LEARNING DISCRETE PROBABILISTIC MODELS FOR APPLICATION IN MULTIPLE FAULTS DETECTION

Luis E. Garza Castañón
Department of Mechatronics and Automation, ITESM Monterrey Campus, Mexico
legarza@itesm.mx

Francisco J. Cantú Ortíz
Research and Graduate Programs Office, ITESM Monterrey Campus, Mexico
cantu@itesm.mx

Rubén Morales-Menéndez
Center of Innovation and Technology Design, ITESM Monterrey Campus, Mexico
rmm@itesm.mx

Keywords: Fault Detection, Bayesian Networks, Machine Learning, Power Networks.

Abstract: We present a framework to detect faults in processes or systems based on probabilistic discrete models learned from data. Our work is based on a residual generation scheme, where the prediction of a model for process normal behavior is compared against measured process values. The residuals may indicate the presence of a fault. The model consists of a general statistical inference engine operating on discrete spaces, and represents the maximum entropy joint probability mass function (pmf) consistent with arbitrary lower order probabilities. The joint pmf is a rich model that, once learned, allows us to address inference tasks, which can be used for prediction applications. In our case the model allows the one step-ahead prediction of process variable, given its past values. The relevant dependencies between the forecast variable and past values are learnt by applying an algorithm to discover discrete bayesian network structures from data. The parameters of the statistical engine are also learn by an approximate method proposed by Yan and Miller. We show the performance of the prediction models and their application in power systems fault detection.

1 INTRODUCTION

The problem of fault detection in processes has received great attention in last decades, and a wide variety of methods have been developed, most of them based on fault detection and isolation (FDI) techniques or in knowledge-based methods (Venkatasubramanian et al., 2003). FDI is based on the use of analytical redundancy rather than physical redundancy. In FDI the redundancy in static and dynamic relationships between process inputs and outputs is exploited (Frank, 1990). The methods used by FDI can be summarized in parity space approach, state estimation approach, fault detection filtering, and parameter identification approach. In every case, a mathematical model of process is required, either in state-space or input-output form, but most of the time these models are linear systems. Since many processes exhibits a nonlinear dynamics, several methods have been developed to deal with nonlinearities such as: decoupling approach, nonlinear observers and nolinear parity spaces (Zhang and Ding, 2005). These methods are limited to work well in a small region around the point of operation or are adequate just for a limited class of nonlinear systems.

In the other hand, Knowledege-based methods rely on qualitative model descriptions in the form of neural networks, Bayesian networks, fuzzy logic or qualitative reasoning. Neural networks are widely used in fault detection and diagnosis (Xu and Chow, 2005) but they represent black box models and can not deal with missing information. Fuzzy logic uses a database with IF-THEN rules which use linguistic variables. The problem with fuzzy logic is that can not deal with incomplete information in explicit form and the overall dimension of rules may blow up strongly even for small processes (Isermann, 1997). The methods based in qualitative reasoning require a set of qualitative differential equations between process variables not easy to obtain for complex processes. Other machine learning approaches used in fault detection can be found in (Sedighi et al., 2005; Davy et al., 2006). Bayesian networks (BNs) have been lately used in fault detection and diagnosis (Yongli et al., 2006; Matsuura and Yoneyama, 2004), as they represent robust models for nonlinear systems able to deal with missing information and noise. A potential problem in BNs is the time for inference process in large domains.

A recent trend is the combination of methods to take advantage of the best aspects of every approach (Gentil et al., 2004). Our work is mainly focus in this direction.

Our fault detection method is based on a prediction model obtained from the process normal behavior time series. We can find in technical literature many approaches using machine learning techniques for time series prediction. For instance, in (Luque et al., 2007) an evolutionary approach is applied to learn a set of rules to predict local behavior of time series. In (Chen and Zhang, 2005) an adaptive network based fuzzy inference system (ANFIS) is used to predict chaotic and traffic flow time series. In (Vanajakshi and Rilett, 2007) a support vector machine (SVM) approach is used to predict traffic flow time series. In (Ma et al., 2007) evolving recurrent neural networks are presented which predict chaotic time series. None of these methods address the problem of missing information.

In our approach, we generate residuals by comparing actual measurements against a prediction given by a normal behavior model. The model structure and parameters are learned by applying machine learning techniques. The residuals behavior indicate the existence of a fault.

We test our approach by diagnosing multiple-faults events in a large power transmission network and show promising results.

2 OUR APPROACH

A general overview of the proposed approach is shown in Figure 1. Basically we generate residuals from the comparison between a process normal behavior model and the actual process values. We substitute the classical models of process normal behavior (eg. discrete linear models) with a discrete probabilistic function, whose parameters and structure are learned off-line from normal behavior process data. The probabilistic function is a general statistical inference engine, which allows inference to know the future value of a process variable, given its past values. In our case, we predict the one step-ahead value of the process variable given a set of past values. The set of relevant process variable values having direct influence on the forecast variable, are learned off-line by using an algorithm to learn discrete Bayesian networks. The output of this algorithm is a graphical causal structure, which is simplified by selecting the Markov blanket of the forecast process variable. This kind of compact probabilistic models are robust to noise, incomplete information and nonlinearities.

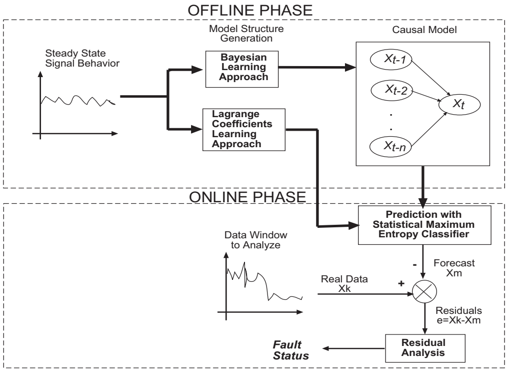

OFFLINE PHASE
Model Structure Generation
Causal Model
Steady State Signal Behavior
Bayesian Learning Approach
Lagrange Coefficients Learning Approach
X_{t-1}
X_{t-2}
X_t
X_{t-n}
ONLINE PHASE
Prediction with Statistical Maximum Entropy Classifier
Data Window to Analyze
-
Forecast Xm
Real Data Xk
+
Residuals e=Xk-Xm
Fault Status
Residual Analysis

Figure 1: An overview of the fault detection approach based on machine learning models.

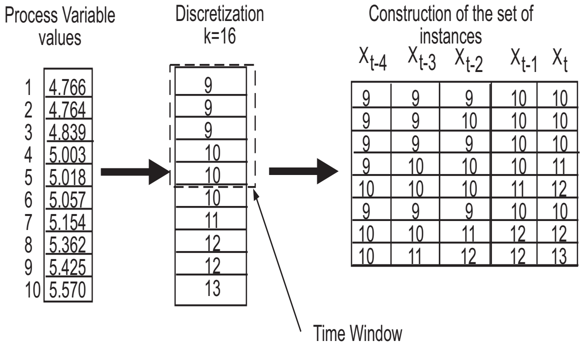
In the decision and isolation step, we generate residuals from the comparison between the output of the probabilistic model and actual process variable values. The identification of the fault is performed by a comparison of the residuals against a set of given thresholds.

The architecture of the method is split in two phases: the off-line phase and the online phase. The off-line phase learns the model structure and parameters, and the online phase take the decision regarding the presence of a fault.

2.1 The Off-line Phase

The off-line phase generates a discrete process normal behavior model from data, by applying machine learning techniques which learn both: the model structure and the parameters. The models can include several variables having an influence over the state of the process. The procedure to generate the models starts with the discretization of continuous variables, by using fixed bins or fuzzy clustering. The fixed interval width discretization, merely divides the range of observed values in equal sized bins. The general idea with multivariate discretization approach based on the *fuzzy C-means* algorithm (Wang, 1997), is that rather than discretizing independently each variable, we find the centroids of the *c* clusters defined by the user, and assign each instance of the multivariate series to the closest cluster [1].

[1] According to a defined metric. We use a simple Euclidean distance metric



	Process Variable values	Discretization k=16
1	4.766	9
2	4.764	9
3	4.839	9
4	5.003	10
5	5.018	10
6	5.057	10
7	5.154	11
8	5.362	12
9	5.425	12
10	5.570	13

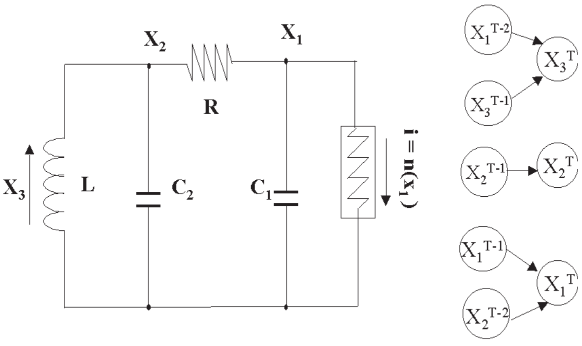
X_{t-4}	X_{t-3}	X_{t-2}	X_{t-1}	X_t
9	9	9	10	10
9	9	10	10	10
9	9	9	10	10
9	10	10	10	11
10	10	10	11	12
9	9	9	10	10
10	10	11	12	12
10	11	12	12	13

Figure 2: Selection of attributes with $M_d = 5$.

The process of discretization allows the use of standard discrete Bayesian networks learning algorithms and the implementation of the algorithm to learn the general statistical inference engine parameters.

Once the discretization phase has been achieved, the next issue in the construction of the model, is the specification of the set of attributes and the instances, to be supplied to the algorithm that learns the discrete Bayesian network structure. This is not a trivial issue, since possibly we do not know anything about the lagged dependencies in the process variable dynamics. If we have observed a sample of N data for the variable X, the forecast or prediction variable X_t may depend on any of the past values $X_{t-1}, X_{t-2}, \dots, X_{t-N}$. We solve this problem by selecting an initial set of attributes M_d [2] and keep adding attributes until a causal structure can be found. Although it is possible that different causal structures can be found, even a trivial structure with just two nodes, we can test each structure and select the more accurate. If a causal structure cannot be found with a discretization policy, then increase the number of bins, in fixed discretization policy, or increase the number of clusters, in the *fuzzy C-means* discretization policy, and again do the iterative selection of the size of attributes. An example of the selection of the attributes in a time series is shown in figure 2, with $M_d = 5$. The input to the discrete bayesian networks learning algorithm is thus a set of instances having the form $\{X_{t-M_d-1}, \dots, X_t\}$. Notice we are not assuming beforehand anything regarding independence of variables or specific time dependencies. The algorithm that learns the Bayesian network structure tries to find such dependencies.

When the causal structure of the set of M_d attributes is found, we select our model from the Markov blanket of the prediction variable X_t. The Markov blanket in a BN consists of node's parents, its children and its children's parents. The Markov blanket forms a natural feature selection, as all features outside the Markov blanket can be safely deleted from the BN. We exploit this feature to produce a much smaller causal structure for our forecast model, without compromising the classification accuracy.



(a) (b)

Figure 3: (a) Chua's electric circuit, (b) Learned graphical models from data.

The prediction variable is the M_dth attribute, has $\mathcal{P}$ parents (variables influencing directly its value) and no children (other variables over which the forecast variable have an influence). We enforce this by specifying a variable ordering to the BN learning algorithm. For instance, Figure 3 shows the models obtained for an electrical circuit which behaves as a chaotic system. X_1 represents electrical current across the inductance L and X_2 and X_3 represent voltages at capacitors C_1 and C_2.

After we obtain the relevant past values for the forecast variable, we learn the parameters of the statistical inference engine based on the maximum entropy principle. This method can be stated as follows: Consider a random feature vector $\hat{F} = (\mathbf{F}, \mathbf{C})$, $\mathbf{F} = (F_1, F_2, \dots, F_N)$, with $F_i \in \mathcal{A}_i$ and $\mathcal{A}_i$ the finite set $\{1, 2, 3, \dots, |\mathcal{A}_i|\}$, and $\mathbf{C} \in \{1, 2, \dots, K\}$. Denote the full discrete feature space by $\mathcal{G} \equiv \mathcal{A}_1 \times \mathcal{A}_2 \cdots \times \mathcal{A}_N \times \mathcal{C}$. Suppose we are given knowledge of all $(N(N-1)/2)$ pairwise pmf's $\{P[F_m, C], \forall m\}$ and wish to constrain the joint pmf $P[\mathbf{F}, \mathbf{C}]$ to agree with these. The pairwise probabilities typically are estimated from training set co-occurrence counts. The maximum entropy (ME) joint pmf consistent with these pairwise pmf's has the Gibbs form:

$$P[C = c | F = f] = \frac{exp\left(\sum_{i=1}^{N} \gamma(F_i = f_i, C = c)\right)}{\sum_{c'=1}^{K} exp\left(\sum_{i=1}^{N} \gamma(F_i = f_i, C = c')\right)} \tag{1}$$

[2] M_d is also the size of the time window, and the instances are formed sliding the time window through the complete time series. In a time series with N data we can have $N - M_d + 1$ instances

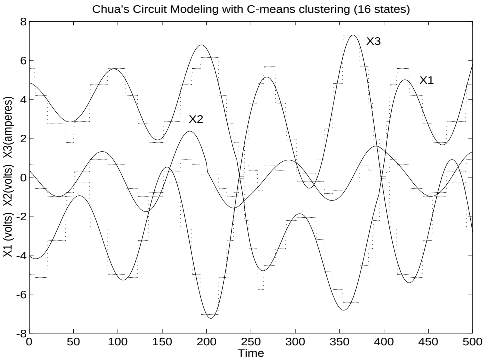


Figure 4: Modeling Chua's circut parameters with a C-Means clustering discretization method.

where

- F is the set of relevant past values for the forecast variable,
- C is the set of predicted variables.

The subset of model parameters (Lagrange multipliers) $\{\gamma(C_i = c_i, F = f), i = 1,\ldots,N, c_i = 1,\ldots,K, f = 1,\ldots,K\}$ are learned with a deterministic annealing algorithm. Where N is the number of relevant past values for the prediction variable, K is the number of discretization bins.

We need to supply following inputs to the Lagrange coefficients learning algorithm:

- A training set of $\mathcal{P}+1$ attributes with M instances,
- a training set support size $\mathcal{G}_s << \mathcal{G}$,
- an annealing parameter η,
- an annealing threshold ε,
- an annealing initial temperature T_{max} and final temperature T_{min}
- a ρ learning-rate parameter.

The inference engine provides a probability distribution of the forecast variable, given the evidence of relevant past values of forecast variable. We select the discrete state with highest probability and to make a comparison against the real data, we substitute the state by its correspondent real value. An example of modeling is shown in Figure 4.

2.2 The Online Phase

In order to perform process fault detection, the observations or measurements obtained from the process, have to be compared against the prediction given by the normal behavior model. From this comparison, the residuals are generated and then analyzed to give a decision about the behavior of the component.

If we denote X_t as the measurement of a component variable at time t, and $\hat{X}_t$ as the prediction of the component variable given by the steady state model, then the residual e_t is computed from:

$$e_t = X_t - \hat{X}_t \qquad (2)$$

The differences between the steady-state model and the real data, e_t, are transformed to a filtered version of residuals, using the equation:

$$\overline{e_t} = \overline{e}_{t-1} + \lambda * (|e_t| - \overline{e}_{t-1})$$

The value of λ, between 0 and 1, represents the smoothing factor of the residuals. We refer to the average value of a set of filtered residuals as the error weighted moving average (EWMA) index. An example of EWMA residuals behavior in Chua's electrical circuit is shown in figure 5 under normal circumstances, and in figure 6 under an additive fault.

The fault decision is accomplished by comparing the actual filtered residuals against the limit thresholds of each fault mode. The limit thresholds are calculated previously from process data. In our case, we perform intensive simulations in a power transmission network which include single faults and different combinations of multiple faults.

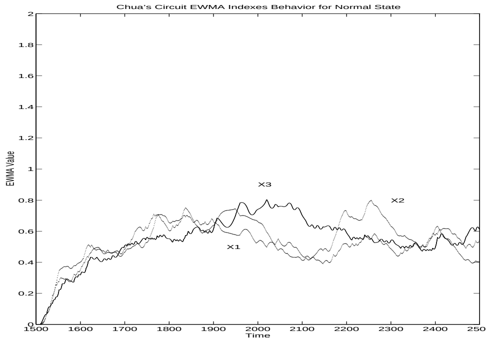


Figure 5: EWMA residuals behavior in normal operation of the three parameters in Chua's circuit.

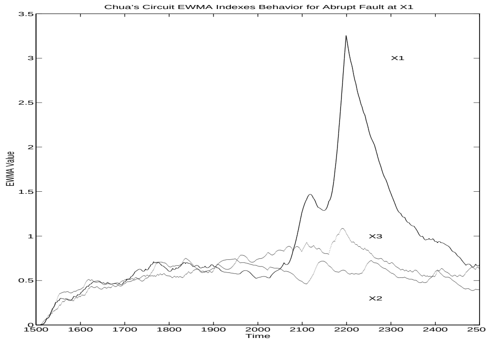


Figure 6: EWMA residuals behavior in an additive fault at X_1 in Chua's circuit.

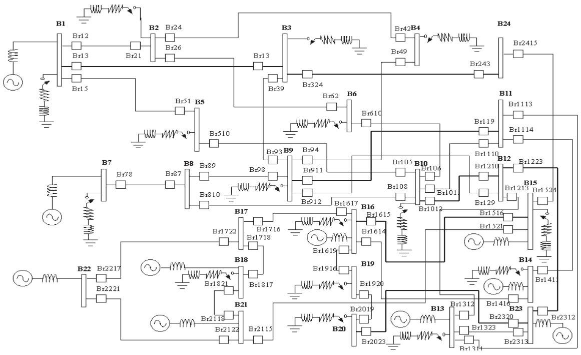


Figure 7: The electrical power network test system.

3 CASE STUDY

We illustrate the application of our approach in a simulated power transmission network, shown in fig. 7. The system consists of 24 nodes, 34 lines and 68 breakers. The electrical power network is supplied with the energy produced by three-phase generators. Ideally, the generators supply the energy to three-phase balanced loads, which means that every load has an identical impedance. In a balanced circuit, each phase has the same magnitude of voltage, but displaced 120 electrical degrees. In all simulations we include dynamic behavior by varying resistive-inductive loads in several nodes.

A fault in a electric network is any event that interfere with the normal flow of current. The faults in an electrical power network can be divided in two types: *symmetrical* faults and *unsymmetrical* faults. The symmetrical faults involve the three phases of the system, are relatively easy to evaluate, and represent about the 5 % of the fault cases. The unsymmetrical faults involve some kind of unbalance, and include line to ground faults and line to line faults. The line to ground faults represent about 70 % of the faults, and the line to line faults represent about 25 % of the cases (Grainger and Stevenson, 1994).

The diagnosis in large power networks is a difficult task, mainly due to overwhelming amount of data, the cascaded effect, and the uncertainty in the information. The main protection breakers of a node can be opened (as a secondary protection) by faults at neighbor nodes, giving rise to ambiguous diagnoses. The voltage measurements at a given node, are also perturbed by faults at neighbor nodes.

With our modeling approach, we represent the steady state dynamics of continuous signals (e.g. voltages) in every node, and detect different types of faults: symmetrical faults (e.g. a three-phase to ground fault) and unsymmetrical faults (e.g. a line-to-ground fault).

To evaluate the degree of success in the identification of the faulty components, we ran a set of 48 simulations in the power network. We randomly simulate simultaneous different types of faults in several nodes. The type of faults included *symmetrical* and *unsymmetrical* faults.

Table 1: Performance evaluation by type of fault.

Fault Type	*Correct*	*Wrong*	*% Accuracy*
A-B-C-GND	18	0	100.0
A-B-GND	12	0	100.0
A-GND	16	3	84.2
A-B	18	4	81.8
B-C	22	0	100.0
NO FAULT	20	7	74.0

The results obtained (see table 1) show that we were able to determine with great accuracy the *symmetrical* faults, but we have problems with false positive detections and line-to-line faults.

We also performed an evaluation with a level of 30 % of random missing information in the same test nodes data. The steady state models were learned with a training set of data with just 10% of random missing information. The computed EWMA indices remain almost in the same values (±2%) computed without missing information. The evaluation with missing information, delivered the same fault identification as the evaluation without missing information.

4 DISCUSSION

This approach is intended to work with data coming from multiple sources. The intention is to build, with this data, models which are robust to incomplete information and non-linearities. We have tested in some examples the capabilities of model to approximate nonlinear dynamics. The accuracy of the model, is related mainly to the level of discretization and the learning time of model's parameters. If we increase the level of discretization, we also need to increase the set support $\mathcal{G}_S$ of model's parameters learning algorithm, with the consequence of rising significatively the learning time. For instance, with 16 states and a set support size of 50 elements, learning time was 7.5 hours (using a desktop computer with a 1.3 GHz processor clock). If we increase the number of states to 32, the learning time was 12.5 hours. If we just increase the set support size for 16 states, from 50 to 80 elements, the learning time increases to 15 hours.

In summary, we do not think we have a restriction on the kind of applications we can tackle due to the accuracy of the model. All we need is a level of accuracy enough to distinguish between normal operation and every type of fault. We think that a level of discretization of at most 32 states, will cover many of the fault detection applications.

5 CONCLUSIONS AND FUTURE WORK

We have presented a new approach to detect faults based on models learned by machine learning techniques. The model represents the process normal behavior and is used in a residual generation scheme where model output is compared against actual process values. The residuals generated from this comparison are used to indicate the existence of a fault. The compact learned models are robust to noise, missing information and nonlinearities. We apply our method in a very difficult domain, as it is an electrical power network. The noise in data, the cascaded effect, and the perturbation by neighbor nodes, makes the diagnosis task hard to achieve. We have shown good levels of accuracy in the determination of the real faulted components and the mode of fault, in multiple events, multiple mode fault scenarios, where missing information was given. We determine in experimental simulations that wrong node state identifications were mainly due to the overlapping between EWMA indices thresholds, giving rise to ambiguous fault decisions. We plan to reach higher levels of success with the help of more reliable signal change detection methods.

REFERENCES

Chen, D. and Zhang, J. (2005). Time series prediction based on ensemble anfis. In *Proceedings of the fourth International Conference on Machine Learning and Cybernetics*. IEEE.

Davy, M., Desorbry, F., Gretton, A., and Doncarli, C. (2006). An online support vector machine for abnormal events detection. In *Signal Processing 86 (2006)*. Elsevier.

Frank, P. (1990). Fault diagnosis in dynamic systems unisg analytical and knowledge based redundancy a survey and new results. In *Automatica*. Elsevier.

Gentil, S., Montmain, J., and Combastel, C. (2004). Combining fdi and ai approaches within causal-model-based diagnosis. In *IEEE Transactions on Systems, Man and Cybernetics, part B*. IEEE.

Grainger, W. and Stevenson, W. (1994). *Power Systems Analysis*. McGraw-Hill, USA.

Isermann, R. (1997). On fuzzy logic applications for automatic control, supervision, and fault diagnosis. In *IEEE Transactions on Systems, Man, and Cybernetics*. IEEE.

Luque, C., Valss, J., and Isasi, P. (2007). Time series forecasting by means of evolutionary algorithms. In *Proceedings of the Parallel and Distributed Processing Symposium 2007*. IEEE.

Ma, Q., Zheng, Q., Peng, H., Zhong, T., and Xu, L. (2007). Chaotic time series prediction based on evolving recurrent neural networks. In *Proceedings of the fourth International Conference on Machine Learning and Cybernetics*. IEEE.

Matsuura, J. P. and Yoneyama, T. (2004). Learning bayesian networks for fault detection. In *International Workshop on Machine Learning for Signal Processing*. IEEE.

Sedighi, A., Haghifam, M., and Malik, O. (2005). Soft computing applications in high impedance fault detection in distribution systems. In *Electric Power Systems Research 76 (2005)*. Elsevier.

Vanajakshi, L. and Rilett, L. (2007). Support vector machine technique for the short term prediction of travel time. In *Proceedings of the 2007 Intelligent Vehicles Symposium*. IEEE.

Venkatasubramanian, V., Rengaswamy, R., k. Yin, and Kavuri, S. (2003). A review of process fault detection and diagnosis part 1, part 2 and part 3. In *Computers and Chemical Engineering*. Elsevier.

Wang, L. (1997). *A Course in Fuzzy Systems and Control*. Prentice Hall, USA.

Xu, L. and Chow, M. (2005). Power distribution systems fault case identification using logistic regression and artificial neural network. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*.

Yongli, Z., Limin, H., and Jinling, L. (2006). Bayesian networks-based approach for power systems fault diagnosis. In *IEEE Transactions on Power Delivery*. IEEE.

Zhang, P. and Ding, S. X. (2005). A simple fault detection scheme for nonlinear systems. In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*. IEEE.