International Conference on INTELLIGENT SYSTEM APPLICATION TO POWER SYSTEMS 1994,
Montpellier, France

EVALUATION OF DIFFERENT AI-METHODS FOR FAULT DIAGNOSIS IN POWER SYSTEMS

K. Bieler
ETH Zürich, Switzerland
8092 Zürich
Tel.:+411 632 41 87
e-mail: bieler@eeh.ee.ethz.ch

H. Glavitsch
ETH Zürich, Switzerland
8092 Zürich
Tel.:+411 632 41 85
e-mail: haglav@eeh.ee.ethz.ch

Abstract: Out of various methods for fault diagnosis this paper presents a selection of them, namely Expert Systems, Artificial Neural Networks expecially Multilayer Perceptron and Kohonen Feature Map, and Hidden Markov Models, which have been investigated as to their usefulness for the above purpose. An evaluation of the methods is given and numerical performance figures are presented. Advantages and disadvantages are discussed. Furthermore it is shown that the evolution of a fault can be made visible.

Keywords: Fault Diagnosis, Alarm Processing, Expert System, Artificial Neural Network, Multilayer Perceptron, Kohonen Feature Map, Hidden Markov Models

Introduction

Electric power systems including their substations are becoming more complex. On the other hand our society is demanding an adequate quality of service, i.e. a high availability of power and energy which forces a utility to take highly sophisticated measures to meet these requirements.
One of these measures is the detection and location of a fault and its subsequent clearing, as quickly as possible. Changes of circuit breaker status, excess of current limits, variation of frequency and voltage, responses of relays, etc. are already reported to a control center by standard equipment and cause fault messages and alarms. It is then up to the operator to interpret these alarms and its depends on his expertise and ability to perform an accurate diagnosis. Based on similar events as they occurred previously the operator is able to learn and to acquire knowledge, i.e. a pure heuristic process. However, what makes it difficult is the fact that messages are inconsistent and erroneous and that key parameters are lacking. On top of it, alarms appear in larger quantities in a short time (showers of alarms) and need to be interpreted on the spot. Hence such heuristic procedures cannot be maintained. For an efficient control operation a support system is needed which is to be based on fault models and systematic search procedures as they are offered by artificial intelligence methods.

Procedures for the solution of the problem

It is difficult to come up with a purely logic formulation of the alarm problem since the connection matrix is incorrect whenever there is an incorrect topology. Hence, any workable approach must include some modelling of the uncertainties. In one or the other way this is done by the use of expert systems, artificial neural networks, Markov models and other methods of pattern recognition. In the case of expert systems the experience and knowlege of numerous experts and/or cases is embedded in the knowledge base. Other methods require learning through a series of cases which relieves the operator or the expert from setting up a knowledge base.
In any case it is necessary to present or file the alarms in chronological order and to convert the messages to some compact form using a convenient alphabet. The automatic process as it will be conceived will start from a vector which contains the information of the alarms. One way of creating such a vector is by allocation a 1 (one) or 0 (zero) to a given position (coordinate) of the vector for each of the messages, creating a pattern.
Hence it is quite obvious to approach the problem by pattern recognition.

Expert Systems (ES)

Basically ES are well qualified for the diagnosis of faults in an electrical power system because there is sufficient information in a qualitative form of knowledge and data, however there is no analytical process to solve the problem of fault diagnosis. In an ES for fault diagnosis the knowledge of operators, switching engineers, relay specialists and others is stored in a knowledge base. With the help of an ES human thinking is imitated. It is not a conventional computer program but a system that makes logical conclusions with the help of a knowledge base. The knowledge is presented in the form of rules and facts in so-called rule based systems. [1] ,[2], [3], [4] show prototypes of ES for fault diagnosis and alarm processing. [5] is an application of an ES in practice.
In cooperation with ABB Switzerland the authors have acquired experiences with ES for fault diagnoses [6], [7]. ABB has developed an ES written in Prolog to analyse fault messages and the ETH has developed a generator also written in Prolog to generate alarms because some of the disturbances in the power system occur very seldom. Therefore there is a need to de-

velop an event generator to make a thorough testing of the system possible. An input to the generator is a description of the disturbance, similar to the output of the alarm interpreter.
The ES allows for the interpretation of the alarms to be performed locally, at the substation level. It aims at decentralizing the alarm interpretation from the higer level control centers and performing it at least partially at the substation level. The ES assumes the event time stamps to be accurate. Another peculiarity is the temporal aspect of the problem. Its findings are based on a set of historic and often redundant pieces of information coming form relays and breaker operations rather than on the insufficiently informative disparity in the input-output pattern of the system variables. In other words the nature of the problem relates more redundant rather than insufficient information.
The real world disturbances form three different substations have been interpreted and the system has shown very good results. But the ES has problems to analyse an alarm shower if the messages are inconsistent, in particular if there is no redundancy.

Methods including the fact that messages contain uncertainties

The paper is focussed on methods which take uncertainties into a account. Two classes are considered, namely artificial neural networks (ANN) and Hidden Markov Models where it is characteristic that the model has to be trained. This feature alleviates the user from formulating a knowledge base. On the other hand training requires some preparation of the data and the models require a structure in some of the methods. The formulation of the structure is, as it stands today, a heuristic process.

Preparation of data for training of ANN

ANN have to be trained. For that purpose a set of training patterns has to be built and each pattern consists of the alarm messages caused by a fault case.
There is a certain number of possible different fault messages in a power system. Those messages which are activated are indicated by the number 1 and the others by the number 0. A defined sequence of all possible fault messages is then attached. That means that each fault case is represented as a sequence consisting of 0 and 1. Every fault case delivers its own vector or a so-called pattern.
In practice it is difficult to have enough messages of fault cases because the electric supply companies do not store all the fault protocols or hesitate to make them public. Another problem is the availibility of the fault cases. It is not possible to have all desirable fault cases in a substation. These problems can be eliminated by a simulator.
With such a simulator [7], [6] fault messages of 80 fault cases have been generated for the present study. There are 27 possible different fault messages. Each of these 80 fault cases generates an input vector with 27 elements consisting of 0 and 1 whether the message is present or not.
Each fault case can be devided into fault types. These are named as follows: 60 of the simulated fault cases are e.g. faults of the name z1, 10 of z2 and 10 of z3, see Table 1.

Fault types	Fault names
Zones	z1, z2, z3
Poles	1, 2, 3
Earth	with, without
Locality	near, middle, end, far
Sorts	ln, lf, lh
Phases	r, s, t

Table 1: Fault types of a KFM. (ln: faults on the line removed by auto-reclosure; lf: faults on the line not removed by auto-reclosure; lh: high-impedance faults on the line)

The fault messages can be attached to the fault names of each fault type because the fault cases of these fault messages are known, respectively simulated.

Multilayer Perceptron (MP)

One of the most popular ANN is the MP. It can also be used for alarm interpretation. For that purpose the network is first trained with a set of fault patterns. Then a new pattern is mapped and allocated to an output neuron which represents a fault name. In this way a fault can be evaluated. Each fault type is represented by an own MP. However, it is also possible to treat all fault types and alarms by one single MP.
An MP was set up for alarm interpretation with the given fault types and alarms. It consisted of three layers where the first representing the input layer which contained 27 neurons corresponding to the length of the pattern. The hidden layer had 15 neurons. The number of the output neurons depends on the number of the fault names of a fault type. The fault type 'Earth' for example is divided into faults with earth contact and such without earth contact. Therefore two output neurons are enough. All other fault types require three output neurons besides the fault type 'Locality' with four output neurons. The activation function of each neuron in the hidden and output layer is the sigmoid function (formula 1):

$$f(x) = \frac{1 - e^{x}}{1 + e^{-x}} \qquad (1)$$

Each of the MP is trained with the 80 simulated fault patterns and with the corresponding output vector. The learning rate μ was set to 0.2 and the momentum α to 0.5. The number of iterations for each MP was under 300 in order to meet an error limit of 0.0001.

To analyse a fault the fault vector is mapped to each MP. Then it is possible to interpret the kind of the fault of each fault type.
It has to be realized that an MP is not able to produce error-free results as it will be shown in Tab. 2 further down which is a general finding of any pattern recognition scheme.

Kohonen Feature Map (KFM)

The MP as described above requires the setup of a certain structure, e.g. how layers and neurons are connected which is a heuristic process.
In contrast the KFM also requires training, but very little as far as a advance structure is concerned. The training is based on similarities of fault messages, i.e. the input vectors and the similarities result in a formation of clusters which are made visible in a map.
The KFM is applied if the data elements may even be related to each other in a highly nonlinear fashion. The process in which such a map can be formed is an unsupervised learning algorithm. Like other unsupervised classification methods, KFM may also be used to find clusters in the input data and to identify an unknown data vector within one of the clusters.
The map is represented graphically which is a great advantage of the KFM.

Theory of the Kohonen Feature Map

A KFM maps neighboring input vectors (vectors are neighbored, if the Euclidian distance is small) also on neighboring output neurons. That means that the relation of the neighborhood remains. The output neurons are arranged on one or two dimensional grids. A KFM consists only of an input and an output layer.

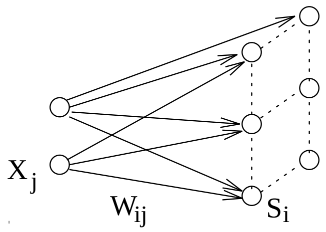

Figure 1: Model of a KFM

An input pattern activates the output neurons with the help of the weights as follows:

$$d_i = \sum_{j=1}^{N} (X_j - W_{ij})^2 \tag{2}$$

The neuron with the smallest activated value d_i represents the input vector on the grid. The unsupervised learning algorithm calibrates the weights in the way that 'similar' input vectors are mapped topologically near together on the output grid. The learning algorithm is the following:

1. choose $W_{ij}(0)$ randomly.
2. choose an input vector $X_j = X_j(t)$ randomly.
3. calculate all $d_i = \sum_{j=1}^{N}(X_j(t) - W_{ij}(t))^2$.
4. $i^* = argmin_i d_i : d_{i^*} = min_i d_i$
5. $W_{ij}(t+1) = W_{ij}(t) + \mu(t) * \Lambda_{ii^*}(t)(X_j(t) - W_{ij}(t))$
6. t $\longleftarrow$ t+1, goto 2

The neighborhood function Λ_{ii^*} is 1 for $i = i^*$ and decreases with the distance $|r_i - r_{i^*}|$ of the activation center. Moreover the learning rate $\mu(t)$ and the range $\Lambda_{ii^*}(t)$ are functions decreasing with the time t.
It is obvious that the similar output neurons can be related to certain fixed fault regions because similar input vectors and fault cases are mapped also to topologically close output neurons. A new input vector, or a new fault case, is then mapped in a certain region. Every region represents a fault case and in this way the fault can be detected.

Formation of the Kohonen Feature Map

The KFM used in this paper defines a mapping from the input data space onto a regular two-dimensional (20x20) array of nodes. With every node i a parametric reference vector d_i is associated. The lattice type of the array is defined as rectangular. The input layer consists of 27 neurons because the length of the input vector has this length.
Each of the 80 fault cases has the activation center somewhere on the grid of the 400 output neurons. With this dimensioning the fault cases are not too restrictive to map somewhere on the grid but they are forced however to decide clearly where to map the fault cases on the grid.
60 of the simulated fault cases are e.g. faults of the name z1, 10 of z2 and 10 of z3. Each fault is interpreted with the maps. Therefore 18 maps have to be built representing each of them by a fault name. Every map is to be built in that way that the regions correspond to the fault names.
The fault messages can be attached to the fault names of each fault type because the fault cases of these fault messages are known, respectively can be simulated. The regions which represent the fault names of a fault type can be made visible. For that purpose all of the 80 fault cases which belong to the same fault name are mapped and all output values are added. In this way regions with similar output values are formed. These regions represent the fault names. Figure 2 shows this mechanism forming the maps for faults with, and without earth contact respectively.

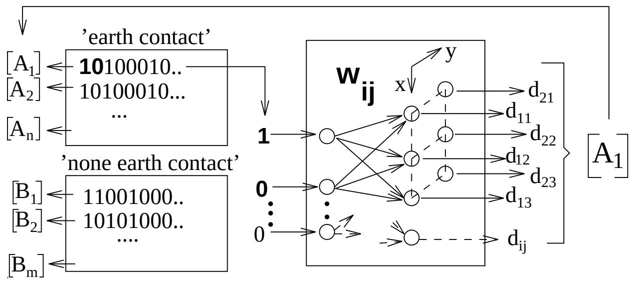

Figure 2: Forming of a KFM

In the study faults have been separated into two groups for which corresponding files have been set up.

One of them consists of faults with earth contact and the other one of faults without earth contact. Each pattern of the file with e.g. earth contact faults is mapped by using the trained weights whereby each pattern stimulates all output neurons. This means that for each ouput neuron the dissimilarities between the input vector and the weights are calculated (formula 2). Thus each pattern forms a matrix A. The first pattern forms the matrix A_1, the last one the matrix A_n. All matrices which are built from the patterns in the file with earth contact are added up.

$$[A_{with}] = [A_1] + [A_2] + \ldots + [A_n] \tag{3}$$

Likewise the matrices for faults without earth contact are built:

$$[B_{without}] = [B_1] + [B_2] + \ldots + [B_m] \tag{4}$$

These matrices can be made visible by allocating dark colors for low matrix values and light colors for high matrix values. In this way the matrix values are arranged rectangularly. The scales of the following pictures show the index values of the rows and columns of the matrix.

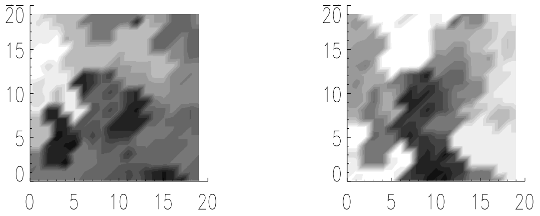

Figure 3: Maps of the fault type **earth**. The map which represents faults with earth contact (A_{with}) is on the left side. The other map represents faults without earth contact ($A_{without}$)

In this way the maps with and without earth contact are built (see figure 3). The map on the left side the dark region represents the faults with earth contact and the light region all other faults. In the second map the dark region represents the faults without earth contact and the light region all other faults. Unfortunately it may happen that some trained fault cases are mapped in the false region. By subtracting matrix $[B_{without}]$ form matrix $[A_{with}]$ the matrix $[C_{with}]$ is built which represent faults with earth contact and the subtraction of the matrix $[A_{with}]$ from the matrix $[B_{without}]$ is built which represent faults without earth contact (see figure 4).

$$[C_{\mathbf{with}}] = [A_{\mathbf{with}}] - [B_{without}] \tag{5}$$

$$[C_{\mathbf{without}}] = [B_{\mathbf{without}}] - [A_{with}] \tag{6}$$

Regions representing faults with earth contact have small matrix values and faults without earth contact

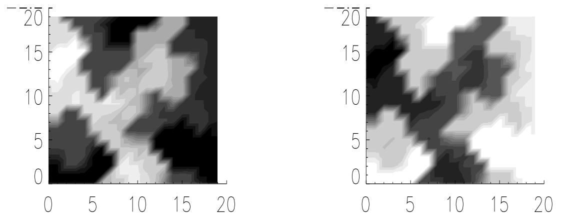

Figure 4: Exact maps of the fault type **earth**. The map which represents faults with earth contact (C_{with}) is on the left side. The other map represents faults without earth contact ($C_{without}$)

have large ones. By a 3-dimensional presentation of the matrix $[C_{with}]$ (Figure 5) a landscape with hills and valleys is formed. Faults with earth contact are in the valleys and the other ones on the hills. All of the 80 simulated fault cases are mapped correctly. Such maps are formed for all fault types.

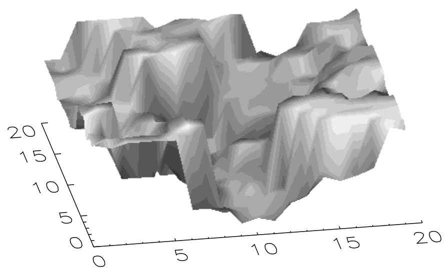

Figure 5: Hill landscape of a KFM

The maps of fault types with more than two properties e.g. the fault type **zone** contains the properties 'faults in the first zone (z_1)', 'faults in the second zone (z_2)' and 'faults in the third zone (z_3)' are formed in the following way. First the matrices A_{z1},A_{z2} and A_{z3} are built as it is shown in formula 3 and 4. These three matrices can be represented as figure 3. The regions can be then produced more exactly by the following procedure (comparable with formula 7 and 6).

$$\begin{aligned} C_{\mathbf{z1exact}} &= (A_{\mathbf{z1}} - A_{z2}) + (A_{\mathbf{z1}} - A_{z3}) \\ C_{\mathbf{z2exact}} &= (A_{\mathbf{z2}} - A_{z1}) + (A_{\mathbf{z2}} - A_{z3}) \\ C_{\mathbf{z3exact}} &= (A_{\mathbf{z3}} - A_{z2}) + (A_{\mathbf{z3}} - A_{z1}) \end{aligned} \tag{7}$$

In this way maps for all properties of every fault type can be built.

Interpretation of the maps

A fault case, respectively a fault pattern, is mapped and all dissimilarities between the input pattern and the weights (formula 2) for all output neurons are built. The neuron with the smallest dissimilarity value

represents then the fault. The x and y coordinates of this neuron is drawn in every map of every fault name. The region in which the fault is drawn in represents the fault name. If the fault lies in a valley as in the example above, it is a fault with an earth contact. The deeper or higher the place of the fault the more certain the declaration of the fault type is. Faults lying in an intermediate layer however cannot be classified. Thus a boundary layer which delimits the regions is made. Fault places above this layer are e.g. faults without earth contact and faults below of the layer are faults with earth contact.
It is to be realized that the interpretation of such maps is not straightforward. The setting of boundary layers and blending regions which identify the undefinable places require some experience. However, the continued exposure to this problem and some knowledge about the faulted system will allow to acquire the necessary expertise.
A numerical interpretation is given in Tab. 2.

Evolution of faults

With the help of the described method the evolution of a fault case can be made visible. For that purpose the first of the alarm messages is mapped. That means the input vector, i.e. the fault pattern consists only of one active message. The other 26 bits of the input vector are zero. This simple pattern is mapped somewhere on the map but in blending regions because it is not possible to make a secure statement of the fault with one fault message. Then the next message is added. The input vector consists now of two active bits and is also mapped. This pattern can be mapped at a different point than the first one. The process is repeated till all messages have been considered. A point-to-point connection is formed when all mapped patterns are joined. First the faults are in blended regions but then they propagate to dark or light ones. Figure 6 shows the evolution of a fault in the third zone.

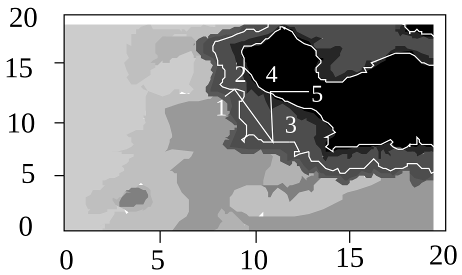

Figure 6: Evolution of a fault

The map is divided into three regions. The light one represents faults in the first zone (z1), the dark region faults in the third zone (z3) and the remaining area represents faults in the second zone (z2). The fault development begins at the boundary between z1 and z2 in an uncertain region and propagates then along this boundary and remains in this uncertain region. The fault then jumps to the boundary between z2 and z3. If all information is there the fault jumps into the third zone. The fault place is in a definite region because there is enough information to permit a confident explanation of the fault.

Hidden Markov Models (HMM)

A great disadvantage of the KFM is the defined length of the input vector. If there is a fault not all the possible 27 fault messages appear. To circumvent this shortcoming HMM is a method to process time variable messages effectively. The method is used successfully for speech recognition.

Theory of the HMM

The setup of the model is based on some knowledge about the process. Only a sequence of output symbols $O = O_1 O_2 ... O_T$ is observed but there is no knowledge how this sequence is built. The task is to find a model which imitates the process as closely as possible. The model should imitate the sequence of the output values with a probability which is as high as possible. First the topology of the model has to be fixed with states and transitions connecting the states. Every state can produce all of the 27 possible messages the so-called output alphabet V_k with a certain possibility. At any time step the state changes with a certain probability to another one. A HMM is characterised as follows:

1. N: the number of states
2. M: the number of the output alphabet V_k
3. A=$\{a_{ij}\}$: the transition probabilities
4. B=$\{b_i(k)\}$: the output probabilities
5. π_{ij}: the initial state probabilities

Often a HMM is given in compact form $\lambda = (A, B, \pi)$. After fixing the states and transitions the model has to be verified that the sequence produced by this model has a high probability.
There are three algorithms associated with HMM:

- The forward algorithm computes the probability that a HMM generates a given output sequence $O = O_1 O_2 ... O_T$. The probability $P(O|\lambda)$ is the probability that the output sequence O is generated along the state sequence $Q = q_1 q_2 ... q_T$. Such a state sequence begins with the initial state and ends with the final state.

  $P(O|\lambda)$ is calculated with the forward variable $\alpha_t(i) = P(O_1 O_2 .. O_T | q_t = S_i | \lambda)$ as follow:

  - $\alpha_1(i) = \pi_i b_i(O_1); 1 < i < N$
  - $\alpha_{t+1}(j) = (\sum_{i=1}^{N} \alpha_t(i) a_{ij}) b_i(O_{t+1});$ $1 < t < T - 1$
  - $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

- The Baum-Welch-algorithm for optimizing the parameter set a_{ij} and $b_i(k)$ of training sequences. There is as yet no analytical procedure to re-estimate the model parameters A and B. An iterative method is employed to choose λ such that $P(O|\lambda)$ is locally maximized. The re-estimated model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ is better than or equal to the previous model λ in the sense that $P(O|\overline{\lambda}) \succeq P(O|\lambda)$. The value $\overline{a_{ij}}$ is the ratio of the expected number of transitions from state q_i to state q_j to the expected number of transitions from state q_i. $\overline{b_i(k)}$ is the ratio of the expected number of times in state j and observing symbol V_k to the expected number of times in state j.

- The Viterbi algorithm calculates the most probable state sequence of a HMM given as output sequence $O = O_1 O_2 ... O_T$. This algorithm is not used in this paper.

A very good introduction to HMM is given by [8]

Preparation of the data

The number of fault messages causing by a fault case may vary and hence HMM can process fault patterns with variable length as opposed to KFM. The messages are processed chronologically. Every possible message is coded as a number. The message indicating the response of the phase R is e.g. coded as the number 1, the response of the phase S as the number 2, the trip signal of the circuit breaker as number 10 and so on. The alarm messages of a fault case can be presented as a sequence of numbers, so-called vector codes such as **1 2 5 10 4 2 21**.

Taking again the study case treated above one can proceed as follows. The 27 possible messages form the output alphabet V_k. The 80 fault patterns resulting from the 80 simulated fault cases are coded and are allocated to the fault types, or fault names as for the KFM. Thus the necessary training rules are generated.

Implementation

The goal to form an HMM is the imitation of an observation with the best possible model. If the length of the output sequence is defined every output symbol can be allocated to a state (this is the so-called classical Markov model). However to imitate a variable output sequence with one model it is not possible to allocate every symbol to a state. The states become hidden. Such a model has to produce all output sequences with a variable length. If a model represents all phase-R-fault cases each fault pattern has a different number of fault messages depending on the fault location for example. But all alarm messages are a temporal process as conceived in speech recognition that means they start at a point and end at a final one. Therefore left-right-models (Bakis models) are used. There is only an initial state in which every process starts with the initial probability $\pi = 1$, several intermediate states and a final state where the process ends. The states are connected with non-recurrent loops except in the states themselves.

In the study different models have been checked-out, e.g. Bakis-, Sphinx model [10]. Thereby best results have been obtained with a Bakis model with 5 states (Figure 7).

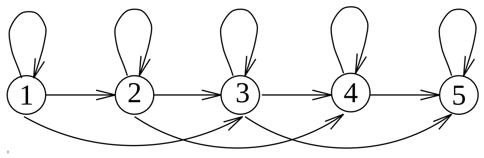

Figure 7: Bakis-Model

Each name of a fault is represented by a HMM. The topology of all of them is the same. The transition probability matrix A and the output probability matrix B is formed by training.

The training process is the following: Every name of a fault is represented with a topologically equal HMM. The transition and output probabilities A and B are trained such that all patterns belonging to the same name of a fault are produced with as high a probability as possible.

Interpretation of a fault

There are six types of faults (Tab.1) for analysis. Every fault type is divided into fault names. Every fault name is represented by a HMM. A new fault case or the new fault pattern is treated as follows: The fault pattern is run through each HMM of a fault type. Each HMM produces an output probability. The HMM with the largest probability value represents then the fault. Figure 8 shows for example the analysis of the fault type 'zones'. There are three possiblities where the fault can be. Either in the first (z1) or in the second (z2) or in the third (z3) zone. The output probabilities of each HMM are calculated namely $P(O|\lambda_{z_1})$, $P(O|\lambda_{z_2})$ and $P(O|\lambda_{z_3})$. The model with the highest probability imitates the pattern sequence of the fault case better than the others. The fault name represented by the HMM with the highest output probability is the most probable one belonging to the fault case. The other 5 fault types are treated equally.

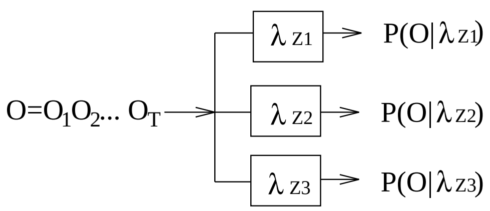

Figure 8: Interpretation of the zones

Results and comparisons of the methods

It is difficult to compare the methods directly because they are different by nature. Whereas in an ES the knowledge of experts is written down as rules the other methods do not need this knowledge. They learn by themselves but require some expertise in interpreting the results.

In order to be able to compare results of the methods studied numerically 825 test cases have been generated. They are derived from 80 training cases with varying patterns. New messages have been added and old one have been removed. This is quite realistic since in reality messages may have errors, missing data, etc. The fault cases have been tested with the given test data base. 492 patterns of 825 test cases are faults with an earth contact and 333 without an earth contact. The numerical result are listed in Tab.2.

The recognition rate of the KFM and HMM is very high where the KFM is even somewhat superior. Some caution has to be exercised in comparing the methods directly because the topology of all the HMM is the same and the messages are handled temporally. The input vectors of the KFM however have a fixed length and are arranged chronologically.

fault-type	fault-name	number of cases	% correct KFM	HMM	MP
earth	with	492	98	97	90
	without	333	98	87	100
sorts	ln	318	99	76	99
	lf	411	100	89	97
	lh	96	100	100	100
poles	1	213	97	100	99
	2	522	94	89	77
	3	90	80	80	63
phases	r	275	91	88	76
	s	275	94	88	77
	t	275	96	89	77
locality	n	243	100	89	99
	m	243	98	99	99
	e	243	100	86	100
	f	96	100	100	100
zones	z1	729	100	100	100
	z2	43	100	90	100
	z3	53	100	100	100

Table 2: Testresults

Advantages and disadvantages of the methods

There is a large amount of expert experience and knowledge in an ES. Rules can easily be changed or added if required. It is easy to integrate new knowledge by implementing new rules. Changes of the topology of a substation can be treated by adding new facts without changing the rules. The experience acquired in the research of ES for fault diagnosis is quite rich and there are already ES in operation, however ES show also a great disadvantage. New or inconsistent alarm messages do not fit an existing ES. If there are no rules to treat such messages the ES fails, however, this judgement must not be taken in that stringent manner. An ES may be complemented by stochastic features such that it may generate answers with probabilities attached.

With the KFM it is possible to make a fault or the evolution of a fault visible. Each new and inconsistent fault is formed to the region of the map where the most similar trained fault cases are located. Dependent on the position where the fault is mapped the explanation of the fault is certain or uncertain. If the database is large enough it is easy to create a well clustered map. KFM are independent of the knowledge and experience of the experts. The treatment of a fault is very fast. But the KFM needs an input vector with a fixed length and the implementation of time dependent messages is difficult to realize. Moreover, changes of the topology are difficult to handle because new KFM have to be trained again with a new database.

With HMM time dependent messages can easily be processed. However it is difficult to find a good topology of a HMM because the number of the fault messages of a fault name is variable. A further advantage is that a HMM always produces an answer. The fault case can be a new one or a very inconsistent one. HMM are not dependent on the knowledge of experts. However changes of the topology of a substation entail a lot of work because the HMM have to be trained again with a new set of fault cases.

Conclusion

The problem of fault diagnosis has been treated from various directions. Expert systems have been mentioned, however, their performance is widely known and their characteristics could be included in the final evaluation. The main emphasis of the paper was on methods which allowed the inclusion of uncertainies, i.e. incorrect messages, messages of variable lengths, etc. Thereby the Kohonen Feature Map and Hidden Markov Models were in the foreground. They are particularly suitable for handling inconsistent and new messages. The great advantage of HMM is the possibility of handling the alarm messages chronologically. KFM and HMM evaluate data from single lines, transformers, current breakers, etc. because their topologies do change rarely. Both methods process all fault messages that means historical, new and also inconsistant ones. In a way it can be said that KFM and HMM understand alarm messages.

On the other hand ES are best suited for substations or control where the topology may change. In a hybrid manner an ES may process the output of KFM and HMM.

The numerical results show quite high recognition rates which seem practical for an implementation in substations. At this point, as it must be realized, these tools constitute some aids which have to be developed further.

Acknowledgment

The authors wish to thank Ulrike Glavitsch, Informationssysteme ETH-Zürich, Switzerland, for providing software for the HMM.

References

[1] Bruce F. Wollenberg, Toshiaki Sakaguchi
Artificial Intelligence in Power System Operations
Proceedings of the IEEE, December 1987
Vol.75, No. 12

[2] C. Fukui, J. Kawakami
An Expert System For Fault Section Estimation Using Information From Protective Relays And Circuit Breakers
IEEE Transactions On Power Delivery
Vol. PWRD-1, No.1 pp. 83-90,

[3] M. Munneke, T.S. Dillon
Implementation of an Alarm Processing Expert System in a Regional Control Centre
Proceedings of the Tenth Power Systems Computation Conference, 1990

[4] M.Stalder, A. Germond
An Expert System in Prolog for Alarm Handling in a Substation
Proceedings of the Ninth Power Systems Computation Conference
pp. 973-978, September 1987

[5] P. Fauquembergue, J.V. Esquevin
An Expert System as an On-Line Aiding Tool for The Maintenance of Protection Systems
CIGRE, Proceedings of Study Committe 34, Colloquium and Meeting
21-26/09/1991, Foz do Iguaçu, Brazil

[6] V. Milŏsević
SALI: a Substation ALarms Interpreter
Proceedings of The First International Conference on the Practical Application of Prolog, London, April 1992

[7] K. Bieler, H. Glavitsch, V. Milŏsević
SAIS: A Substation Alarm Interpreter and Simulator
ABB Technical Report, Document Ref. No.: ABB CHCRC 93-05, Baden-Dättwil, Switzerland, 1992

[8] L.R. Rabiner
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition
Proceedings of the IEEE
Vol. 77, No. 2, February 1989

[9] T. Kohonen
Self-Organization and Associative Memory
Springer Verlag,
Berlin-Heidelberg-NewYork-Tokio
3 edition, 1989

[10] K.F. Lee
Automatic Speech Recognition: The Development of the SPHINX System
Kluwer Academic Publishers, Boston,1989