Available online at www.sciencedirect.com
ScienceDirect
Journal of Environmental Sciences 2010, 22(6) 851–857
JOURNAL OF ENVIRONMENTAL SCIENCES
ISSN 1001-0742
CN 11-2629/X
www.jesc.ac.cn

Assessment of physically-based and data-driven models to predict microbial water quality in open channels

Minyoung Kim[1], Charles P. Gerba[2], Christopher Y. Choi[3,*]

1. Agricultural Safety Engineering Division, Department of Agricultural Engineering, National Academy of Agricultural Science, Rural Development Administration, 249 Seodun-dong, Gwonson-gu, Suwon, 441-707, Korea. E-mail: mykim75@korea.kr
2. Department of Soil, Water, and Environmental Science, the University of Arizona, Rm 429 Shantz Bldg #38, P.O. Box 210038, Tucson, AZ 85721, USA
3. Department of Agricultural and Biosystems Engineering, the University of Arizona, 1177 E. Fourth Street Shantz Bldg. #38, Rm 403, Tucson, AZ 85721, USA

Received 17 October 2009; revised 28 December 2009; accepted 08 January 2010

Abstract

In the present study, a physically-based hydraulic modeling tool and a data-driven approach using artificial neural networks (ANNs) were evaluated for their ability to simulate the fate and transport of microorganisms in a water system. To produce reliable data, a pipe network was constructed and a series of experiments using a fecal coliform indicator (*Escherichia coli* 15597) was conducted. For the physically-based model, morphological (pipe size, link length, slope, etc.) and hydraulic (flow rate) conditions were used as input variables, and for ANNs, water quality parameters (conductivity, pH, and turbidity) were used. Both approaches accurately described the fate and transport of microorganisms (physically-based model: correlation coefficient (R) in the range of 0.914 – 0.977 and ANNs: R in the range of 0.949 – 0.980), with the exception of one case at a low flow rate (q = 31.56 cm^3/sec). This study also indicated that these approaches could be complementarily utilized to assess the vulnerability of water facilities and to establish emergency plans based on hypothetical scenarios.

Key words: transport; open channel; artificial neural networks; *Escherichia coli*

DOI: 10.1016/S1001-0742(09)60188-1

Introduction

Several studies have attempted to identify a system that can reliably monitor for and control against the accidental or intentional release of biological agents into public water systems (Ostfeld and Salomons, 2004). A great knowledge of the behavior and survival of microbial contaminants in piped systems will enable us to establish reliable assessments of the risks posed by the contaminants and the design of efficient and effective techniques to mitigate these problems (Garsdal et al., 1995).

Predictive water quality models are generally divided into two categories: those which are physically-based and those which are data-driven. The traditional hydraulic-modeling approach is often termed "physically-based modeling" (or "knowledge-driven modeling") because these models aim to explain the underlying processes that determine water quality and flow. In contrast, data-driven models are based on a limited knowledge of the process and rely on data to describe input and output characteristics (Solomatine, 2002).

MOUSE, one of the physically-based models, was developed by the Danish Hydraulic Institute (Hoersholm, Denmark). This software package models comprehensive surface runoff, open channel flow, pipe flow, water quality, and sediment transport in urban drainage systems, storm water sewers, and sanitary sewers (Roysted et al., 1999). It has been used to study combined, sanitary, and complex sewer overflows, and it has also aided in designing new site developments, in creating regulatory consenting procedures, and in diagnosing existing storm water and sanitary sewer systems. Of the MOUSE modules available, one, a deterministic modeling tool called MOUSE TRAP (Transport of Pollutants), has proved particularly useful in this study (DHI, 2003).

Although physically-based models are more widely applicable, the user must know all underlying physical processes to operate any of these software packages successfully (Bhattacharya and Solomatine, 2000). The large amount of specific data required, much of which is not readily available, also impedes the application of these models. In addition, physically-based models can often only approximate complicated real-world situations (Arnell, 1996; Kite et al., 1996). Therefore, as an alternative approach, researchers have sought data-driven modeling programs, and in the last decade these programs have become popular because sufficient data has become

* Corresponding author. E-mail: cchoi@ag.arizona.edu

available. Today, Artificial Neural Networks (ANNs) in particular appears to be the most popular data-driven modeling method.

ANNs is an information processing system composed of many nonlinear, densely interconnected processing elements, or neurons, that are arranged in groups called layers. The basic structure of ANNs usually consists of three layers: the input layer, wherein the data are introduced to the network; the hidden layer(s), wherein the data are processed; and the output layer, wherein the results of both given and processed input parameters are produced. To establish the interconnections among neurons, known inputs and outputs are presented to the ANNs to train them in an ordered manner. ANNs have been used extensively to automate many processes, from grading potatoes to matching fingerprints (Chtioui et al., 1999; Michaelides et al., 2001).

The present study has the following objectives: (1) investigating the selection of appropriate input and output parameters in MOUSE and ANN models; (2) evaluating their ability to predict microbial water quality (as compared to predictions based on field-data); and (3) discovering their potential use in designing decision-making tools.

1 Materials and methods

1.1 Description of field study

A laboratory-scale model of a water system was constructed at the Agricultural Research Center of the University of Arizona in Tucson, AZ, USA. This system was designed to generate experimental data sets that could then be used to validate the computational results. The connected main and sub-main (10.16 cm diameter) were constructed from PVC pipes. A water-feeding pipe (1.27 cm diameter), made from the same material, was installed in the ground and connected to each inlet point. The overall slope of the system was uniformly set to 0.7% to create gravity-driven flow. The layout and geometry of this pipe system is described in Fig. 1.

A full-featured software program named LoggerNet (Campbell Scientific Inc., Logan, Utah, USA) was used to facilitate programming, communications, and data retrieval between the datalogger and a computer. Solenoid valves, controlled by a datalogger, generated steady-state flow patterns (31.56, 63.10, and 94.65 cm^3/sec) over time. Prior to the microbial transport study, a tracer test using NaCl was conducted to characterize the hydraulic and transport properties, and to quantify NaCl, a CS547A-L conductivity (EC) probe (San Diego, USA) was used.

Escherichia coli ATCC 15597 (American Type Culture Collection, Sparks, MD, USA) was used to represent a fecal indicator bacterium in each experiment. A total of 500 mL of the *E. coli* suspension (with a predefined concentration of $\approx 10^8$ colony forming units per milliliter (cfu/mL)) was pump-injected into each inlet point. Each injection lasted about 40 sec. Continuous water sampling at the outlet was conducted over time, and samples were

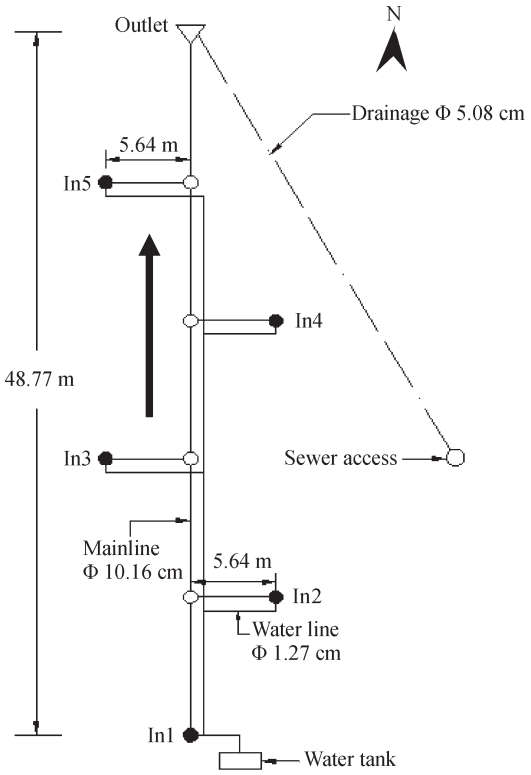

Fig. 1 Schematic of experimental setup with five inlets (In1–In5) for feeding water and injecting the sodium chloride and *E. coli* solution and one outlet for collecting water samples.

diluted and cultured onto plates of Eosin Methylene Blue Agar Levine using the spread plate method (APHA, 1989). After an incubation period (overnight) at 35°C, the *E. coli* on each plate were counted. The same water samples were used for pH and turbidity readings using a Corning 445 pH meter and HACH 2100AN turbidimeter (San Diego, CA, USA), respectively. Breakthrough curves were obtained by collecting and analyzing water samples over time at an outlet.

1.2 Physically-based model: MOUSE

Network information, hydraulic data, and boundary conditions were the three of the input parameters used for computation in MOUSE HD (hydrodynamics) and AD (advection-dispersion) modules. The physical dimensions of each component (such as pipe diameter, length, slope, etc.) were measured in the field and imported into MOUSE.

The temporal and spatial distributions of computed pipe flow discharges, water levels, and cross-sectional flow areas were computed by MOUSE HD to provide data that a MOUSE AD sub-module needed to simulate the fate and transport of *E. coli* in MOUSE TRAP. The die-off of *E. coli* due to environmental and physical variations was considered over time. However, as expected, the effect of those variations was slight because the transport distance and reaction time were short. AD module computations met the boundary conditions at all external boundaries (i.e.,

the time-variant *E. coli* and NaCl concentrations and the various flow rate conditions at each inlet point).

Manning's number was originally set as a default value (0.0125) for the PVC pipes, but this value was then adjusted so as to better represent the pipe conditions, which were smooth and clean. The number of computational nodes and the dispersion coefficient were experimentally determined by matching simulated and measured *E. coli* concentration curves.

1.3 Data-driven model: Artificial neural networks

The basic structure of ANNs usually consists of three layers: the input layer, where the data are introduced to the network; the hidden layer(s), where the data are processed; and the output layer, where the results of both given and processed input parameters are produced. The interconnection between neurons is accomplished by using known inputs and outputs which are then presented to the ANNs to train them in an ordered manner. ANNs have been used extensively to automate many processes, from grading potatoes to matching fingerprints (Chtioui et al., 1999; Michaelides et al., 2001).

This study introduced two types of neural network algorithms and compared their performance in simulating the microbial distribution within an open channel system. The two models analyzed were Backpropagation (BP) and generalized regression neural network (GRNN). The unique approaches of these two models have been used in many applications, such as function estimation, pattern recognition, clustering, forecasting, optimization, association, and control (Geeraerd et al., 1998; Hajmeer et al., 2001).

Networks trained with experimental data that adequately represent the overall characteristics of the critical physical processes will achieve a higher generation ability. To accomplish this goal, the 495 data points used in this study (each 55 data points of *E. coli*, pH, EC under three flow rates) were divided into three subsets: a training set (72% of the data), a validation set (8% of the total), and a test set (20% of the total). In this study, the k-fold cross-validation method was used to improve the generation of the network and prevent over-fitting (Bishop, 1995). All data were randomly divided into $k = 12$ subsets of equal size and different network architecture and selected parameters were identified and trained. Each time one of subsets from the training set was left out during the training, but the omitted subset was used for validation. This procedure was repeated until no further decrease in error occurred among 12 subsets. After training and validation, the network architecture having the smallest error over 12 subsets was selected and then evaluated using the test set.

Each model, BP and GRNN, was created with NeuralWorks Professional II/PLUS (Carnegie, Pennsylvania, USA) software, version 5.22, and the Neural Network Toolbox for MATLAB (Natick, Massachusetts, USA) software, respectively. These packages allow users to develop their own models by providing different networks and control parameters.

1.3.1 Backpropagation

Backpropagation is based on searching an error surface (an error as a function of ANN weights) using a gradient descent for points with a minimum error, E (Eq. (1)).

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}(x_i) - y(x_i))^2 \tag{1}$$

where, $\hat{y}(x_i)$ and $\underline{y}(x_i)$ represent the predicted and the actual values, respectively. The term "backpropagation" refers to the way the error computed at the output side is propagated backward from the output layer. In its most common configuration, the backpropagation network has three layers: an input layer, a hidden layer, and an output layer. Time series data for pH, turbidity and conductivity were fed into an input layer, and corresponding *E. coli* concentrations were provided in an output layer.

Several factors are evaluated to achieve the best architectural performance for the neural network. These factors include the following: a number of hidden PEs (processing elements); transfer functions (linear, TanH, sigmoid, DNNA (digital neural network architecture, sine); update rules (delta-rule, normalize cumulative delta, extended delta-bar-delta, Quickpro, Maxpro, and delta-bar-delta); and the effect of a number of iterations.

The numbers of input and output PEs are usually fixed by the particular application of users, but the number of hidden PEs must be specified. Network performance can be considered a quadratic function of the number of hidden PEs; thus, a decrease in the number of PEs could result in increased performance, just as an increase reported by German et al. (1992), Basheer and Hajmeer (2000).

For preprocessing input data, three normalization methods were considered to ensure that the statistical distribution of values for each net input and output is roughly uniform. In addition, the values were scaled to match the range of the input neurons, which means that along with any other transformations performed on network inputs, each input should be normalized as well (Mendelsohn, 1993). Normalized input and output data were then fed into the initialized network to prevent larger numbers from overriding smaller ones and to prevent premature saturation of hidden nodes, which impedes the learning process. Because time-variant *E. coli* concentrations have an exceptionally large range compared to pH, turbidity, and conductivity values, the logarithmic values of *E. coli* data were taken prior to normalization.

When the neural group is provided with data, the neurons in the first layer propagate the weighted data through the hidden layers. The collective effect on each of the hidden nodes is summed up by performing the dot product of all values of input nodes and their corresponding interconnection weights. Once the net effect at one hidden node is determined, the activation at that node is calculated using a transfer function to yield an output. The amount of activation obtained represents the new signal that is then transferred forward to the subsequent layer (e.g., either the hidden or the output layer). The same procedure for calculating the net effect is repeated for each hidden node.

In this "learning" process, weights are connected and updated to generate a desired effect. The learning (update) rule is the mathematical equation that determines the increment or decrement by which the PE weights change during the learning phase. In addition, a transfer function is typically a non-linear function that transforms the weighted sum of the effective inputs to a potential output value. When designing a network, the initial transfer function applies to each layer of the network. The cycle of applying an input, calculating an output, computing an error, and changing the weights constitutes one iteration of the network. German et al. (1992) showed that a lengthy training time with excessive iterations will have a large bias component of error. The network will fit noise into the data, leading to poor simulation results. Therefore, the above governing factors were carefully examined to achieve the best performance of BP architecture by a trial-and-error method.

1.3.2 Generalized regression neural network (GRNN)

GRNN was originally proposed and developed by Specht (1991). GRNN computes its output using the "nearest-neighbor" method. In this approach, the forecast for an input vector X is the weighted average of the outputs in the training sample. The closer an input vector in the training sample is to X, the larger the weight of its corresponding output vector (Marquez and Hill, 1993). Equation (2) summarizes the GRNN logic in an equivalent nonlinear regression formula:

$$E[y|X] = \frac{\int_{-\infty}^{\infty} y f(X, y)\,\mathrm{d}y}{\int_{-\infty}^{\infty} f(X, y)\,\mathrm{d}y} \tag{2}$$

where, $E[y|X]$ is the expected value of the output y given an input vector X, y is the output predicted by GRNN, X is the input vector $(x_1, x_2, ..., x_n)$ which consists of n predictor variables, and $f(X, y)$ is the joint probability density function of X and y (Tsoukalas and Uhrig, 1997).

GRNN consists of four layers, including the input, pattern, summation, and output layers. Each data input layer (pH, turbidity, and conductivity readings) corresponds to an individual process parameter. The input layer is fully connected to the second pattern layer, where each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns. Each pattern layer is connected to the two neurons in the summation layer: S_j (numerator) and S_d (denominator), resulting in the simulated *E. coli* concentrations through an output layer which is then compared to the measured *E. coli* concentrations.

The data provided to GRNN as well as to BP were obtained from MOUSE. The pre-processing of input variables is prepared by normalizing all values. Unlike NeuralWare, which already has a function to normalize input variables, MATLAB requires an additional program such as Excel to compute the normalized values. Since the GRNN is a supervised training process, the network "learns" by examining the relationship between each input vector X (pH, turbidity and conductivity) and corresponding output y (*E. coli* concentrations) during the training step.

The smoothing factor, σ, is the most important computing parameter for the GRNN's performance. Theoretically, it is not possible to determine the true σ because the underlying parent distribution is not known. However, the optimum σ for the GRNN built from a training data set can be automatically approximated using the trial-and-error method. Therefore, once the input and output variables are normalized and a network is set, an estimated σ is used in further computations. Any mean square error (MSE) obtained can be used to choose the next estimated σ; this process is repeated until a target MSE is achieved.

2 Results and discussion

The ranges of pH and turbidity change based on their spatial (where to be injected) and temporal (when to be measured) variations were 6.850–8.395 for pH and 0.121–11.925 for turbidity. While pH values did not significantly change over time, turbidity values varied due to metabolic acids produced by *E. coli* and TSB media used to grow *E. coli*.

The standard criteria for governing factors of model performance were controlled by several parameters, including the shape of the breakthrough curve, a peak concentration point, travel time, and dispersion pattern. Based on these criteria, Manning's number was gradually changed, beginning with the default value (0.0125). With various flow conditions, a range from 0.0100 to 0.0125 was used to describe the inner pipe conditions. In addition, the number of computational nodes ranged from 10 to 32 in each branch, depending on the geometric conditions of the given span.

Overall, the computational values are in good agreement with the experimental data. However, one weakness of MOUSE is its limited capability to predict the tailing of microbial concentrations due to the interaction between the microorganisms and the pipe wall surface, which depends upon hydraulic conditions. Once a certain number of microorganisms were injected into the water system, small portions usually attached to the pipe surface and were released slowly over time (Strauss, 2004). In addition, MOUSE produced negative values with sudden increases or decreases in *E. coli* concentrations caused by numerical inputs and corresponding results, and this phenomenon is common in mathematical software. Nevertheless, MOUSE was able to simulate the amount and duration of peak concentrations of *E. coli* and to predict the commencement time of detecting *E. coli* concentrations.

Among the various parameters which govern the performance of BP, the effect of the transfer function, as one of the examples, is shown in Fig. 2. For a given flow condition (q = 63.10 cm^3/sec), all other factors (number of hidden layers, number of hidden PEs, learning rate, etc.) were fixed to see how the transfer function affected model performance. Five transfer functions were compared; it was concluded that the hyperbolic tangent (TanH) function was superior to others.

Figure 3 shows the effect of smoothing factors (σ),

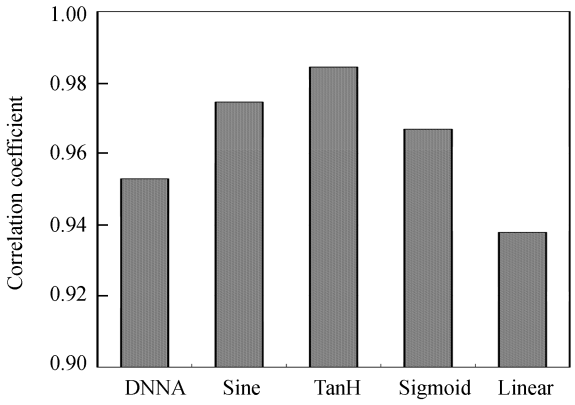

Fig. 2 Effects of transfer functions on correlation coefficient to predict *E. coli* concentrations (q = 63.10 cm^3/sec).

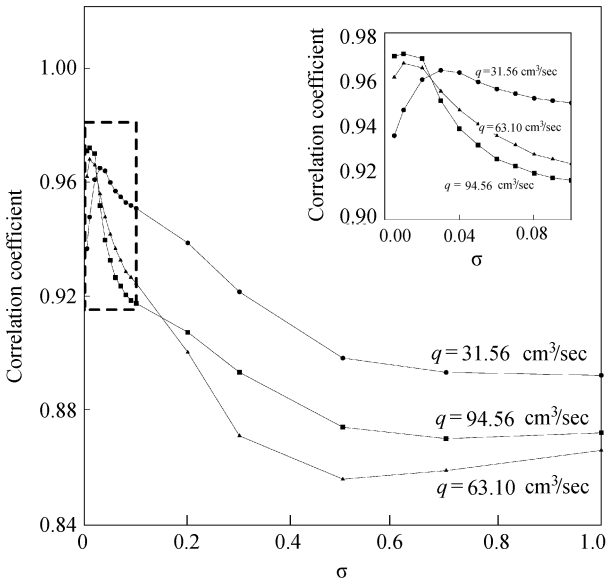

Fig. 3 Response of correlation coefficients to a change in smoothing factor (σ) in GRNN.

which is the most significant parameter in GRNN in estimating the concentration curves. The smoothing factor is significant in that, as the value of this parameter becomes smaller (or larger), the regression performed by GRNN becomes more local (or global). A range of values from 0.005 and 0.03 showed large fluctuations in correlation coefficients, but values higher than 0.03 did not improve the model performance. In addition, after σ = 1.0, correlation coefficients stayed the same. The best GRNN performance was recorded with smoothing factors of 0.03 (q = 31.56 cm^3/sec), 0.01 (q = 63.10 cm^3/sec) and 0.010 (q = 94.56 cm^3/sec), respectively.

Table 1 Compared performance of MOUSE, BP and GRNN models

Model	q (cm^3/sec)	Correlation coefficient
MOUSE	31.56	–0.509
	63.10	0.977
	94.65	0.914
BP	31.56	0.980
	63.10	0.949
	94.65	0.963
GRNN	31.56	0.964
	63.10	0.967
	94.65	0.971

Figure 4 demonstrates how well MOUSE, GRNN, and BP performed in matching the experimental curves at various flow rates. Except for one case with a low flow rate (q = 31.56 cm^3/sec) and a tailing zone, the overall pattern of each breakthrough curve is in excellent agreement with the experimental data. In a normal Y-axis (the upper-right inset of each figure), MOUSE, BP and GRNN performed well to simulate microbial spatial and temporal distribution. Logarithmic depictions were intended to emphasize the tailing phenomenon, which is a unique characteristic for microorganisms. While BP and GRNN performed the simulation effectively, MOUSE was not able to demonstrate the tailing phenomena in all three flow rates presented.

Table 1 shows the resulting performance of each model using correlation coefficients. Overall, performances between BP and GRNN under various flow conditions were not significantly different. MOUSE revealed its limitation for a low flow rate of 31.56 cm^3/sec, causing a significant error to predict *E. coli* concentration profiles, but it behaved fairly well under higher flow conditions.

3 Conclusions

The capability of physically-based and data-driven models, i.e., MOUSE and ANNs, was applied and compared to predict microbial transport through an open channel system. The physically-based model, MOUSE, requires network data, hydraulic data, and boundary conditions as part of the input variables. It is capable of accurately predicting microbial concentration patterns throughout the channel network. However, MOUSE is limited in its ability to simulate microbial tailing due to the attachment and detachment of bacteria to the pipe surface and is less accurate at a low flow rate. In contrast, for ANNs, a data-driven model, three water quality parameters (pH, turbidity, and conductivity) are used to predict *E. coli* concentrations under various hydraulic conditions. Generally, the results were consistent with the experimental data in representing microbial spatial and temporal distribution, including tailing zones. ANNs, however, require a series of data sets, either from a series of field experiments or physically-based models.

This study further revealed that these physically-based and data-driven models can be used to complement one another. A "what-if" scenario where biological agents are intentionally or accidentally introduced into water systems can be generated using MOUSE, and these computational results then allow ANNs to identify the source of pathogenic release in water systems (e.g., drinking water, sewer, and irrigation systems). Identification of the contaminant source location and time is an essential step to minimize (or isolate) the affected area to prevent further contamination and mitigate potential damage. These modeling tools can be utilized under various emergency situations with further development and validation for real-world water distribution and collection systems in conjunction with newly-developed event monitors, water

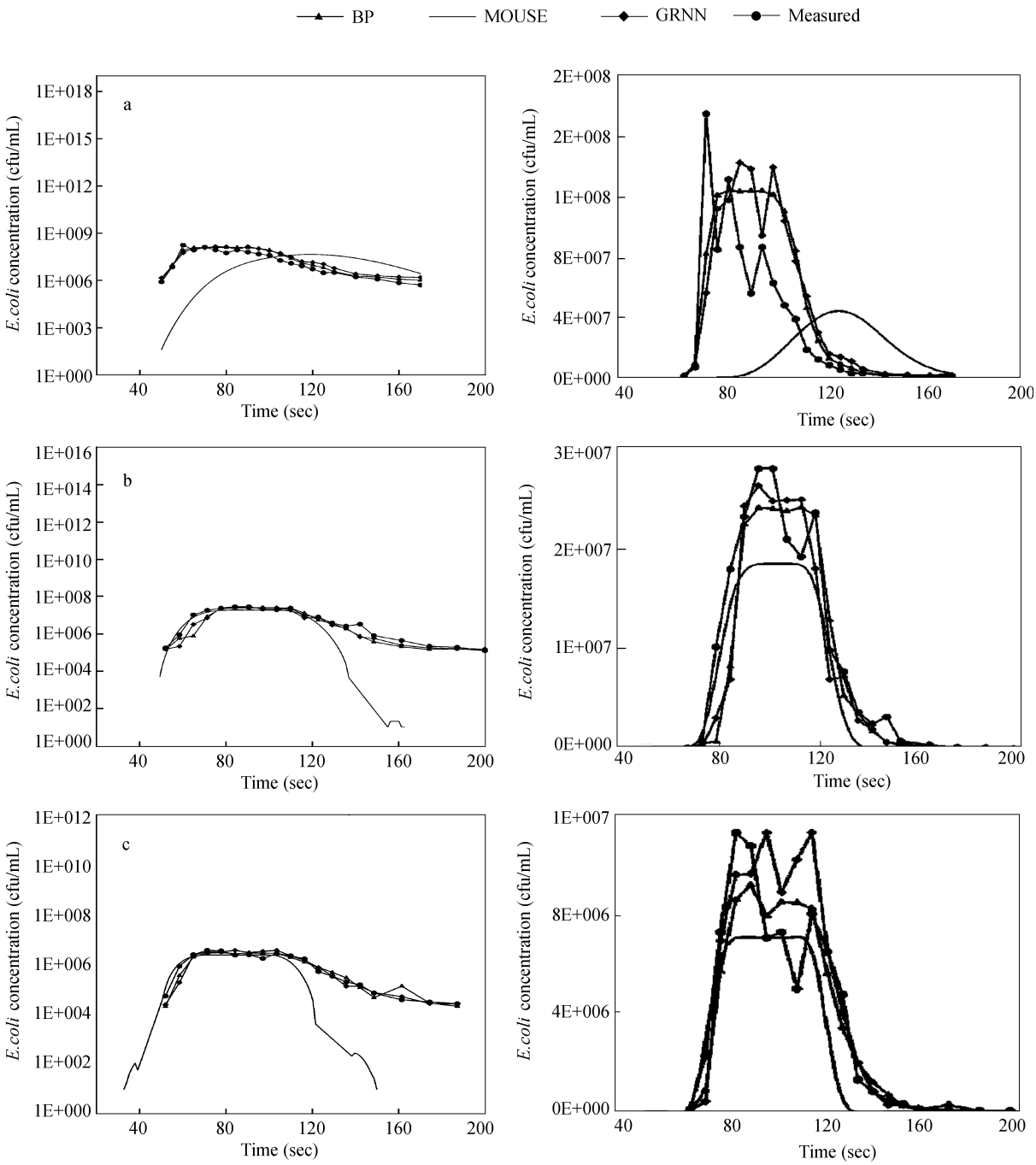

Fig. 4 Performance comparison of MOUSE, BP, and GRNN models to measured *E. coli* concentrations in logarithmic *Y*-axis under the flow rate of (a) 31.56, (b) 63.10, (c) 94.65 cm^3/sec, respectively. The right figures were depicted in a normal *Y*-axis.

quality sensors, and chemical and biological sensors.

References

APHA (American Public Health Association), 1989. Standard Methods for the Examination of Water and Wastewater (17th ed.). *American Public Health Association*. 9–61.

Arnell N W, 1996. Global Warning, River Flows and Water Resources. Wiley, Chichester.

Basheer I A, Hajmeer M, 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43: 3–31.

Bhattacharya B, Solomatine D P, 2000. Application of artificial neural network in stage-discharge relationship. In: Proceeding of the 4th International Conference on Hydroinformatics. Iowa City, IA. 1–7.

Bishop C M, 1995. Neural Networks for Pattern Recognition (1st ed.). Clarendon, Oxford.

Chtioui Y, Panigraphi S, Francl L, 1999. A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease. *Chemometrics and Intelligent Laboratory Systems*, 48: 47–58.

DHI Software (2003). MOUSE TRAP Technical Reference Advection-Dispersion Module. Portland, OR.

Garsdal H, Mark O, Dørge J, Jepsen S E, 1995. MOUSETRAP:

Modeling of Water Quality Processes and the Interaction of Sediments and Pollutants in Sewers. *Water Science and Technology*, 31(7): 33–41.

Geeraerd A H, Herremans C H, Cenens C, Van Impe J F, 1998. Application of artificial neural networks as a nonlinear modular modeling technique to describe the bacterial growth in chilled food products. *International Journal of Food Microbiology*, 44: 49–68.

German S, Bienenstock E, Doursat R, 1992. Neural networks and the bias/variance dilemma. *Neural Computation*, 4: 1–58.

Hajmeer M N, Basheer I A, Marsden J L, Fung D Y C, 2001. New approach for modeling generalized microbial growth curves using artificial neural networks. *Journal of Rapid Methods and Automation in Microbiology*, 8: 265–284.

Kite G W, Ellehoj E, Dalton A, 1996. GIS for large-scale watershed modeling. In: Geographical Information Systems in Hydrology (Singh V P, Fiorentino M, eds.). Kluwer Academic Publishers, Dordrecht, The Netherlands. 237–268.

Marquez L, Hill T, 1993. Function approximation using backpropagation and general regression neural networks. Institute of Electrical and Electronics Engineers, 607–615.

Mendelsohn L, 1993. Preprocessing Data for Neural Networks.

Michaelides S H, Pattichis C S, Kleovoulou G, 2001. Classification of rainfall variability by using Artificial Neural Networks. *International Journal of Climatology*, 21: 1401–1414.

Ostfeld A, Salomons E, 2004. Optimal layout of early warning detection stations for water distribution systems security. *Journal of Water Resources Planning and Management*, 130: 377–385.

Roysted U E, Melhuus M, Lindholm G J, 1999. Monitoring and Model Calibration for the Sewer Network in Oslo. In: 3rd DHI Software User Conference. 3–8.

Solomatine D P, 2002. Data-driven modeling: paradigm, methods, experiences. In: Proceeding of the 5th International Conference on Hydroinformatics. Cardiff, UK. 757–763.

Specht D F, 1991. A general regression neural network. IEEE Transaction on Neural Networks. 2: 568–576.

Strauss J L, 2004. Detachment of *Escherichia coli* from saturated porous media in laboratory columns. M.S. Thesis. University of New Hampshire, UK.

Tsoukalas L H, Uhrig R E, 1997. Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc., New York.