2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation

Paper Number: 085100

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, Steven Hoff, Brian Zelle, Minda Nelson

Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, Iowa

Written for presentation at the
2008 ASABE Annual International Meeting
Sponsored by ASABE
Rhode Island Convention Center
Providence, Rhode Island
June 29 – July 2, 2008

Abstract. *The quantification of diurnal and seasonal gas (NH_3, H_2S, 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. Backpropogation, Diurnal, Gas, GRNN, PM_{10}, Seasonal, Swine buildings.

Introduction

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, season, building characteristics, ventilation rate, animal size and density, manure handling system, and weather conditions (Jacobson et al., 2005). However, due to a lack of data, none of the existing models consider the diurnal, seasonal, and climate variations of odor and gas emission rates from animal buildings. It has been common to use randomly measured data or the mean or geometric mean of measured data during the daytime at any time of the year as the emission rates for model inputs (Lim et al., 2000; Jacobson et al., 2005). 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 (Hoff et al., 2006; Guo et al., 2007; Sun, 2008a). 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. Arogo et al. (2003) attempted but could not assign empirical ammonia emission factors to estimate the average ammonia emission rates from various barns because of the many variables affecting air emissions. The under- or overestimated predictive results showed that using emission factors for all animals in all regions was not appropriate if direct and long-term measurements from a substantial number of representative animal feeding operations have not been conducted.

The regression analysis method uses standard least-squares multivariate regression equations to predict GPCER. The purpose of multiple regression analysis is to establish a quantitative relationship between various predictor variables (e.g., weather and animal conditions, production systems, etc.) and air emissions. This relationship is used to understand which predictors have the greatest effect and to forecast future values of the equation response when only the predictors and the direction of their effects are known. Sun (2005) developed statistical multiple-linear regression models to predict diurnal and seasonal odor and gas concentrations and emissions from confined swine grower-finisher rooms. However, the main weakness of this method is that the complex and sometimes nonlinear relationships of multiple variables can make statistical models complicated and awkward (Comrie, 1997). Moreover, these models seem very dependent on the specifics of the experiment situation. Therefore, it is difficult to apply the developed model to the data from other experiments. The only way to establish a robust set of equations is to sample hundreds of animal feeding operations under different meteorological conditions. The lack of sufficient data is the main cause of the uncertainty of the statistical regression models.

The process-based models (also called mechanical models) determine the movement of elements (e.g., nitrogen, carbon, and sulfur) into, through, and out of the livestock production system, investigate the underlying chemical and physical phenomenon, and identify the effects of changing one or more variables of the system. In many cases, this modeling method uses mass balance equations to describe the mechanisms of gaseous emissions and estimate their characteristic and amount at each transformation stage. Recently, Zhang et al. (2005) established a comprehensive and predictive ammonia emission model to estimate ammonia emission rates from animal feeding operations using a process-based modeling approach. The main processes treated in the model included nitrogen excretion from the animals, animal housing, manure storage, and land application of manure. The results showed that the sensitivity analysis of various variables (e.g., manure production system, animal housing designs, and environmental conditions) needs to be quantified and that additional model validation is needed to improve model predictive accuracy. Other researchers also studied the process of mass (ammonia) transport and developed mechanical models for swine feeding operations (Aarnink and Elzing, 1998; Ni et al., 2000; Kai et al., 2006). Although there has been considerable value in the development and application of mechanistic modeling of ammonia volatilization from the main individual sources, some circumstances of gaseous emissions are not well understood and several parameters are difficult to determine experimentally. For example, adsorption, absorption, and desorption of ammonia from various materials in animal barns might be another emission source, but this mechanism is not easily acquired. Additionally, the 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 (PM_{10}) from swine finishing buildings.

Materials and Methods

Experiment Data

The NH_3, H_2S, CO_2, and PM_{10} 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 NH_3 analyzer (model 17C, TEI, Franklin, Mass.), a pulsed fluorescence SO_2 detector (model 45C, TEI, Franklin, Mass.), and two photoacoustic infrared CO_2 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. PM_{10} concentrations were measured continuously using two tapered-element oscillating microbalance (TEOM) ambient PM_{10} 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., 2006). 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 AU = 500 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., 2008b) 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_{out}), animal units (AU), total building ventilation rate (VR), and indoor temperature (T_{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.

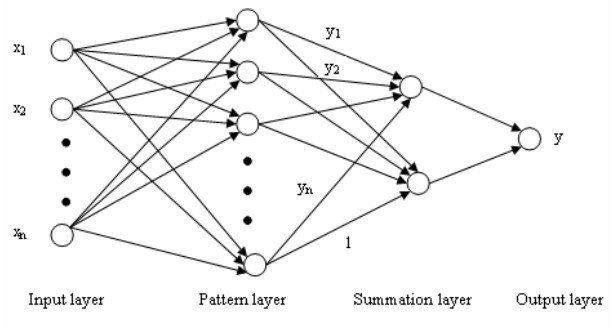


Figure 1. Generalized regression neural network architecture.

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:

$$E\left[\frac{y}{x}\right]=\frac{\int_{-\infty}^{\infty} yf(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy} \qquad (1)$$

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*:

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

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 σ (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:

$$D_i^2 = \left(x - x^i\right)^T \left(x - x^i\right) \quad (3)$$

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

$$\hat{f} = \frac{\sum_{i=1}^{n} y^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad (4)$$

The smoothing factor σ, considered as the size of the neuron's region, is a very important parameter of GRNN. When σ is large, the estimated density is forced to be smooth and in the limit becomes a multivariate Gaussian with covariance σ^2 I (I = unity matrix), whereas a smaller value of σ 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):

$$\mathrm{RMSE} = \left(\frac{1}{N}\sum_{i=1}^{N}\left[A_i - T_i\right]^2\right)^{1/2} \quad (5)$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|T_i - A_i\right| \quad (6)$$

$$\mathrm{R}^2 = \frac{\sum_{i=1}^{N}\left[A_i - \bar{T}\right]^2}{\sum_{i=1}^{N}\left[T_i - \bar{T}\right]^2} \quad (7)$$

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 NH_3 concentrations under different measurement months (Jan., Apr., and July). The mean NH_3 concentrations during the winter were much higher than the NH_3 levels in the summer, and large diurnal NH_3 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.

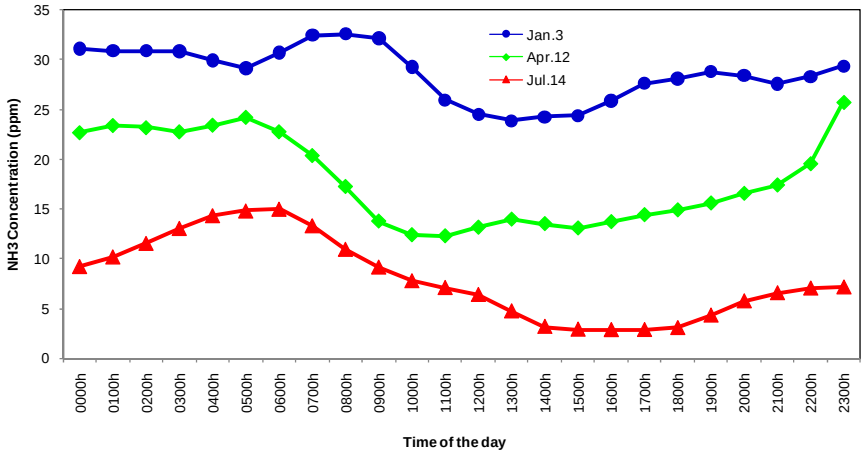


Figure 2. Different diurnal and seasonal variation patterns of NH_3 concentrations from the deep-pit swine finishing building (hourly averages presented for three selected days).

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 NH_3 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.

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 (Kecman, 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 expect for the combination of *logsig* and *purelin*. The *tansig* and *logsig* functions were employed in this research.

Table 1. Results using different numbers of neurons in the hidden layer.[a]

No. of Neurons	R^2 of Predicted vs. Actual					Avg. R^2	Elapsed Time[b]
5	0.7926	0.7862	0.7898	0.8090	0.7906	0.7936	10.3905
10	0.8079	0.8283	0.7830	0.8207	0.8337	0.8147	11.1335
20	0.8697	0.8491	0.8610	0.8579	0.8596	0.8595	12.7646
40	0.8901	0.8796	0.8878	0.8951	0.8826	0.8870	33.8522
50	0.9281	0.9358	0.9194	0.9080	0.9263	0.9235	34.7166
70	0.9136	0.8786	0.8797	0.8968	0.9077	0.8953	52.8628
90	0.9556	0.9541	0.9313	0.9366	0.9622	0.9480	71.3110
120	0.9272	0.9646	0.9415	0.9305	0.9347	0.9397	99.6590
150	0.9400	0.9367	0.9448	0.9186	0.9359	0.9352	138.6449

[a] The testing network was a two-layer network with *tansig* and *logsig* transfer functions. Five training times were used for each training process. The training algorithm was *trainrp*.

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

Table 2. Results using different transfer functions.

Transfer Functions[a]	R^2 of Predicted vs. Actual					Avg. R^2	Max. R^2
tansig, *logsig*	0.9103	0.8985	0.9122	0.9284	0.9185	0.9136	0.9284
tansig, *tansig*	0.8883	0.8740	0.8478	0.8799	0.8880	0.8756	0.8883
logsig, *logsig*	0.8905	0.9067	0.9136	0.9067	0.8874	0.9010	0.9136
logsig, *tansig*	0.8845	0.8910	0.8908	0.8829	0.8556	0.8810	0.8910
tansig, *purelin*	0.8951	0.8759	0.8898	0.9023	0.8971	0.8920	0.9023
logsig, *purelin*	0.8571	0.8361	0.8331	0.8248	0.8337	0.8370	0.8571

[a] The first term indicates the transfer function for the hidden layer; the second term indicates the transfer function for the output layer. The testing network was a two-layer network with the *trainlm* algorithm and 50 neurons in the hidden layer. Five training times were used for each training process.

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*, *traingdx*, *trainingda*, *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 *traingda* (gradient descent BP with adaptive learning rate) had the fastest training speed, followed by *trainscg* (scaled conjugate gradient BP) and *traingdx* (gradient descent BP with momentum and adaptive learning rate). The *trainoss* (one step secant BP), *trainrp* (resilient BP), *traincgp* (conjugate gradient BP with Polak-Ribiére updates), *trainbfg* (BFGS quasi-Newton BP), and *traincgf* (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 *traingda* and *traingdx* 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, 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 NH_3 concentration ANN model. However, it has a drawback in that it requires the storage of large matrices. If this is the case, the *trainrp* algorithm may be a good alternative due to its small memory requirement. The optimal parameters of the BP neural network model for the NH_3 concentrations are summarized in table 4.

Table 3. Results using different training algorithms.[a]

Training Algorithm	R^2 of Predicted vs. Actual					Avg. R^2	Elapsed Time[b]
traingd	0.7447	0.8658	0.8127	0.9042	0.8541	0.8363	140.9733
traingdx	0.8447	0.7370	0.7583	0.7992	0.7872	0.7853	23.4912
traingda	0.7019	0.7385	0.7381	0.7120	0.7410	0.7263	18.3326
trainrp	0.8528	0.8356	0.8277	0.8146	0.8186	0.8299	30.6993

trainlm	0.9032	0.9222	0.9118	0.8705	0.8935	0.9002	34.0856
trainbfg	0.7913	0.7961	0.8101	0.8232	0.8206	0.8083	33.6143
trainscg	0.8119	0.8187	0.8090	0.8450	0.8097	0.8189	20.1881
trainoss	0.7935	0.7807	0.7722	0.7233	0.7728	0.7685	24.1788
traincgf	0.7932	0.7344	0.8269	0.8240	0.8230	0.8003	41.2675
traincgp	0.7236	0.8523	0.7988	0.8150	0.8085	0.7996	32.4685

[a] The testing network was a two-layer network with *tansig* and *logsig* transfer functions 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 3.0G processor and 3.0 Gb RAM.

Table 4. Optimal parameters of the BP ANN model.

Parameter	Value/Function/Method
Network architecture	2-layer network
Input features	T_{out}, AU, VR, and T_{in}[a]
Layer neurons	4-50-1 (input-hidden-output layer)
Missing data	Substituting the neighborhood mean
Data normalization	*mapstd* function
PCA[b]	*processpca* function
Transfer function	*tansig* (hidden layer); *logsig* (output layer)
Training algorithm	*trainlm*

[a] T_{out} = outdoor temperature (°C), AU = animal units, VR = ventilation rates ($m^3\ s^{-1}$), T_{in} = indoor temperature (°C);
[b] PCA = principal component analysis.

GRNN Model Development

The only parameter particular to the GRNN is the use of the smoothing factor σ, which significantly affects network performance. Table 5 summarizes the results for the NH_3 concentration GRNN model using different smoothing factor values. The σ values 0.05 and 0.1 can fit data very closely, with higher R^2 values than when using the larger σ, but the larger smoothing factor can make the function approximation smoother.

Table 5. GRNN results using different smoothing factors.[a]

σ	R^2 of Predicted vs. Actual					Avg. R^2	Elapsed Time[b]
0.05	0.9702	0.9946	0.9536	0.9449	0.9471	0.9621	5.2021
0.1	0.9188	0.8965	0.9227	0.9194	0.9013	0.9117	5.6713
0.3	0.7546	0.7466	0.7375	0.7323	0.7116	0.7365	5.3201
0.5	0.6466	0.6988	0.6533	0.6635	0.6959	0.6716	5.4735
1	0.4840	0.5125	0.4922	0.5003	0.5006	0.4979	5.5726

[a] Five training times for each training process.
[b] The elapsed time (s) indicates the time of one training. The computer had an Intel Pentium 3.0G processor and 3.0 Gb RAM.

Statistical Performance of Predictive Models

The statistical performance of the developed predictive models are 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 (*A* -

A_{min})/(A_{max} - A_{min}) and (T - T_{min})/(T_{max} - T_{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 (R^2, MAE, and RMSE) was within 2% change in every case. The results shown here were derived from the best network after the tests.

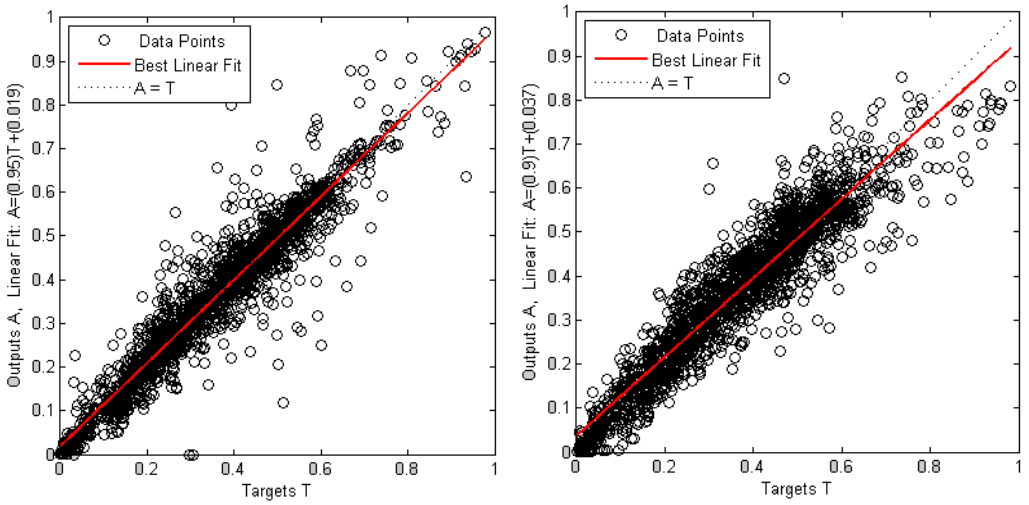
Table 6. Statistical performance of developed predictive models.

Model[a]	Number of Data Points[b]	GRNN			BPNN		
		R^2	MAE	RMSE	R^2	MAE	RMSE
NH_3Con (ppm)	8048	0.9946	1.92	3.12	0.9074	2.60	3.58
NH_3ER (kg d^{-1})	7973	0.9774	0.80	1.35	0.8825	1.38	2.12
H_2SCon (ppb)	7479	0.9167	102.54	181.37	0.8281	158.01	227.03
H_2SER (kg d^{-1})	7366	0.9258	0.08	0.14	0.8115	0.13	0.19
CO_2Con (ppm)	8500	0.9838	184.20	302.78	0.9785	242.93	376.19
CO_2ER (kg d^{-1})	8215	0.9410	144.02	217.29	0.8691	159.51	223.23
PM_{10}Con (μg m^{-3})	9187	0.8570	125.52	241.60	0.7726	180.86	290.40
PM_{10}ER (kg d^{-1})	9289	0.8719	0.07	0.14	0.6689	0.09	0.16

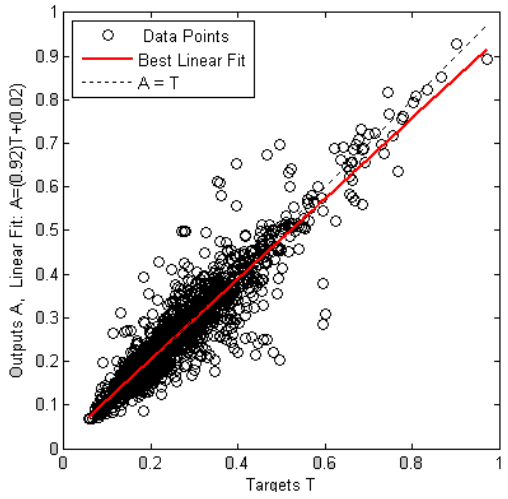
[a] Con and ER indicate the concentrations and emission rates, respectively.
[b] Indicates the number of total hourly averaged data points. The test data for the predictive models were 25% of the total data.

All the GRNN and BPNN predictive models, except for the PM_{10} concentration and emission BPNN models, had excellent predicting abilities with high R^2 values (81.15% to 99.46%) and low MAE and RMSE values, which implies that these models were well-developed (table 6). The high R^2 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).

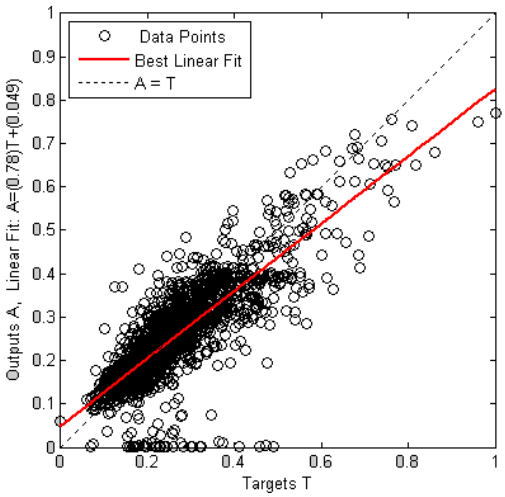
All the GRNN predictive models had higher R^2 values and lower MAE and RMSE values than the BPNN models. This demonstrates that the GRNN models outperformed the BPNN models. Thus, the GRNNs were able to predict diurnal and seasonal gas and particulate matter concentrations and emissions more effectively.



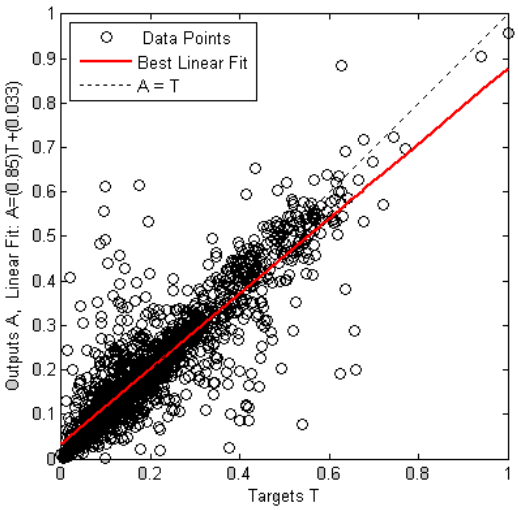
(a) NH_3Con- GRNN (b) NH_3Con- BPNN



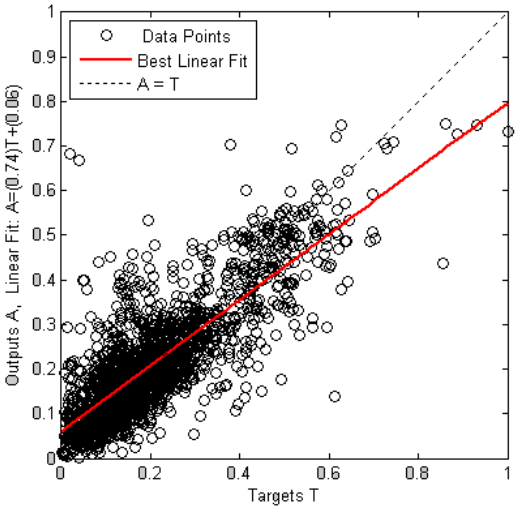
(c) NH_3ER- GRNN



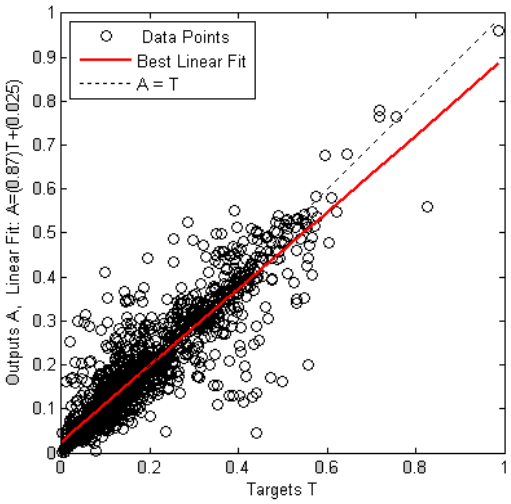
(d) NH_3ER- BPNN



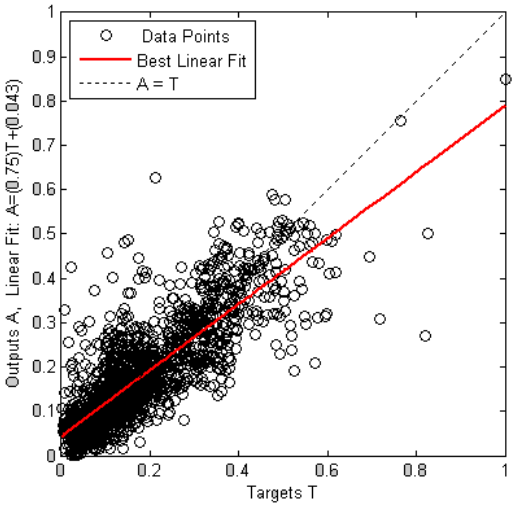
(e) H_2SCon- GRNN



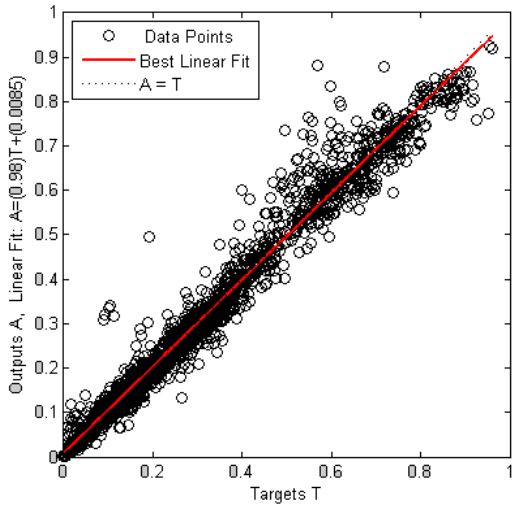
(f) H_2SCon- BPNN



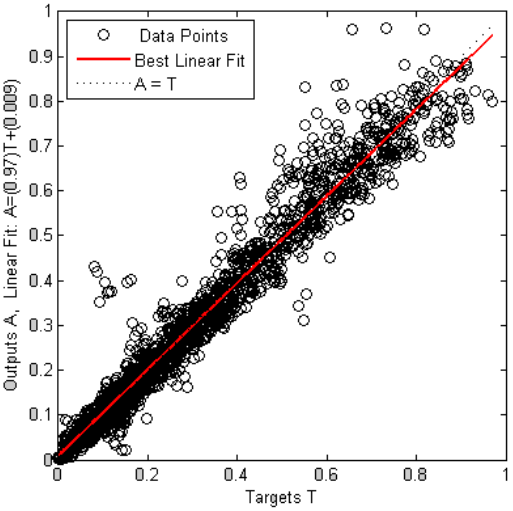
(g) H_2SER- GRNN



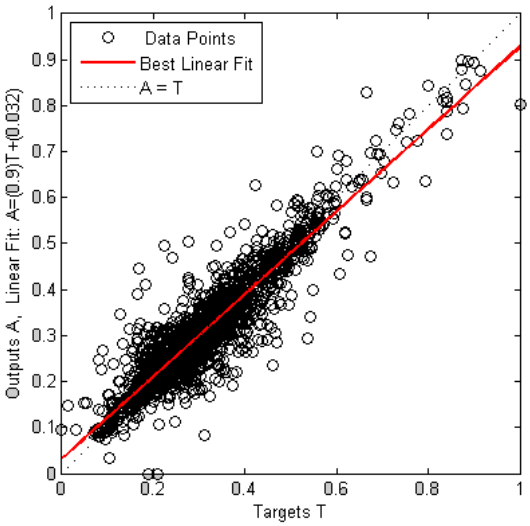
(h) H_2SER- BPNN



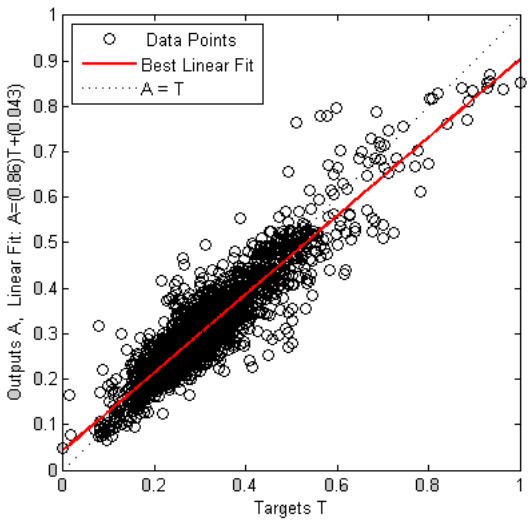
(i) CO_2Con- GRNN



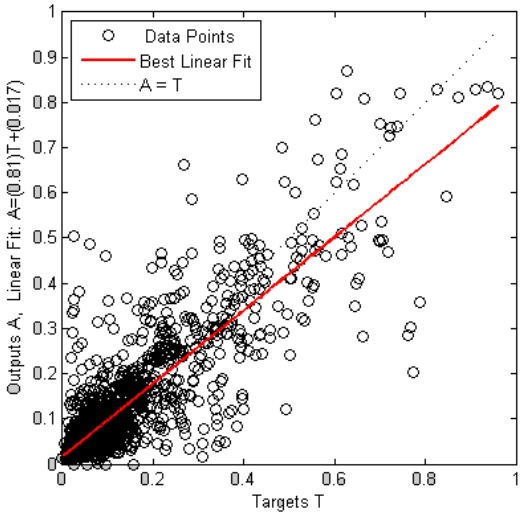
(j) CO_2Con- BPNN



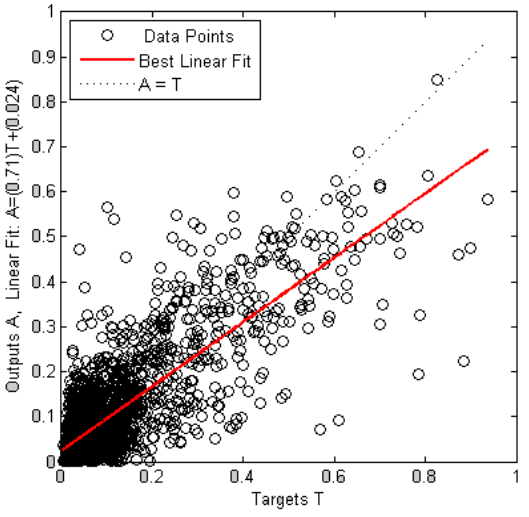
(k) CO_2ER- GRNN



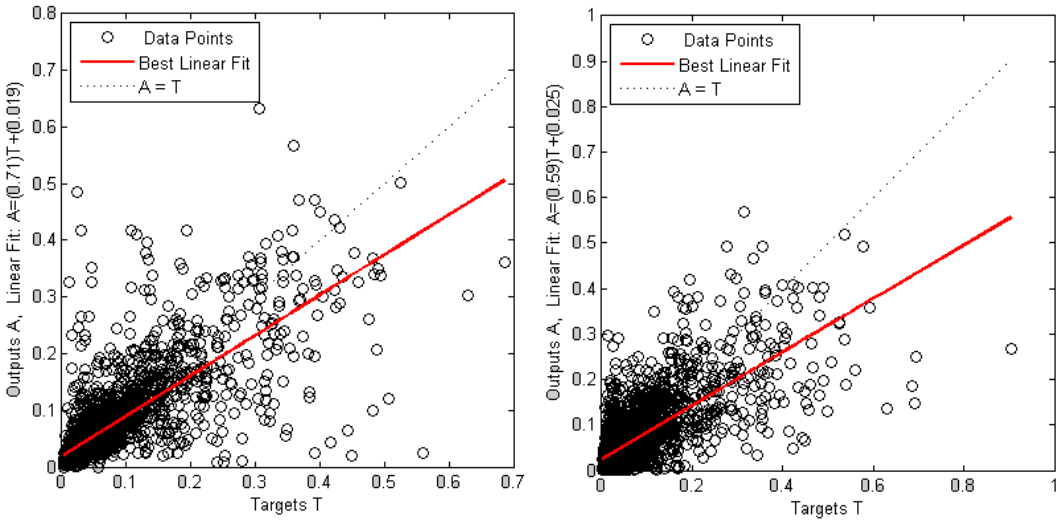
(l) CO_2ER- BPNN



(m) PM10Con- GRNN



(n) PM10Con- BPNN



(o) PM10ER- GRNN

(p) PM10ER- BPNN

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).

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_3, H_2S, CO_2, and PM_{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.

Acknowledgements

The authors would like to thank the USDA-IFAFS program for funding the research that resulted in the calibration data presented in this paper. The six-state APECAB project (Aerial Pollutant Emissions from Confined Animal Buildings) was directed by Dr. Larry Jacobson, University of Minnesota, and Dr. Albert Heber, Purdue University.

References

Aarnink, A. J. A., and A. Elzing. 1998. Dynamic model for ammonia volatilization in housing with partially slatted floors, for fattening pigs. *Livestock Prod. Sci.* 53(2): 153-169.

Arogo, J., P. W. Westerman, and A. J. Heber. 2003. A review of ammonia emissions from confined swine feeding operations. *Trans. ASAE* 46(3): 805-817.

Comrie, A. C. 1997. Comparing neural networks and regression models for ozone forecasting. *J. Air Waste Mgmt. Assoc.* 47(6): 653 663.

Kai, P., B. Kaspers, and T. van Kempen. 2006. Modeling source of gaseous emissions in a pig house with recharge pit. *Trans. ASABE* 49(5): 1479-1485.

Kasabov, N. K. 1998. *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, Mass.: MIT Press.

Kecman, V. 2001. *Learning and Soft Computing*. Cambridge, Mass.: MIT Press.

Grivas, G., and A. Chaloulakou. 2006. Artificial neural network models for prediction of PM_{10} hourly concentrations, in the greater area of Athens, Greece. *Atmos. Environ.* 40(7): 1216-1229.

Guo, H., W. Dehod, J. Agnew, J. R. Feddes, C. Lague, and S. Pang. 2007. Daytime odor emission variations from various swine barns. *Trans. ASABE* 50(4): 1365-1372.

Heber, A. J., J. Ni, T. T. Lim, P. Tao, A. M. Schmidt, J. A. Koziel, D. B. Beasley, S. J. Hoff, R. E. Nicolai, L. D. Jacobson, and Y. Zhang. 2006. Quality assured measurements of animal building emissions: Gas concentrations. *J. Air and Waste Mgmt. Assoc.* 56(10): 1472-1483.

Hoff, S. J., D. S. Bundy, M. A. Nelson, B. C. Zelle, L. D. Jacobson, A. J. Heber, J. Ni, Y. Zhang, J. A. Koziel, and D. B. Beasley. 2006. Emissions of ammonia, hydrogen sulfide, and odor before, during, and after slurry removal from a deep-pit swine finisher. *J. Air Waste Mgmt. Assoc.* 56(5): 581-590.

Hooyberghs, J., C. Mensink, G. Dumont, F. Fierens, and O. Brasseur. 2005. A neural network forecast for daily average PM_{10} concentrations in Belgium. *Atmos. Environ.* 39(18): 3279-3289.

Jacobson, L. D., H. Guo, D. R. Schmidt, R. E. Nicolai, J. Zhu, and K. A. Janni. 2005. Development of the OFFSET model for determination of odor-annoyance-free setback distances from animal production sites: Part I. Review and experiment. *Trans. ASABE* 48(6): 2259-2268.

Lim, T. T., A. J. Heber, and J. Ni. 2000. Odor setback guideline for swine production systems. In *Odors and VOC* Emissions 2000. Alexandria, Va.: Water Environment Federation.

Ni, J., J. Hendriks, C. Vinckier, and J. Coenegrachts. 2000. Development and validation of a dynamic mathematical model of ammonia release in pig house. *Environ. Intl*. 26(1-2): 97-104.

Sousa, S. I. V., F. G. Martins, M. C. M. Alvim-Ferraz, and M. C. Pereira. 2007. Multiple linear regression and artificial neural network based on principal components to predict ozone concentrations. *Environ. Modeling and Software* 22(1): 97-103.

Specht, D. F. 1991. A general regression neural network. *IEEE Trans on Neural Networks* 2(6): 568-576.

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.

Sun, G., H. Guo, J. Peterson, B. Predicala, and C. Laguë. 2008a. Diurnal Odor, Ammonia, Hydrogen Sulfide and Carbon Dioxide Emission Profiles of Confined Swine Grower/Finisher Rooms. *J. Air Waste Mgmt. Assoc*. (accepted).

Sun, G., S. J. Hoff, B. C. Zelle, and M. A. Smith. 2008b. Forecasting daily source air quality using multivariate statistical analysis and radial basis function networks. *J. Air Waste Mgmt. Assoc*. (accepted).

Zhang, R. H., T. R. Rumsey, J. R. Fadel, J. Arogo, Z. Wang, G. E. Mansell, and H. Xin. 2005. A process-based ammonia emission model for confinement animal feeding operations: Model development. In *Proc. 14th Intl. Emission Inventory Conf.* Research Triangle Park, N.C.: U.S. EPA, Office of Air Quality Planning and Standards.

Nomenclature

AU = animal units

BP = backpropagation

BPNN = backpropagation neural network

Con = concentration

ER = emission rate

GPCER = gas and PM_{10} 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

R^2 = coefficient of determination

tansig = tangent sigmoid transfer function

T_{in} = indoor temperature (°C)

T_{out} = outdoor temperature (°C)

trainbfg = BFGS quasi-Newton BP training algorithm

traincgf = conjugate gradient BP with Fletcher-Reeves updates training algorithm

traincgp = conjugate gradient BP with Polak-Ribiére updates training algorithm

traingd = gradient descent BP training algorithm

traingda = gradient descent BP with adaptive learning rate training algorithm

traingdx = gradient descent BP with momentum and adaptive learning rate training algorithm

trainlm = Levenberg-Marquardt BP training algorithm

trainoss = one step secant BP training algorithm

trainrp = resilient BP training algorithm

trainscg = scaled conjugate gradient BP training algorithm

VR = ventilation rates (m^3 s^{-1})