

Learning Models of Plant Behavior for Anomaly Detection and Condition Monitoring

A. J. Brown, V. M. Catterson, M. Fox, D. Long, and S. D. J. McArthur, *Senior Member, IEEE*

Abstract—Providing engineers and asset managers with a tool which can diagnose faults within transformers can greatly assist decision making on such issues as maintenance, performance and safety. However, the onus has always been on personnel to accurately decide how serious a problem is and how urgently maintenance is required. In dealing with the large volumes of data involved, it is possible that faults may not be noticed until serious damage has occurred.

This paper proposes the integration of a newly developed anomaly detection technique with an existing diagnosis system. By learning a Hidden Markov Model of healthy transformer behavior, unexpected operation, such as when a fault develops, can be flagged for attention. Faults can then be diagnosed using the existing system and maintenance scheduled as required, all at a much earlier stage than would previously have been possible.

Index Terms—Cooperative systems, Decision support systems, Hidden Markov models, Intelligent systems, Learning systems, Monitoring, Partial discharges, Power systems, Power transformers.

I. INTRODUCTION

Condition monitoring of electrical plant equipment is important for safely prolonging the life of costly assets such as power transformers. However, many condition monitoring systems produce too much data for engineers to view and assess, leading to useful indicators of health being overlooked. This could be solved with a condition monitoring architecture capable of anomaly detection, diagnosis, and prognosis, extracting as much information as possible from condition data.

Multi-agent systems (MAS) technology offers a way of creating such an architecture, capable of linking disparate sensors and analysis techniques in a flexible, extensible way. Agents have previously been used for power engineering applications such as feeder automation [1], substation automation [2], and microgrid control [3].

Previous work resulted in the COndition Monitoring Multi-Agent System (COMMAS) [4], a system which uses classifiers trained on a set of known defects to identify partial discharge causing faults. It was developed as a multi-agent system, in part to facilitate future extensibility, and its diagnostic capabilities would be enhanced by the inclusion of anomaly detection and prognostic ability.

Different transformers can display very different behavior while still being classed as healthy. As an example, some transformers may display low levels of partial discharge which are of no concern. A disadvantage of the classification approach is that a system which has been trained to recognize defects will

be unable to differentiate between these differing behaviors and so may provide misleading results.

If normal operating behavior were learned on a per-transformer basis, then anomalous behavior of *individual* transformers could be detected and flagged. This would provide the first stage in diagnosing a previously unseen fault and would also provide an immediate form of monitoring, in anomaly detection, when retrofitting equipment. Classification attempts could follow based on the assumption that a defect is then present.

Several authors have described methods for learning abstract behavior classifications from sensed data. Littman et Al. [5] showed how to learn human behavior classifications on the basis of accelerometer data. Koenig et al [6] and Fox et al. [7] showed how robot behaviors can be learned using the robots ability to sense the structure of its environment.

The remainder of this paper is structured as follows: Section II describes an approach to learning normal behavior of a transformer, based on the work described in [7], and how this model of behavior can be used for detecting anomalies. Section 3 follows with a description of the previous operation of COMMAS, and how anomaly detection can integrate with this functionality. Section IV details an experiment showing how this works in practice, using data and a model from an in-service transformer. Finally, Section V explains how prognostic abilities can follow from the learned model of transformer behavior, and how future work will seek to integrate this within the condition monitoring system.

II. LEARNING A BEHAVIORAL MODEL

The first step towards detecting anomalous behavior is to understand normal operation for a particular transformer. This can be done by learning a model, using data collected from the plant in question for training. The data used for this system is partial discharge data.

A. Partial Discharge Data

A partial discharge (PD) is caused by an electric field surrounding a conductor exceeding the dielectric strength of the conductor's insulation. This results in an electrical discharge that does not fully bridge the gap between conductors.

Defects causing partial discharge can be introduced during manufacture or may be the result of degradation over time. In the latter case, normal aging of a transformer may cause a low level of discharge activity, which may occur regularly over a number of years. This would not be classified as defective or abnormal behavior until the pattern of activity alters and

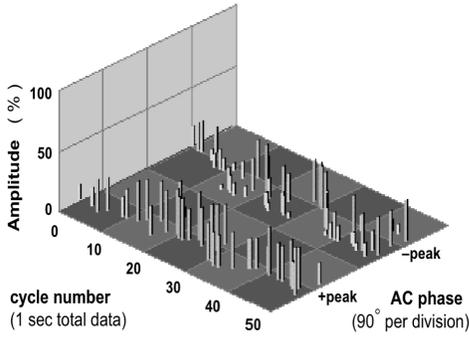


Fig. 1. The 3-dimensional phase-resolved graph

it reaches a damaging level. However, six broad classes of partial discharge-causing defect have been identified: bad contacts, floating components, protrusions, rolling particles, surface discharges, and suspended particles [8].

A partial discharge radiates EM energy in the ultra-high frequency (UHF) range, meaning PD activity can be detected by mounting UHF sensors on the transformer. Graphing each discharge in terms of voltage cycle number and phase position of occurrence gives a *phase-resolved plot*, generally showing one second of discharge activity (see Figure 1).

The particular monitoring hardware being used for this work records phase-resolved discharge data to files. One file contains a single day's activity as separate *events*. Discharges within a second of each other are counted as the same event, while a period of no activity between two discharges would cause two separate events to be recorded. A day's file may contain hundreds of events, such as that shown in Figure 1.

B. The Automatic Construction of a Behavioral Model

In any physical system there is a gap between its actual operation in the physical world and its behavioral state as perceived through its sensors. The consequence is that, in reality, the system moves through states that are *hidden* from direct perception. Furthermore, because of the inability of the system to accurately perceive its state, and the uncertainty in the physical world, the transitions between these hidden states are probabilistic and there is a probabilistic association between states and observations.

A convenient target representation for learned behavioral models, based on these physical systems, is the Hidden Markov Model (HMM) [9], the structure of which is defined below.

A stochastic state transition model is a 5-tuple, $\lambda = (\Psi, \xi, \pi, \delta, \theta)$, with:

- $\Psi = \{s_1, s_2, \dots, s_n\}$, a finite set of *states*;
- $\xi = \{e_1, e_2, \dots, e_m\}$, a finite set of *evidence items*;
- $\pi : \Psi \rightarrow [0, 1]$, the prior probability distributions over Ψ ;
- $\delta : \Psi^2 \rightarrow [0, 1]$, the transition model of λ such that $\delta_{i,j} = \text{Prob}[q_{t+1} = s_j | q_t = s_i]$ is the probability of transitioning from state s_i to state s_j at time t (q_t is the actual state at time t);
- $\theta : \Psi \times \xi \rightarrow [0, 1]$, the sensor model of λ such that $\theta_{i,k} = \text{Prob}[e_k | s_i]$ is the probability of seeing evidence e_k in state s_i .

A fundamental step in constructing a HMM is to determine the set of observations and states that describe the causal structure of the system. A key novelty in the work of Fox et al [7], is the construction of these sets through unsupervised learning. First, feature vectors are obtained from the raw sensor data. These are then clustered into an observation space using a clustering technique based on the well-known Self-Organizing Map principle (SOM) [10]. The clustering process discretizes the data into a finite set of observations to which a second clustering stage is applied. This uses vector scalar product as a similarity metric and simply groups together observations with a scalar product below a specified threshold.

The choice of feature vectors is an important step. The work described here uses the phase-resolved UHF-signal amplitude measurements, as shown in Figure 1. Phase-resolved plots are divided into 6 segments, coinciding with the rising positive segment of the voltage cycle, the peak positive segment, the falling positive segment and then the same 3 segments for the negative half-cycle. Calculations are made of the mean, standard deviation and kurtosis for the signal in each of these segments. Data is not collected continuously, but in bursts following triggering discharges, so the density of phase-resolved plots varies in time according to the amount of PD activity. The relative density is reflected by also creating features representing the numbers of events in an interval to the left and to the right of each phase-resolved plot.

Having identified the observation and state sets, the final stage in the process is the re-estimation of the probability distributions of the HMM by applying Expectation Maximization (EM) [11], [12]. Learning is based on observation histories, presented as sequences. In the experiment presented in this paper individual days of data taken from a healthy transformer are individually presented to the EM learning process.

The EM process is seeded with an initial transition model in which it is assumed that there is an equal probability of the system transitioning from any state into any state accessible from it. This initial next state probability distribution is improved by re-estimation. Figure 2 shows an example of a model learned from data generated by a healthy transformer, the EM process having converged. In the figure it can be seen that there are two states which are darker colored: these are artificial states representing the start and end of any trajectory.

C. Detecting Anomalies

The identification of anomalies depends on characterizing the pattern of typical trajectories through the learned model and comparing the trajectories of new data sets with those characteristics. Trajectories of data can be mapped to the most likely sequence of states visited in the learned HMM using the Viterbi algorithm [13]. This algorithm identifies both the most likely sequence of states and the associated probability that the model assigns to the sequence as an explanation of the input observations. This probability becomes very small as sequences grow longer (simply because the space of possible sequences grows exponentially as the sequence length grows). Therefore, the probability is usually mentioned using the log of the actual value. This value is referred to as the *log-likelihood*.

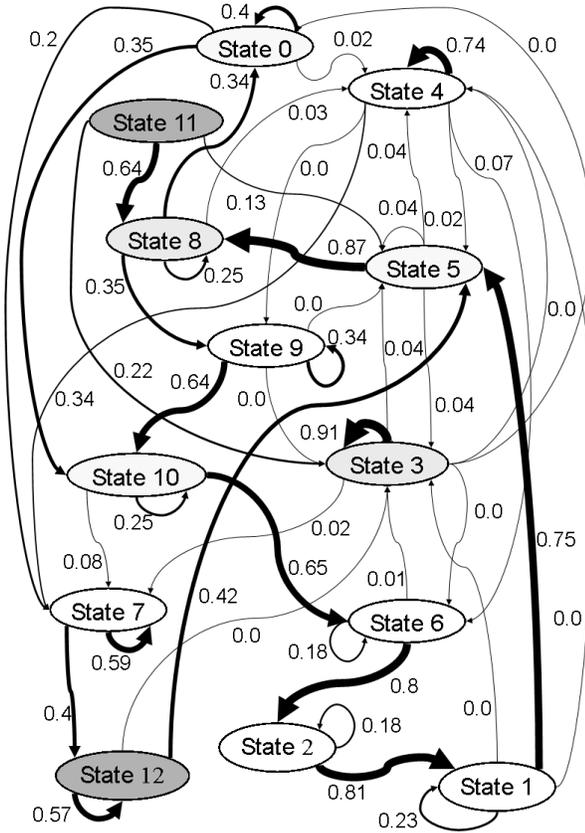


Fig. 2. A transition model learned from the training data

Work performed by Gough [14] has demonstrated that an important characteristic of typical trajectories is the trend in log-likelihood of the evolving sequence. Here, this is refined, and the mean gradient of the decreasing log-likelihood is focused upon. Specifically, anomalous behavior usually results in the Viterbi algorithm finding unusual state sequences to explain the atypical observation sequences. These often have atypical gradients in their log-likelihood curves. The reason for this is that either unlikely transitions are required to explain unusual sequences of otherwise likely observations, or very unlikely observations are made which lead the Viterbi algorithm to identify the most likely self-transitions on single states as good explanations (see state 3 in Figure 2). In the former case, the trajectories tend to have faster decreasing log-likelihood than normal, while in the latter case they tend to have slower decreasing log-likelihood.

Therefore incremental changes in log-likelihood for the new data are compared with the mean change in the learned data. Alerts are raised when these values deviate significantly from the mean (the sensitivity can be adjusted), but full alarms are triggered when a sequence of n consecutive observations includes at least k that lie on the same side of the mean. The choice of n and k is made in order to set sensitivity. Most interestingly, it is possible to report the points in time at which the significant changes in gradient occur, because the observations are associated with specific time points.

III. CONDITION MONITORING MULTI-AGENT SYSTEM

The original set of diagnostic modules, collectively called the Condition Monitoring Multi-Agent System (COMMAS), focused on classifying the type of defect causing partial discharge activity. It operated on the assumption that a defect was present, with an engineer having performed the task of anomaly detection by identifying transformers with potential problems.

The system was designed as an agent system, to take advantage of benefits such as flexible communications between agents and system extensibility, as it was anticipated that the system's functionality would increase over time. Any new tasks, such as a new method of data interpretation, could easily integrate into the system by developing it as an agent which uses the Directory Facilitator's standard service location facilities [15] to find other appropriate agents. This is important for a condition monitoring system, which may be required to handle various types of data across different types of plant and still produce reliable information.

A. The Diagnostic Agents

Originally, the COMMAS diagnostic agents would pick up and process all recorded partial discharge data. This was split into four stages: data monitoring, interpretation, corroboration, and information, shown in Figure 3.

The data monitoring stage comprised two separate agents: a Data Formatting Agent and a Feature Extraction Agent. The first detects updates to the current day's file of PD activity, and reads the latest data from it. The Data Formatting Agent then passes this to the Feature Extraction Agent, which calculates 101 features of the data. These include basic, deduced, and statistical features which have been shown to relate to the type of the causing defect [16].

Feature Vectors are then passed to the interpretation agents for classification. Three different classifiers are used: K-Means clustering, C5.0 rule induction, and a back propagation neural network. Each of the classifiers produces a defect type and probability, as well as historical information about their accuracies. All this information is collected by the Diagnosis Agent, which employs a Bayesian Belief Network to combine all the evidence and determine the most likely existing defect.

Agents within multi-agent systems can be distributed across various locations. This capability is exploited to keep the majority of data processing close to the data source. Thus the data monitoring, interpretation and corroboration stages described above are located on-site at the substation. Only transformer information of interest to the user need enter the communication network for transmission to the display-handling Engineering Assistant Agent located off-site.

B. Combining Anomaly Detection and Diagnostics

The integration of an anomaly detection system with COMMAS would make the system much more practical and beneficial. This addition would mean that operation would continue until an unusual behavior was detected in a transformer being monitored. Assuming that this change was caused by

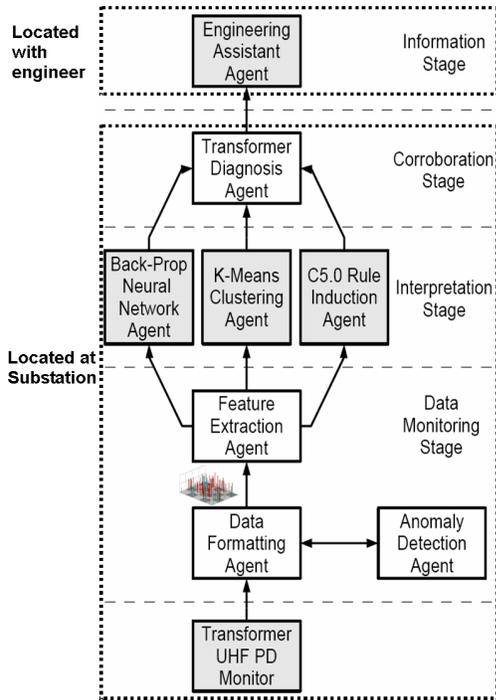


Fig. 3. The Agent Architecture of COMMAS with Anomaly Detection

a previously unseen defect within the transformer, COMMAS could then attempt to classify the problem.

The MAS architecture on which COMMAS was designed makes the combination of the two separate systems straightforward. Wrapping the anomaly detection system within an agent enables communication between this and the existing COMMAS framework. This new agent shall henceforth be referred to as the Anomaly Detection Agent (see Figure 3).

When anomalies are observed in a transformer's behavior, the detection software generates a time-stamp. This indicates the point at which the change in behavior was first identified. When such a change is recognized a message, containing the time-stamp, will be sent from the Anomaly Detection Agent to the Data Formatting Agent, informing it to begin sending data to COMMAS for diagnosis of the fault which may now exist. Upon receiving this trigger message, the Data Formatting Agent will first search through previously examined raw data files to find those which represent events occurring after the time-stamp. These will be formatted and the data sent in turn to the Feature Extraction Agent. From this point on COMMAS will function as before, attempting to identify the defect which has caused each of these files to be produced. Future partial discharge data can be sent directly to COMMAS for diagnosis under the assumption that a fault now exists within the transformer.

This adapted system makes the supervision of transformer operation much easier for engineers. Previously unseen faults are generally detected by the detailed study of formats of the raw sensor data, for instance the phase-resolved format shown in Figure 1. This requires many hours of analysis for even a short period of a transformer's activity and, with numerous transformers to monitor, defects can exist for a prolonged

period of time before being discovered. Hence, serious damage can occur before maintenance takes place, previously having resulted in catastrophic failure. This new approach will alert those concerned to changes within monitored transformers in a timely fashion. A warning message can be sent to the EAA when anomalies are discovered flagging that the behavior of the relevant transformer has changed from its normal. The detailed diagnosis of COMMAS would then follow once the relevant data files had been analyzed.

This rapid detection of changes in behavior will allow problems to be identified quickly and necessary maintenance to be carried out at a much earlier stage reducing the risk of severe damage to equipment. The input of the diagnosis from COMMAS will help to decide how immediate the maintenance should be and where it should be targeted, making the procedure more efficient. The anomaly detection also means that transformers that are deemed to be behaving normally can be allowed to continue operating without periodic maintenance which can often be unnecessary.

IV. TESTING THE ANOMALY DETECTION

The goal of this experiment was to show that a learned HMM can be used to detect anomalies in a system on which it has been trained.

A. Data

UHF data was captured from a transformer displaying low levels of partial discharge, deemed non-damaging by experts, and used to learn a model of healthy behavior.

Two known defects were created in the laboratory: a bad contact between a loose nut and bolt (labeled BC), and a metallic particle rolling across the surface of insulation (labeled RP). A voltage was applied to each defect in turn until they started discharging, and UHF data was captured. This created two sets of known defective partial discharge data: one set from the BC defect, and one from the RP defect.

To create data for testing the anomaly detection, discharge events from the known defective data were inserted into 14 previously unseen files of good behavior captured from the transformer. Runs of 200–250 discharge events from each of the defects were inserted into each day's data, creating a test set of 28 files (14 with the BC defect and 14 with the RP defect). The initiation of the defective events varied from the 200th event in the file to the 690th event.

B. Experiment

The experiment entailed presenting the runs of events containing serious discharges to the Viterbi Algorithm, given the learned model of healthy behavior. Anomalies would be detected if the log-likelihood of a sequence deviated significantly from those of the sequences of normal behavior. In this experiment, the values of n and k , discussed in Section II, are set to $n = 30$ and $k = 25$.

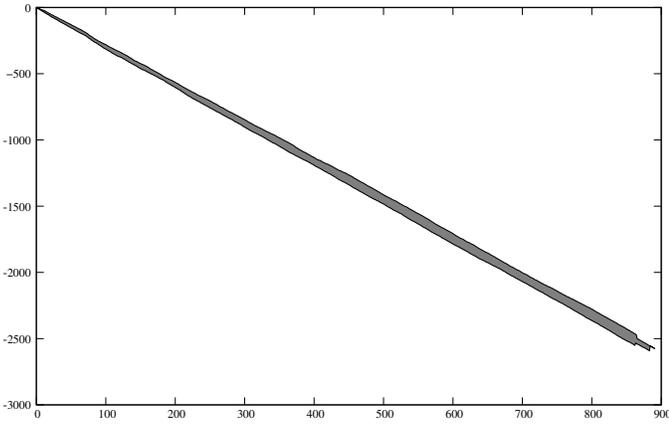


Fig. 4. The envelope of Viterbi sequence probabilities for the training data

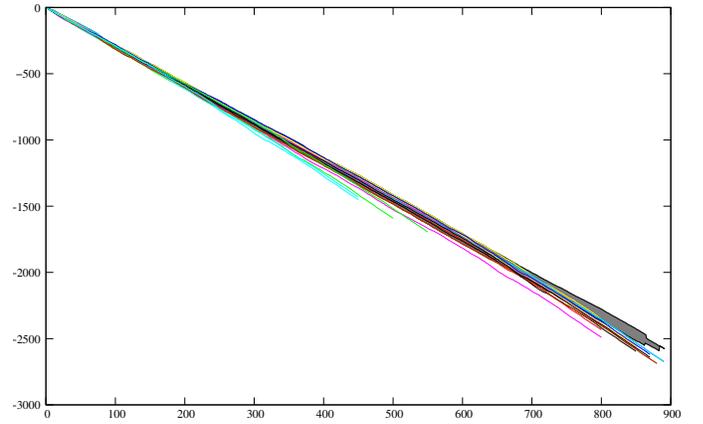


Fig. 6. The Viterbi sequence probabilities for the rolling particle error data

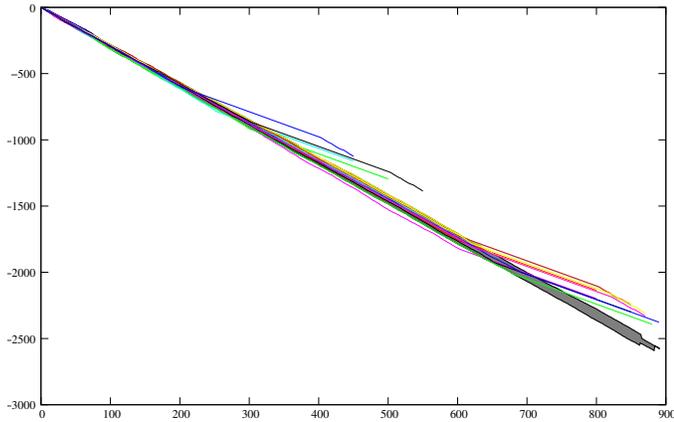


Fig. 5. The Viterbi sequence probabilities for the broken contact error data

C. Results

The envelope of log-likelihood for the training data is shown in Figure 4. The x-axis indicates the observation number, while the y-axis represents the log-likelihood of the sequence so far having been seen given the model. The linearity of the curve is striking; the narrowness of the envelope highlights the consistency in the input data reflected in the model.

The error data log-likelihood is shown plotted in figures 5 and 6, for bad contact errors and rolling particle errors respectively. It can be seen that the bad contact errors consistently generate shallow gradient curves from the point at which the errors are injected, while the rolling particle errors generate less pronounced deviations and all of these are steeper gradient curves. This is due to the bad contact errors leading to a sequence of uncommon observations which rapidly drive the transition model into state 3 in Figure 2, so that the low observation probability is compensated by the high probability of self-transition. The rolling particle errors produce a different (but more likely) observation that leads to a lower probability path through the states.

Table I shows, for the trajectories used as test data, the observation number at which the error data was introduced (and which type of error was used), the observation number at which the anomaly detection proposed the error first occurred and the time lag between the introduction of the error and

the time at which it was first detected. Since the detection of an anomaly requires a window of 30 readings, there is necessarily a lag between the point at which an anomaly occurs and the point at which it is detected, but the time of the start of that window can then be identified as the time at which the anomaly is first present. Furthermore, only 25 out of the last 30 observations must lie on one side of the mean in order to trigger an alarm. This explains why, in a few cases, anomalies appear to be detected at an observation *earlier* than the one at which the error was actually injected: the consistency in the data following the anomaly outweighs the small sample of normal data that is included in the start of the window.

V. FUTURE WORK

A. Retraining Classifiers

Experiments were carried out with the SOM of the HMM learning code to determine if an additional classification technique for COMMAS could be derived from this. Since this is an unsupervised technique, none of the learned states were labeled with responsible defects automatically, so these labels had to be added manually after clustering. This proved to be relatively straightforward as, in most cases, each cluster could be associated with a particular defect with a reasonable level of certainty.

As described in Section II, the features used in learning a HMM are not of the same format of those used for classification and diagnosis in COMMAS. Both formats of feature vector were presented to the SOM for classification. The cluster landscapes formed using the smaller set of features were able to separate the various defects from each other with a comparable accuracy to those formed with the features more traditionally used in representing partial discharge [16].

This result may point towards a less statistical feature set capable of representing partial discharge. Fewer features would be advantageous as they could provide greater generality, thereby increasing the accuracy of future classifiers. In order to test this theory, the classification agents of COMMAS will be retrained using feature vectors of the shorter format. If COMMAS can then provide a similar standard of results to its current operation, it will be possible to conjoin the

TABLE I
RESULTS SHOWING THE TIME TAKEN TO DETECT ANOMALIES

ID	Type	Error	Detected	Alarm	Time
Oct24.ed1	BC	300	301	Full	155s
Oct24.ed2	BC	250	252	Full	160s
Oct25.ed1	BC	600	601	Full	155s
Oct25.ed2	BC	600	602	Full	160s
Oct26.ed1	BC	650	643	Full	115s
Oct26.ed2	BC	650	652	Full	160s
Oct27.ed1	BC	690	692	Full	160s
Oct27.ed2	BC	680	682	Full	160s
Oct28.ed1	BC	600	601	Full	155s
Oct28.ed2	BC	600	601	Full	155s
Oct29.ed1	BC	620	618	Full	140s
Oct29.ed2	BC	620	621	Full	155s
Oct30.ed1	BC	300	289	Full	95s
Oct30.ed2	BC	200	202	Full	160s
Oct24.ed1	RP	300	333	Warn	165s
			415	Full	12m5s
Oct24.ed2	RP	250	283	Warn	165s
			365	Full	12m5s
Oct25.ed1	RP	600	633	Warn	165s
			715	Full	12m5s
Oct25.ed2	RP	600	633	Warn	165s
			715	Full	12m5s
Oct26.ed1	RP	650	683	Warn	165s
			765	Full	12m5s
Oct26.ed2	RP	650	683	Warn	165s
			765	Full	12m5s
Oct27.ed1	RP	690	723	Warn	165s
			805	Full	12m5s
Oct27.ed2	RP	680	713	Warn	165s
			795	Full	12m5s
Oct28.ed1	RP	600	633	Warn	165s
			715	Full	12m5s
Oct28.ed2	RP	600	633	Warn	165s
			715	Full	12m5s
Oct29.ed1	RP	620	653	Warn	165s
			644	Full	270s
Oct29.ed2	RP	620	653	Warn	165s
			645	Full	275s
Oct30.ed1	RP	300	333	Warn	165s
			415	Full	12m5s
Oct30.ed2	RP	200	233	Warn	165s
			315	Full	12m5s

previously discussed anomaly detection and diagnosis components further, with the original feature vectors simply passing onwards from the anomaly detection agent to COMMAS when deviances from normal behavior are observed.

B. Prognosis

So far, only a HMM for normal behavior has been discussed. However, it is equally possible to learn models based upon other types of behavior. Models can be learnt to represent operation with various known defects. This would allow a more relevant model to be substituted for the previous model of normal behavior when anomalous behavior is detected and a responsible defect diagnosed. This updated model would then allow a form of prognosis for future events if activity were allowed to continue without intervention.

If particular states or state sequences are known to be failure modes it is possible, using the Viterbi Algorithm, to predict the likelihood of the transformer entering these failure

modes if the current internal fault is not repaired. This could greatly aid decisions on the priority of future maintenance tasks.

VI. CONCLUSIONS

This paper has described a method of anomaly detection which is appropriate for identifying changes in behavior within transformer units. The technique begins by learning a HMM of normal behavior from UHF sensor data which is then used to determine whether future behavior is in sync with this model. By encapsulating the technique within an agent, messages can then easily be sent to COMMAS for the identification of underlying faults which may be causing these behavioral changes.

The overall system will alert engineers to unfamiliar behavior and greatly reduce the time needed to detect newly formed defects. It will also guide maintenance by identifying the fault, allowing decisions to be made on its seriousness and maintenance schedules to be adjusted accordingly.

REFERENCES

- [1] D. M. Staszkesky, D. Craig, and C. Befus, "Advanced Feeder Automation is Here," *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 56–63, Sept./Oct. 2005.
- [2] Q. H. Wu, D. P. Buse, P. Sun, and J. Fitch, "An Architecture for E-Automation," *IEEE Computing and Control Engineering Journal*, vol. 14, no. 2, pp. 38–43, Apr./May 2003.
- [3] A. L. Dimeas and N. D. Hatzigiorgyriou, "Operation of a Multiagent System for Microgrid Control," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1447–1455, Aug. 2005.
- [4] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The Design of a Multi-Agent Transformer Condition Monitoring System," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1845–1852, Nov. 2004.
- [5] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proc. of 17th Conf. on Innovative Applications of AI (IAAI)*, 2005.
- [6] S. Koenig and R. G. Simmons, "Unsupervised Learning of Probabilistic Models for Robot Navigation," in *Proceedings of the International Conference on Robotics and Automation*, 1996, pp. 2301–2308.
- [7] M. Fox, M. Ghallab, G. Infantes, and D. Long, "Robot Introspection through Learned Hidden Markov Models," *Artificial Intelligence*, vol. 170, no. 2, pp. 59–113, 2006.
- [8] G. P. Cleary and M. D. Judd, "An Investigation of Discharges in Oil Insulation using UHF PD Detection," in *Proceedings of the 14th IEEE Int. Conf. on Dielectric Liquids (GRAZ)*, July 2002, pp. 341–344.
- [9] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.
- [10] T. Kohonen, "Self-organized formation of topologically correct feature maps," in *Readings in Machine Learning*, J. W. Shavlik and T. G. Dietterich, Eds. San Mateo, CA: Kaufmann, 1990, pp. 326–336.
- [11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
- [12] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.
- [13] G. D. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973.
- [14] J. Gough, "Opportunistic plan execution monitoring and control," Ph.D. dissertation, University of Strathclyde, UK, 2006.
- [15] Foundation for Intelligent Physical Agents, "FIPA Agent Management Specification," 2004, available from <http://fipa.org/specs/fipa00023>.
- [16] E. Gulski, "Computer-Aided Recognition of Partial Discharges using Statistical Tools," Ph.D. dissertation, Delft University, Delft, The Netherlands, 1991.