Journal of Korea TAPPI
Vol. 33. No. 5, 2001
Printed in Korea

Application of Artificial Neural Networks to Diagnosis of Web Breaks in Papermaking

Jin Hee Lee and Hak Lae Lee†

1. Introduction

Paper mill is a large capital-intensive unit, therefore, it is prime requisite that throughput per unit time should be increased and product quality should also be high and stable. Improvement of production for a paper mill is evaluated by productivity and the uniformity of products. Quality uniformity can be evaluated through quality control. Process stability and process runnability for a papermaking process, however, is rather elusive but very important. There are many disturbing reasons for process stability, and many of them have their roots in web breaks, fluctuations in wet end, frequent paper grade change, shut down and startup and so on. Moreover, there are many environmental registrations that cause deterioration of production efficiency which include reduction of fresh water consumption and effluent discharge and contamination of the process white water. Among these reasons, web breaks is the most significant cause for the loss of process stability and production efficiency for the paper machine.

Web break is an ever-serious problem in many paper mills and complex phenomenon with several causes that can vary considerably from one process to another. A web break on the paper machine severely impairs production efficiency and products quality. Production loss is the principal concern during sheet break, but there are also extra works required to clean, rethread, and restart the paper machine. Bringing the machine back on line consumes a great deal of time and raw materials.[1] The reuse of large amount of broke after web breaks impair also products quality badly.

Reducing the number of web breaks on paper machine, therefore, is one of the largest efficiency improvement methods available to today' s papermaker. According to a research, a reduction of just one 15-minute break a day can affect profitability substantially, if daily production is 800 tons/day and virgin pulp price is 400 $/ton, this can save as much as $1 million annually per paper machine.[2]

Thus, it becomes essential to understand the dynamics of paper breaks and to introduce a realistic model for analyzing these breaks. A characteristic feature of paper breaks, however, is the virtual impossibility of predicting their exact time of occurrence and their duration. Therefore, it is not possible to conceive a simple mathematical model that would predict the breaks. Instead, if more reliable main causes are revealed, counter measures for web breaks reduction can be planed easily.

• Department of Forest Products, College of Agriculture and Life Sciences, Seoul National University, Suwon 441-744, Republic of Korea.
† Corresponding author: e-mail: lhakl@plaza.snu.ac.kr

Artificial neural networks are relatively new data analysis tools and have applications in areas with great complexity. They are best employed when the underlying fundamental equations are either not known or too difficult to use. Typical early examples involve modeling the prices of stocks and shares or controlling complex chemical processes. Nowadays neural networks are being used in investment analysis, process control, signature analysis, image recognition, monitoring on several industries and so on.[3]

Artificial neural networks are mathematical models capable of performing non-linear statistical calculations and can be thought of as similar to multiple correlations. Neural networks can be used to find relationships in data, and neural logic can learn and extract the essence of the relationships among the inputs and outputs from the data supplied to it during the training process.

2. Overview of the Web Break Problem

Web breaks are very infrequent events. Even at a low rate, however, web breaks cause significant losses in production. The low frequency of breaks also makes the evaluation of the runnability of a particular paper grade difficult.

A sequence of breaks may be characterized by their average duration and frequency. Break duration is the time elapsed between the moments when its continuity is restored in the paper machine. This duration, varying from a few minutes to hours, depends on the break type. Generally wet breaks have longer duration and cause more severe problem than dry breaks.

Many studies[4-7] have indicated that web breaks are due to distinct flaws or defects in the paper web. Especially for papers made of mechanical furnish, shives and the like create such defects. Therefore, good pulp screening is essential to improve runnability of newsprint machine. Disturbances in the wet end chemistry of the paper machine, edge cuts, and other physical defects are also common causes of breaks. In paper coating and printing, water and heat soften paper. Low moisture content makes paper brittle. Water can cause breaks by wetting flaws that would otherwise be harmless.

The break frequency seems to depend on the basis weights.[8] As breaks occur in the open draws where the sheet is not supported, great efforts have been devoted to developing means to reduce or to close the open draws, not only in new machines but also by equipment retrofit in existing machines. The ultimate goal would be to provide uninterrupted web support from the headbox to the calender.[9]

Web breaks occur when adhesion of the sheet to contacting surface is greater than its cohesiveness. This happens when the local strength is too low somewhere in the web or the momentary load is too high which can be caused by excessive operating tension, fluctuating tension control, or out-of-round unwind rolls. Similarly, low strength can be caused by any number of conditions, including poor formation or the presence of defects.[3]

The most frequent causes of breaks in the wet end of a machine have been identified as stock quality variations, low web strength, low dryness, lumps, slime, and water drops.[10] However, there is no physically based predictive model to forecast break occurrence for given operating conditions, and despite recent developments on these subjects, anticipating or preventing the triggering event remains an elusive goal. Sheet breaks continue to be a frustrating problem that affects paper machine performance.

Runnability is a term used to denote the expected (mean) frequency of web breaks for a given material under a specified loading condition. Runnability can be measured in units of breaks/100 rolls, breaks/length of sheet, breaks/area of sheet, or breaks/day. From an operations viewpoint, breaks/100 rolls or breaks/day are the most convenient measures since production is often evaluated in rolls or units of time. The most appropriate independent measure of runnability is based on web area, which is easily converted to other units for management reports. Roisum[5-6] had studied the level of statistical confidence in laboratory runnability and troubleshooting and diagnosis of web breaks.

A runnability problem composed of numerous individual breaks will have multiple causes of varying severity. If runnability is to be improved, a wide range of diagnostic skills and analysis tools must be used to shift through the complex interaction of multiple contributing problems. Lindström *et al.*[10] researched to cope with an avalanche of breaks by logic tree on a high-speed fine paper machine. A paper machine operating at less than desirable efficiency is often affected by a sudden increase of break frequency. Two reasons can be discerned for such an occurrence: firstly, some of the elementary rules of papermaking may be habitually disregarded. Secondly, the elementary principles of troubleshooting may be neglected. All successful troubleshooters worked by the logic diagrams, whether these are written out or are from experience.

A probabilistic model using Markov chain[11] has recently been proposed on the basis of correlations between paper break characteristics, machine speed, and broke reuse rate, derived from a large set of actual mill data. Khanbaghi *et al.*[12] have performed a statistical analysis of paper break data from a typical paper mill. They have shown that the paper break process could be reasonably modeled as a continuous Markov chain with three states, an operating state, a failure state of type one resulting from breaks in the wet area, and a failure state of type two resulting from breaks in the dry area of the paper mill. They used a more restricted set of data, and subsequently established an empirical dependency of failure rates on machine operating speed and broke recirculation ratio. Miyanishi and Hirotaka[1] had used a neural network diagnosis of web breaks on a newsprint machine in combination with results from principal component regression analysis suggested that web breaks could be reduced by manipulating wet-end chemistry variables.

3. Artificial Neural Networks

Neural networks have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced.

Biological nervous systems and mathematical theories for learning have inspired neural networks. The brain consists of a large number (approximately 10^{11}) of highly connected elements (approximately 10^4 connections per element) called neurons. These neurons have three principal components: the dendrites, the cell body (soma) and the axon. The dendrites (a branching input structure) are tree-like receptive networks of nerve fibers that carry electrochemical signals into the cell body. The cell body effectively sums and thresholds these incoming signals. The axon (a branching output structure) is a signal long fiber that carries the signal from the

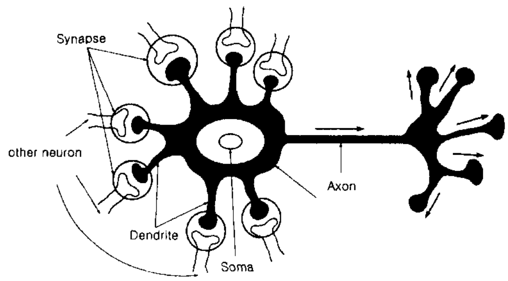

Fig. 1. Schematic drawing of biological neurons.

cell body out to other neurons. The point of contact between an axon of one cell and a dendrite of another cell is called a synapse. Fig. 1. is a simplified schematic diagram of two biological neurons.

The axons of one cell connect to the dendrites of another via a synapse. When a neuron is activated, it fires an electrochemical signal along the axon. This signal crosses the synapses to other neurons, which may in turn fire. A neuron fires only if the total signal received at the cell body from the dendrites exceeds a certain level (the firing threshold).

The strength of the signal received by a neuron (and therefore its chances of firing) critically depends on the efficacy of the synapses. Each synapse actually contains a gap, with neurotransmitter chemicals poised to transmit a signal across the gap. Hebb[3], who is one of the most influential researchers into neurological systems, postulated that learning consisted principally in altering the "strength" of synaptic connections. Thus, from a very large number of extremely simple processing units (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform extremely complex tasks.

There are two key similarities between biological and artificial neural networks. First, the building blocks of both networks are simple computational devices (although artificial neural networks are much simpler than biological neurons) that are highly interconnected. Second, the connections between neurons determine the function of the network.

There is no universally accepted definition of an artificial neural network. But perhaps most people in the field would agree that a neural network is a network of many simple processors ("units"), each possibly having a small amount of local memory. The units are connected by communication channels ("connections"), which usually carry numeric (as opposed to symbolic) data, encoded by any of various means. The units operate only on their local data and on the inputs they receive via the connections. The restriction to local operations is often relaxed during training.

Some neural networks are models of biological neural networks and some are not, but historically, much of the inspiration for the field of neural networks came from the desire to produce artificial systems capable of sophisticated, perhaps "intelligent", computations similar to those that the human brain routinely performs, and thereby possibly to enhance our understanding of the human brain. Most neural networks have some sort of "training" rule whereby the weights of connections are adjusted on the basis of data. In other words, neural networks "learn" from examples (as children learn to recognize dogs from examples of dogs) and exhibit some capability for generalization beyond the training data.

Neural networks normally have great potential for parallelism, since the computations of the components are largely independent of each other. Some people regard massive parallelism and high connectivity to be defining characteristics of neural networks, but such requirements rule out various simple models, such as simple linear regression, which are usefully regarded as special cases of neural networks. It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection which has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the activation of the neuron. The activation signal is passed through an activation function (a transfer function) to produce the output of the neuron. If the step activation function is used (i.e. the neuron' s output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0) then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Actually, the step function is rarely used in artificial neural networks. Those weights may be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain.

If a network is to be of any use, there must be inputs (which carry the values of variables of interest in the outside world) and outputs (which form predictions, or control signals). Inputs and outputs correspond to sensory and motor nerves such as those coming from the eyes and leading to the hands. However, there may also be hidden neurons which play an internal role in the network. The input, hidden and output neurons need to be connected together. A simple network has a feedforward structure: signals flow from inputs, forwards through any hidden units, eventually reaching the output units. Such a structure has stable behavior. However, if the network is recurrent (contains connections back from later to earlier neurons) it may be unstable, and has very complex dynamics. Recurrent networks are very interesting to researchers in neural networks, but so far it is the feedforward structures which have proved most useful in solving real problems.[17]

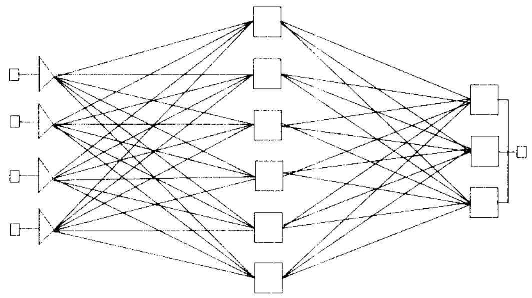

Fig. 2 . A typical feedforward neural network.

A typical feedforward network is shown in Fig. 2. Neurons are arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks which are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

When the network is used, the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

2.1 Using Neural Networks

The type of problem amenable to solution by a neural network is defined by the way they work, and the way they are trained. Neural networks work by feeding in some input variables, and producing some output variables. They can therefore be used where you have some known information, and would like to infer some unknown information. Another important requirement for the use of a neural network therefore is that you know (or at least strongly suspect) that there is a relationship between the proposed known inputs and unknown outputs. This relationship may be noisy but it must exist. In general, if you were using a neural network you won' t know the exact nature of the relationship between inputs and outputs - if you knew the relationship, you would model it directly. The other key feature of neural networks is that they learn the input/output relationship through training. There are two types of training used in neural networks, with different types of network using different types of training.

In supervised learning, the network user assembles a set of training data. The training data contains examples of inputs together with the corresponding outputs, and the network learns to infer the relationship between the two. Training data is usually taken from historical records. The neural network is then trained using one of the supervised learning algorithms (of which the best known example is back propagation), which uses the data to adjust the network' s weights and thresholds so as to minimize the error in its predictions on the training set. If the network is properly trained, it has then learned to model the (unknown) function that relates the input variables to the output variables, and can subsequently be used to make predictions where the output is not known.

If it has decided on a problem to solve using neural networks, data for training purposes will need to gather. The training data set includes a number of cases, each containing values for a range of input and output variables. The first decisions will need to make which variables to use, and how many (and which) cases to gather.

The number of cases required for neural network training frequently presents difficulties. There are some heuristic guidelines, which relate the number of cases needed to the size of the network (the simplest of these says that there should be ten times as many cases as connections in the network). Actually, the number needed is also related to the (unknown) complexity of the underlying function, which the network is trying to model. As the number of variables increases the number of cases required increases nonlinearly, so that with even a fairly small number of variables (perhaps fifty or less) a huge number of cases are required.

Neural networks are also noise tolerant. However, there is a limit to this tolerance: if there are occasional outliers far outside the range of normal values for a variable they may bias the training. The best approach to such outliers is to identify and remove them (either discarding the case, or converting the outlier into a missing value).

3.2. Feature Selection Method in Neural Networks

Artificial neural networks can be used successfully to detect faults in rotating machinery, using statistical estimates of the vibration signal as input features. In any given scenario, there are many different possible features that may be used as inputs for the artificial neural network. One of the main problems facing the use of artificial neural networks is the selection of the best inputs to the artificial neural network, allowing the creation of compact, highly accurate networks that require comparatively little preprocessing.

Yacoub and Bennani[20] proposed a feature selection measure and an architecture optimization procedure for Multi-Layer Perceptrons (MLP). The algorithm presented in that research employs a heuristic measure named Heuristic for Variable Selection (HVS). Their new measure allows us to identify and select important variables in the features space. This can be achieved by eliminating redundant features and those that do not contain enough relevant information. The proposed measure is used in a new procedure aimed at selecting the "best" MLP architecture given an initial structure. Application results for two generic problems: regression and discrimination, demonstrates the proposed selection algorithm' s effectiveness in identifying optimized connectionist models with higher accuracy. Finally, an extension of HVS, named εHVS, is proposed for discriminative features detection and architecture optimization for Time Delay Neural Networks models (TDNN).

Kwak *et al.*[21] presented an automated vision system for detect and classify surface defects on leather fabric by feature selection method. Visual defects in a gray-level image are located through thresholding and morphological processing, and their geometric information is immediately reported. Three input feature sets are proposed and tested to find the best set to characterize five types of defects: lines, holes, stains, wears, and knots. Two multilayered perceptron models with one and two hidden layers are tested for the classification of defects. If multiple line defects are identified on a given image as a result of classification, a line combination test is conducted to check if they are parts of larger line defects. Experimental results on 140 defect samples show that two-layered perceptrons are better than three-layered perceptrons for this problem. The classification results of this neural network approach are compared with those of a decision tree approach. The comparison shows that the neural network classifier provides better classification accuracy despite longer training times.

3.3 Various Applications Using Neural Networks

As artificial neural networks had again vitalized by Kohonen, Hopfield, and Rumelhart early in the 1980' s, several studies had been tried on pulp and paper industry in the 1990' s. Since then, there are various applications of artificial neural networks on pulp and paper industry. Rudd[22] had introduced a neural network system for advanced process control of pulp mill brown stock washer in 1991. After this study,[23-26] he published several papers on the prediction and control of paper machine properties, and pulping process.

Beaverstock and Hinson[27] suggested neural networks as technologies for control applications and claimed that neural networks could be trained to predict that output having its own 'realities.' Dayal *et al.*[28] investigated the use of neural networks and Partial Least Squares (PLS) regression method to build empirical models for Kappa number using historical data from an industrial, continuous Kamyr digester. The neural network results were comparable to PLS result but no insight into the process could be obtained from the neural network models. In terms of predictive ability, the totally empirical PLS and neural network model were similar to other the semi-empirical H factor model.

Scharchanski and Dodson[29] had developed a new simulator for paper forming using neural networks paradigms. The new simulator was influenced by various parameters involved in paper forming, that include fiber concentration, fiber propensity to flocculate, fiber flexibility, as well as parameters describing the forming conditions. The simulation package seems to provide sufficient flexibility to model the normal range of commercial paper structures. This allows interpretations of furnish and process changes through quantitative parameter changes, and an improved range of design tools for the engineering of papers. Qian *et al.*[30] demonstrated that a complex woodchip refining system can be modeled and implemented in a feed-forward neural network. The neural network can learn about the relationships among process variables using mill data from the distributed control system and operator logbooks. Besides this approach does not require the enormous efforts that are required to develop mechanistic models or rule-based expert systems. Therefore, they suggested that a neural network can be successfully used in steady-state modeling and optimization of complex industrial processes.

Sui *et al.*[31] carried out model based pulp quality control of TMP refiner. In their study, neural network modeling technique was used to learn and capture the process characteristics on line in real time. Baines *et al.*[32] studied predicting boiler emissions with neural networks. O' Neill *et al.*[33] investigated the techniques of multiple linear regression and neural networks analysis for seeking out relationships between the paper properties of tensile index and light scattering coefficient and wood, pulp and fiber properties. And then they have found that neural network analysis results in a better-fitted model for data with the R^2 values.

Milosavljevic and Heikkilä[34] applied feed-forward neural networks on modeling a scrubber and resulted that neural networks are efficient at approximating the nonlinearities in the system, which predicts the outlet water temperature from a scrubber. Using the neural network for simulating a scrubber, it is possible to avoid complicated fluid dynamics phenomena of the gas-liquid flow but still to offer good non-classical solutions in predicting operational parameters.

In addition to above studies, several researchers[35-37] have demonstrated artificial neural networks application and their effectiveness on pulp and paper processes. Several researches[38-45] about paper quality prediction have also been studied.

Literature Cited

1. Miyanishi, T. and Shimada, H., TAPPI JOURNAL, 81(9):163-170 (1998).
2. The web site: www.conmark.com.
3. Hagan, M. T., Demuth, H. B., and Beale, M., Neural Network Design, PWS Publishing Company, Chapter 1 (1995).
4. Niskanen, K., Papermaking Science and Technology Book 16 Paper Physics, Finnish Paper Engineers' Association and TAPPI, Chapter 5 (1998).
5. Page, D. H., and Seth, R. S., Tappi, 65(8): 92 (1982).
6. Roisum, D. R, Tappi J., 73(1):97 (1990).
7. Roisum D. R., Tappi J., 73(2):101 (1990).
8. Moilanen, P., and Lindqvist, U., Tappi J., 79(9):88 (1996).
9. Rounsley, R. R., Tappi, 66(9):69 (1983).
10. McDonald, J. D., Daunais, R., Pikulik, I. I., and Pye, I. T., Pulp Paper Canada, 92(3):T62-T66 (1991).
11. Lindström, R., Manfield, W. H., Tkacz, A. F., and Mardon J., APPITA J., 47(2):163-171 (1994).
12. Haykin, S., Neural Networks: A Comprehensive Foundation, NY, MacMillan, 548-556 (1994).
13. Khanbaghi, M., Malhame, R., Perrier, M., and Roche, A., Pulp Paper Canada, 82nd Annual

Meeting Proceedings: A1-A6 (1996).
14. Hopfield, J. J., Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Science, 79:2554-2558 (1982).
15. Rumelhart, D. E., McClelland, J. L., Parallel Distributed Processing, Explorations in the microstructure of cognition, Vol. 1, Cambridge, MA, The MIT Press (1986).
16. DARPA Neural Network Study, AFCEA International Press, p. 60 (1988).
17. Haykin, S., Neural Networks: A Comprehensive Foundation, NY, MacMillan, p. 2 (1994).
18. Nigrin, A., Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press, p. 11 (1993).
19. Zurada, J. M., Introduction To Artificial Neural Systems, Boston, PWS Publishing Company, p. (1992).
20. Yacoub, M., and Bennani, Y., International J. Neural Systems, 10(5): 379-395 (2000).
21. Kwak, C. J., Ventura, J. A., and Karim T. S., J. Intelligent Manufacturing, 11(5):485-499 (2000).
22. Rudd, J. B., Tappi J., 74(10):153-159 (1991).
23. Schweiger, C. A., and Rudd, J. B., 1994 TAPPI Process Control Symposium:145-153.
24. Rudd, J. B., 80th Annual Meeting, Technical Section, CPPA: B169-173 (1994).
25. Rudd, J. B., 1999 TAPPI Process Control Conference Proceedings.
26. Rudd, J. B., 2001 TAPPI Process Control, Electrical and Information Conference & ISA-PUPID 40th Annual Symposium, Session 19.
27. Beaverstock, M. C., and Hinson, M. C., 1993 TAPPI Process Control Conference: 201-205.
28. Dayal, B. S., Macgregor, J. F., Taylor, P. A., Kildaw, R., and Marcikic S., Pulp Paper Canada, 95(1):26-32 (1994).
29. Scharcanski, J., and Dodson, C. T. J., APPITA, 48(5):347-350 (1995).
30. Qian, Y., Tessier, P., and Dumont, G. A., Tappi J., 78(6):167-174 (1995).
31. Sui, O. S., Sanche, L., Mills, C., Smith, W., Smith, G., and Douglas T., 1998 TAPPI Pulping Conference Proceedings: 195-308.
32. Baines, G. H., Hayes, R. L., and Stabell J. L., TAPPI J., Vol.80(5):57-61 (1997).
33. O' Neill, P., Mohiddin, S., and Michell, A. J., 52nd APPITA Annual General Conference Proceedings, Vol. 2: 571-579 (1998).
34. Milosavljevic, N., and Heikkila, P., TAPPI J., 82(3):197-201 (1999).
35. Zhu, J., Xia, Q., and Rao, M., 82nd Annual Meeting, Pulp and Paper Technical Association of Canada: A29-A33 (1996).
36. Zhu, J., Xia, Q., Rao, M., and Zurcher, J., Pulp Paper Canada, 98 (9):35-38 (1997).
37. Aguiar, H. C., Filho, R. M., Pasquali, S. M., and De Jesus, R., 52nd APPITA Annual General Conference Proceedings:447-451 (1998).
38. Sasaki, T., Position correspondence identification algorithm in basis weight profile control, International CD symposium 97 cross directional web measurement, controls and actuator systems in paper machines, Finnish Society of Automation: 139-144 (1997).
39. Edwards, P. J., Murray, A. F., Papadopoulos, G., Wallace, A. R., Barnard, J., and Smith, G., TAPPI J., 82(7):145-151 (1999).
40. Kumar, A., and Hand, V. C., 1999 TAPPI Process Control Conference Proceedings.
41. Sarimveis, H., and Retsina, T., 86th Annual Meeting, Pulp and Paper Technical Association of Canada:A27-A30 (2000).
42. Stephenson, K., 2001 TAPPI Process Control, Electrical and Information Conference Proceedings.
43. Edwards, P. J., and Murray, A. F., 2001 TAPPI Process Control, Electrical and Information Conference Proceedings.
44. Lenz, S., Koerbitz, R., and Rudd, J. B., 2001 TAPPI Process Control, Electrical and Information Conference Proceedings.
45. DeCicco, J., 2001 TAPPI Process Control, Electrical and Information Conference Proceedings.
46. Swingler, K., Applying Neural Networks A Practical Guide, ACADEMIC PRESS, Chapter 3(1996).