Development and Comparison of Backpropagation and Generalized Regression Neural Network Models to Predict Diurnal and Seasonal Gas and PM 10 Concentrations and Emissions from Swine Buildings

Gang Sun
Iowa State University

Steven J. Hoff
Iowa State University, hoffer@iastate.edu

Brian C. Zelle
Iowa State University

Minda A. Smith
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/abe_eng_pubs

Part of the Agriculture Commons, and the Bioresource and Agricultural Engineering Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/abe_eng_pubs/351. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
Abstract
The quantification of diurnal and seasonal gas (NH₃, H₂S, and CO₂) and PM10 concentrations and emission rates (GPCER) from livestock production facilities is indispensable for the development of science-based setback determination methods and evaluation of improved downwind community air quality resulting from the implementation of gas pollution control. The purpose of this study was to employ backpropagation neural network (BPNN) and generalized regression neural network (GRNN) techniques to model GPCER generated and emitted from swine deep-pit finishing buildings as affected by time of day, season, ventilation rates, animal growth cycles, in-house manure storage levels, and weather conditions. The statistical results revealed that the BPNN and GRNN models were successfully developed to forecast hourly GPCER with very high coefficients of determination (R²) from 81.15% to 99.46% and very low values of systemic performance indexes. These good results indicated that the artificial neural network (ANN) technologies were capable of accurately modeling source air quality within and from the animal operations. It was also found that the process of constructing, training, and simulating the BPNN models was very complex. Some trial-and-error methods combined with a thorough understanding of theoretical backpropagation were required in order to obtain satisfying predictive results. The GRNN, based on nonlinear regression theory, can approximate any arbitrary function between input and output vectors and has a fast training time, great stability, and relatively easy network parameter settings during the training stage in comparison to the BPNN method. Thus, the GRNN was characterized as a preferred solution for its use in air quality modeling.

Keywords
Backpropogation, Diurnal, Gas, Generalized regression neural network, PM10, Seasonal, Swine buildings

Disciplines
Agriculture | Bioresource and Agricultural Engineering

Comments
This article is from Transactions of the ASABE 51, no. 2 (2008): 685–694.

This article is available at Iowa State University Digital Repository: http://lib.dr.iastate.edu/abe_eng_pubs/351
Development and Comparison of Backpropagation and Generalized Regression Neural Network Models to Predict Diurnal and Seasonal Gas and PM$_{10}$ Concentrations and Emissions from Swine Buildings

G. Sun, S. J. Hoff, B. C. Zelle, M. A. Nelson

Abstract: The quantification of diurnal and seasonal gas (NH$_3$, H$_2$S, and CO$_2$) and PM$_{10}$ concentrations and emission rates (GPCER) from livestock production facilities is indispensable for the development of science-based setback determination methods and evaluation of improved downwind community air quality resulting from the implementation of gas pollution control. The purpose of this study was to employ backpropagation neural network (BPNN) and generalized regression neural network (GRNN) techniques to model GPCER generated and emitted from swine deep-pit finishing buildings as affected by time of day, season, ventilation rates, animal growth cycles, in-house manure storage levels, and weather conditions. The statistical results revealed that the BPNN and GRNN models were successfully developed to forecast hourly GPCER with very high coefficients of determination ($R^2$) from 81.15% to 99.46% and very low values of systemic performance indexes. These good results indicated that the artificial neural network (ANN) technologies were capable of accurately modeling source air quality within and from the animal operations. It was also found that the process of constructing, training, and simulating the BPNN models was very complex. Some trial-and-error methods combined with a thorough understanding of theoretical backpropagation were required in order to obtain satisfying predictive results. The GRNN, based on nonlinear regression theory, can approximate any arbitrary function between input and output vectors and has a fast training time, great stability, and relatively easy network parameter settings during the training stage in comparison to the BPNN method. Thus, the GRNN was characterized as a preferred solution for its use in air quality modeling.

Keywords. Backpropagation, Diurnal, Gas, Generalized regression neural network, PM$_{10}$, Seasonal, Swine buildings.

To address gaseous pollutants generated by livestock and poultry industries, atmospheric dispersion models have been a useful tool for regulatory agencies and state planners to determine reasonable science-based setback distances between animal production facilities and neighboring residences. In addition, environmental researchers and livestock producers can use models to evaluate downwind community air quality impacts resulting from the implementation of gas pollution control (Hoff et al., 2006). The accuracy of dispersion model predictions relies largely on the accuracy of source emission rates, which are highly variable because they depend on time of the day, seasons, building characteristics, ventilation rate, animal size and density, manure handling systems, and weather conditions (Jacobson et al., 2005). However, due to the lack of data, none of the existing models consider the diurnal, seasonal, and climate variations of odor and gas emission rates from animal buildings. Some researchers simply use randomly measured data or the mean or geometric mean of some data measured during the daytime at any time of the year as the emission rates for the model input (Lim et al., 2000; Jacobson et al., 2005), which may result in great uncertainties in predictions. Thus, there is a great need to obtain source gas and PM$_{10}$ concentration and emission rate (GPCER) profiles for the time period of interest (e.g., an hour or a day) to ensure the accuracy of atmospheric dispersion models.

Several studies have investigated diurnal and seasonal odor and gas emission rates from different types of swine production buildings (Sun, 2005; Hoff et al., 2006; Guo et al., 2007). However, direct and long-term measurements of odor, gas, and PM$_{10}$ concentrations and emissions at all animal operations are not practical since every gas source is different and animal and weather conditions change constantly. In the absence of effective and efficient means to directly measure GPCER from each livestock production facility, development of source GPCER mathematical prediction models might be a good alternative to provide reasonably accurate estimates. Three modeling approaches have been proposed for predicting source GPCER: the emission factors method, the multiple regression analysis method, and the process-based modeling method.

Emission factors, expressed by the amount of each substance emitted per animal, are multiplied by the number of animal units to get average air emissions from animal operations. Argo et al. (2003) attempted but could not assign em-
For all animals in all regions was not appropriate if direct and estimated predictive results showed that using emission factors many variables affecting air emissions. The under- or overestimated ammonia emission rates from various barns because of the practical ammonia emission factors to estimate the average gas release process is very complex due to abundant nonlinear relationships between gaseous emissions and the many variables that cause gas production. Therefore, a major effort would be required in future process-based model studies. Due to the absence of adequate information available about the process of gas pollutant production, a black-box modeling approach using artificial neural networks (ANN) would be a powerful and promising tool for air quality prediction. Black-box models do not need detailed prior knowledge of the structure and different interactions that exist between important variables. Meanwhile, their learning abilities make the models adaptive to system changes. In recent years, there has been an increasing amount of applications of ANN models in the field of atmospheric pollution forecasting (Hooyberghs et al., 2005; Grivas et al., 2006; Sousa et al., 2007). The results show that ANN black-box models are able to learn nonlinear relationships with limited knowledge about the process structure, and the neural networks generally present better results than traditional statistical methods. In the literature, little attention has been paid to forecasting source air quality within and from animal buildings. The overarching goal of this project was to develop backpropagation and generalized regression neural network models (black-box models) to predict diurnal and seasonal concentrations and emissions of ammonia, hydrogen sulfide, carbon dioxide, and particulate matter less than or equal to 10 μm (PM10) from swine finishing buildings.

MATERIALS AND METHODS

EXPERIMENT DATA

The NH3, H2S, CO2, and PM10 data were collected from two identical deep-pit swine finishing buildings in Iowa from January 2003 to April 2004. Each building had one room and was designed to house 960 pigs ranging in weight between ~20 and 120 kg. Slurry was stored in a 2.4 m deep concrete holding pit below a fully slatted floor and was designed to store manure for one year.

An instrument trailer (Mobile Emission Laboratory, MEL) was used to monitor gas and particulate matter concentrations, environmental data, and barn airflow rates. A chemiluminescence NH3 analyzer (model 17C, TEL, Franklin, Mass.), a pulsed fluorescence SO2 detector (model 45C, TEL, Franklin, Mass.), and two photoacoustic infrared CO2 analyzers (model 3600, MSA, Pittsburgh, Pa.) were used to measure gas concentrations at 12 locations within two buildings (“north barn” and “south barn”). A solenoid switching system enabled gas samples to be delivered to each analyzer simultaneously in 10 min switching increments, i.e., each location was monitored for 10 min every 120 min. PM10 concentrations were measured continuously using two tapered-element oscillating microbalance (TEOM) ambient PM10 monitors (model 1400a, Rupprecht & Patashnick, Albany, N.Y.). Environmental parameters (e.g., temperature, relative humidity, and static pressure) and total building ventilation rates were monitored simultaneously. The total ventilation rates were measured by recording the on/off status of four single-speed tunnel fans, and the on/off status and fan rpm levels of all variable-speed fans (two pit fans, one sidewall fan, and one tunnel fan). The ventilation rate of each fan was obtained in situ using a FANS unit, for which calibration equations were developed as a function of static pressure and fan rpm levels for the variable-speed fans (Heber et al., 2002, 2005; Hooyberghs et al., 2005).
Gas and PM$_{10}$ emission rates were determined by multiplying the total airflow rate of the exhaust fans by the increase in gas and PM$_{10}$ concentrations between the building ventilation inlet and outlet. The total building emissions were calculated from three emission locations (the blended pit ventilation fans, the sidewall fan, and the tunnel fans) and were expressed on an animal unit basis by dividing the total emissions by the total animal units ($1\text{AU} = 500\text{kg}$). During the whole measurement period, approximately three complete production cycles of pigs raised from 20 to 120 kg were monitored.

The hourly average gas concentrations were determined based on the 10 min sampling data using interpolation, while the hourly gas emissions were obtained by multiplying real-time ventilation rates by the interpolated gas concentrations. Pig weight was measured twice for each group (entering and leaving), and linear interpolation was used to estimate intermediate weights.

The original data set of hourly average GPCER values from the north barn included 7366-9289 lines and four variables. The data set presented diurnal (hourly) and seasonal (16 continuous measurement months) variations of gas and PM$_{10}$ concentrations and emission rates. A multivariate statistical analysis (Sun et al., 2008) was conducted, and from this analysis it was determined that four main variables were significant contributors to the GPCER models. These four input variables include: outdoor temperature ($T_{\text{out}}$), animal units (AU), total building ventilation rate (VR), and indoor temperature ($T_{\text{in}}$).

**BACKPROPAGATION NEURAL NETWORK**

The multilayer perceptron (MLP) is the most common and successful neural network architecture with feed-forward network topologies in atmospheric science modeling applications; while the most common supervised learning technique used for training artificial neural networks is the multilayer backpropagation (BP) algorithm (Kecman, 2001). The term “backpropagation” refers to the process by which derivatives of network error, with respect to the networks, are fed back to the network and used to adjust the weights so that the error decreases with each iteration and the neural model gets closer and closer to producing the desired outputs. In this way, BP offers a method of minimizing errors between obtained outputs and desired target values.

There are generally four steps to develop a BP neural network for modeling: (1) preprocess the data, (2) create the network object, (3) train the network, and (4) simulate the network response to new inputs. In this research, the first step (preprocess) was done to scale the inputs and targets to fall within a specified range (from 0 to 1) in case the higher values would drive the training process and mask the contribution of lower valued inputs, as well as to perform a principal component analysis to eliminate redundancy of the data set. In the second step (network construction), the data set was divided into training, validation, and test subsets: one-half for the training set, one-fourth of the data for the validation set, and one-fourth for the test set. The training set was used for computing the gradient and updating the network weights and biases. The validation set was used for improving generalization. The test set was used for validating the network performance. The data in each subset were selected randomly, and then a network was created. The third step (network training) initialized and trained the network. A total of five trainings were conducted. Finally, the trained network was employed to simulate the test data. The performances of the network in each training process and the best network with the highest prediction performances were recorded.

**GENERALIZED REGRESSION NEURAL NETWORK**

The generalized regression neural network (GRNN) is a neural network architecture that can solve any function approximation problem if sufficient data are given. Figure 1 is a schematic of the GRNN architecture with four layers: an input layer, a hidden layer (pattern layer), a summation layer, and an output layer.

The main function of a GRNN is to estimate a linear or nonlinear regression surface on independent variables, i.e., the network computes the most probable value of an output $y$ given only training vectors $x$ (Specht, 1991). Specifically, the network computes the joint probability density function (pdf) of $x$ and $y$. The expected value of the output $y$ given the input vector $x$ is given by:

$$
E[y|x] = \frac{\int y f(x,y)dy}{\int f(x,y)dy}
$$

(1)

When the density $f(x,y)$ is not known, it must usually be estimated from a sample of observations of $x$ and $y$. The probability estimator $\hat{f}(x,y)$ is based on sample values $x^i$ and $y^i$ of the random variables $x$ and $y$:

$$
\hat{f}(x,y) = \frac{1}{(2\pi)^{(p+1)/2} \sigma^{(p+1)}} \frac{1}{n} \sum_{i=1}^{n} \exp \left\{ -\frac{(x-x^i)^2 + (y-y^i)^2}{2\sigma^2} \right\} \exp \left\{ -\frac{(y-y^i)^2}{2\sigma^2} \right\}
$$

(2)

where $n$ is the number of sample observations, and $p$ is the dimension of the vector variable $x$.

A physical interpretation of the probability estimate $\hat{f}(x,y)$ is that it assigns sample probability of width $\sigma$ (smoothing factor or “spread”) for each sample $x^i$ and $y^i$, and the probability estimate is the sum of those sample probabilities.

The squared distance between the input vector $x$ and the training vector $x^i$ is defined as:

$$
D^2 = (x-x^i)^T (x-x^i)
$$

(3)
and the final output is determined by performing the integration in equation 4. This result is directly applicable to problems involving numerical data.

\[
\hat{f} = \frac{\sum_{i=1}^{N} y_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{N} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}
\]  

(4)

The smoothing factor \(\sigma\), considered as the size of the neuron's region, is a very important parameter of GRNN. When \(\sigma\) is large, the estimated density is forced to be smooth and in the limit becomes a multivariate Gaussian with covariance \(\sigma^2\) I (I = unity matrix), whereas a smaller value of \(\sigma\) allows the estimated density to assume non-Gaussian shapes, but with the hazard that wild points may have a great effect on the estimate (Specht, 1991). Therefore, a range of smoothing factors and methods for selecting those factors should be tested empirically to determine the optimum smoothing factors for the GRNN models.

**Performance Indicators and Software**

The root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (\(R^2\)) between the modeled output and measures of the training and testing data set are the most common indicators to provide a numerical description of the goodness of the model estimates. They are calculated and defined according to equations 5, 6, and 7, respectively (Sousa et al., 2007):

\[
RMSE = \left(\frac{1}{N} \sum_{i=1}^{N} [A_i - T_i]^2\right)^{1/2}
\]  

(5)

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |T_i - A_i|
\]  

(6)

\[
R^2 = \frac{\sum_{i=1}^{N} [A_i - \bar{T}]^2}{\sum_{i=1}^{N} [T_i - \bar{T}]^2}
\]  

(7)

where

- \(N\) = number of observations
- \(T_i\) = observed value
- \(A_i\) = predicted value
- \(\bar{T}\) = average value of the explained variable on \(N\) observations.

RMSE and MAE indicate the residual errors, which give a global idea of the difference between the observed and predicted values. \(R^2\) is the proportion of variability (sum of squares) in a data set that is accounted for by a model. When the RMSE and MAE are at the minimum and \(R^2\) is high (\(R^2 \geq 0.80\)), a model can be judged as very good (Kasabov, 1998). Neural Network toolbox 5.1 and Statistics toolbox 6.1 in Matlab 7.4 (R2007a) were used in the present study to develop ANN models.

**Results and Discussion**

**Diurnal and Seasonal Data**

Central Iowa climate information based on monthly measurement averages in 2003 could be separated into three typical weather conditions: warm weather (June, July, Aug.; 22.6°C to 27.9°C), mild weather (Apr., May, Sept., Oct.; 10.1°C to 16.4°C), and cold weather (Jan., Feb., Mar., Nov., Dec.; -7.4°C to 2°C). Figure 2 shows three different diurnal and seasonal variation patterns of NH3 concentrations under different measurement months (Jan., Apr., and July). The mean NH3 concentrations during the winter were much higher than the NH3 levels in the summer, and large diurnal NH3 variations between day and night were observed in April. Diurnal and seasonal fluctuations of other air pollutants also existed. These variations indicated that the gaseous concentrations and emissions during different periods of the day and different seasons must be obtained and considered in air dispersion models for setback distance determination in lieu of random data sampled from snapshot measurements.

**BPNN Model Development**

The development of a good BP neural network model depends on several important parameters determined using trial-and-error methods. The BP ANN model of NH3 concentration is presented here as an example showing how to choose these parameters step by step. Other predictive models followed this modeling process and methods.

![Figure 2. Different diurnal and seasonal variation patterns of NH3 concentrations from the deep-pit swine finishing building (hourly averages presented for three selected days).](image-url)
The initial problem faced in this study was deciding on the BP network architecture, i.e., the number of layers and neurons in the hidden layer as well as the type of activation functions for the layers. A three-layer BP network was constructed to determine if its prediction performance was superior to a two-layer network. Unfortunately, the results were almost the same. It is worth noting that the bigger network architecture would need more computation and could cause overfitting of the data. In practical applications, one rarely encounters a structure more complex than a two-layer network. Thus, a two-layer BP network was employed, which could produce solutions arbitrarily close to the optimal solution.

Networks are sensitive to the number of neurons in their hidden layers. The optimum number of neurons required is problem dependent, being related to the complexity of the input and output mapping, the amount of noise in the data, and the amount of training data available. Too few neurons lead to underfitting, while too many neurons contribute to overfitting, in which all training points are well-fitted but the fitting curve oscillates widely between these points. Currently, there is no guiding rule to determine how many neurons to use in the hidden layer (Keeman, 2001). The only method available is to try different numbers of neurons to observe how the results look. Table 1 gives the predictive model results (e.g., \( R^2 \) between the predicted and actual values) using different numbers of neurons in the hidden layer. The initial number of neurons was 5, and the number was increased until a relatively stable and optimal value was achieved. It can be seen that 40 to 70 neurons in the hidden layer produced high \( R^2 \) results (around 0.90). The predictive performance improved slightly with increasing numbers of neurons (90 to 150), but the training time increased significantly. When the network had 5 or 10 neurons in the hidden layer, the \( R^2 \) decreased to 0.80. Thus, 40 or 50 were determined as the optimum number of neurons in the hidden layer to avoid low predictive results caused by too few neurons or the overfitting performance from too many neurons.

Note that networks with threshold units are hard to train because the threshold units are not continuous; a small change in the weights does not cause any change in the output. Sigmoid transfer functions are usually preferable to threshold activation functions. With sigmoid units, a small change in the weights produces a change in the output, which makes it possible to tell whether that change in the weights was good or bad. There are three sigmoid transfer functions often used for BP networks: tansig (hyperbolic tangent sigmoid) transfer function, logsig (log-sigmoid) transfer function, and purelin (linear) transfer function. The tansig transfer function, which can produce both positive and negative values, tended to yield faster training than the logsig transfer function, which can produce only positive values. Table 2 summarizes the BP network performance (e.g., \( R^2 \)) using different transfer functions. In general, all of the transfer function combinations tested obtained nearly the same network performance except for the combination of logsig and purelin. The tansig and logsig functions were employed in this research.

Once the BP network was constructed and the weights and biases were initialized, the network was ready for training. Neural Network toolbox 5.1 in Matlab offers several training algorithms, such as traingd, traindx, trainda, trainrp, trainlm, trainbfg, trainscg, trainoss, traincgf, and traincgp, which are used for training BP networks. Their characteristics deduced from the experiments are shown in table 3. It was observed that the traingd (gradient descent BP) algorithm had the lowest training speed compared to all other algorithms, whereas trainda (gradient descent BP with adaptive learning rate) had the fastest training speed, followed by trainscg (scaled conjugate gradient BP) and traindx (gradient descent BP with momentum and adaptive learning rate). The trainoss (one step secant BP), trainrp (resilient BP), traincgp (conjugate gradient BP with Polak-Ribière updates), trainbfg (BFGS quasi-Newton BP), and traincfg (conjugate gradient BP with Fletcher-Reeves updates) algorithms could obtain relatively fast training speeds, but their prediction performances were not as good as those made by the trainlm (Levenberg-Marquardt BP) algorithm, which was capable of achieving very satisfying statistical results with the highest \( R^2 \) and the smallest mean square error among the other algorithms.

Furthermore, although the trainda and traindx algorithms trained the BP network much faster than the trainlm algorithm, the performances of the former algorithms were very sensitive to the proper setting of the learning rate and momentum. A large learning rate may lead to faster convergence, but it may also cause strong oscillations near the optimal solution or even diverge, while excessively small learning rates result in very long training times. The purpose of adding momentum was to allow the network to respond not only to the local gradient, but also to recent trends in the error surface and allow the network to ignore small features in the error surface. Without momentum, the network can get stuck in a shallow local minimum. Conversely, with momentum,
Table 3. Results using different training algorithms.\[a\]

<table>
<thead>
<tr>
<th>Training Algorithm</th>
<th>R² of Predicted vs. Actual</th>
<th>Avg. R²</th>
<th>Elapsed Time[b]</th>
</tr>
</thead>
<tbody>
<tr>
<td>trainbfg</td>
<td>0.7913</td>
<td>0.7961</td>
<td>0.8232</td>
</tr>
<tr>
<td>trainbsp</td>
<td>0.8119</td>
<td>0.8187</td>
<td>0.8450</td>
</tr>
<tr>
<td>trainoss</td>
<td>0.7935</td>
<td>0.7807</td>
<td>0.7233</td>
</tr>
<tr>
<td>trainscg</td>
<td>0.7932</td>
<td>0.7344</td>
<td>0.8269</td>
</tr>
<tr>
<td>traincg</td>
<td>0.7236</td>
<td>0.8523</td>
<td>0.8150</td>
</tr>
</tbody>
</table>

\[a\] The testing network was a two-layer network with tanlin and logsl function and 50 neurons in the hidden layer. Five training times were used for each training process.

\[b\] Elapsed time (s) indicates the time of one training. The computer had an Intel Pentium 3.0 G processor and 3.0 Gb RAM.

Table 4. Optimal parameters of the BP ANN model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/Function/Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network architecture</td>
<td>2-layer network</td>
</tr>
<tr>
<td>Input features</td>
<td>T₀, AU, VR, and Tₙ[a]</td>
</tr>
<tr>
<td>Layer neurons</td>
<td>4-50-1 (input-hidden-output layer)</td>
</tr>
<tr>
<td>Missing data</td>
<td>Substituting the neighborhood mean</td>
</tr>
<tr>
<td>Data normalization</td>
<td>mapstd function</td>
</tr>
<tr>
<td>PCA[b]</td>
<td>processcpa function</td>
</tr>
<tr>
<td>Transfer function</td>
<td>tanlin (hidden layer); logsl (output layer)</td>
</tr>
<tr>
<td>Training algorithm</td>
<td>trainlm</td>
</tr>
</tbody>
</table>

\[a\] T₀ = outdoor temperature (°C), AU = animal units, VR = ventilation rates (m³ s⁻¹), Tₙ = indoor temperature (°C).

\[b\] PCA = principal component analysis.

the network can slide through such a minimum. The optimal learning rate and momentum can only be acquired experimentally using the trial-and-error method. Therefore, the trainlm algorithm was suitable for training the NH₃ concentration ANN model. However, it has a drawback in that it requires the storage of large matrices. If this is the case, the trainlp algorithm may be a good alternative due to its small memory requirement. The optimal parameters of the BP neural network model for the NH₃ concentrations are summarized in table 4.

**GRNN Model Development**

The only parameter particular to the GRNN is the use of the smoothing factor \( \sigma \), which significantly affects network performance. Table 5 summarizes the results for the NH₃ concentration GRNN model using different smoothing factors.

<table>
<thead>
<tr>
<th>Model[a]</th>
<th>Number of Data Points[b]</th>
<th>GRNN</th>
<th>BPNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH₃Con (ppm)</td>
<td>8084</td>
<td>0.9946</td>
<td>1.92</td>
</tr>
<tr>
<td>NH₃ER (kg d⁻¹)</td>
<td>7973</td>
<td>0.9774</td>
<td>0.80</td>
</tr>
<tr>
<td>H₂SCon (ppb)</td>
<td>7479</td>
<td>0.9167</td>
<td>102.54</td>
</tr>
<tr>
<td>H₂SER (kg d⁻¹)</td>
<td>7366</td>
<td>0.9258</td>
<td>0.08</td>
</tr>
<tr>
<td>CO₂Con (ppm)</td>
<td>8500</td>
<td>0.9838</td>
<td>184.20</td>
</tr>
<tr>
<td>CO₂ER (kg d⁻¹)</td>
<td>8215</td>
<td>0.9410</td>
<td>144.02</td>
</tr>
<tr>
<td>PM₁₀Con (µg m⁻³)</td>
<td>9187</td>
<td>0.8570</td>
<td>125.52</td>
</tr>
<tr>
<td>PM₁₀ER (kg d⁻¹)</td>
<td>9289</td>
<td>0.8719</td>
<td>0.07</td>
</tr>
</tbody>
</table>

\[a\] Con and ER indicate the concentrations and emission rates, respectively.

\[b\] Indicates the number of total data points. The test data for the predictive models were 25% of the total data.

**STATISTICAL PERFORMANCE OF PREDICTIVE MODELS**

The statistical performance of the developed predictive models is given in table 6, and scatter plots of predicted values (output \( A \)) versus respective observed values (target \( T \)) for the GRNN and BPNN models are illustrated in figure 3. The data presented in figure 3 were normalized using \( (T - T_{\text{min}})/(T_{\text{max}} - T_{\text{min}}) \). The intercept and slope of the least squares line between predictions and observations are also displayed. It is worth mentioning that a series of random tests was conducted to evaluate the effectiveness of the models. The results showed that all the models were quite stable. The value of each performance indicator (R², MAE, and RMSE) was within 2% change in every case. The results shown here were derived from the best network after the tests.

All the GRNN and BPNN predictive models, except for the PM₁₀ concentration and emission BPNN models, had excellent predicting abilities with high R² values (81.15% to 99.46%) and low MAE and RMSE values, which implies that these models were well-developed (table 6). The high R² indicates that a majority of the variability in the air pollutant outputs could be explained by the four input variables (outdoor and indoor temperature, building ventilation rate, and animal units).

All the GRNN predictive models had higher R² values and lower MAE and RMSE values than the BPNN models. This demonstrates that the GRNN models outperformed the BPNN models. Thus, the GRNNs were able to predict diurnal and seasonal gas and particulate matter concentrations and emissions more effectively.
Figure 3. Scatter plots (a) to (p) of predicted values (output A) versus respective observed values (target T) for the GRNN and BPNN models (Con and ER indicate the concentrations and emission rates, respectively) (continued on next page).
Figure 3 (continued from previous page). Scatter plots (a) to (p) of predicted values (output A) versus respective observed values (target T) for the GRNN and BPNN models (Con and ER indicate the concentrations and emission rates, respectively) (continued on next page).
SUMMARY AND CONCLUSIONS

Backpropagation and generalized regression neural network methods were employed to explore the complex and highly nonlinear relationships between air pollutants and four variables (outdoor temperature, animal units, ventilation rate, and indoor temperature) on the measurements of diurnal and seasonal NH\textsubscript{3}, H\textsubscript{2}S, CO\textsubscript{2}, and PM\textsubscript{10} levels and emissions from deep-pit swine buildings.

It was found that the obtained results of BPNN and GRNN predictions were in good agreement with the actual measurements, with coefficient of determination ($R^2$) values between 81.15% and 99.46% and very low values of systemic performance indexes. The good results indicated the ANN technologies were capable of accurately modeling source air quality within the livestock production facilities and emissions from these production facilities.

The process of constructing, training, and simulating the BP network models was very complicated. Likewise, determining the best values for several network parameters, such as the number of layers and neurons, type of activation functions and training algorithms, learning rates, and momentum, were difficult. The effective way of obtaining good BP modeling results was to use some trial-and-error methods and thoroughly understand the theory of backpropagation. Conversely, for the GRNN models, there was only one parameter (the smoothing factor) that needed to be adjusted experimentally. Moreover, the BP network performance was very sensitive to randomly assigned initial values. However, this problem was not faced in GRNN simulations. The GRNN approach did not require an iterative training procedure as in the backpropagation method. The local minima problem was also not faced in the GRNN simulations. Other significant characteristics of the GRNN in comparison to the BPNN were the excellent approximation ability, fast training time, and exceptional stability during the prediction stage. Thus, the GRNN technology outperformed BP, which has been demonstrated in this study. It can be recommended that a generalized regression neural network be used instead of a backpropagation neural network in source air quality modeling.
REFERENCES


Sun, G. 2005. Monitoring and modeling diurnal and seasonal odor and gas emission profiles for swine grower/finisher rooms. MS thesis. Saskatoon, Saskatchewan, Canada: University of Saskatchewan, Department of Agricultural and Bioresource Engineering.


NOMENCLATURE

AU = animal units
BP = backpropagation
BPNN = backpropagation neural network
Con = concentration
ER = emission rate
GPCER = gas and PM10 concentration and emission rate
GRNN = generalized regression neural network
logsig = log sigmoid transfer function
MAE = mean absolute error
PCA = principal component analysis
purelin = linear transfer function
RMSE = root mean square error
R2 = coefficient of determination
tansig = tangent sigmoid transfer function
Tin = indoor temperature (°C)
Tout = outdoor temperature (°C)
trainbfg = BFGS quasi-Newton BP training algorithm
trainscg = scaled conjugate gradient BP training algorithm
trainscg = conjugate gradient BP with BFGS quasi-Newton updates training algorithm
traincgf = conjugate gradient BP with Fletcher-Reeves updates training algorithm
traincgd = conjugate gradient BP with Polak-Ribière updates training algorithm
trainng = gradient descent BP training algorithm
trainng = gradient descent BP with adaptive learning rate updates training algorithm
trainlm = Levenberg-Marquardt BP training algorithm
trainoss = one step secant BP training algorithm
trainrprop = resilient BP training algorithm
trainscg = scaled conjugate gradient BP training algorithm
VR = ventilation rates (m3 s-1)