

# Net-Centric Surveillance Tolerance Band Reduction Potential

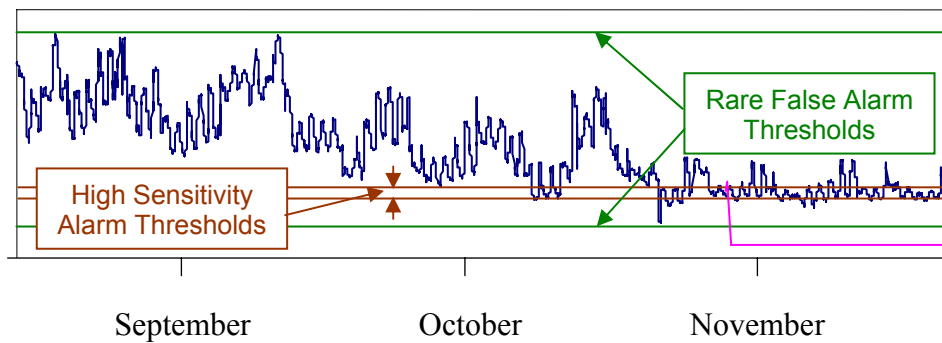
Copyright © 2004 by Brainlike Surveillance, Inc.

## Data-Based Surveillance Challenges

Data-based surveillance entails observing one or more data values at a time, comparing them to expected values, and producing an alert if they are unexpected. Deciding how to proceed after an alert falls outside the surveillance scope of this report.

Key challenges for data-based surveillance include (a) identifying analytic procedures for determining alarm thresholds that are accurate, and (b) making them efficient, automated, and affordable. The simplest conventional analytic methods for determining alarm thresholds involve gathering a sample that contains typical values for measurements to be monitored, then calculating statistics such as means and standard deviations for each measurement, and then calculating fixed alarm thresholds based on such statistics. The simplest and most commonly used such thresholds are the mean value plus or minus a constant times the standard deviation value. Unfortunately, these methods seldom yield results that precisely and quickly identify real problems, while at the same time producing very few false alarms [1-25].

Figure 1 illustrates a case in point [20,21]. The figure shows electricity meter values measured every 15 minutes over a three month period. Surveillance for this kind of data might include being on the lookout for meter excursions that might be caused by an electrical malfunction, such as an insulation breakdown. Such an excursion is shown around the November time frame in the figure. If surveillance were effective, the problem would be identified and resolved before major damage occurred. For example, the sharp positive spike shown in the excursion occurred over a two-hour period in the figure, during which the problem might have been found and resolved. The steep negative spike reflects the kind of major breakdown that would follow if the problem had not been found. The two sets of thresholds shown in the figure illustrate the drawback of fixed threshold monitoring. The broader set would be hopelessly insensitive to developing problems, while the narrower set or would generate many false alarms to the point of being useless. Similar problems with fixed threshold monitoring exist in a variety of other applications.



**Figure 1. Typical Time Series Surveillance Challenge**

In some applications, trained experts can supplement data-based surveillance to the point of adding substantial precision. For example an electricity monitoring expert paying attention to the excursion in Figure 1 might well have recognized the developing problem in time to have initiated preventive action. However, the difficulty with using trained experts is that training them and deploying enough of them to keep track of all monitored measurements can be prohibitively expensive.

### Net-Centric, Auto-Adaptive Surveillance Prospects

Net-centric surveillance has been defined as a process that (a) receives two or more input sensor values via a computer network, (b) correlates current and recent sensor values to produce warning signals, and (c) displays the warning signals for initiating preventive action [26]. By using these correlations, net-centric surveillance can very precisely estimate the expected value of any given sensor output at any given time. In turn, these estimates can provide the basis for determining very precise threshold values.

Auto-adaptive surveillance continuously and automatically adapts alarm thresholds to changing conditions. When used in conjunction with net-centric surveillance, auto-adaptive methods achieve profound improvements in surveillance accuracy. The previously described meter monitoring setting is an excellent case in point. Along with the single meter readings shown in Figure 1, over 40 other meter readings were also observed every 15 minutes. Figure 2 shows the tolerance bands that auto-adaptive surveillance produced, along with the same excursion that was shown in Figure 1. As shown in Figure 2, the tolerance bands were sufficiently narrow to detect the excursion immediately. Moreover, the tolerance bands were sufficiently precise to produce a tolerable number of false alarms over the entire three month period [20].

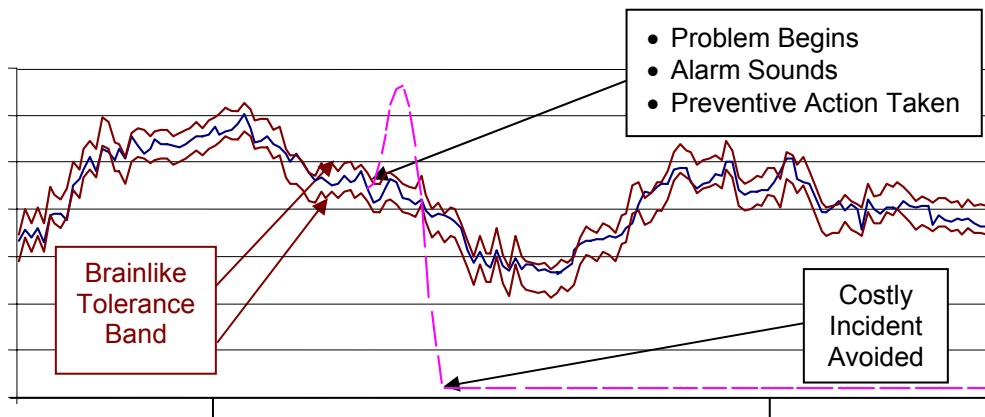


Figure 2. Auto-Adaptive Time Series Tolerance Bands

Results like those in Figure 2 have been obtained for many different surveillance applications. The dramatic auto-adaptive surveillance improvements shown in Figure 2 have been demonstrated in application performance monitoring, structural testing, and acoustical sensing applications to cite a few [1,18,19].

### **Auto-Adaptive Surveillance Basics**

The following explanation is an excerpt from [22], which in turn summarizes an extensive research and development bibliography [1-17]. The auto-adaptive technology illustrated in this report adapts to changing conditions automatically and continuously. Monitoring systems based on the technology produce alarms only when observed sensor levels differ significantly from their expected values. Expected values are updated continuously in order to reflect changing conditions. By adjusting expected values continuously, the technology reduces false alarms and reduces information to its monitoring essence. It also operates automatically, removing the need for re-tuning.

Resulting precision improvements and cost reductions have prompted significant venture funding for commercial development of software products that monitor computer servers. These products, which are currently satisfying corporate server farm customers, receive multiple performance indicators arriving in arrays containing up to several hundred values at each arrival time. Arrival times for each array may be separated by a few seconds or less.

Along with its proven advantages when implemented as software, the technology offers larger — thus far unexploited — advantages when implemented as hardware [1-2]. Kernel technology is based on an algorithm that is compact, separable, fast, and designed for on-chip implementation [3-4]. When implemented in hardware, the kernel algorithm can receive multiple sensor outputs arriving as arrays, just as in software. However, hardware arrival times for each array can be only a few microseconds or less.

When run in either software or hardware, the kernel algorithm performs three basic operations between each sensor array arrival time. First, it computes expected sensor array values. Second, it determines if the sensors' expected values differ significantly from their actual values, and produces alarm signals accordingly. Third, and most important, the kernel algorithm updates its learned parameters efficiently.

Efficient, real-time learning operation distinguishes this technology from all known alternatives. Learning is most important because it allows monitoring and control systems to operate more effectively by adapting to changing conditions. Efficient learning is especially important for data reduction at the micro-electronic level, because it can be implemented on a chip for fast, compact, and rugged operation. Data reduction at the micro-electronic level is critically important because sensor array transmission capacity is often limited and inter-chip transmission capacity is extremely limited.

Learning is also important for control at the micro-electronic level, because it can be tailored to suit numerical optimization. When implemented in this way, the kernel

algorithm can identify optimal parameter values far faster than conventional methods. Related applications include electronic antenna and sensor array pointing, among many others inside and outside the shallow water intrusion detection domain.

The auto-adaptive technology illustrated in this report resembles regression analysis in that it computes the expected value of each monitored variable as a function of all other current values of other monitored variables. It resembles auto-regressive, moving average analysis in that it also uses recent values to compute expected values. It also resembles empirical Bayes methods in that it updates learned parameters by combining current information with prior information at each time point. It resembles Davidon-Fletcher Powell numerical optimization as well, in that it operates efficiently on the inverse of correlation matrices instead of requiring conventional matrix inversion. The technology combines these features to produce a robust, efficient, and fully automated monitoring process.

When used in this way, auto-adaptive technology can add monitoring value by increasing estimation precision in two distinct ways. First, each imputed value for each input is computed not merely as a function of recent values for that input only, as in typical time series applications. Instead, each imputed value is calculated as a function of other recent and current input values. Second, and more important, at each time point an auto-adaptive kernel continuously and automatically updates learned estimation model parameters including means, correlations, regression weights, and deviance metrics.

Kernel operation is fast and compact. When implemented in software, any given kernel can receive 100 inputs, produce estimates, and update learned parameters in less than 100 milliseconds, and over 5,000 kernel models can reside on a conventional server. When implemented on chips, kernels will operate orders of magnitude more quickly and reside in orders of magnitude less space.

The enabling technology has been developed to solve a variety of technical problems by performing certain technical correction functions automatically, without which auto-adaptive operations could not be sustained. Some functions correct for developing linear redundancies and related numerical problems. Others automatically identify and replace deviant and missing values. Others automatically reduce recent input values to smooth trend features that will not produce estimation excursions.

As part of successful efforts to develop the enabling technology into software products, a broad variety of such problems have already been solved. In the process, a scalable platform for hardware development, simulation, and testing has evolved as well. While other problems will emerge as part of developing operational hardware, extensive software development experience to date on essentially the same algorithms offers a distinct advantage.

## Net-Centric Surveillance Tolerance Band Reduction Factors

$\chi$	Number of Sensors per Array as well as the Number of Arrays ( $N$ )							
	2	4	8	16	32	64	128	256
<b>0.01</b>	1.00	1.00	0.99	0.95	0.88	0.79	0.66	0.53
<b>0.05</b>	1.00	0.97	0.88	0.76	0.62	0.48	<b>0.36</b>	0.26
<b>0.10</b>	0.98	0.91	0.77	0.62	0.48	0.35	0.26	0.18
<b>0.30</b>	0.89	0.69	0.51	0.37	0.26	0.19	0.13	0.10
<b>0.36</b>	0.85	0.63	0.45	0.33	0.23	0.17	0.12	0.08
<b>0.50</b>	0.75	0.51	0.36	<b>0.25</b>	0.18	0.13	0.09	0.06
<b>0.60</b>	0.66	0.43	0.30	0.21	0.15	0.10	0.07	0.05
<b>0.70</b>	0.57	0.36	0.24	0.17	0.12	0.08	0.06	0.04
<b>0.80</b>	0.46	0.28	0.19	0.13	0.09	0.06	0.04	0.03
<b>0.90</b>	0.32	0.19	0.13	0.09	0.06	0.04	0.03	0.02
<b>0.95</b>	<b>0.22</b>	0.13	0.09	0.06	0.04	0.03	0.02	0.01
<b>0.99</b>	0.10	0.06	0.04	0.03	0.02	0.01	0.01	0.01

**Figure 3. Net-Centric Tolerance Band Reduction Factors**

Figure 3 shows the extent that tolerance bandwidths could be reduced if multiple sensors and multiple sensor arrays were used in a network-centric configuration system. Each non-gray cell in the table is the proportionate tolerance band reduction that would be achieved if  $N$  arrays were monitored in a field, each array were made up of  $N$  component sensors, and each pair of sensors had a product-moment correlation coefficient of  $\chi$ . For any such cell, the value of  $N$  is the cell's column label and the value of  $\chi$  is that cell's row label.

As the Figure 3 values show, (a) including only two sensors in a surveillance network can dramatically reduce tolerance bandwidths when correlations are high (see the black cell in the lower left); (b) including many sensors can dramatically reduce tolerance bandwidths even when correlations are low (see the black cell in the upper right); and (c) using moderate numbers of sensors can dramatically reduce tolerance bandwidths when correlations are moderate (see the black cell in the middle).

### Mathematical Derivations

The tolerance band reduction values in Figure 3 are based on combining and comparing sensor values in certain ways, and they are based on simplifying assumptions about the ways the sensor values are inter-correlated. This section specifies the ways that sensor values would be combined and compared to produce the tolerance band reduction values shown in Figure 3. Also addressed are the assumed correlational structure, and development of the necessary formulas for computing the numerical data presented in Figure 1. Material presented in this section is available in greater detail elsewhere [1-25].

The Figure 3 values are based on assuming that each pair of sensors under consideration has exactly the same product-moment correlation value, which is denoted by  $\chi$  in this section. The values shown in the first column of Figure 3 are these same  $\chi$  values, and they are used along with the formulas derived below to compute all other Figure 3 entries. This assumed correlational structure is well known in the statistical literature as compound symmetry, among other terms, and in the psychometric literature as parallel structure, along with other designations C.

The Figure 3 values are also based on assuming that several composite sensor array values would typically be computed and compared with estimated composite values. These estimated composite values, in turn, are computed as functions of other observed composite array values. In general, the problem of identifying the best possible composites for comparing two or more arrays is not simple. Matters become especially complicated when nonlinear relationships exist. The same is true even when correlations are strictly linear, but compound symmetry is not satisfied. However, compound symmetry simplifies matters greatly, and it satisfies the needs of this paper for demonstrating how precision can be increased as the number of sensors per array and the number of arrays increases. Indeed, when compound symmetry is satisfied simple derivations indicate that optimal composites for comparison may be obtained simply by summing the sensor values in any given array to produce a single composite array value. Equally straightforward analysis shows that optimal expected values for any given sensor array composite can be obtained by simply accumulating composite values for all other sensor array composites.

The Figure 3 data are also restricted to the relatively simple case of discovering an anomaly in a single array such that each sensor in the array would be perturbed in the same way, but other sensors in other arrays would not be perturbed at all. One notices that the Figure 3 values show dramatic increases in monitoring sensitivity under these conditions, which are realistic in some settings. However, it should be noted that if all sensors in all arrays were perturbed in the same way, which may be realistic in other settings, then no such increases in sensitivity would be expected by comparing composite sensor arrays with each other.

Two derivations are provided below. The first shows the extent to which a tolerance band can be reduced for one array by using estimated values from two or more parallel arrays. The second derivation indicates the extent that inter-array correlations can be increased by increasing the number of component sensors in each array. In the first derivation, the number of component sensors in a sensor array is denoted by  $j^{[1]+}$ . In keeping with Figure 3, which is based on assuming that the number of sensors in each array is the same as the number of sensor arrays being compared,  $j^{[1]+}$  is also used in the second derivation to denote the number of sensor arrays being compared. In cases where distinct counts of sensor arrays and sensors per array are to be considered, these same derivations would be applicable, except that distinct  $j^{[1]+}$  values are needed in that situation.

Predicting one sensor array from other parallel sensor arrays.

(Array orders are shown in subscript parentheses below, all one-dimensional arrays with T superscripts are column vectors, and all other one-dimensional arrays are row vectors.)  
For

$$\mathbf{J}_{(j^*)} = (\mathbf{J}_{(j^{1+})}^{[1]}, \mathbf{J}_2),$$

having covariance matrix

$$\boldsymbol{\chi} = \begin{bmatrix} 1 & \chi & \cdots & \chi \\ \chi & 1 & \cdots & \chi \\ \vdots & \vdots & \ddots & \vdots \\ \chi & \chi & \cdots & 1 \end{bmatrix}, \quad (1)$$

let

$$\mathbf{M}_{(2)} = \mathbf{J} \mathbf{c},$$

where

$$\mathbf{c} = \begin{bmatrix} \mathbf{1}_{(1 \times j^{1+})}^T & \vdots & \mathbf{0}_{(1 \times j^{1+})}^T \\ \cdots & & \cdots \\ 0 & \vdots & 1 \end{bmatrix}. \quad (2)$$

Then  $\mathbf{M}$  has covariance matrix,

$$\begin{aligned}
\mathbf{v}^{[M]} &= \mathbf{c}^T \boldsymbol{\chi} \mathbf{c} \\
&= \mathbf{c}^T \begin{bmatrix} 1 + (j^{[1]+} - 1)\chi & \chi \\ \vdots & \vdots \\ 1 + (j^{[1]+} - 1)\chi & \chi \\ j^{[1]+} \chi & 1 \end{bmatrix} \\
&= \mathbf{c}^T \begin{bmatrix} 1 + (j^{[1]+} - 1)\chi \mathbf{1}^T & \vdots & \chi \mathbf{1}^T \\ \dots & \dots & \dots \\ j^{[1]+} \chi & \vdots & 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{1}_{(j^{[1]+})} & \vdots & 0 \\ \dots & \dots & \dots \\ \mathbf{0}_{(j^{[1]+})} & \vdots & 1 \end{bmatrix} \begin{bmatrix} 1 + (j^{[1]+} - 1)\chi \mathbf{1}^T & \vdots & \chi \mathbf{1}^T \\ \dots & \dots & \dots \\ j^{[1]+} \chi & \vdots & 1 \end{bmatrix} \\
&= \begin{bmatrix} j^{[1]+} [1 + (j^{[1]+} - 1)\chi] & j^{[1]+} \chi \\ j^{[1]+} \chi & 1 \end{bmatrix},
\end{aligned}$$

so that its correlation matrix is

$$\begin{aligned}
\boldsymbol{\chi}^{[M]} &= \tilde{\mathbf{D}}(\mathbf{v}^{[M]})^{-1/2} \mathbf{v}^{[M]} \tilde{\mathbf{D}}(\mathbf{v}^{[M]})^{-1/2} \\
&= \begin{bmatrix} 1 & \frac{j^{[1]+} \chi}{\{j^{[1]+} [1 + (j^{[1]+} - 1)\chi]\}^{-1/2}} \\ \frac{j^{[1]+} \chi}{\{j^{[1]+} [1 + (j^{[1]+} - 1)\chi]\}^{-1/2}} & 1 \end{bmatrix}.
\end{aligned} \tag{3}$$

Thus, when  $j^{[1]+} = 1$ ,  $\boldsymbol{\chi}_{2,1}^{[M]} = \chi$ , as expected. To see how  $\boldsymbol{\chi}_{2,1}^{[M]}$  behaves asymptotically, note that it may also be expressed as



$$\begin{aligned}\chi_{2,1}^{[M]} &= \frac{\chi}{\{[1 + (j^{[1]+} - 1)\chi]/j^{[1]+}\}^{1/2}} \\ &= \frac{\chi}{[(1 - \chi)/j^{[1]+} + \chi]^{1/2}},\end{aligned}$$

which clearly approaches  $\chi^{1/2}$  as  $j^{[1]+}$  approaches infinity.

Finally, to see how tolerance bandwidth depends on  $j^{[1]+}$ , note that the standard error of  $J_2$  given  $M_1$  is

$$v_{2,1}^{[M]1/2} = (1 - \chi_{2,1}^{[M]2})^{1/2} v_2^{[M]1/2},$$

so that the tolerance bandwidth reduction proportion is

$$\frac{v_2^{[M]1/2} - v_{2,1}^{[M]1/2}}{v_2^{[M]1/2}} = 1 - (1 - \chi_{2,1}^{[M]2})^{1/2}. \quad (4)$$

Thus, the level of tolerance bandwidth reduction for any given  $j^{[1]+}$  value can be obtained by first calculating  $\chi_{2,1}^{[M]}$  as a function of  $j^{[1]+}$  in conjunction with equation (3) and then calculating the equation (4) as a function of  $\chi_{2,1}^{[M]}$ .

#### Correlation coefficients between two composite sensor array values.

Let

$$\mathbf{J}_{(j^{[P]+})}^{[P]} = (\mathbf{J}_{(j^{[1]+})}^{[1]}, \mathbf{J}_{(j^{[1]+})}^{[2]}),$$

have covariance matrix

$$\boldsymbol{\chi}^{[P]} = \begin{bmatrix} \chi & \vdots & \mathbf{1}^T \mathbf{1} \chi \\ \dots & & \dots \\ \mathbf{1}^T \mathbf{1} \chi & \vdots & \chi \end{bmatrix},$$

where  $\boldsymbol{\chi}$  has the same parallel form as equation (1), and let

$$\mathbf{M}_{(2)}^{[P]} = \mathbf{J}^{[P]} \mathbf{c}^{[P]},$$

where

$$\mathbf{c}^{[P]} = \begin{bmatrix} \mathbf{1}_{(1 \times j^{[1]+})}^T & \vdots & \mathbf{0}_{(1 \times j^{[1]+})}^T \\ \dots & & \dots \\ \mathbf{0}_{(1 \times j^{[1]+})}^T & \vdots & \mathbf{1}_{(1 \times j^{[1]+})}^T \end{bmatrix}.$$

Then

$$\begin{aligned} \mathbf{v}^{[P]} &= \mathbf{c}^{[P]T} \chi \mathbf{c}^{[P]} \\ &= \mathbf{c}^{[P]T} \begin{bmatrix} [1 + (j^{[1]+} - 1)\chi] \mathbf{1}^T & \vdots & j^{[1]+} \chi \mathbf{1}^T \\ \dots & & \dots \\ j^{[1]+} \chi \mathbf{1}^T & \vdots & [1 + (j^{[1]+} - 1)\chi] \chi \mathbf{1}^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{1}_{(j^{[1]+})} & \vdots & \mathbf{0}_{(j^{[1]+})} \\ \dots & & \dots \\ \mathbf{0}_{(j^{[1]+})} & \vdots & \mathbf{1}_{(j^{[1]+})} \end{bmatrix} \begin{bmatrix} [1 + (j^{[1]+} - 1)\chi] \mathbf{1}^T & \vdots & j^{[1]+} \chi \mathbf{1}^T \\ \dots & & \dots \\ j^{[1]+} \chi \mathbf{1}^T & \vdots & [1 + (j^{[1]+} - 1)\chi] \chi \mathbf{1}^T \end{bmatrix} \\ &= \begin{bmatrix} j^{[1]+} [1 + (j^{[1]+} - 1)\chi] & (j^{[1]+})^2 \\ (j^{[1]+})^2 \chi & j^{[1]+} [1 + (j^{[1]+} - 1)\chi] \end{bmatrix}. \end{aligned}$$

Thus,

$$\begin{aligned} \chi_{2,1}^{[P]} &= \frac{(j^{[1]+})^2 \chi}{j^{[1]+} [1 + (j^{[1]+} - 1)\chi]} \\ &= \frac{\chi}{\chi + (1 - \chi)/j^{[1]+}}, \end{aligned} \tag{5}$$

which clearly approaches 1 as  $j^{[1]+}$  approaches infinity.

Finally, the level of tolerance bandwidth reduction for any given  $j^{[1]+}$  value can be obtained by first calculating  $\chi_{2,1}^{[P]}$  as a function of  $\chi$  in equation (5), then replacing using that value for  $\chi$  in the previous section, and proceeding as indicated at the end the previous section.

## References

(Most of these references can be accessed via [www.Brainlike.com](http://www.Brainlike.com).)

1. R.J. Jannarone, *Concurrent Learning and Information Processing: A Neuro-computing System that Learns during Monitoring, Forecasting, and Control*. Chapman & Hall, New York, 1997.
2. R.J. Jannarone, *Concurrent Learning and Performance Information Processing System*, U.S. patent #5,835,902, 1998.
3. K. Naik, *A Concurrent Information Processing Parallel Design Analysis*, Unpublished Masters Thesis, University of South Carolina, 1996.
4. R.J. Jannarone, "Conjunctive Item Response Theory Kernels," *Psychometrika*, Vol. 51, pp. 449-460, 1986.
5. R.J. Jannarone, "Locally dependent models: conjunctive item response theory," In W.J. van der Linden & R.K. Hambleton III (Eds.), *Handbook of Modern Item Response Theory*, Springer-Verlag, New York, pp. 465-480, 1996.
6. R.J. Jannarone, K.F. Yu, and J.E. Laughlin, "Easy Bayes Estimation for Rasch Type Models," *Psychometrika*, Vol. 55, pp. 449-460, 1990.
7. R.J. Jannarone, K.F. Yu, and Y. Takefuji, "Conjunctoids: Statistical Learning Modules for Binary Events," *Neural Networks*, Vol. 1, pp. 325-337, 1988.
8. R.J. Jannarone, "Concurrent Information Processing, I: An Applications Overview," *Applied Computing Review*, Vol. I, pp. 1-6, 1993.
9. G.E. Weeks, W.E. Daniel, R.E. Edwards, Jr., S.N. Palakodety, S.S. Joshi, D. Qian, & R.J. Jannarone, "Hollidge Gauge Failure Testing using Concurrent Information Processing Algorithm," *Proceedings of the International Topic Meeting on Nuclear and Hazardous Waste Management*, American Nuclear Society, Inc., La Grange Park, IL, Vol. 1, 1998.
10. K. Ma, *Statistical Neural Network Models: Theory and Implementation for Visual Pattern Recognition*, Unpublished Doctoral Dissertation, University of South Carolina, 1991.
11. S.N. Joshi, *A General-Purpose Concurrent Information Processing Prototype*, Unpublished Masters Thesis, University of South Carolina, 1994.
12. S.S. Palakodety, *A Special-Purpose Concurrent Information Processing System*, Unpublished Masters Thesis, University of South Carolina, 1994.
13. Y. Hu, "Automated Real-Time Neural Computing for Defense Waste Processing," *Proceedings of the International Topic Meeting on Nuclear and Hazardous Waste Management*, American Nuclear Society, Inc., La Grange Park, IL, Vol. 1, pp. 534-540, 1992.
14. R.J. Jannarone, *Multi-Kernel Neural Network Concurrent Learning, Monitoring, and Forecasting System*, U.S. patent #6,216,119, 2001.
15. R.J. Jannarone, *Concurrent Learning and Performance Information Processing System*, U.S. patent #6,289,330, 2001.
16. J. Voorhees. "The Limits on Wireless Security: 802:11 in early 2002," SANS Institute (<http://rr.sans.org/wireless/limits.php>), January 30, 2002.
17. E. Amoroso, *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*. Intrusion.Net Books, 1999.

18. "Brainlike Condition Monitoring," published at [www.Brainlike.com](http://www.Brainlike.com).
19. "Brainlike Shallow Water Intrusion Detection," published at [www.Brainlike.com](http://www.Brainlike.com).
20. "Brainlike Advantage," published at [www.Brainlike.com](http://www.Brainlike.com).
21. "Brainlike Monitoring Improvement Illustration," published at [www.Brainlike.com](http://www.Brainlike.com).
23. "Brainlike Enabling Technology Basics," published at [www.Brainlike.com](http://www.Brainlike.com).
24. "Brainlike Return on Investment Analysis," published at [www.Brainlike.com](http://www.Brainlike.com).
25. "Network-Centric Surveillance Short Course," published at [www.Brainlike.com](http://www.Brainlike.com).
26. "Net-Centric Surveillance Course." [www.Brainlike.com](http://www.Brainlike.com).
27. F.A. Graybill, *Matrices with Applications in Statistics*, 2nd Edn., Wadsworth International, Belmont, CA, 1983.
28. F.M. Lord & M.R. Novick, *Statistical Theories of Mental Tests*. Wiley, New York, 1968.