The Trial of a Self-Learning Alarm Processor and Generator

John Ypsilantis, PhD
CEO, Heuristics Australia Pty Ltd
Sydney, Australia
http://www.heuristics.com.au *

Abstract: Intelligent decision support in the operation of power systems has been an active research topic since the mid 1980s, with recent interest in learning systems building on this foundation. This paper describes the trial of a self-learning intelligent alarm processor, The primary aims of the trial were to demonstrate the technology and the operating platform, and to verify that the system is sufficiently stable and reliable for control room operations. A summary of the trial results is presented, and the paper concludes with an assessment of the techniques.

1 Introduction

Despite the impressive advancement of computer technology and performance in recent years, there are certain classes of problem which remain intractable and require the application of human expertise. Several examples of this class of problem can be drawn from utility operations.

Intelligent decision support in the operation of power systems has been an active research topic since the mid 1980s [1]. Recent interest in self-adaptive or self-learning systems has built on these foundations. Learning intelligent decision support systems, or *IDSS*, are now being proposed for various power system operations tasks.

It is important to note that an IDSS assists with manual operations. Instead of automating tasks, an IDSS aims to assist a human with a decision making process.

This paper describes the trial of a self-learning intelligent alarm processor, *LIAP*. The primary aims of the trial were to demonstrate the technology and the operating platform, and to verify that the system is sufficiently stable and reliable for control room operations.

2 LIAP

2.1 Architecture

LIAP is essentially a "black box" open system which interfaces with a SCADA master station. It accepts alarm text streams, identifies and then suppresses secondary or redundant alarms. LIAP can also generate diagnosis information via its own alarms. An important feature of LIAP is its ability to detect and suppress interspersed sequences of redundant alarms.

The learning and alarm processing functions are carried out in parallel, from the same input data stream. The learning function is completely unsupervised in regard to alarm blocking, and is partially supervised in regard to intelligent alarm generation.

Figure 1 summarises the main functional components of LIAP.

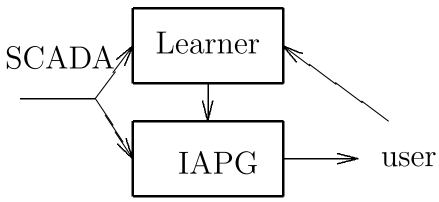

Figure 1: LIAP block diagram

2.2 Learner

The learning component of LIAP was initially implemented using the sequence tree algorithm. This algorithm, and its application to alarm processing, is described in [2] and [3].

The sequence tree algorithm detects candidate alarms which respond to others. LIAP may block such alarms in the future.

An alternative, the *sequence graph* algorithm, was subsequently implemented and used in the trial. The sequence graph algorithm achieves the same objectives as the sequence tree, but it uses a graph search technique instead of a tree analysis.

Another major advantage offered by the sequence graph algorithm is true unsupervised learning. Alarm strams must be partitioned before they can be used with the sequence tree algorithm. This is not necessary with a sequence graph.

Both the sequence tree and sequence graph algorithms exhibit nonmonotonic learning. Any learned information which subsequently becomes invalid, is “forgotten.” An example of this is the decomissioning of telemetry equipment at a substation, which would result in the cessation of certain alarms or alarm sequences.

2.3 Intelligent Alarm Generator

The output of the sequence tree and sequence graph algorithms may be used to generate rudimentary diagnosis and reporting of conditions that may otherwise be lost in the raw stream of alarms. This function is termed *intelligent alarm generation.*

Intelligent alarm generation is in essence the replacement of a blocked sequence of alarms with a single, descriptive message. LIAP allows the end user to associate such messages with learned sequences. These are then reported in context whenever the sequence, possibly interleaved with other alarms, is detected in the alarm stream.

LIAP maintains the associated messages as the knowledge base evolves. As with the learning functions, messages may be added or changed on-line, in parallel with inference.

3 Trial

3.1 Overview

The objectives of the trial were:

- To test LIAP with realistic data,

- To compare the performance of the sequence graph and sequence tree algorithms in this application,
- To test the parallel learning and inference techniques, and
- To verify stability and reliability of the software.

The trial was carried out at the Energy Australia control room in Sydney. Energy Australia is the largest power distribution utility in New South Wales.

LIAP was deployed on a very modest hardware platform. This comprised a Pentium 75 personal computer with 16Mb of RAM and 0.5Gb of disk space, operating under Linux.

3.2 Interface Issues

System interface was a primary consideration in the trial. In order to eliminate the possibility of interference with normal SCADA operations, LIAP was interfaced serially via one of the incident logger ports. Input was therefore a stream of plain text alarms delimited by line breaks.

The processed alarm stream can be viewed via a web browser, an X-windows client or a plain text client. Any number of these can be used concurrently, which facilitates the integration of the system with a SCADA, while at the same time minimising the impact on core SCADA functions and performance.

A small, isolated LAN was used to access the processed alarm stream during the trial. While this may be suitable for a full, live deployment of LIAP, a more practical approach would be to use the same LAN as the SCADA operator workstations. The LIAP client task would then appear as another window on these workstations.

4 Results

4.1 Phase 1

The trial was carried out in two phases. In the first phase, LIAP was deployed with the sequence tree algorithm and its performance was monitored over several months.

A number of issues arose during the first phase:

- Memory usage rapidly increased to the point where it was no longer possible to maintain the working databases in RAM.
- The sequence tree exhibited efficient learning, but it also exhibited a tendency to learn very specialised rules. This resulted in a large knowledge base.

Fig 2 is the learning curve that resulted from phase 1.

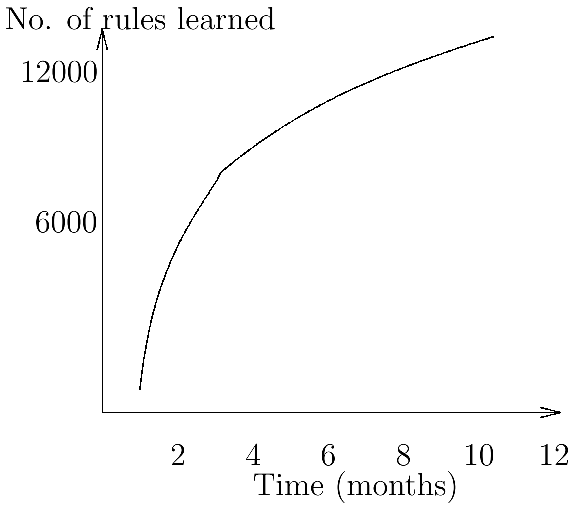

Figure 2: Learning curve for sequence tree

4.2 Phase 2

Phase 2 of the trial was carried out using the sequence graph algorithm and a disk-based database. The second phase was conducted over a similar duration, and the resulting learning curve was compared to that found in the first phase.

The sequence graph exhibited a similar learning period, but also demonstrated a significant improvement in learning efficiency. The resulting number of rules at the end of the trial was approximately 1/40th of the number found using the sequence tree. This also implies that the rules generated by the sequence graph are significantly more general than those from the sequence tree.

The learning curve for Phase 2 of the trial is shown in Figure 3.

The disk-based database did not affect LIAP's performance. However, the smaller number of rules was also a significant contributing factor.

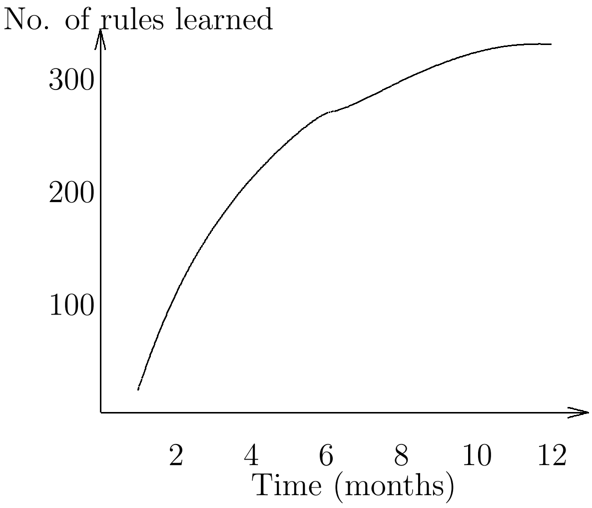

Figure 3: Learning curve for sequence graph

4.3 Discussion

The learning curves in both phases suggest that LIAP substantially completed most of its learning after 6 months. It is interesting to note that both learning algorithms exhibited the same characteristics in this regard. This figure appears to be a good benchmark in regard to estimating the amount of training data required to achieve effective performance.

The sequence graph has an added advantage in that it is completely unsupervised. In contrast, the sequence tree implementation of LIAP required classification of alarms as either unblockable or potentially blockable.

LIAP performed well throughout the trial. The computer and operating system exhibited appropriate stability for live implementation, and the system was not affected by the main Year 2000 date transitions on 1 January 2000 and 1 March 2000.

5 Conclusion

The trial has demonstrated that LIAP is stable and useable. LIAP can perform well on modest hardware.

The sequence graph algorithm is clearly superior to the sequence tree in this application, and has the added advantage in that it is totally unsupervised.

The learning techniques used for LIAP may be applied elsewhere. Future directions include the application to analysis of other sequence based problems in utility trading. Application to problems arising in general e-commerce and internet operation is also possible.

6 Acknowledgements

The author gratefully acknowledges the cooperation of Energy Australia, particularly Mr Bruce Edney, Mr Greg Hasting and Mr Glenn Kershaw of Energy Australia for their assistance throughout this project.

References

[1] B. F. Wollenberg. "Feasibility study for an energy management system intelligent alarm processor." IEEE Trans. Power Systems vol. PWRS–1, May 1986 pp. 241–247.

[2] J. Ypsilantis and H. Yee. "Machine learning of sequences-of-events: application to power distribution." Electric Power Systems Research, vol. 23, 1992 pp. 115–122.

[3] J. Ypsilantis and H. Yee. "Machine learning of rules for a power system alarm processor." Proceedings of the IEE International Conference on Advances in Power System Control, Operation and Management (APSCOM'91), Hong Kong, November 1991 pp. 321–325. (IEE Publication Number 348).

[4] J. Ypsilantis and H. Yee. "Genetic learning applied to distribution system fault diagnosis and alarm processing." Proceedings of the International Conference on Expert System Applications to Power Systems (ESAP'93), Melbourne, Australia 1993.

Biography

John Ypsilantis recceived his BSc (Pure Mathematics and Computer Science) in 1984, BE (hons) and PhD in Electrical Engineering in 1986 and 1993 respectively, all from the Univerity of Sydney, Australia. He has worked as a senior R&D engineer, manager and consultant on several SCADA and human-machine interface projects, and he is the CEO of Heuristics Australia Pty Ltd. Dr Ypsilantis' interests include the application of adaptive intelligent decision support to the supervision of power and gas transmission and distribution, internet and e-commerce.

Dr Ypsilantis is a Member of the Institution of Engineers, Australia, a Chartered Professional Engineer, an Associate of the Australian Computer Society and a Member of the Institute of Electrical and Electronics Engineers.