matrix R.

Both statistics can be evaluated against fixed thresholds designed on the average run length (ARL). Due to the hybrid nature of ADAPT, poor results have been obtained utilising fixed thresholds. As done previously by Wang and Tsung (2008), FaultBuster tries to improve the detection performances by using a predictor on the PCA subspaces issuing an alarm only after violation of an adaptive threshold.

A multivariate exponentially weighted moving average (MEWMA) is used in each subspace (x^p may be $\hat{x}$ or $\tilde{x}$) to overcome the T^2 chart limitations (Montgomery, 2005).

$$Z_i = \alpha x_i^p + (1-\alpha) Z_{i-1}, \tag{6}$$

where $0 < \alpha \le 1$ and $Z_0 = 0$. The control chart is

$$T^2(Z_i) = Z_i^T \Sigma_{Z_i}^{-1} Z_i, \tag{7}$$

where the covariance matrix is

$$\Sigma_{Z_i} = \frac{\alpha}{2-\alpha}\left[1 - (1-\alpha)^{2i}\right]\Sigma, \tag{8}$$

where Σ is a diagonal matrix containing the eigenvalues of R corresponding to the subspace considered.

The MEWMA signals are monitored by a set of Kalman filters on the four signal features in Tab. 1. To maintain a simple implementation each feature has its own Kalman filter. The models are based on a linear regression updated on-line by least square minimisation each n_{ls} samples, allowing to handle eventual non linearity in the statistics. The detection of both abrupt and progressive faults depends on a correct design of n_{ls}.

Table 1. Features used for residual generation.

Feature	Description
S_i	Value of $\hat{T}^2(\hat{Z}_i)$ or $\tilde{T}^2(\hat{Z}_i)$
$\Delta S_i = S_i - S_{i-1}$	Approximated first derivative
$\Delta^2 S_i = \Delta S_i - \Delta S_{i-1}$	Approximated second derivative
$f = \Delta S_i * \Delta^2 S_i$	Relation between derivatives

A 3σ control chart is used to monitor the Kalman prediction. The control variance which defines the Upper and

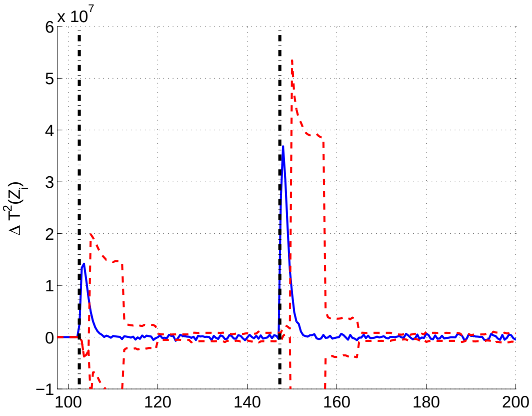

Figure 3. *Exp_675_pb_t2*, $\Delta T^2(Z_i)$ 3σ control chart. The Kalman prediction is the solid line while the UCL and LCL control limits are denoted as dotted lines. Vertical dashed-dotted lines mark the faults.

Lower Control Limit (UCL, LCL) is the estimated Kalman variance. A detection is issued if all the control charts of a subspace register a violation of the same control limit.

Fig. 3 shows the control chart for ΔS_i in *Exp_675_pb_t2*. The *Inverter_2* fails off and afterward the position sensor *ESH273* fails stuck closed. This experiment shows the ability of FaultBuster to detect faults on components not directly observable (Inverter) and multiple faults by the combination of the single feature detectors (Fig. 4).

To manage intermittent faults, after each alarm a reference model has to be stored to establish when the fault disappears. The detection capability of progressive faults with slow dynamics has to be investigated, since some limitations expected due to the adaptivity of the detector.

As response to command, the detection is inhibited for i_m samples. New feature models are fitted on the data, avoiding the computation of a new PCA model.

3.3 Observation identification

After an alarm the anomaly source have tos be identified. Several approaches have been proposed to boost the identification capabilities of the commonly used contribution plots (Mnassri et al., 2008). Since the detection is based on MEWMA signals and on some non-Gaussian sensors, FaultBuster explores and implements an alternative solution.

The PCA model is seen as a Multi-Layer Perceptron network (MLP) where: the output stage represents the PCA projection, the hidden the observations, and the input stage is not used. By retropropagating the quadratic error between the mean of the last n_m normal measures and the abnormal, the contribution rate of each sensor to the abnormality is estimated.

The corresponding MLP realises the mapping $Y = X^T W$ where $W = P$ if the alarm comes from the PC subspace ($W = \hat{P}$ otherwise). Letting x_{ref} given by the mean of the last n_m fault-less observations, and x_{flt} be the faulty observation, the quadratic error is:

$$E = Err^2 = \frac{1}{2}\left(x_{flt}^T W - x_{ref}^T W\right)^2 .$$

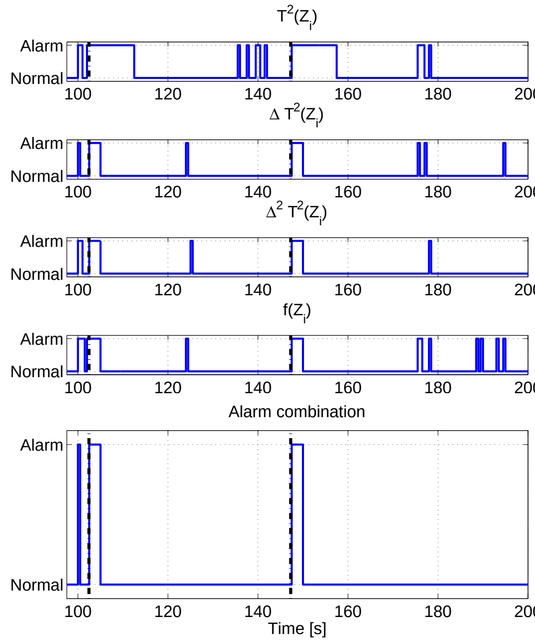

Figure 4. *Exp_675_pb_t2* PC subspace alarm signals. Scalar detector would be inadequate while their combination lead to 2 true positives and a false alarm in $t = 100$.

The hidden node j is responsible for some fraction of the error Err in each of the output nodes to which it connects (Russell and Norvig, 2003). Thus, the values of Err are divided according to the strength of the connection between the hidden node and the output node and propagated back to provide the Δ_j values for the hidden layer. The propagation rule for the hidden nodes j is,

$$\Delta_j = (x_j^{flt} - x_j^{ref}) \sum_i W_{j,i} Err_i. \tag{9}$$

Where Err_i is the ith component of the error vector $Err = (x_{flt}^T W - x_{ref}^T W)$. The term $(x_j^{flt} - x_j^{ref})$ is intended as the first derivative of the not known activation function. The resulting vector Δ is interpreted as the contribution of each sensor to the abnormal situation.

To filter false alarms, Δ is pruned using a threshold. Once fitted a PCA model the backpropagation is tested using normal observations as examples and targets. The maximum Δ_j among all the sensors is stored in Δ_{max}. It represent the maximum expected contribution rate for a sensor in nominal conditions. At run-time, each Δ_j is zeroed if less than Δ_{max}. If this results into a null vector the alarm is classified as false and removed. Fig. 5 shows how the alarms issued by the detectors are filtered, removing the false alarm characterised by zero contributions and leaving the true alarms.

An analogous solution for the PCA has been discussed by Chiang et al. (2001), where each component of the weight matrix has been scaled by the corresponding variance.

$$\Delta_j = \sum_i \frac{t_i}{\sigma_i^2} W_{i,j} (x_j^{flt} - \mu_j)$$

Where t_i is ist column of the score matrix T, σ_i^2 the related variance and μ_j is the off-line learnt mean value of the sensor j. FaultBuster does not use the PCA model to

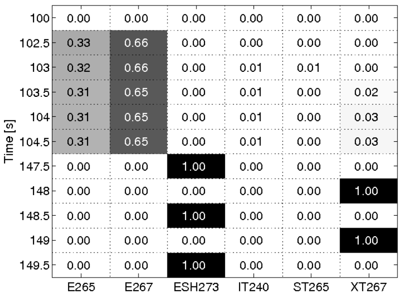

Figure 5. *Exp_675_pb_t2*, sensor identification. The alarm for $t = 100$ is discarded since $\Delta_j < \Delta_{max} \forall j$.

directly estimate the contribution rate, since a new PCA model is not trained after each mode switch.

3.4 Fault diagnosis

PCA can efficiently isolate faults on the monitored variables, but as also described recently by Mnassri et al. (2008) has limitations to diagnose problems in components non directly observable. FaultBuster tries to fill the gap by using a diagnostic module based on probabilistic reasoning and first order logic. Markov logic combines the two by attaching weights to first-order formulae and viewing them as templates for features of Markov networks (Richardson and Domingos, 2006). Weights are tuned by learning and the implementation in FaultBuster is based on Alchemy.

ADAPT and normal industrial systems are in general complex. To manage the number of components and failure modes, the inference is done hierarchically. Using the evidences a general Knowledge Base (KB) outputs the component class, i.e. pump, relay, voltage sensor. By inference on specific component class KBs, the probable faulty component is individuated with its failure mode.

The predicates defined in the KBs are reported in Tab. 2. Each predicate may make use of the variables in Tab. 3. Them describe the facts that the KB is able to interpret. Evidences are generated from a fault pattern as an ordered sequence of discrete contributions in the form of *oocss* and *oocb*. The discrete value is obtained by quantisation of the contribution space (See values in Tab.3).

Table 2. Knowledge base predicates

Predicate	Description
FA	False alarm
oocsb(boolsens)	Boolean variable fault contribution
oocss(contsens,value)	Continuous variable fault contribution
command(cmd)	Command sent
fault(faultClass)	Fault diagnosis

The KB for the component classes is composed by simple first order logic rules not specific to the monitored system.

If a command has been executed recently and an alarm has been issued than it likely to be a false alarm. This rule allows the system to move from one operating point to another avoiding nuisance alarms.

```
command(+c)^oocss(+s,+v)=>FA
```

If no commands has been sent, the presence of a fault have to be investigated, hence a false alarm is unlikely to be.

```
!command(c)=>!FA
!command(c)^oocss(+s,+v)=>fault(+x)
!command(c)^oocsb(+s)=>fault(+x)
```

The inference on the general KB leads to the probability distribution among the fault classes in Tab. 3. Knowing the most likely fault class, the class-specific KBs can be interrogated (Tab. 4). For example, in ADAPT system, the knowledge base relative to a large fan looks like:

```
//Variable declaration
FaultModeLargeFan = {OverSpeed,
UnderSpeed, FailedOff}
[..]
//Predicate declaration
FaultLargeFan(LargeFan, FaultModeLargeFan)
[..]
//Rules
!command(z)^oocss(+s,+v)=>FaultLargeFan(+x,+f)
!command(z)^oocsb(+s)=>FaultLargeFan(+x,+f)
```

Table 4. *Exp_675_pb_t2*, Markov logic inference.

Predicate	Probability
General KB	
Fault(BasicLoad)	0.00
Fault(Battery)	0.00
Fault(BooleanSensor)	0.00
Fault(CircuitBreaker)	0.00
Fault(CommandableCircuitBreaker)	0.04
Fault(Inverter)	**0.94**
Fault(LargeFan)	0.00
Fault(LightBulb)	0.00
Fault(Relay)	0.03
Fault(ContinuousSensor)	0.57
Fault(WaterPump)	0.00
FA	0.00
Inverter KB	
FaultInverter(INV1, FailedOff)	0.00
FaultInverter(INV2, FailedOff)	**0.96**

No specific knowledge about the physical system interconnection has been modeled. This gives generalisation power at the cost of depending on the amount and quality of examples used to train the initial KBs. The diagnostic performances are expected to increase with the amount of information modeled in the single KBs and with the number of faults that occour in the monitored plant. For industrial applications the system can be taught to classify new fault patterns by the plant operators. Planned extension is in the way to use Bond Graph Models to generate interconnection rules to boost the diagnostic performance by evaluating possible failure chains.

4. RESULTS

To demonstrate FaultBuster in action the metrics in (Kurtoglu et al., 2009) has been evaluated in 233 ADAPT scenarios and summarised in Tab.5. The data consists with either nominal, single, double or triple fault, with various relay and circuit breaker open/close (Kurtoglu et al., 2009). Each scenario starts with ADAPT unpowered. Two configurations has been tested: fully adaptive, namely FBuster, and with a bootstrap model, FBuster^M.

The fully adaptive solution shown very low false positive/negative rate and a fair detection accuracy. This is

Table 3. Knowledge base variables

Variable	Description
boolsens	ESH141A, ESH144A, ESH160A, ESH170, ESH171, ESH172, ESH173, ESH174, ESH175, ESH183, ESH184, ESH241A, ESH244A, ESH260A, ESH270, ESH271, ESH272, ESH273, ESH274, ESH275, ESH283, ESH284, ESH341A, ESH344A, ISH136, ISH162, ISH166, ISH180, ISH236, ISH262, ISH266, ISH280, ISH336
contsens	E135, E140, E142, E161, E165, E167, E181, E235, E240, E242, E261, E265, E267, E281, E335, E340, FT520, FT525, IT140, IT161, IT167, IT181, IT240, IT261, IT267, IT281, IT340, LT500, LT505, ST165, ST265, ST515, ST516, TE128, TE129, TE133, TE228, TE229, TE328, TE329, TE500, TE501, TE502, TE505, TE506, TE507, TE510, TE511, XT167, XT267
cmd	EY136_OP, EY236_OP, EY336_OP, EY141_CL, EY144_CL, EY160_CL, EY170_CL, EY171_CL, EY172_CL, EY173_CL, EY174_CL, EY175_CL, EY183_CL, EY184_CL, EY241_CL, EY244_CL, EY260_CL, EY270_CL, EY271_CL, EY272_CL, EY273_CL, EY274_CL, EY275_CL, EY283_CL, EY284_CL, EY341_CL, EY344_CL
faultClass	BasicLoad, Battery, BooleanSensor, CircuitBreaker, CommandableCircuitBreaker, Inverter, LargeFan, LightBulb, Relay, ContinuousSensor, WaterPump
value	Big, Medium, Small

Table 5. DX Competition results metrics

Metric	FBuster	FBuster^M	ProDiagnose
Detection Accuracy	74%	83.1%	88.33%
False Positives Rate	2.53%	15.56%	7.32%
False Negatives Rate	38.96%	17.53%	13.92%
Classification Errors	236	217	76
Mean Time To Detect (ms)	14553.3	17789.9	5873
Mean Time To Isolate (ms)	48893.5	54104.6	11988

related to its capacity to better tune a reference PCA model scaled on the actual working status. False negative and detection rate are influenced strongly by the inability to detect fault in the first 120 samples. This limitation has been removed in FBuster^M by providing an off-line model. Results confirm the increase of detection rate by a leverage of the false and missed alarms. Isolation performance confirm the need to introduce process knowledge to boost the diagnosis.

5. CONCLUSION

FautlBuster was born to diagnose complex industrial system where limited or no process knowledge were available. FaultBuster combined the performances of a statistical model based detector and of a probabilistic first order logic inference engine. The system demonstrated good detection capabilities in the DX 09 Diagnostic Challenge. Both detection and diagnosis modules have been based on knowledge directly extracted from example data to explore the capabilities of a pure data-driven system. The detection module needs minor refinements, whereas to better diagnose, the KBs have to embed process knowledge or have to be trained on larger example sets. FaultBuster demonstrated to be computationally lightweight since the inference was executed only after confirmed alarms.

ACKNOWLEDGEMENTS

Giovanni Mazzuto and Fabio Virgulti are acknowledged for having contributed to the solutions in FaultBuster.

REFERENCES

Al Ghazzawi, A. and Lennox, B. (2008). Monitoring a complex refining process using multivariate statistics. *Control Engineering Practice*, 16(3), 294–307.

Bakshi, B.R. (1998). Multiscale pca with application to multivariate statistical process monitoring. *AIChE Journal*, 44(7), 1596–1610.

Caponetti, F., Bergantino, N., and Longhi, S. (2009). Fault tolerant software: a multi-agent system solution. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*.

Chiang, L., Russell, E., and Braatz, R. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer.

Du, Z., Jin, X., and Wu, L. (2007). Fault detection and diagnosis based on improved pca with jaa method in vav systems. *Building and Environment*, 42(9), 3221–3232.

Jackson, J.E. (1991). *A User's Guide to Principal Components*. Wiley.

Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., and Wright, S. (2009). Benchmarking diagnostic algorithms on an electrical power system testbed. In *International Conference on Prognostics and Heath Management*.

Liu, J., Lim, K.W., Srinivasan, R., and Doan, X.T. (2005). On-line process monitoring and fault isolation using pca. *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 658–661.

Liu, X., Kruger, U., Littler, T., Xie, L., and Wang, S. (2009). Moving window kernel pca for adaptive monitoring of nonlinear processes. *Chemometrics and Intelligent Laboratory Systems*, 96(2), 132–143.

Mnassri, B., El Adel, E., and Ouladsine, M. (2008). Fault localization using principal component analysis based on a new contribution to the squared prediction error. *2008 16th Mediterranean Conference on Control and Automation*, 65–70.

Montgomery, D.C. (2005). *Introduction to Statistical control*. Wiley.

Qin, J. (2009). Data-driven fault detection and diagnosis for complex industrial processes. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62, 107–136.

Ricks, B.W. and Mengshoel, O.J. (2009). The diagnostic challenge competition: Probabilistic techniques for fault diagnosis in electrical power systems. In *Proc. of the 20th International Workshop on Principles of Diagnosis (DX-09)*. Stockholm, Sweden.

Russell, S. and Norvig, P. (2003). *Artificial intelligence: a modern approach*. Prentice Hall.

Wang, K. and Tsung, F. (2008). An adaptive t2 chart for monitoring dynamic systems. *Journal of Quality Technology*, 40(1), 109–123.