

Modeling a scrubber using feed-forward neural networks

NENAD MILOSAVLJEVIC AND PERTTI HEIKKILÄ

GAS-LIQUID FLOWS ARE FOUND in a diverse range of process industries, especially in the paper industry. Paper machines require a large quantity of hot, dry air for the drying process. This air is used for the machine hood ventilation (in conventional paper machines) or directly for paper drying (in a Yankee dryer hood). After the drying process, the mixture of hot air and evaporated water, which has a specific enthalpy several times greater than the air delivered to the drying hood, is conducted through the system of heat exchangers (Fig. 1) called *recuperators*.

The last stage of the system of the heat exchangers usually represents a scrubber. In a counterflow scrubber, the cold process water is sprayed into a humid exhaust air stream flowing upward. Due to heat and mass transfer, the humid exhaust air temperature is reduced while the temperature of the process water is increased. Using a scrubber as a part of the heat recovery process in the paper industry, up to 40% of the waste energy can be utilized (1, 2), and at the same time, emissions of heat and humidity into the atmosphere can be significantly reduced.

In their work, Milosavljevic *et al.* (2) presented a physical model for the scrubber process. Experimental results in their work agree well with the numerical simulation based on a simplified one-dimensional physical model. By combining simple relationships to predict the droplet size

and the one-dimensional model, they offer a promising approach for designing a scrubber for heat recovery from paper machines. Due to the complexity of the subject, they have used a number of simplifying hypotheses. However, further study and improvements are needed. A physical process simulation generally requires the solution of a governing differential equation for a number of different initial and/or boundary conditions. Milosavljevic *et al.* (2) used a set of nonlinear equations, numerically solved using the Runge-Kutta method, knowing the initial conditions and relations needed to estimate the mass and heat transfer coefficient.

In this work, an attempt is made to simulate a physical process using an artificial neural network. The neural networks are useful when the physical phenomena of the process are completely unknown or are too complex for solving with a physical model. When using a neural network, no physical laws are employed. One only needs enough experimental results to make a model using this nonclassical method.

The idea is to use a neural network to predict the outlet temperature of the process water after the scrubber. Based on this information, it is possible to determine other parameters needed to design the scrubber for heat recovery from paper machines. The results of neural network simulations were compared with those obtained by the physical model by Milosavljevic *et al.* (2).

ABSTRACT

Mathematical modeling in the pulp and paper industry is largely based on physical models that contain many empirical correlations and assumptions. A technique that can complement these physical models, and that does not require simplifying assumptions, is a neural network. Neural networks can be used to model systems where physical models are not available or to improve existing physical models with various techniques of hybrid modeling. Such models have an advantage of the robustness of physical models combined with the higher accuracies offered by empirical techniques such as neural networks.

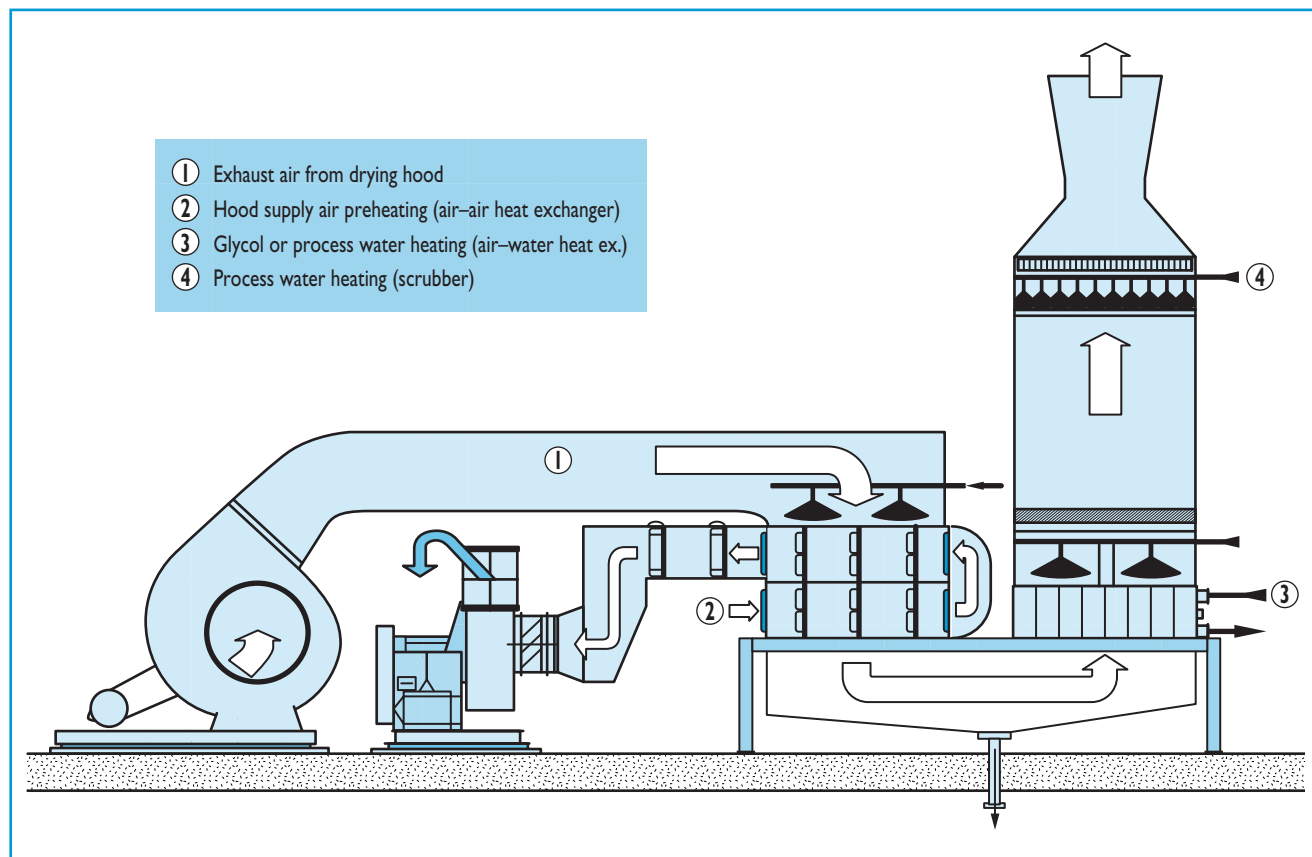
Most systems in paper drying are quite nonlinear and are often too complicated to be accurately described with physical models. Neural networks are powerful tools that can solve a variety of nonlinear modeling problems. Using an example of a scrubber, this paper illustrates the versatility of this technique. Neural networks were used to predict the optimum operational conditions in designing a scrubber for heat recovery from paper machines. In this study, feed-forward neural networks are applied to estimate the outlet water temperature of the scrubber process. The networks are trained with data obtained from experiments carried out on two pilot scrubbers. The accuracy of the neural network model is significantly higher than that of the physical model.

Application:

The neural network is very useful when the physical phenomena of the process are unknown or too complex to be solved with a physical model.

ARTIFICIAL NEURAL NETWORK MODELS FOR THE DESIGN OF A SCRUBBER

In this study, the feed-forward neural network, whose architecture is shown



1. Heat recovery pressurized tower

in Fig. 2, is used to predict the outlet process water temperature from the scrubber process. The neural networks consisted of seven input neurons in the input layer and one output neuron in the output layer. The neural networks have only one hidden layer.

The inputs were mass flow rate of the supply air (x_1), temperature of the supply air (x_2), inlet humidity of the supply air (x_3), mass flow rate of the water (x_4), inlet water temperature (x_5), height of the scrubber (x_6), and water pressure in the nozzle (x_7); the output was the outlet water temperature from the scrubber. The data sets consisted of 259 vectors based on the data obtained from the experimental measurements on two pilot scrubbers carried out in Valmet Pansio, Air Systems, Turku (pilot scrubber no. 1) and the Technical Research Centre of Finland (VTT Energy), Jyväskylä (pilot scrubber no. 2). The network design comprises two layers, one of nonlinear process-

ing units followed by an output layer of linear processing units, with adjustable connection strengths (weights) between them.

An input vector, $x_p = (x_{p1}, x_{p2}, \dots, x_{pN})$, is applied to the input layer of the network (3, 4). The input units distribute the values to the hidden-layer units. Here, different vectors are labeled by the subscript p . The net input to the j th hidden unit is as follows:

$$a_{pj}^h = \sum_i w_{ji}^h x_{pi} + b_j^h \quad (1)$$

where w_{ji}^h is the weight on the connection from the i th input unit to the j th hidden unit, and b_j^h is the bias term of the hidden layer. The bias is a weighted unit input to each neuron, thus adding a constant term to the net input of the node. The superscript h refers to quantities on the hidden layer.

Assuming that the activation of this neuron is equal to the net input, then the output of this neuron is as

follows:

$$\xi_{pj} = \tanh_j^h(a_{pj}^h) \quad (2)$$

The feed-forward neural network was trained with a hyperbolic tangent sigmoid transfer function. The tanh-sigmoid function is a fully differentiable function that allows neurons to be trained with back propagation.

If the output layer has a linear (identity) activation function, as should be the case for function approximation tasks, the output of the network would be given by the following equation:

$$o_{pk} = \sum_j w_{kj}^o \xi_{pj} + b_k^o \quad (3)$$

where w_{kj}^o is the weight on the connection from j th hidden layer to k th output layer, b_k^o represents bias terms of the output layer, and the superscript o refers to the quantities of the output layer.

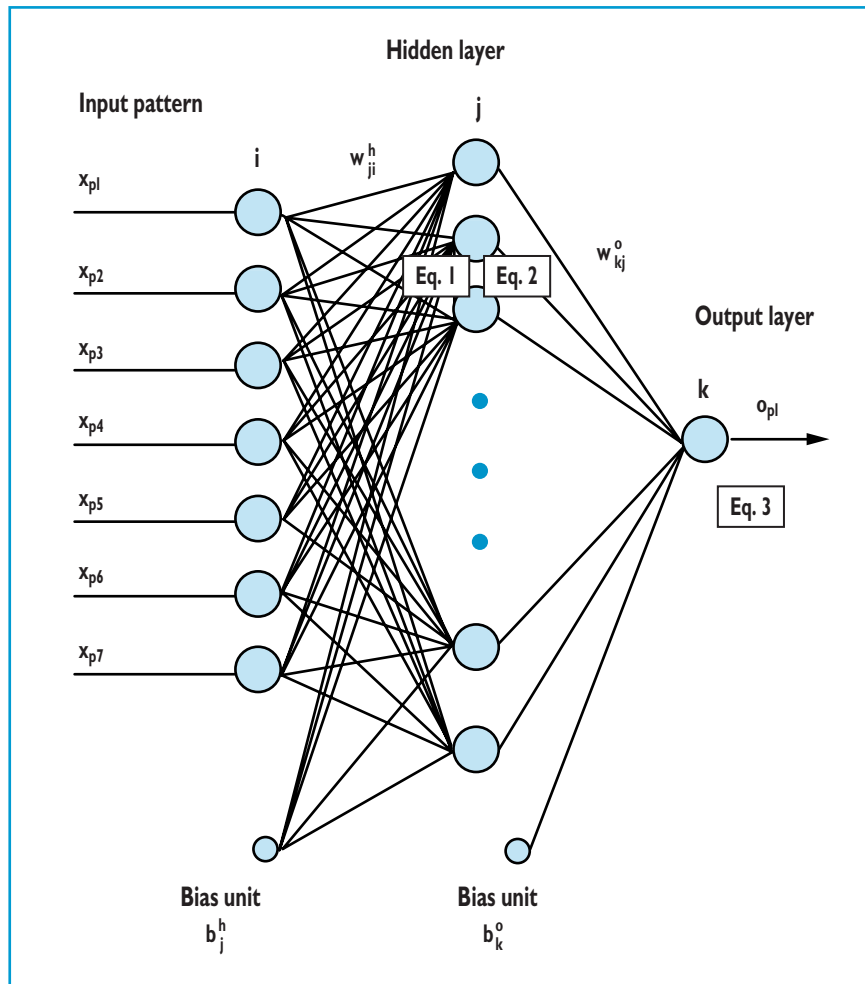
The weight parameters in these equations are found so that they are variables in a numerical method. This numerical method finds values for the weight parameters that produce the best fit of the output of the network to the experimental data.

Neural networks are often trained by the back-propagation method. However, when the training task is considered from an engineering and practical point of view, this task becomes an optimization problem, with the objective of minimizing prediction errors. Thus, it may be considered to be a least-squares regression problem. To solve such optimization problems, there are zeroth-order methods, such as Hooke and Jeeves' method, first-order methods such as the steepest-descent method, second-order methods such as Newton's method, higher-order methods, and a stochastic method such as simulated annealing. Back propagation is an approximation of the steepest-descent method, and therefore even the steepest descent can be expected to work faster in most circumstances (5).

The Levenberg-Marquardt method (6, 7) is used to learn appropriate weights and biases associated with each interconnection. This method uses an interpolation between the approaches based on the maximum neighborhood (a "trust region") in which the truncated Taylor series gives an adequate representation of the nonlinear model.

The limitation of this technique is its large memory requirement compared with first-order techniques. The neural networks were trained by the Levenberg-Marquardt software of the MATLAB Neural Network Toolbox 2.0.

We have adopted the root-mean-squared (rms) normalized error to evaluate discrepancies between predicted and measured values:



2. Schematic of the three-layer, feed-forward neural network

$$\text{rms} = \sqrt{\frac{\sum_{p=1}^P \sum_{k=1}^K (d_{pk} - o_{pk})^2}{PK}} \quad (4)$$

where d_{pk} is the predicted outlet water temperature and o_{pk} is the experimental outlet water temperature from the neural network. The subscript p refers to the p th training vector, and the subscript k refers to the k th output unit. In our case, only one outlet unit was presented: $K=1$ (outlet water temperature from the scrubber), so Eq. 4 can be reduced to the form:

$$\text{rms} = \sqrt{\frac{\sum_{p=1}^{P_{\max}} (T_{w_{nn}} - T_{w_{\exp}})^2}{P_{\max}}} \quad (5)$$

In Eq. 5, the sum-squared error is

represented by:

$$\text{SSE} = \sum_{p=1}^{P_{\max}} (T_{w_{nn}} - T_{w_{\exp}})^2 \quad (6)$$

where

$P_{\max}$ = maximum number of experiments
 $T_{w_{nn}}$ = outlet temperature from the scrubber predicted by nn
 $T_{w_{\exp}}$ = measured outlet temperature from the scrubber.

RESULTS AND DISCUSSION

In this work, neural network models used to simulate the outlet water temperature from the scrubber process were tested. For the neural network training, the data set consists of the results obtained from the experimental measurement, which has been carried out on the two pilot scrubbers. Note that the neural net-

Process parameter	Pilot scrubber no. 1	Pilot scrubber no. 2
Mass flow rate of inlet air, kg/m ² /s	1.0–3.15	0.55–2.15
Temperature of inlet air, °C	51–62	99.3–203.6
Humidity of inlet air kg/kg	0.013–0.139	0.138–0.409
Mass flow rate of water, kg/m ² /s	1.0–3.15	1.053–4.22
Temperature of water, °C	2.2–37.5	5–40
Height of column, m	1.3–5.3	3–5
Nozzle pressure, kPa	100–300	200–300

I. Ranges of process parameters for pilot scrubbers no. 1 and 2

work can be used as a predictive tool only over the range of process variables used in training the network. Interpolation within the ranges covered by the training set produces acceptable results, whereas extrapolation is not possible. The ranges of parameters trained with neural network models in this study are given in **Table I**.

The measurements on the pilot scrubber no. 1 covered the ranges of process parameters for the typical paper and board machines where the temperature and humidity of the exhaust air are relatively low ($T=50$ – 60°C and humidity= 0.1 – 0.14 kg/kg). In the case of Yankee dryers used in producing tissue paper, the exhaust air outlet conditions are higher ($T=200^{\circ}\text{C}$ and humidity= 0.4 kg/kg). Therefore, another set of measurements, with different process parameters, was carried out on pilot scrubber no. 2.

The results obtained from the measurements on the two pilot scrubbers were divided into two groups of data. The first group is used for training purposes, and the second group is used to test the performance of the trained networks. The neural network model was trained with 199 training instances. After the network had been trained, a testing set consisting of 60 vectors was used to evaluate its accuracy. The adjustable parameter in the neural network training was the number of neurons in the hidden layer. Various sizes of neural networks with one hidden layer were trained and tested. The results of neural network simulations for the training and testing sets for pilot scrubbers, as well as the results from simulations with a physical model (2), are given in **Table II**. The configurations of neural networks in Table II are described with three numbers (i, j, k), where i represents a number of input parameters, j represents a number of neurons in the hidden layer, and k represents a number of output units.

The configuration of a neural network (7,7,1) gave the best results. A (7,7,1) network model resulted in an error sum squared of 0.23992 (rms error of 0.0347) on the 199 training instances and an error sum squared of 0.14841 (rms error of 0.04973) on the 60 testing instances.

Training neural networks with a higher number of neurons in the hidden layer resulted in decreasing the sum squared error of the training instances but increasing the sum squared error of the testing instances. This means that the number of the neurons in the hidden layer should be chosen as low as possible to avoid overfitting the results. The physical model (2) resulted in error sums squared of 1.09671 and 0.25232 (rms errors of 0.07424 and 0.06485) of the training and testing instances, respectively. Neural network configurations with 5–9 neurons in the hidden layer gave better results compared with those obtained with the physical model.

The test data set consists of 60 experiments, which have been used for the simulation to determine the outlet water temperature of the scrubber. Each experiment consists of seven measured input values and one measured output value from the scrubber. The input values are as follows:

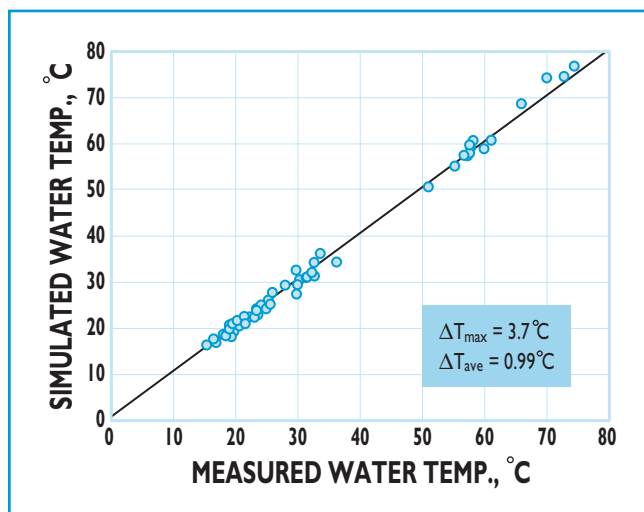
- mass flow rate of inlet air, kg/m²/s
- temperature of inlet air, °C
- humidity of inlet air, kg_{water}/kg_{dry air}
- mass flow rate of water, kg/m²/s
- inlet temperature of water, °C
- height of the column, m
- spray nozzle pressure, kPa.

Configuration of neural network	TRAINING SET		TEST SET	
	SSE ^a	rms ^b	SSE	rms
(7,4,1)	0.45371	0.04775	0.34897	0.07626
(7,5,1)	0.24461	0.03506	0.19205	0.05657
(7,6,1)	0.28239	0.03767	0.20224	0.05805
(7,7,1)	0.23992	0.03472	0.14841	0.04973
(7,8,1)	0.16019	0.02837	0.24283	0.06362
(7,9,1)	0.15853	0.02822	0.24520	0.06393
(7,10,1)	0.09225	0.02153	0.78000	0.11402
(7,12,1)	0.09371	0.02170	0.41697	0.08336
Physical model by Milosavljevic et al.	1.09671	0.07424	0.25232	0.06485

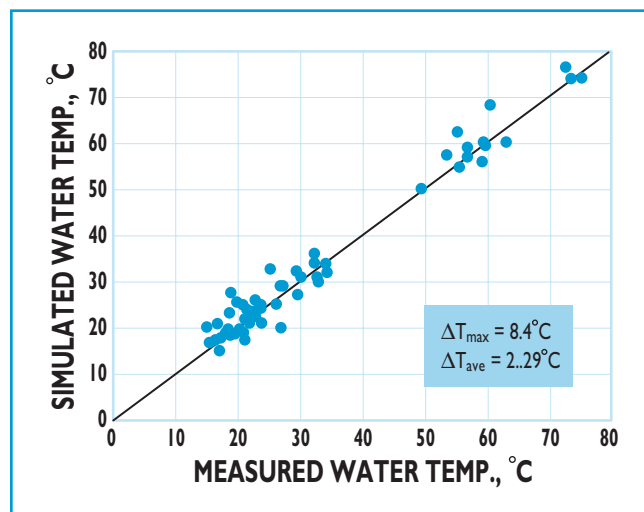
^a Sum-squared error

^b Root-mean-squared error

II. Neural network simulation results and results from simulation with physical model by Milosavljevic et al. (2)



3. Neural network model with configuration (7,7,1)



4. Physical mathematical model (2)

The output value is the outlet water temperature (in °C). The results of the simulation obtained with the neural network with configuration 7,7,1 and the physical model (2) are shown in Figs. 3 and 4, respectively.

When the scrubber neural network model was used for the scrubber simulation, the average deviation between the predicted and experimental outlet water temperature was only 0.99°C (max. deviation was 3.7°C). When the scrubber process was simulated with the physical model, the average temperature deviation was 2.29°C (max. deviation was 8.5°C). More accurate results obtained by the neural network simulation model can lead to reduced equipment costs and better energy utilization of the humid exhaust air.

CONCLUSIONS

This example illustrates how neural networks are a powerful technique in developing empirical models with less effort and time than physical models. Most of the physical models of heat and mass transfer depend on several empirical correlations. The approach taken in this paper shows that neural networks are efficient at approximating the nonlinearities in this system, which predicts the outlet water temperature from a scrubber. The neural network model required

none of the simplifying hypotheses and assumptions that are used in physical models (e.g., droplet distribution) nor did it require assumptions in estimating heat and mass transfer coefficients. Using the neural network for simulating a scrubber, it is possible to avoid complicated fluid dynamics phenomena of the gas-liquid flow but to still offer good non-classical solutions in predicting operational parameters. Several similar systems in pulp and paper drying can benefit from neural network technology if applied properly.

Neural network results were compared with those obtained by a physical model of the system, which was developed after several years of work. The physical model is quite accurate, and its normalized rms error is on the order of 0.065–0.075 (average temperature deviation=2.29°C). The neural network model took much less effort, and its accuracy is even better, with a normalized rms error of 0.035–0.050 (average temperature deviation=0.99°C). This better model can lead to reduced equipment costs and better energy utilization from the humid exhaust air. A similar project in the drying section could lead to much larger savings in energy consumption and more stable quality of the product. **TJ**

Milosavljevic and Heikkilä are R&D engineers at Valmet Corp., Air Systems Pansio, FIN-20240 Turku, Finland.

The authors would like to thank Valmet Corp., Pansio, Turku, Finland, and the Technical Research Centre of Finland (VTT Energy), Jyväskylä, for contributing the pilot scrubbers (nos. 1 and 2) for this study. We would also like to thank Abhay Bulsari for having read this paper and helping us to present it in the appropriate form.

Received for review March 25, 1997.

Accepted Jan. 15, 1998.

LITERATURE CITED

1. Skansi, D. and Milosavljevic, N., *Proceedings of the 39th International Annual Conference KoREMA*, KoREMA, Zagreb, Croatia, 1994, p. 363.
2. Milosavljevic, N., Malinen, P., Heikkilä, P., et al., *Appl. Thermal Eng.* 17(6): 569(1997).
3. Zurada, J. M., *Introduction to Artificial Neural Systems*, West Publishing Co., St. Paul, MN, 1992, PP. 181-196.
4. Freeman, J. A. and Skapura, D. M., *Neural Network Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Amsterdam, 1991, PP. 93-96.
5. Bulsari, A. B., *Neural Networks for Chemical Engineers*, Elsevier, Amsterdam, Netherlands, 1995, PP. 33-76.
6. Marquardt, D. W., *J. Soc. Indust. Appl. Math.* 11: 431(1963).
7. Levenberg, K., *Quart. Appl. Math.* 2: 164(1944).