Proceedings of the 2001 IEEE Aerospace Conference, Big Sky Montana, March 2001.

Anomaly Detection for Advance Military Aircraft Using Neural Networks[1]

Tom Brotherton and Tom Johnson
The Intelligent Automation Corporation
10299 Scripps Trail, PMB 231
San Diego, CA 92131
(858) 679-4140
Tom.Brotherton@iac-online.com

Abstract— Automated *Prognostics and Health Management* (PHM) is a requirement for the advanced military aircraft. PHM is the key to achieving true *condition-based maintenance*. PHM processing strategies include modules for the detection, diagnosis and prognosis of known fault conditions. However in real operations there will also occur faults and other off-nominal operations that were never anticipated nor ever encountered before. We call these events *anomalies*. Missing the presence of an anomaly could potentially be catastrophic with the loss of the pilot and aircraft. We have developed a neural net approach for performing *anomaly detection*. The neural net anomaly detector 'learns' to recognize consistent sets of multiple input sensor signal patterns from known nominal data. It is generic and has been applied to a variety of aircraft subsystems and for fusion with other detectors with excellent results. Presented here are a description of the neural net anomaly detector and the application to advanced military aircraft.

1. Table of Contents

2. 1. Introduction

Automated *Prognostics and Health Management* (PHM) is a requirement for advanced military and commercial aircraft. PHM is the key to achieving true *condition-based maintenance*. The PHM system will potentially save money, aircraft downtime, and even lives by providing for the right parts to be in the right place at the right time.

Advanced military aircraft PHM processing strategies include modules for the detection, diagnosis and prognosis of known fault conditions. However in real operations there will also occur faults and other off-nominal operations that were never anticipated nor ever encountered before. We have called these events *anomalies*. This is particularly true with new military aircraft but is also important for legacy aircraft. Missing the presence of an anomaly could potentially be catastrophic with the loss of the pilot and aircraft. An important part of the overall system is the inclusion of *anomaly detection*. The role of the *Anomaly Detector* (AD) is to flag and report these unanticipated and never seen before events.

Table 1. Table of acronyms

ACRONYM	MEANING
AD	Anomaly detector
APU	Auxiliary power unit
BU	Basis unit
CD-RBF	Class dependent – Radial basis function
EGT	Exhaust gas temperature
FF	Fuzzy factor
LMS	Least mean square
LVQ	Linear vector quantizer
MLP	Multi-layer perceptron
NN	Neural net
NNAD	Neural net anomaly detector
PHM	Prognostics and health management
RBF	Radial basis function

Several different approaches for performing anomaly detection are under development. Described here is a neural net approach for performing this processing. Neural networks have been shown to be ideally suited for solving detection and classification problems when all fault classes are known. However in the real world often the number and labels of the classes of interest may be unknown. This is particularly true for the anomaly detection problem where in its initial form there is only one known class; the *nominal* (or normal) class. A-priori it is not known which of the signals being monitored will give rise to a potential fault. Thus traditional indicators of known faults (for example a single spectral peak) cannot be exploited for detection and classification. All signals must be monitored. This makes anomaly detection a "harder" problem to solve then traditional classification problems. The neural net used for the AD is a radial basis function (RBF) neural net [1,2]. In our approach the architecture of the RBF neural net is constrained so that groups of hidden unit basis functions within the neural net are associated with only a single class [3,4]. This allows for the neural net to be used for anomaly detection. In addition to the detector output, the processing gives a measurement of the "distance from nominal" for each of the signals that are input to the AD. These distance measures are then passed down-stream to *reasoners* that isolate the exact nature of the anomaly. We have successfully applied the anomaly detector to a variety of problems that include applications where up to 50 different sensor measurements were monitored simultaneously.

The approach is generic and has been applied to a variety of problems including advanced aircraft subsystems and for the fusion of multiple detectors [5,6] with excellent results.

The outline of the paper is as follows. Section 2 contains a detailed description of the approach. Application of the neural net AD to aircraft subsystem data is contained in Section 3. Section 4 contains a summary and conclusions.

3. Neural Network Anomaly Detection

Presented here is a detailed description of the neural net anomaly detector (NNAD). NNAD uses radial basis function (RBF) neural nets (NN) to form a statistical model of "nominal" data. As new data enters into the system, it is compared to the RBF NN model. If data falls within the boundaries defined by that model, then it is flagged as "nominal". If is does not, then it is flagged as an "anomaly". Figure 1 shows a simplified example of that processing.

In Figure 1, the input signal data is 2 dimensional. The RBF NN model of the nominal data has two basis functions that are represented by the two ellipses in the figure. Here the basis functions are Gaussian in shape to give a continuous degree of membership measure from each of the basis functions centers. The two ellipses represent constant degree of membership contours that may be used as a detection threshold.

In the figure 1, the small green and red circles represent test samples from nominal and anomaly data respectively. The green circles all fall inside of the detection threshold so they are classified as 'nominal'. The red circles fall outside of the detection threshold and they are declared as anomalies. One of the red circles is clearly far away from the detection threshold ellipse and thus clearly an anomaly. The other red circle is much closer. Is it indeed an anomaly or is it a false alarm?

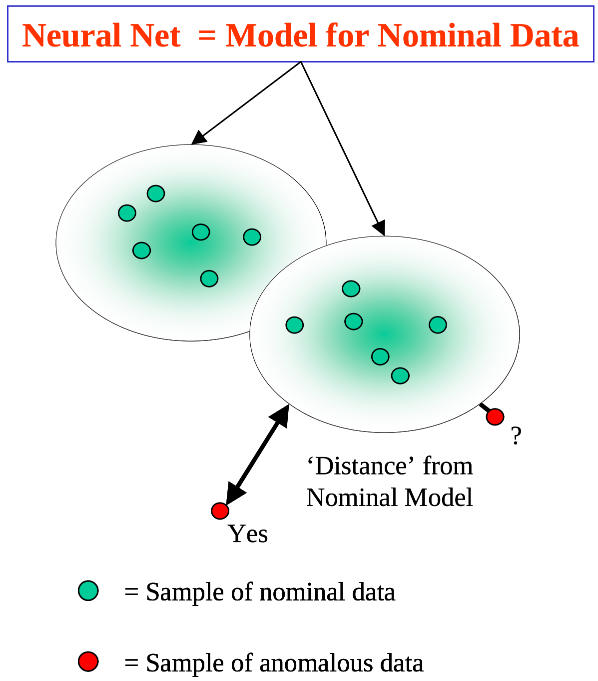


Figure 1 Simplified example of the NNAD

That's the basic approach to anomaly detection using the neural net. Of course with the real data we are dealing with many more features (6-50 for aircraft applications) and the number of basis functions, particularly for transient data, is much larger (typically 80+). In addition the basis functions need not be Gaussian in shape, so that the processing becomes still more complicated.

Figure 2 shows a high-level flow diagram for the overall processing flow for the development and running of NNAD. Training of neural nets and adjustment of processing and detection parameters are all performed off line (the processing in the blue box of figure 2). Once the system has been trained, those parameters and neural nets are used to perform the anomaly detection.

Each of the major functions shown in Figure 2 is described in more detail in the following sections.

Mode Detection

Typically subsystem components have different operating characteristics depending on different environments and different operating conditions. Theoretically with sufficient training data, all these different characteristics can be handled with a single neural net. However that network will

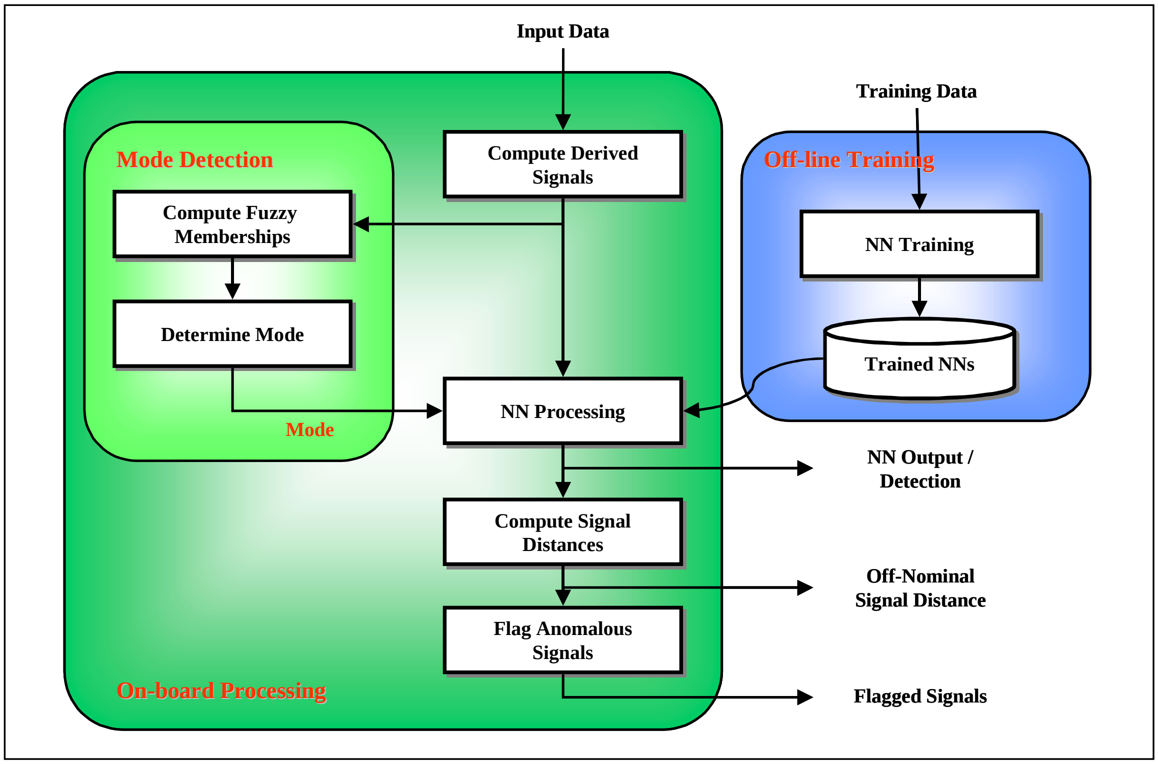


Figure 2 High level flow diagram for the development of NNAD

become extremely large and may exhibit poor performance for conditions that do not have much associated training data. One way to handle this problem is to break the problem into different *regimes*. An example are "low temperature" and "high temperature" regimes. Another example of a set of regimes could be "shaft speed" or "throttle rate of change". The *mode* of operation is defined as the combination of the regimes. We break the overall anomaly detection problem up into smaller problems by considering and exploiting these different operating modes.

Currently mode definition is done by hand. However modes can also be identified by multivariate statistical changes in the data [7]. Using the data makes more sense from the statistical processing point of view. Once the data has been segmented out for a particular mode, then processing tailored for that mode can be developed. Processing tailored to a specific regime / mode will always gives better results then general processing.

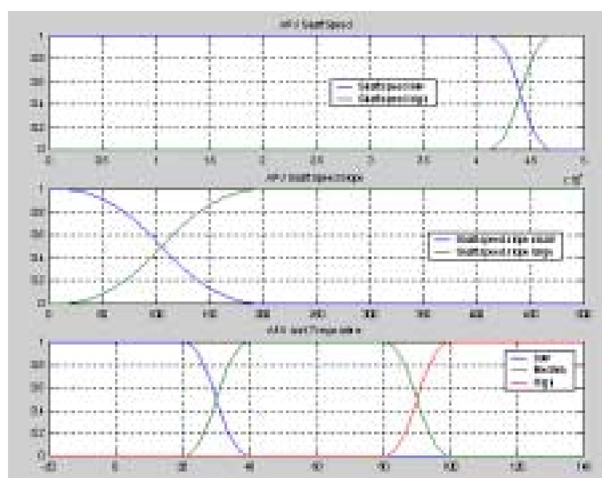


Figure 3 Regime and mode example

We use a fuzzy logic approach to regime / mode classification. Figure 3 shows an example of regimes defined for an auxiliary power unit (APU). There are three different regimes shown that have to do with the component operating condition and environment. They are shaft speed, rate of change of shaft speed, and air temperature. For example the bottom plot is air temperature. The three curves on that plot correspond to the fuzzy membership functions of "low", "medium", and "high" outside temperatures. The mode of operation is then just defined as the outer product of each of the individual regimes. An advantage of using the fuzzy approach is that regimes and modes all overlap so that there are no hard boundaries between regimes and modes and transition between regimes and modes is "graceful". Also it is easy to see that fairly complex set of "rules" to determine different modes are the result of defining and combining very simple regimes.

Neural Networks

The heart of the processing are the neural networks used to form the model of nominal signal data. The particular neural net that we've used is the Radial Basis Function (RBF) neural network (NN) [1,2]. The RBF NN is essentially a nearest neighbor type of classifier. Thus it has several properties that make it ideal for performing anomaly detection. These are not found with multi-layer perceptron neural networks.

The architecture for the standard RBF NN is shown in Figure 4. There are two steps involved with "training" the RBF neural network. The first step is clustering of the input data used to form the hidden-layer *basis unit* functions (BUs). All of the input training vectors for all classes are lumped together at this stage. The data is then clustered into one of several candidate BUs using a clustering algorithm such as the *linear vector quantization* (LVQ) algorithm. There are a variety of techniques to perform to perform this clustering that are included in our program. We have found for anomaly detection the k-means algorithm [8] gives good results in reasonable time.

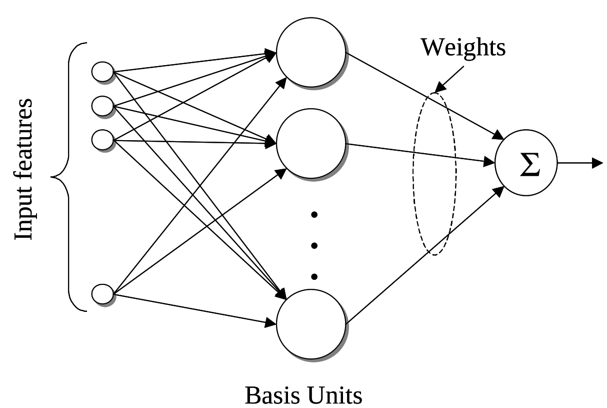


Figure 4 Standard RBF Neural Net Architecture

For NNAD described here these basis functions take the form of multidimensional Gaussian distribution functions. The mean and variance of each dimension is estimated from the data. Following clustering is a *least mean-square* (LMS) weighting of the BU outputs to form the desired function approximation for classification.

During clustering we force the basis units in the RBF NN to be associated with only a single class of data. For the NNAD only nominal data is used so that each of the BUs is used to represent some portion of the overall feature / feature trajectory space. For transient data the number of BUs can be quite large. However for general classification this will include sets of basis units associated with different known fault categories.

The output from the RBF NN can be determined in two ways. The first is simply the final output of the neural net as described above. The second is to select the basis unit that has the maximum activation. The BU with the highest activation will be the BU that's "nearest" to the set of input signals. This is possible because all of the BUs are associated with only the nominal data class. For NNAD we use both methods for getting the neural net output. The LMS output is used for the overall detection and the nearest basis unit is used for the individual signal detections. In effect the RBF NN neural net is a nearest-neighbor classifier with the BUs defining prototype models for different segments of the signal data. As other classes are added, additional BUs are added. These too will be associated with just a single class. We call this architecture a class dependent – radial basis function (CD-RBF) neural net [4].

In estimating the RBF neural net parameters for anomaly detection, an estimate of the multidimensional centroid (mean) of each basis unit as well as the multidimensional variance is made. The widths can be artificially expanded by application of a constant we denote as a *fuzzy factor*. The width of one dimension of the basis unit is found as:

$$width = \sqrt{fuzzy\ \ factor \times \mathrm{var}\,iance\ estimate} \qquad (1)$$

Several different kinds of basis units can be considered for the CD-RBF. The "traditional" choice is that of a Gaussian functional. That is what we have used for NNAD to date. However, for other signal types, different shaped basis unit functions that are "matched" to the different data underlying distributions may be more appropriate. For example when magnitude spectral data is input to the system a better choice may be a basis function that has a Rayleigh distribution shape. This is because the magnitude spectral data has a Rayleigh distribution. The neural net can be made into a fuzzy logic net by using fuzzy logic membership functions for the basis unit functions [9,10,11]. Other basis functions and mixtures of basis functions, depending on the distributions of the signals that are being input / modeled, can be considered as well.

The CD-RBF has several interesting properties that are important for classification (diagnostics), prediction (prognostics) and anomaly detection problems.

• *Interpolation:* When the CD-RBF uses Gaussian shaped functions as the basis nodes internal to the neural net, in essence the neural net is a *sum-of-Gaussians* model. The final output of the neural net is the linear sum of the basis functions. The combination provides for a smooth transition between the basis functions, which are derived directly from the input data. Thus the output is a smooth transition of the inputs even when the inputs represent different transition states. This is important for filling in "gaps" that may exist in the training data as is almost always the case for real data problems.

• *Anomaly detection*: The net can detect when some new event that the network has never encountered before during training is present at the inputs as described above. This comes about because of the nearest-neighbor property. When signal data is input to the system, it is matched against the nominal model developed. If the input signals do not fall into any of the basis units, then an anomaly detection is made.

• *Neural net visualization*: the CD-RBF neural net allows for easy visualization of "why" the network performed as is did. This is not possible with MLP neural networks, since the individual nodes are not specifically tied to unique classes or the unique partitionings of the features that are a constraint during clustering for the CD-RBF. For anomaly detection processing, it may allow insight into what measurements are important for identifying certain types of subsystem component state conditions. The visualization step creates plots showing the distance off nominal of each feature vs. time. Examples of these plots are shown below in the "Application" section.

• *Classifier degree of membership*: No "hard" decisions need be made regarding the particular class that the basis unit is associated with. Thus, classes that are similar would both have high degrees of membership, and a "hard" decision on what is happening can be postponed for further analysis, for example by a downstream expert system. It would also indicate that there is not sufficient information for discrimination between those two classes and that additional work needs to be performed to develop other signal measurements or derived signals to differentiate the overlapping classes.

Training

The neural nets can be trained using standard RBF training techniques as described above. However there is a unique problem when training for anomaly detection in that there is only a single class for which the basis functions are designed to model. For most trainers, a training goal is to achieve "no errors" when applying the trained neural net to the data that was used for training. This is fine when there are several contender classes, as the width of the BUs is controlled so that there is little / no overlap of classes. For anomaly detection this implies that no training events are ever flagged as an anomaly. Just making the width of the BUs very large ensures this will always be true, but at the expense of no anomaly detections. There are two ways to address this issue.

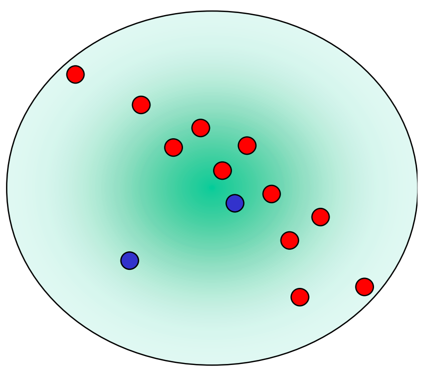
A first approach is to simply allow for false detections (false alarms) during training. Training to a set false alarm rate is included in NNAD. Typically in the results that we have developed, we allow a 2% false alarms during training. These false alarms can be removed by fusion with other ADs [6]. Allowing for false alarms during training is in fact required as the number of training samples grows. Consider data sampled from 2D Gaussian distribution. As the number of samples selected increases, more and more points will fall into the tails of the distribution. If the training criterion is to have no false alarms, then the encompassing detection hyper-ellipse will grow larger and larger and will loose all sensitivity with respect to detecting true anomalies.

A second approach is to artificially constrain the "width" of the variance estimates for the NN. This can be accomplished by adjusting the fuzzy factor FF as described above. During training several fuzzy factors maybe considered. The program automatically selects the "best" versus a tradeoff with the number of basis units used. Initially the neural net trainer tries to model the data with a single basis unit. If the desired performance level is achieved, training stops. If not, additional basis units are added and training is restarted. Basis units are added until the desired training performance criteria is met or the maximum number of basis units (set by the user) are used.

To model the data using a large fuzzy factor (i.e. FF>1) only single basis unit may only be required. It will have a width that is proportional to all the data used for training. Figure 5. shows an example of such a RBF model for the 2 input signal example. The red dots represent the set of points that correspond to "nominal" data that were used for training. As seen those data follow a fairly narrow trajectory. However it's also seen that a single, large BU is sufficient to model the training data.

Consider the blue dots in Figure 5. There are two test cases that are input to the system. On of the blue dots indeed looks to be "nominal" as it is bunched in with the other nominal data points used for training. The second blue dot however, does appear to be a true anomaly. It is not detected as an anomaly because it is well inside the nominal data basis unit.

- Large FF
  - Small number of EBUs

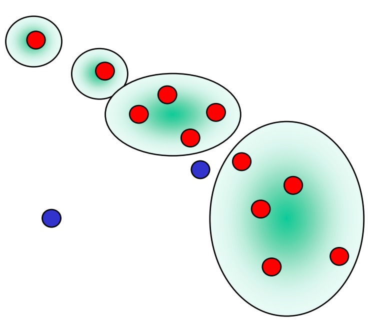


- Very general
- Missed detections

Too General ?

Figure 5 The RBF NN model found when a large fuzzy factor is used.

- Small FF
  - Large number of EBUs



- More sensitive to outliers
- More false alarms

Over Trained ?

Figure 6 The RBF NN model found when a small fuzzy factor is used.

One solution to this problem is to restrict the size of the basis unit to be smaller using the fuzzy factor constraint. Setting the fuzzy factor to be smaller than 1.0 will accomplish this. Figure 6 shows the same data set and the RBF NN model found by setting the fuzzy factor to be small (<1). As seen in the figure there are now 4 BUs used to model the data. This appears to be a better model. The blue dot that was thought to be an anomaly is clearly detected as such with this model of the data. However now there is a false alarm in that the blue dot that is nestled among the training data is also flagged as an anomaly.

This is not that serious of a problem. Additional training data that covers this region of "nominal data space" will result in a basis unit that this blue dot will be included in. With more training data there should be no "holes" in the space of nominal events. With the real problem using 49+ different signals and limited data sets the chances of "holes" in the training data increases.

Compute Signal Distances

When a detection is made, we compute the distance "off nominal" of all the input signals. Figure 7 shows an example of how this processing is done for the two-signal case. In figure 7 the red dot represents the test sample under consideration. Note that no single signal needs to be significantly off nominal for a detection to be made. Rather it is the aggregate signal set that gives rise to the detection.

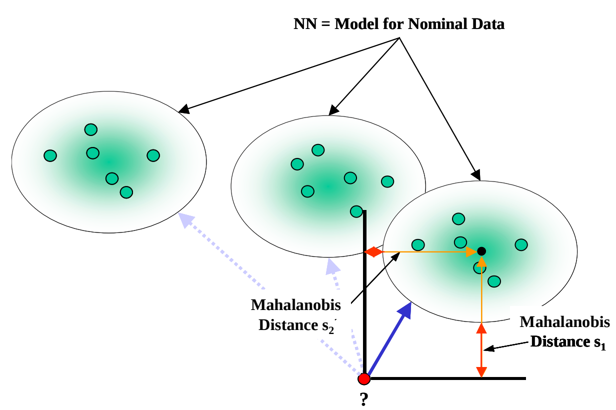


Figure 7 Off-nominal distance calculation

All of the neural net models developed for the subsystem data have between 6 and 100+ basis units. The first step in the processing is to determine which of those basis units is the "closest" to the sample point being tested. The distance computed is the Mahalanobis distance to the each of the clusters. The Mahalanobis distance is used as it accounts for not only the centers of each of the basis units, but also the spread. In Figure 7, the dark blue arrow represents the basis unit that is closest to the sample point. It is the BU that gives the largest output. This BU is selected for the next step in the processing. The basis unit is the most like the set of input signal in a nearest neighbor sense, and thus gives rise to the minimum off nominal distances. Selecting the closest basis unit for <u>each</u> signal individually is <u>not</u> correct. The detection and distance are a function of the set of signals.

The distance is then computed for each of the individual signals as the Mahalanobis distance from the center of the basis function that was selected. In figure 7 the yellow arrows in the figure indicate the distances for the two input signals (s_1 and s_2) to the center of the nearest BU. The red arrows represent the Mahalanobis distance that is reported.

4. APPLICATION TO ADVANCED MILITARY AIRCRAFT SUBSYSTEMS

Considered here is processing of two data sets that are advanced fighter aircraft related. One is collected from the hydraulic system for the flight control surfaces. The second is for the auxiliary power unit (APU).

Hydraulic Data

The hydraulic data used here consisted of seven different data sets. Six of the data sets represented ‘nominal’ data. The seventh data set is anomaly data. Turning off the accumulator in the hydraulic system created the anomaly data. The nominal data sets represent different levels of stick movement by the pilot varying from ‘no movement’ to ‘severe’. There were 8 channels in the data that correspond to different pressure measurements within the system. Figures 8 and 9 show examples of the data sets. Figure 8 is nominal data under heavy stick movement conditions. Figure 9 is anomaly data with heavy stick movement.

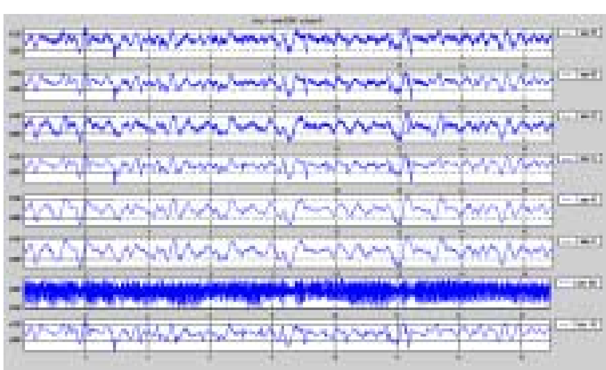


Figure 8 Example nominal hydraulic system data

Turning off the accumulator changes time constants in the response of the system to pilot stick movements. This can be seen visually as a change in the second order of the statistics of the data. As seen the two data sets are similar, varying possibly in their second order statistics.

Five of the nominal data sets were used for training the neural net anomaly detector. One of the nominal and the anomaly data were used for testing the system. The Figure 10 shows the results of the anomaly detector output when the input data is test nominal data.

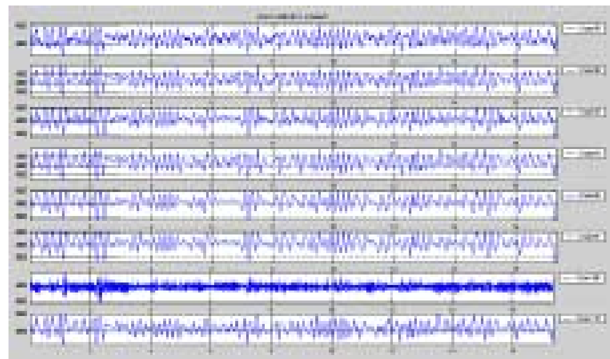


Figure 9 Example anomaly hydraulic system data

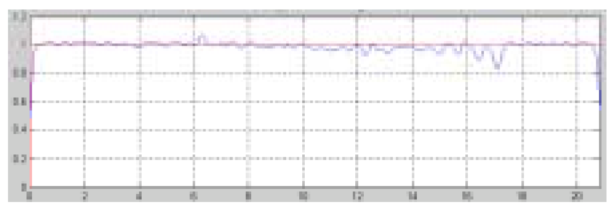


Figure 10 NNAD detector output for nominal test data input.

In the figure the blue line corresponds to the ‘raw’ neural net output. The red line corresponds to the thresholded output or the detector output. Determining the detection threshold is performed automatically during training. The neural net raw outputs normally vary between zero and one with a one indicating a positive response for the data classification. For the example here a value near one indicates nominal data. The thresholded output (the red line) is binary; it is 1 when the input data is determined to be nominal and it is 0 when the input data is flagged as an anomaly. As seen the data is classified correctly as ‘nominal’.

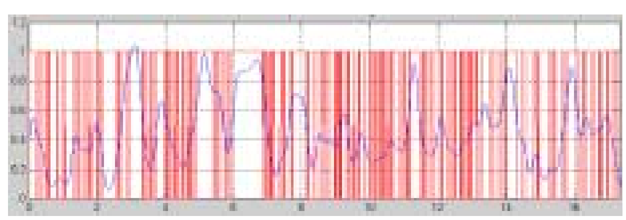


Figure 11 NNAD detector output for anomaly test data input.

Figure 11 shows the NNAD output when the anomaly test data set is input to the system. As seen the raw output wanders between 0.1 and 1. The detector output is also bouncing around between 0 and 1. This is as expected as the nature of the fault causes the system to pass through nominal and anomaly. The detector at this point is operating on each input scan independently. If the output of figure 11 were input to an M of N detector (i.e. accounting for the number of detections that occur over time), then a solid detection would be made. However as seen in a companion paper, fusion with a second anomaly detector also cleans up the output [6].

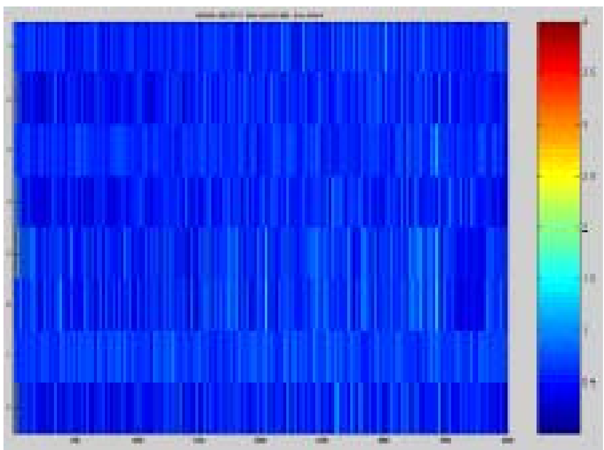


Figure 12 NNAD off-nominal distance measure for nominal input test data

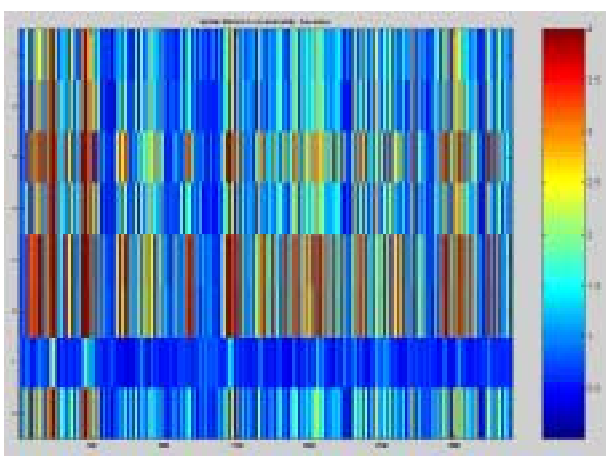


Figure 13 NNAD off-nominal distance measure for anomaly input test data

For both of the data sets processed, the off-nominal distance measure was calculated and displayed. Those results are shown in figures 12 and 13. Figure 12 shows the results for the nominal test data. The figures show input signal along the y-axis and time on the x-axis. The distance off nominal is color encoded as indicated on the side bar. The display saturates with an individual detection threshold that corresponds to roughly two-sigma (i.e. the two-sigma hyper-ellipse of the multidimensional Gaussian distribution).

Figure 12 show the off-nominal distance measures for the nominal test data set. As expected, all the signals are “close” to nominal as indicated by the blue color on the display. Figure 13 shows the off-nominal distance measures for the anomaly test data. As seen the individual channels drop outs follow closely to the detector outputs with lots of transitions between ‘close’ to nominal to saturated (the deep red color). This is as expected.

Auxiliary Power Unit Data

The second data sets processed were from an auxiliary power unit (APU) from an advanced fighter aircraft. That data contained several sets of nominal data as well as real anomalies. Three nominal data sets were used for training. Testing was performed on an independent anomaly data set. One of the training data sets was used for the nominal results presented below.

There were a variety of signals measured from the APU to use for anomaly detection. 6 signal channels were selected for input to NNAD. They are shown in Table 2. Figure 14 is an example of one of the data sets. As seen in the plot the data is highly non-stationary.

In order to focus processing on the different segments of the data, the data was broken up into smaller consistent data sets using the mode definitions described above. The modes used correspond roughly correspond to APU start / shutdown and running in high, medium, and low inlet temperature conditions.

CHANNEL	DESCRIPTION
1	Shaft speed
2	Fuel flow
3	Oil temperature
4	Inlet temperature
5	Exhaust gas temperature (EGT)
6	Compressor Discharge Pressure

Table 2 APU input signals

Figure 15 shows the NNAD detection output for nominal data. The color-coding is the same as that used for the hydraulic data. As seen, with the exception of a few false alarms the data is flagged as nominal. These false alarms can be easily removed using an M of N type detector, median filtering or by fusion with another AD [6].

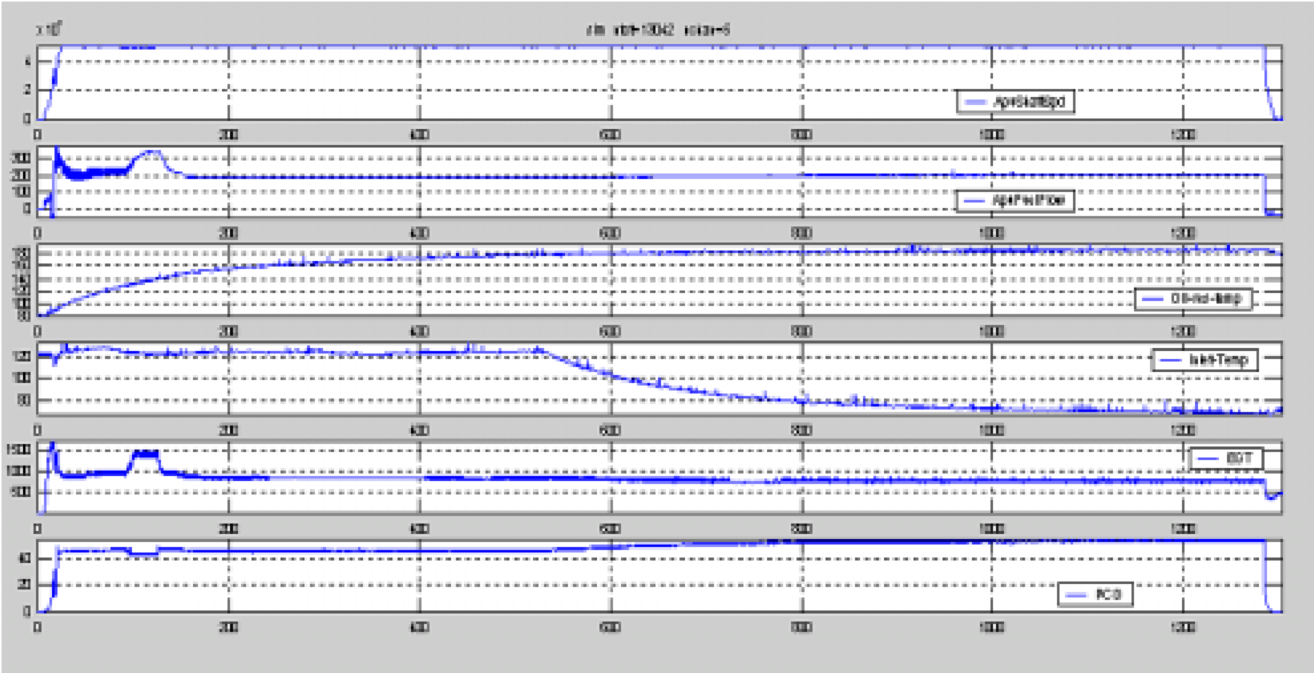


Figure 14 APU input data

Figure 16 shows the NNAD detection output for the anomaly data. Both the red and blue traces are pegged at ‘0’ (the blue lies on top of the red). The detector is very sure the input data is not nominal data.

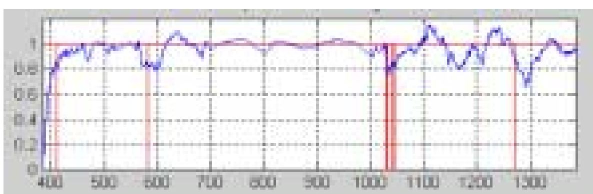


Figure 15 APU nominal data NNAD detector output.

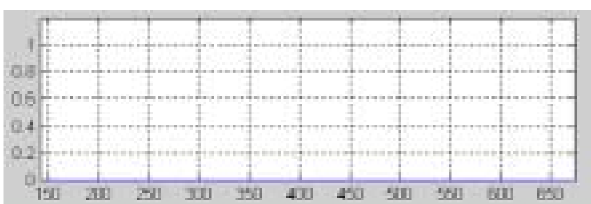


Figure 16 APU anomaly data NNAD detector output.

Figures 17 and 18 show the corresponding off-nominal distance measures for the nominal and anomaly data sets respectively. For the nominal data all the individual signal off-nominal distances are close to 0 as is expected. For the anomaly data, as seen several of the channels are saturated at the detection threshold. Closer examination shows that the EGT is the ‘farthest’ off nominal by quite a large margin. Indeed the fault present in this data are unreliable EGT measurements.

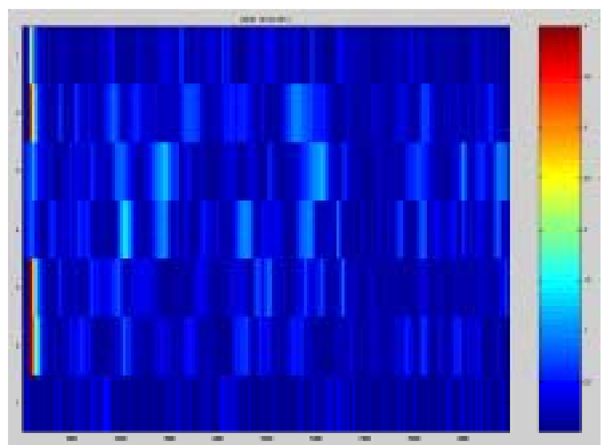


Figure 17 APU off-nominal distance measure for nominal data input

5. SUMMARY AND CONCLUSIONS

Automated Prognostics and Health Management (PHM) is a requirement for the advanced military aircraft. PHM is the key to achieving true condition-based maintenance. An important part of the overall PHM processing is the ability to detect operating conditions and potential faults that were never anticipated nor ever encountered before. We call these events anomalies. Presented here was a description of a neural net approach for performing anomaly detection. We call the algorithm the neural net anomaly detector or NNAD. NNAD fuses input signal data by using the entire set of input signals to form a model of nominal data. It is the entire set of signals that are used for the detection.

Applications of NNAD to processing of data from two different advanced fighter aircraft subsystems were presented. In both those applications NNAD worked well and performance was as expected.

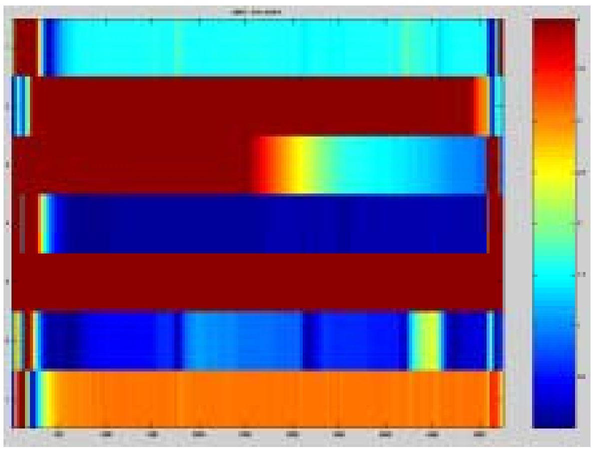


Figure 18 APU data off-nominal distance measures for anomaly test data.

The neural net approach is generic and ideal for anomaly detection problems. It has been applied to processing JSF119 engine data and space shuttle main engine data with equally good results. In those applications there were over 50 signals input. It was also been applied for fusion of two different anomaly detectors developed for advanced military aircraft [6].

NNAD uses a class dependent – radial basis function neural net (CD-RBF) to form a model of the data used for training. As described here, only nominal data has been used. However the approach is easily extended to include known fault classes as well. The CD-RBF also allows the user to gain insight into the data and to visualize "why" the neural net has performed as it has. Those results appeared here as off nominal distance presentations of all the features vs. time. For classification problems, the approach can be extended for performing automated rule extraction (or as is more commonly known as data mining) [4].

6. REFERENCES

[1] Hush, D. R. and B. G. Horne, "Progress in Supervised Neural Nets – What's New Since Lippman?," *IEEE Signal Processing Magazine*, Jan. 1993.

[2] Haykin, S, *Neural Networks: A Comprehensive Foundation*, second edition, Prentice Hall, 1999.

[3] Brotherton, T.W., T. Johnson, and G. Chadderdon, "Classification and novelty detection using linear models and a class-dependent elliptical basis function neural network," *Proc. of The Int'l Joint Conf. on Neural Networks* (IJCNN), Anchorage, AK, May 1998.

[4] Brotherton, T.W., G. Chadderdon, and P. Grabill, "Automated Rule Extraction for Vibration Analysis of Rotating Machines", *The 1999 IEEE Aerospace Conference,* Snowmass, UT, March 1999.

[5] Brotherton, T.W. and E. Mears, "Applications of Neural Nets to Feature Fusion," *The 26th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 1992.

[6] Brotherton, T.W. and R. Mackey, "Anomaly Detector Fusion Processing for Advanced Military Aircraft", *The 2001 IEEE Aerospace Conference,* Big Sky, MT, March 2001.

[7] Mackey, R., "Generalized Cross-Signal Anomaly Detection on Aircraft Hydraulic System", *The 2001 IEEE Aerospace Conference,* Big Sky, MT, March 2001.

[8] Kohonen, T., Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[9] P. K Simpson, "Fuzzy Min-Max Neural Networks - Part 1: Classification," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5., 1992.

[10] P. K. Simpson, "Fuzzy Min-Max Neural Networks - Part 2: Clustering," *IEEE Trans. on Fuzzy Systems*, Vol. 1 No. 1., 1993.

[11] P. Simpson and T. W. Brotherton, "Fuzzy Neural Network Machine Prognosis," *AeroSense, The SPIE's Int'l Symposium on Aerospace / Defense Sensing & Control and Dual-Use Photonics*, Orlando, FL April 1995.

Tom Brotherton received his B.S. degree from Cornell University (1974), an M.A.Sc. from the University of Toronto (1976) and the Ph.D. from the University of Hawaii (1982) all in electrical engineering. He was an assistant professor in the Information and Computer Sciences Department at the University of Hawaii in 1983 and with the Orincon Corp from 1983-1999. He is currently a VP of the Intelligent Automation Corp. (IAC) in San Diego, CA. IAC is involved with the development of aircraft and aircraft equipment monitoring software and hardware. Dr. Brotherton is also on the editorial board for the IEEE Press series on Biomedical Engineering. His interests are in the development of adaptive signal and data fusion techniques for machine condition, fault monitoring, and prognostics as well as medical systems applications.

Tom Johnson received his B.S.E.E. degree from the United States Naval Academy (1984) and his M.S.E.E. degree from the Naval Postgraduate School (1992). Tom served in the U.S. Navy until 1995 with tours on the USS Long Beach (CGN-9), USS Ainsworth (FFT-1090) and USS Enterprise (CVN-65). Following his service time, he worked with Orincon Corp. from 1995 – 1999 and Logicon Information Solutions from 1999-2000. Presently Tom works with Intelligent Automation Corp. (IAC). IAC is involved with the development of aircraft monitoring software and hardware. His current interests are neural network applications for machine conditioning and fault monitoring.