Journal of Hydrology (2008) **349**, 364–375

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jhydrol

A coupled model tree–genetic algorithm scheme for flow and water quality predictions in watersheds

Ami Preis [1], **Avi Ostfeld** *

Faculty of Civil and Environmental Engineering, Technion — I.I.T, Haifa 32000, Israel

Received 6 September 2007; received in revised form 7 November 2007; accepted 9 November 2007

KEYWORDS
Data driven modeling;
Model tree;
Genetic algorithm;
Evolutionary optimization;
Watershed;
Water quality

Summary The rapid advance in information processing systems along with the increasing data availability have directed research towards the development of intelligent systems that evolve models of natural phenomena automatically. This is the discipline of data driven modeling which is the study of algorithms that improve automatically through experience. Applications of data driven modeling range from data mining schemes that discover general rules in large data sets, to information filtering systems that automatically learn users' interests. This study presents a data driven modeling algorithm for flow and water quality load predictions in watersheds. The methodology is comprised of a coupled model tree–genetic algorithm scheme. The model tree predicts flow and water quality constituents while the genetic algorithm is employed for calibrating the model tree parameters. The methodology is demonstrated through base runs and sensitivity analysis for daily flow and water quality load predictions on a watershed in northern Israel. The method produced close fits in most cases, but was limited in estimating the peak flows and water quality loads.

Introduction

Hydrological phenomena are complex processes to describe constituting air, soil, and water, interacting on different spatial and temporal scales. Two basic approaches exist to model hydrological phenomena: physically based and data driven modeling.

A physically based approach requires descriptions of the system's input, the physical laws which govern its behavior, and boundary and initial conditions. Symbolically this can be expressed as:

$$Y(t) = \Phi[X(t)] \tag{1}$$

where t = time, $X(t)$ = input, and $\Phi[X(t)]$ = the system transformation of its input $X(t)$ to its output $Y(t)$. The characteristics of Φ classify the system properties: linear, non-linear, lumped, distributed, deterministic, stochastic, etc. Physically based modeling concentrates on constructing Φ using the physical understanding of the system. Contrary to that data driven modeling seeks to construct Φ empirically by

* Corresponding author. Tel.: +972 4 8292782; fax: +972 4 8228898.
E-mail addresses: preisa@tx.technion.ac.il (A. Preis), ostfeld@tx.technion.ac.il (A. Ostfeld).
[1] Tel.: +972 4 8292630; fax: +972 4 8228898.

0022-1694/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jhydrol.2007.11.013

Nomenclature

D_T total annual precipitation (mm/year),
$D_{t-\Delta t}$ accumulated precipitation (mm) at Δt days prior to day t,
$D^e_{t-\Delta t}(\alpha_i^*)$ accumulated effective precipitation (mm) at Δt days prior to day,
d_t precipitation (mm/day) at day t,
$G(k)$ the k-th generation,
k generation counter,
L_t water quality load at day t (kg/day),
M_t month of year corresponding to day t,
m number of days considered,
N_T^d total nitrogen daily load (kg/day),
N_T^{max} maximum total nitrogen daily load (kg/day),
n number of precipitation domain partitions for the flow model,
n' number of precipitation domain partitions for the water quality model,
P_T^d total phosphorus daily load (kg/day),
P_T^{max} maximum total phosphorus daily load (kg/day),
Q_t flow at day t ($m^3/day \times 10^3$),
Q_t^{max} maximum daily flow ($m^3/day \times 10^3$),
$T_{ave,\ t-\Delta t}$ average air temperature (°C) at Δt days prior to day t,
T_t air temperature (°C) at day t,
T_t^w week of year average temperature corresponding to day t,
t time,
W total source annual watershed water quality yield (kg/year),
W_t source watershed water quality yield (kg/day),
$X(t)$ input,
α_i i-th coefficient calibrated by a genetic algorithm for the flow model,
α_i^* i-th optimal α_i coefficient,
$\alpha_i\ [D_{t-\Delta t}]d_t$ effective daily precipitation,
β_j j-th coefficient calibrated by a genetic algorithm for the water quality model,
β_j^* j-th optimal β_j coefficient,
$\beta_j[D_t\Delta t^e(\alpha_i^*)]W_t$ effective water quality daily load,
Δt (days) time interval antecedent to day t, and
$\Phi[X(t)]$ the system transformation of its input $X(t)$ to its output $Y(t)$.

learning the ''data behavior'' of sets of $X(t)$ and $Y(t)$. A data driven modeling approach can obviously be considered only if sufficient data is available.

The methodology developed and demonstrated in this study is data driven. It is a coupled model tree–genetic algorithm (MT–GA) scheme for predicting flow and water quality constituents in watersheds. The model tree (Quinlan, 1993) constructs linear rules through learning, while the genetic algorithm (Holland, 1975) tunes the model tree parameters. The method is demonstrated through base runs and sensitivity analysis for daily flow and water quality load predictions on a watershed in northern Israel.

Literature review

Data driven modeling is the study of computer algorithms which improve automatically through experience. The most utilized techniques of data driven modeling are artificial neural networks, model trees, fuzzy-rule based systems, and support vector machines.

Since the last decade data driven modeling in both hydrology and water resources research is gaining an increasing interest. This section reviews and classifies the literature on data driven modeling for watershed hydrology into: real time flood forecasting, rainfall–runoff modeling, infilling missing data, coupled data driven–evolutionary optimization models, and water quality prediction models.

Real time flood forecasting

Aqil et al. (2007) compared the Levenberg–Marquardt feed-forward neural network, Bayesian regularization feed-forward neural network, and neuro-fuzzy techniques for flood forecasting. It was shown that no significant differences were found between the forecast accuracies of the three methods. Chen and Yu (2007) presented a model for real-time probabilistic flood stage forecasting. The methodology uses support vector regression, and a probability distribution of forecast error based on a fuzzy inference model. Chiang et al. (2007) used a recurrent neural network model for exploring the effectiveness of merging gauge observations and satellite-derived precipitation on flood forecasting. Pang et al. (2007) introduced a forecasting non-linear perturbation model based on an artificial neural network. Asefa et al. (2006) used support vector machines for hourly stream-flow forecasting. Dawsona et al. (2006) used artificial neural networks for flood forecasting in 850 ungauged catchments in the UK. Filho and dos Santos (2006) applied artificial neural networks to simulate and forecast stage level and stream-flow at the Tamanduateí river watershed in Brazil. Wang et al. (2006) used three forms of coupled artificial neural networks for daily stream-flow forecasting. Wu et al. (2005) presented an application of artificial neural networks for watershed-runoff and stream-flow forecasts. Solomatine and Xue (2004) constructed a coupled artificial neural network-model tree method for flood forecasting, showing an advantage of the coupled model over using only an artificial neural network or a model tree setup. Dorado et al. (2003) proposed an application of genetic programming linked with artificial neural networks for flood forecasting in urban basins. Sivakumar et al. (2002) used phase-space reconstruction and artificial neural networks for river flow forecasting. Coulibaly et al. (2000) used a multilayer feed-forward neural network for real-time reservoir inflow forecasting. Imrie et al. (2000) used an artificial neural network for river flow forecasting in conjunction with a guidance system to the cascade correlation artificial neural

network learning architecture. Zealand et al. (1999) used artificial neural networks for short term forecasting of stream-flow.

Rainfall–runoff modeling

Lin and Wang (2007) compared a non-linear cascaded model to artificial neural networks rainfall–runoff models. Chen and Adams (2006a,b) coupled a semi-distributed form of the Tank model (Sugawara, 1995) with an artificial neural network. Garbrecht (2006) compared the performance of three artificial neural network designs for monthly rainfall–runoff simulation on an 815 km^2 watershed in central Oklahoma. Jia and Culver (2006) compared the performance of a bootstrapped artificial neural network, a maintenance of variance extension, and a modified drainage area ratio, for ungauged watersheds synthetic flow generation. Nilsson et al. (2006) compared artificial neural networks to conceptual rainfall–runoff models for monthly runoff predictions. Ahmad and Simonovic (2005) employed artificial neural networks for predicting the peak flow and shape of a runoff hydrograph using meteorological parameters. Kingston et al. (2005) developed a methodology for incorporating information about relative input contributions in the development of artificial neural networks for rainfall–runoff modeling. Chiang et al. (2004) compared the performance of different types of neural network structures for rainfall–runoff modeling. Hsieh et al. (2004) integrated a geographical information system with an artificial neural network to quantify the similarities of watershed characteristics as an input to rainfall–runoff models. Iorgulescu and Beven (2004) used regression trees for modeling watershed rainfall–runoff relationships. Muttil and Liong (2004) applied genetic programming to generate empirically the underlying equations which connect input to output for the Upper Bukit Timah watershed in Singapore. Solomatine and Dulal (2003) investigated the accuracy performance of artificial neural networks versus model trees for rainfall–runoff modeling, showing a minor advantage to model trees. Zhang and Govindaraju (2003) developed an artificial neural network for rainfall–runoff modeling which accounts within its architecture for the geo-morphological characteristics of the watershed. Tokar and Markus (2000) compared the performance of artificial neural networks models with traditional physical conceptual models for predicting watershed runoff.

Infilling missing data

Coulibaly and Evora (2007) compared six different types of artificial neural networks methods for infilling missing precipitation data, suggesting that the multilayer perceptron network being the most effective at infilling missing daily precipitation records. Teegavarapu and Chandramoulia (2005) improved through an artificial neural network model the inverse distance weighting method for estimating missing precipitation records. Khadam and Kaluarachchi (2004) used support vector machines to reconstruct stream-flow data in a model for quantifying the relative accuracy of calibration data. Sechi and Triverio (2004) and Villafana et al. (2004) developed artificial neural network procedures for input data preparation for watershed runoff prediction.

Coupled data driven–evolutionary optimization models

Chau (2006) developed a particle swarm optimization model to train a neural network for predicting river water levels. Abedini and Nasseri (2004) and Jain and Srinivasulu (2004) developed coupled artificial neural network–genetic algorithm schemes to train neural networks for flow predictions in watersheds.

Water quality prediction models

Amenu et al. (2007) improved the traditional artificial neural network back-propagation feed-forward method by applying the adaptive minimal resource allocation network methodology. The model was applied to an agricultural watershed in central Illinois for predicting daily runoff and nitrate–nitrogen concentration. Shrestha et al. (2007) developed a fuzzy-rule based model for simulating watershed scale nitrate transport. The model uses simulated flows generated by WaSiM-ETH (Schulla and Jasper, 2001) upon which a fuzzy-rule based nitrate transport model was built using simulated annealing. Sahoo et al. (2006) used artificial neural networks to assess flash floods and their attendant water quality parameters using measured data of a Hawaii stream.

Among the models reviewed above, the study of Shrestha et al. (2007) is the closest to this work. This work differs from Shrestha et al. (2007) in using a data driven model for quantifying both the quantity and quality sections and in employing a genetic algorithm to scale model parameters. The two studies are alike in using heuristics (i.e., a fuzzy-rule based model versus a model tree) and evolutionary optimization (i.e., simulated annealing for deriving the fuzzy-rules versus a genetic algorithm for model tree calibration).

Methodology

The methodology is comprised of a coupled model tree–genetic algorithm scheme partitioned into two interconnected sub models for flow and water quality load prediction. This section is built of three sub sections: description of model trees, genetic algorithms, and the proposed methodology.

Model tree (MT)

A model tree is a data driven algorithm (Quinlan, 1993) built of a rule-based predictive structure using a top–down induction approach. The tree is fitted to a training data set by splitting the data into homogeneous subsets based on the data attributes. Thereafter, the tree is constructed with all training cases being predicted by the tree leaves (i.e., each leave is a linear regression model which predicts continuous values for the numerical attributes). The tree is then pruned bottom–up and transformed into a set of if–then rules which simplify the tree structure, and thus improves its ability to classify new instances.

The predictive ability of the tree is measured using a correlation coefficient for the training and validation data sets.

The correlation coefficient equals one in case of a complete fit between measured values and model tree predictions.

Genetic algorithm (GA)

Genetic algorithms (GAs) (Holland, 1975) are heuristic combinatorial search techniques that imitate the mechanics of natural selection and natural genetics of Darwin's evolution principle. The basic idea is to simulate the natural evolution mechanisms of chromosomes (represented by string structures), involving: selection, crossover, and mutation. This is accomplished by creating a random search technique that combines survival of the fittest among string structures with a randomized information exchange.

A typical form of a genetic algorithm involves three main stages: 1. *Initial population generation*: the genetic algorithm generates a bundle of strings (termed population, or generation), with each string (chromosome) being a set of values of the decision variables/optimization parameters. 2. *Computation of strings fitness*: the genetic algorithm evaluates each string's fitness (i.e., the value of the objective function that corresponds to each string). 3. *Construction of new generation*: the genetic algorithm establishes the next generation by performing: selection, crossover, and mutation, where: selection involves the process of choosing chromosomes from the current population for reproduction according to their fitness values; crossover — partial exchange of information between pairs of strings; and mutation — a random change in one of the strings locations. The genetic algorithm parameters are the population size, the mating and mutation rates, and the number of generations.

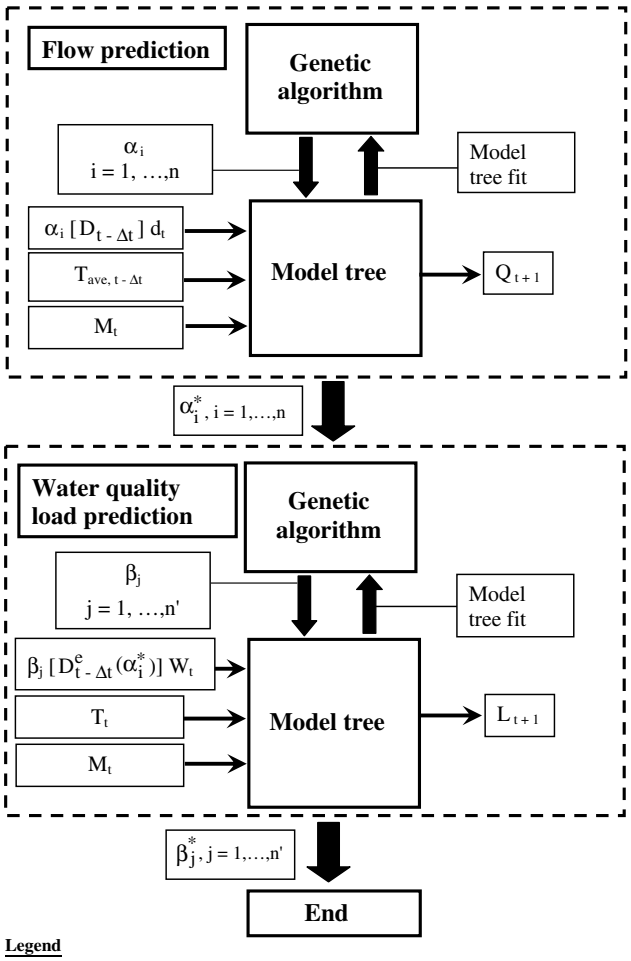
A genetic algorithm pseudo code may take the following form:

1. Initialization
   1.1 Set the generation counter k: = 0;
   1.2 Generate an initial population G (0);
   1.3 Evaluate $G(0)$.
2. Main Scheme
   Repeat
   2.1 Set: $k = k + 1$;
   2.2 Generate $G(k)$ using $G(k-1)$;
   2.3 Evaluate $G(k)$;
   Until stopping conditions are met.

In this study the following GA operations are used: *Selection* — using weighted random pairing; where the better the fitness of a chromosome, the higher is its likelihood to be selected as a parent. *Crossover* — using the one point crossover method where the offspring is a linear combination of its two parents. *Mutation* — through randomly altering one of the chromosome's parameter values. *Elitism* — the best chromosome in each generation is moved unchanged to the next.

The proposed model tree—genetic algorithm (MT—GA) methodology

A schematic description of the proposed model is shown in Fig. 1. The method is comprised of two interconnected schemes of similar structure for predicting flow and water quality loads, respectively.



Legend

n, n' = number of precipitation domain partitions for flow and water quality predictions, respectively, α_i, β_j = i-th and j-th tuned coefficients for flow and water quality load predictions, respectively, $D_{t-\Delta t}$ = accumulated precipitation (mm) at Δt days prior to day t, d_t = precipitation (mm/day) at day t, $T_{ave,\,t-\Delta t}$ = average air temperature (°C) at Δt days prior to day t, T_t = air temperature (°C) at day t, M_t = month of year corresponding to day t, W_t = source watershed water quality yield (kg/day) for day t, Q_{t+1}, L_{t+1} = flow (m^3/day) and water quality load (kg/day) prediction at day t + 1, respectively, α_i^*, β_j^* = optimal i-th and j-th tuned coefficients for flow and water quality load predictions, respectively, $D^e_{t-\Delta t}(\alpha_i^*)$ = accumulated effective precipitation (mm) at Δt days prior to day t.

Figure 1 Schematic description of the proposed methodology.

The prediction of flows (top of Fig. 1) is accomplished through a model tree whose input effective precipitation time series classifier is calibrated by a genetic algorithm. The prediction of water quality loads (bottom of Fig. 1) is attained through using the calibrated effective precipitation time series and by adjusting a vector of parameters which define the fraction of the source water quality loads contributing to the watershed outlet.

Flow prediction

The flow prediction model is comprised of a coupled model tree—genetic algorithm scheme. The model tree has three classifiers: (1) α_i $[D_{t-\Delta t}]$ d_t = effective daily precipitation, where $\alpha_i, 0 \leqslant \alpha_i \leqslant 1$ $(i = 1, \ldots, n)$ = i-th coefficient calibrated by a genetic algorithm, n = number of precipitation

domain partitions, $D_{t-\Delta t}$ = accumulated precipitation (mm) at Δt days prior to day t, and d_t = precipitation (mm/day) at day t; (2) $T_{\text{ave, } t-\Delta t}$ = average air temperature (°C) at Δt days prior to day t; and (3) M_t = month of year corresponding to day t.

For a given precipitation daily time series d_t ($t = 1, \ldots, m$), where m is the number of days considered, number of precipitation domain partition n, a set of coefficient values $\alpha_i, 0 \leqslant \alpha_i \leqslant 1$ $(i = 1, \ldots, n)$, a time interval Δt, time series of the average daily temperatures at Δt days prior to each day t, and the month associated with each day, a model tree is constructed using Cubist (cubist@www.rulequest.com). The fitness of the model tree is evaluated against the training data used to construct it and on an independent validation data set. The resulted fit which is a maximum of 2: a correlation coefficient of 1 for a complete fit of the training data set plus 1 for a complete fit of the validation data set, defines the fitness of the model tree. The genetic algorithm uses the generated fits of the model trees (i.e., a model tree for each set of coefficients α), crossover, and mutation to construct the next generation of strings of candidate α coefficients. The genetic algorithm stops if either a predefined number of generations was attained or if no improvements of the model tree fits were observed at some subsequent generations. The output of the flow model is a set of optimal coefficients α_i^* $(i = 1, \ldots, n)$.

Observations

1. The flow prediction model parameters are the vector of coefficients α, the number of precipitation domain partitions n, and the time interval Δt prior to day t. Finding the appropriate parameter values is in general case dependent. Justification of the parameter values selected for this study is provided through sensitivity analysis, as described in the example application section.
2. The rationale for using the accumulated precipitation prior to day t for predicting the effective rainfall at day t is associated with the upper soil water content. It is assumed that as the assembled precipitation prior to day t increases, so is its impact on the effective rainfall at day t. This was verified through the example application: as the accumulated precipitation prior to day t increased, so did its corresponding optimal α_i coefficient.
3. The reasoning for partitioning the accumulated precipitation prior to day t to a set of discrete domains stems from the need to constrain as of computational efficiency, the dimensionality of the genetic algorithm search space.
4. The classifier of the average air temperature at Δt days prior to day t is a surrogate to Evapotranspiration. The month of the year maps the watershed general conditions (e.g., land use and agricultural management practices, air humidity, radiation, etc.).

Water quality load prediction

As in the flow prediction model, the water quality load prediction model is comprised of a model tree coupled with a genetic algorithm. The model tree has the following classifiers: (1) $\beta_j[D^e_{t-\Delta t}(\alpha_i^*)]W_t$ = effective water quality daily load (kg/day), where: $\beta_j, 0 \leqslant \beta_j \leqslant 1$ $(j = 1, \ldots, n')$ = j-th coefficient calibrated by a genetic algorithm, n' = number of precipitation domain partitions ($n' \neq n$), $D^e_{t-\Delta t}(\alpha_i^*)$ = accumulated effective precipitation (mm) at Δt days prior to day t (i.e., the sum of effective rainfall Δt days prior to day t calculated using the α_i^* coefficients received from the flow prediction model), W_t = source watershed water quality yield (kg/day) for day t as given in Eq. (2) below, (2) T_t = air temperature (°C) at day t; and (3) M_t.

$$W_t = \frac{d_t}{D_T} W \tag{2}$$

where D_T = total annual precipitation (mm/year), W = total source annual watershed water quality yield (kg/year).

For a given source watershed water quality yield time series W_t ($t = 1, \ldots, m$), number of precipitation domain partition n', a set of coefficient values $\beta_j, 0 \leqslant \beta_j \leqslant 1$ $(j = 1, \ldots, n')$, a time series of the accumulated effective precipitation (mm) at Δt days prior to day t, time series of daily temperatures T_t ($t = 1, \ldots, m$), and the month associated with each day t, a model tree is constructed using Cubist. As in the flow prediction model, the fitness of the model tree is evaluated against the training data used to construct it and on an independent validation data set. The resulted fit which is a maximum of 2:1 for a complete fit for the training data set plus 1 for a complete fit for the validation data set, defines the fitness of the model tree. The genetic algorithm uses the generated fits of the model trees (i.e., a model tree for each set of coefficients β), crossover, and mutation to construct the next generation of candidate coefficients. The genetic algorithm stops if either a predefined number of generations was attained or if no improvements of the model tree fits were observed at some subsequent generations. The output of the water quality load prediction model is a set of optimal coefficients β_j^* $(j = 1, \ldots, n')$.

Observations

1. The water quality load prediction model parameters are the vector of coefficients β, and the number of precipitation domain partitions n' ($n' \neq n$). The time interval Δt prior to day t is assumed to be the same as in the flow prediction model.
2. The rationale for using the accumulated effective precipitation prior to day t for predicting the effective water quality daily load is associated with the flow in the watershed governed by the effective rainfall.
3. The daily air temperature at day t acts as a surrogate to water temperature which governs water quality reactions (e.g., absorption on soil). The rationale for using M_t as a classifier is similar to that described for the flow prediction model.
4. The daily source water quality watershed yield as given in Eq. (2) is an assumption for scaling down the annual water quality yield to potential daily water quality yields. Other formulations or assumptions can be used.
5. The daily effective rainfall ties the flow prediction and water quality load prediction models.

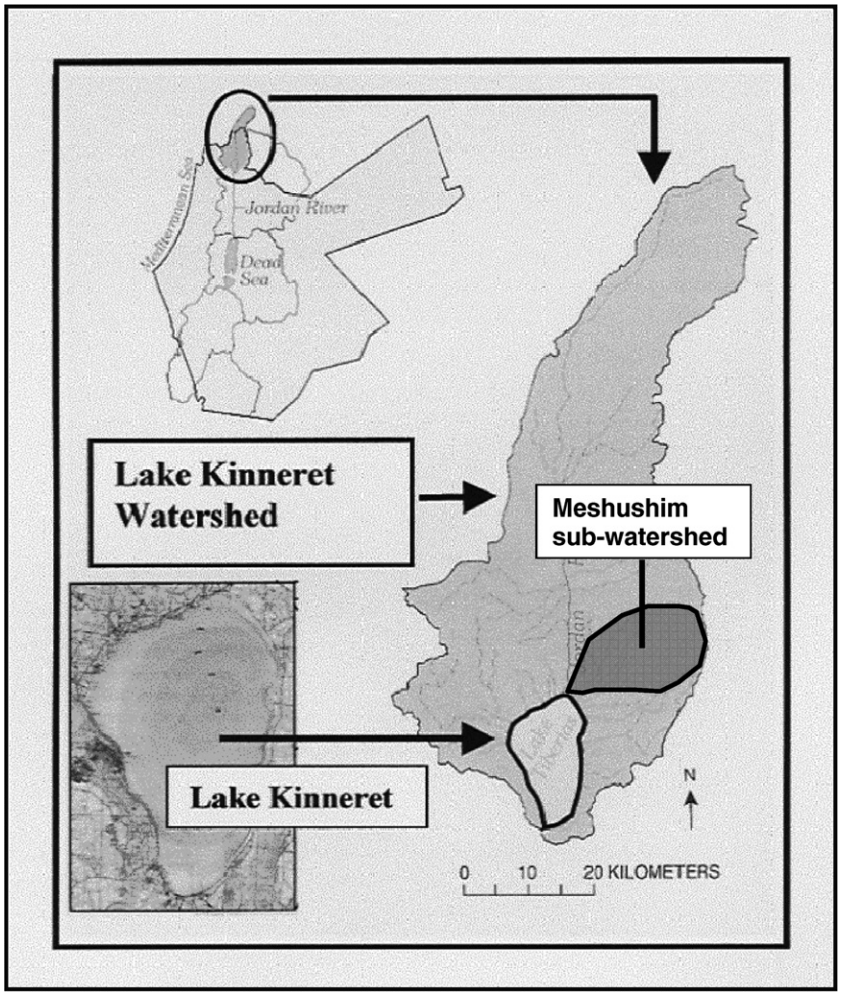


Figure 2 Meshushim sub-watershed location.

Example application

The model was applied to Meshushim watershed, a sub-watershed within the Lake Kinneret (Sea of Galilee) watershed (Fig. 2).

The Lake Kinneret watershed is about 2730 km^2 (2070 km^2 in Israel, the rest in Lebanon), inhabiting about 200,000 people, organized in twenty five municipalities, and three cities (the Israeli part).

The Meshushim watershed has an area of about 140 km^2. The soil consists of an upper layer of rocks, clay and sand. The land is mainly used for agriculture. Urban and industrial zones are located at the center of the area around the city of Katzerin. Rainfall in the basin is measured by Mekorot Company Kinneret Watershed Unit, and by the Israeli Meteorological service on a continuously basis, while monitoring of flow and water quality constituents (e.g., total nitrogen, total phosphorus, nitrate, ammonium, chloride, sodium, etc.) on a daily/weekly basis (Markel and Shamir, 2002). The database of water quality load sources used in this work is that constructed for the Israeli Water Commission by DHV MED (2001).

The data set used in this study spans the years of 1996–2001. 70% of the instances were chosen randomly for training for both the flow prediction and water quality load prediction models. The remaining 30% served for verification.

The training data set is used for calibrating the parameters of the methodology for both the genetic algorithm and the model trees. Once these parameters are set they are not altered any further. These parameters are used on the validation data set to test its performance.

Flow prediction

The results of a base run and sensitivity analysis for the flow prediction model for the Meshushim sub-watershed are summarized in Figs. 3–8.

Fig. 3 shows the outcome of a 'zero run' – a model tree construction without the genetic algorithm component, a base run, there sensitivity analysis runs, and a general analysis run.

The following can be seen from Fig. 3: (1) At the 'zero run', using the week of year average temperature corresponding to day t instead of $T_{ave,\ t-\Delta t}$, the total correlation coefficient received was the lowest (1.80), and the number of classification rules (15) – the highest. (2) At the base run a total correlation coefficient of 1.91 corresponding to nine model tree classification rules was obtained. (3) Sensitivity

Case	Input										Outcome			
	Model tree attributes			Flow prediction model parameters		Genetic algorithm parameters								
	d_t	$T_{ave,\,t-\Delta t}$	M_t	n	Δt	P	g	γ	η	G_c	R^2_T	R^2_V	R^2_{Total}	MTR
R0	+	T_t^w	+	NA	NA	NA	NA	NA	NA	NA	0.91	0.89	1.80	15
BR	+	+	+	7	7	48	7	0.25	0.07	20	0.97	0.94	1.91	9
SA1	+	+	-	7	7	48	7	0.25	0.07	20	0.97	0.93	1.90	9
SA2	+	+	+	7	7	48	7	0.25	0.07	20	0.97	0.93	1.90	**[1]5**
SA3	+	+	+	**5**	7	48	5	0.25	0.07	20	0.97	0.93	1.90	9
GAR	+	+	-	**5**	7	48	5	0.25	0.07	8	0.97	0.93	1.90	**[1]5**

[1] model tree constrained to a maximum of five model tree rules

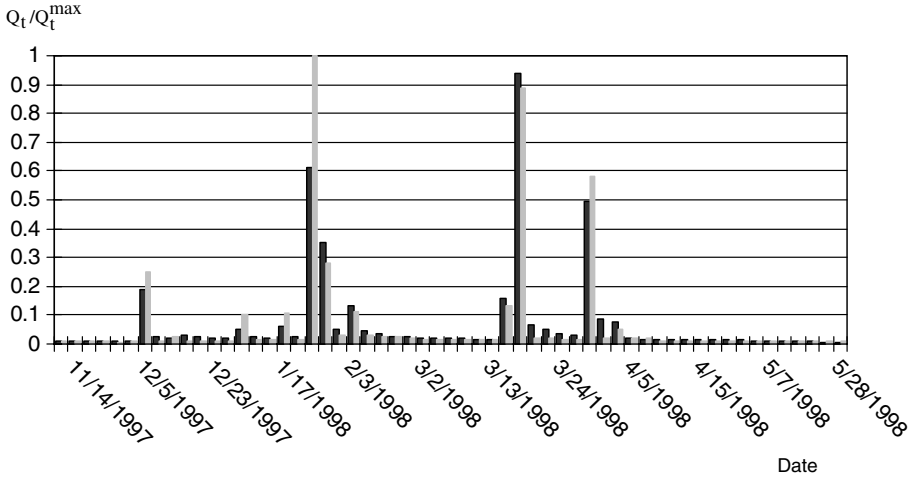
Legend

R0 = run 'zero' - a model tree construction without the genetic algorithm part (T_t^w = week of year average temperature corresponding to day t); BR = base run; SA1 = sensitivity analysis 1; GAR = general analysis run; NA = not applicable; +, - = considered, not considered, respectively; = modification compared to base run; d_t = precipitation (mm/day) at day t; $T_{ave,\,t-\Delta t}$ = average air temperature (°C) at Δt days prior to day t; M_t = month of year corresponding to day t; n = number of precipitation domain partitions; Δt = duration (days) prior to day t; P = population size; g = chromosome size; γ = mating rate; η = mutation rate; G_c = number of generations to convergence; R^2_T = training correlation coefficient; R^2_V = validation correlation coefficient; R^2_{Total} = total correlation coefficient (i.e., $R^2_T + R^2_V$); MTR = number of model tree rules.

Observations and comments

1. The α_i^* (i = 1, …,7) obtained optimal values for the base run were in the range of 0.9687 to 0.123, where the values of 0.9687 and 0.123 corresponded to the highest and lowest weekly accumulated precipitation prior to day t, respectively.
2. The computational time for one genetic algorithm generation on an IBM PC 3.6 GHz, 1GB of Ram was about 1 minute.

Figure 3 Flow prediction model base run and sensitivity analysis results.



Legend

Q_t = daily flow; Q_t^{max} = maximum Q_t among the measured and predicted data for 1997-98;
■ , ■ = measured and predicted, respectively.

Figure 4 Base run: comparison of normalized measured and predicted daily flows on validation data set for 1997—1998.

analysis: at sensitivity analysis 1 (SA1) only the precipitation (mm/day) at day *t*, and the average air temperature (°C) seven days prior to day *t*, served as the model tree classifiers. At SA2 the maximum number of model tree rules was constrained to five. At SA3 the number of precipitation domain partitions was reduced to five. The outcome of all three sensitivity analysis runs was a slight reduction in the total correlation coefficient: 1.90 compared to 1.91 at the base run. (4) At the general analysis run the modifications implemented at each of the sensitivity analysis runs were imposed simultaneously. This resulted in a total correlation coefficient of 1.90 as in SA1—SA3, but a major reduction in the computational effort: eight genetic algorithm generations to convergence compared to twenty at the base run and at SA1—SA3.

Fig. 4 presents a comparison between normalized measured and predicted daily flow values for the validation data set of 1997—98 for the base run. It can be seen from Fig. 4

that predictions versus measurements were in general in good agreement, however less satisfactorily for the high flow records. This deficiency is attributed to the relative small data set used for training. An increase in the training data set will likely improve the model prediction accuracy.

Figs. 5–8 present sensitivity analysis for the parameter values selected for the base run. Fig. 5 shows the tradeoff between the total correlation coefficient results (i.e., training correlation coefficient + verification correlation coefficient) and the employed training percentage. It can be seen from Fig. 5 that a training fraction of 64% or more gave the best total correlation coefficient results. Fig. 6 shows the tradeoff between the total correlation coefficient outcome and the number of precipitation domain partitions. It can be seen from Fig. 6 that a number of precipitation domain partitions of five or more gave the best total correlation coefficient values. Fig. 7 shows the tradeoff between the total correlation coefficient values and the antecedent duration Δt prior to day t. It can be seen from Fig. 7 that an antecedent duration of seven days gave the best total correlation coefficient outcome. Fig. 8 shows the tradeoff between the total correlation

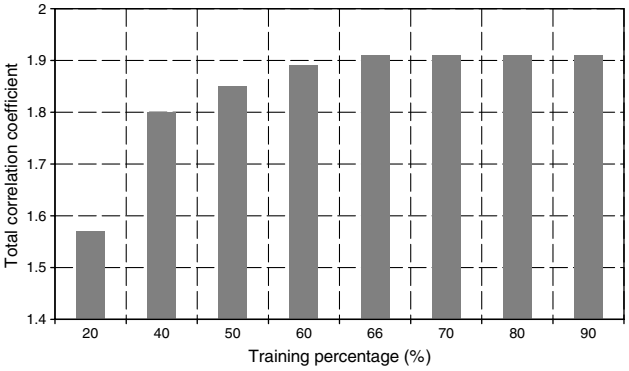


Figure 5 Base run: flow prediction model sensitivity analysis for training percentage.

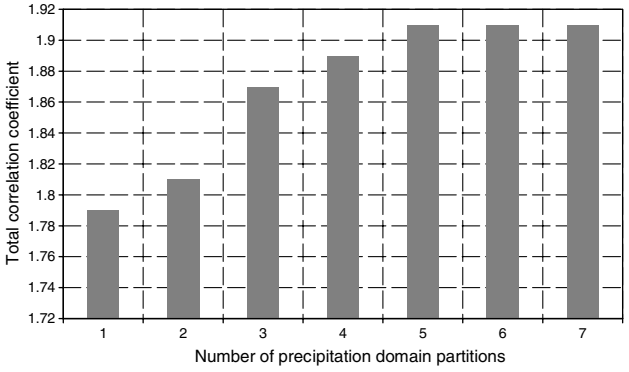


Figure 6 Base run: flow prediction model sensitivity analysis for number of precipitation domain partitions.

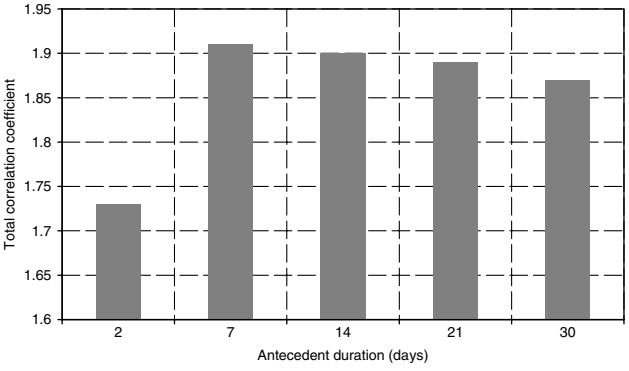


Figure 7 Base run: flow prediction model sensitivity analysis for antecedent duration (days).

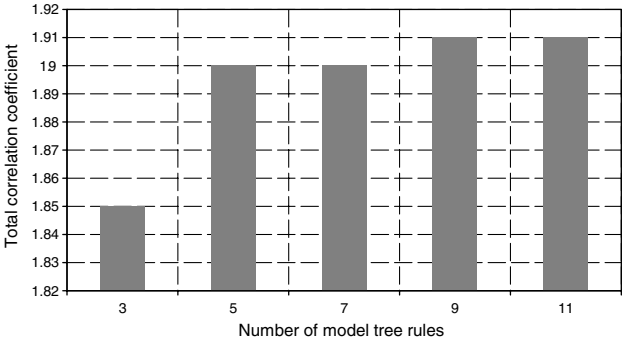


Figure 8 Base run: flow prediction model sensitivity analysis for maximum number of model tree rules.

Rule 1

If

$d_t \leq 12.5$ (mm/day) and $T_{ave,\,t-7} > 14.1$ (^oC)

Then

Q_{t+1} (m^3/day x 10^3) = 1.95 + 0.2 d_t

Rule 2

If

$d_t \leq 12.5$ and $T_{ave,\,t-7} \leq 14.1$

Then

Q_{t+1} = 9.66 + 0.7 d_t - 0.5 $T_{ave,\,t-7}$

Rule 3

If

$d_t \leq 5.9$ and $T_{ave,\,t-7} \leq 11$

Then

Q_{t+1} = - 0.52 + 0.9 d_t + 0.7 $T_{ave,\,t-7}$

Rule 4

If

$5.9 < d_t \leq 12.5$ and $T_{ave,\,t-7} \leq 14.1$

Then

Q_{t+1} = 21.36 + 0.5 d_t - 0.2 $T_{ave,\,t-7}$

Rule 5

If

$d_t > 12.5$

Then

Q_{t+1} = 15.74 - 9.3 $T_{ave,\,t-7}$ + 9.6 d_t

Figure 9 Example of classification rules for the flow prediction model for SA2 (see Fig. 3).

Case	Input										Outcome			
	Model tree attributes			Water quality load prediction model parameters		Genetic algorithm parameters								
	W_t	T_t	M_t	n'	Δt	P	[1]g	γ	η	[2]G_c	R^2_T	R^2_V	R^2_{Total}	MTR
BR	+	+	+	4	7	24	8	0.25	0.15	26	0.96	0.99	1.95	13
SA1	+	+	-	4	7	24	8	0.25	0.15	35	0.96	0.98	1.94	13
SA2	+	+	+	4	7	24	8	0.25	0.15	52	0.96	0.97	1.92	[3]5
SA3	+	+	+	6	7	24	12	0.25	0.15	170	0.96	0.99	1.95	13

[1] four/six β_j coefficients for each of the two total Nitrogen sources considered: graze, and surface runoff.
[2] computational time for one genetic algorithm generation on an IBM PC 3.6 GHz, 1GB of Ram was about 40 seconds.
[3] model tree constrained to a maximum of five model tree rules.

Legend

BR = base run; SA1 = sensitivity analysis 1; +, - = considered, not considered, respectively; = modification compared to base run; W_t = source watershed water quality yield (kg/day) for day t; T_t = air temperature (°C) at day t; M_t = month of year corresponding to day t; n' = number of precipitation domain partitions; Δt = duration (days) prior to day t; P = population size; g = chromosome size; γ = mating rate; η = mutation rate; G_c = number of generations to convergence; R^2_T = training correlation coefficient; R^2_V = validation correlation coefficient; R^2_{Total} = total correlation coefficient (i.e., $R^2_T + R^2_V$); MTR = number of model tree rules.

Figure 10 Total nitrogen water quality load prediction model base run and sensitivity analysis results.

Case	Input										Outcome			
	Model tree attributes			Water quality load prediction model parameters		Genetic algorithm parameters								
	W_t	T_t	M_t	n'	Δt	P	[1]g	γ	η	[2]G_c	R^2_T	R^2_V	R^2_{Total}	MTR
BR	+	+	+	4	7	24	12	0.25	0.15	26	0.92	0.98	1.90	11
SA1	+	+	-	4	7	24	12	0.25	0.15	44	0.92	0.96	1.88	11
SA2	+	+	+	4	7	24	12	0.25	0.15	54	0.92	0.93	1.85	[3]5
SA3	+	+	+	[2]5	7	24	15	0.25	0.15	50	0.92	0.98	1.90	11

[1] four/five β_j coefficients for each of the three total Phosphorus sources considered: graze, surface runoff, and residential-industrial.
[2] computational time for one genetic algorithm generation on an IBM PC 3.6 GHz, 1GB of Ram was about 40 seconds.
[3] model tree constrained to a maximum of five model tree rules.

Legend

BR = base run; SA1 = sensitivity analysis 1; +, - = considered, not considered, respectively; = modification compared to base run; W_t = source watershed water quality yield (kg/day) for day t; T_t = air temperature (°C) at day t; M_t = month of year corresponding to day t; n' = number of precipitation domain partitions; Δt = duration (days) prior to day t; P = population size; g = chromosome size; γ = mating rate; η = mutation rate; G_c = number of generations to convergence; R^2_T = training correlation coefficient; R^2_V = validation correlation coefficient; R^2_{Total} = total correlation coefficient (i.e., $R^2_T + R^2_V$); MTR = number of model tree rules.

Figure 11 Total phosphorus water quality load prediction model base run and sensitivity analysis results.

coefficient results and the number of maximum model tree rules. It can be seen from Fig. 8 that at least nine model tree rules are required to maximize the total correlation coefficient.

Fig. 9 is an example of a model tree outcome. The figure shows the five rules obtained for the general analysis run (GAR) (see Fig. 3). Note that rules 2 and 3 overlap for $d_t \leqslant 5.9$ (mm/day) and $T_{ave,\ t-7} \leqslant 11$ (°C). In such instances the model tree forecasts are averaged to arrive at a final prediction.

Water quality load prediction

Two water quality constituent loads were predicted: total nitrogen and total phosphorus. For the total nitrogen two sources were considered: graze — 14,144 (kg/year), and surface runoff — 42,713 (kg/year), while for the total phosphorus: graze — 2053 (kg/year), surface runoff — 2373 (kg/year), and residential—industrial — 503 (kg/year).

The results of a base run and sensitivity analysis for the total nitrogen and total phosphorus predictions are summarized in Figs. 10—13.

Figs. 10 and 11 summarize base runs and sensitivity analysis for the total nitrogen and total phosphorus predictions, respectively.

The following can be observed from Fig. 10: (1) At the base run a total correlation coefficient of 1.95 corresponding to thirteen model tree classification rules was obtained. (2) Sensitivity analysis: at SA1 only the source watershed water quality yield (kg/day) for day *t*, and the air temperature (°C)

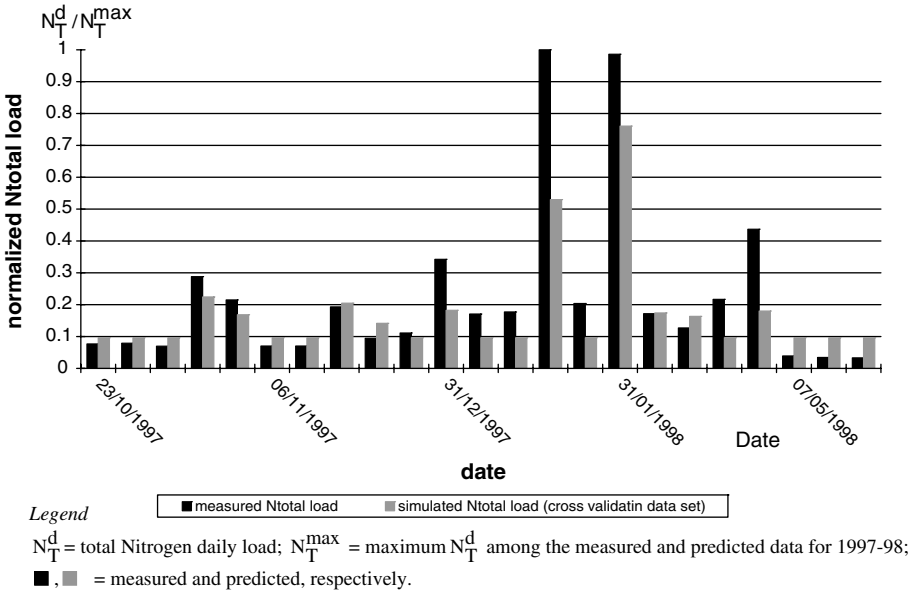


Figure 12 Base run: comparison of normalized measured and predicted total nitrogen loads on validation data set for 1997–1998.

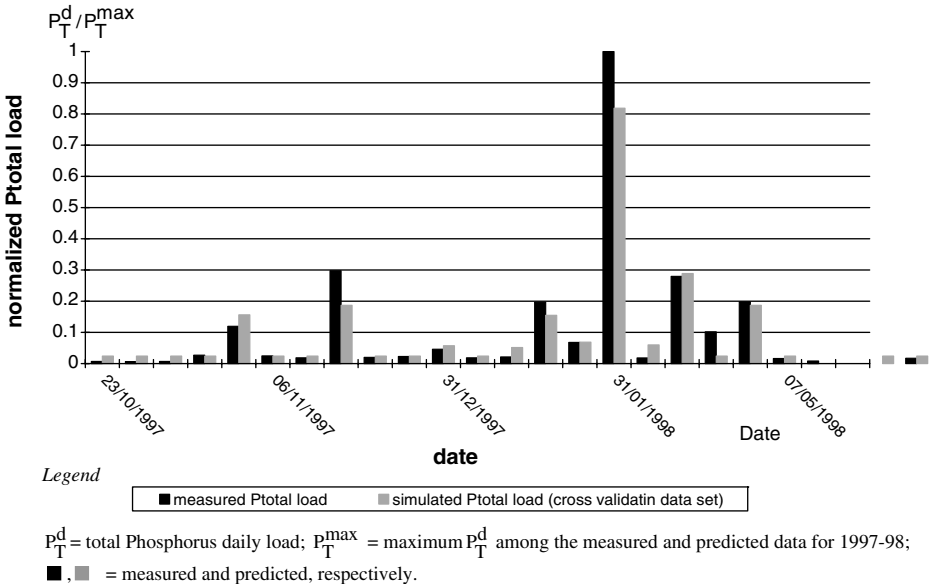


Figure 13 Base run: comparison of normalized measured and predicted total phosphorus loads on validation data set for 1997–1998.

at day *t*, served as the model tree classifiers. At SA2 the maximum number of model tree rules was constrained to five. At SA3 the number of precipitation domain partitions was increased to six. SA1 and SA2 resulted in a reduction in the total correlation coefficient of 1.94 and 1.92, respectively. At SA3 the total correlation coefficient remained unchanged, but the required computational effort increased substantially.

Similar observations can be seen for the total phosphorus load prediction, as shown in Fig. 11: a total correlation coefficient of 1.90 with 11 classification rules for the base run; reduction of the total correlation coefficient at SA1 and SA2; and the same total correlation coefficient for SA3, which required additional computational effort.

Figs. 12 and 13 show comparisons for the base run between measured and predicted values for normalized total nitrogen and total Phosphors, respectively for the validation data set of 1997–1998. It can be seen from both figures that predictions versus measurements were in general in good agreement, however less satisfactorily for the high loads. This deficiency, as observed for the flow prediction (see Fig. 4), is attributed to the relative small data set used for training. An increase in the training data set will likely improve the model prediction accuracy.

Conclusions

1. A coupled data driven (model tree)—evolutionary scheme (genetic algorithm) for flow and water quality load predictions in watersheds was developed and demonstrated. The model tree was used for prediction of flow and water quality loads while the genetic algorithm for calibrating the model tree parameters.
2. The employment of data driven modeling for simulation of complex physical systems is receiving an increasing interest as the result of the growing availability of data. Most of the developed schemes are for real-time flood forecasting and for rainfall—runoff modeling using artificial neural networks. There are almost no models which use model trees, and none which use model trees for water quality load predictions in watersheds. The advantage of using a model tree is in its outcome simplicity: a set of linear rules which can be easily accessed and potentially physically interpreted.
3. As in every data driven modeling implementation one of the most important challenges is to tie the data driven technique to the most important physical governing processes of the system. This task is extremely complex for simulating natural systems like flow and water quality behavior in watersheds. In this study it was assumed that the most important governing phenomena driving both the flow and the water quality loads is the effective rainfall.
4. The predictions received by the developed model were in general in good agreement with measurements. However, the model was less successful in predicting high flows and water quality loads. This is an inherent deficiency of a data driven technique whose accuracy is primarily dependent on the data set used for its training. The larger a data set, the higher likelihood of receiving better predictions. It is anticipated that increasing the number of training instances for the proposed model will improve its prediction accuracy.
5. Research challenges for extending this study are in more closely incorporating the system's physics. This can be accomplished for example by using a physically based model in conjunction with a data driven modeling technique. An evolutionary algorithm can tie both, yet the computational effort expected to run such a setup is anticipated to be very high. The construction of such a framework is thus not straightforward; hence, additional amendments will need to be undertaken.

Acknowledgements

This work was funded by the Israeli Water Commission, and by the Technion Grand Water Research Institute (GWRI).

References

Abedini, M.J., Nasseri, M., 2004. Spatiotemporal rainfall forecasting via ANNS coupled with GA. In: Liong, Phoon, Babovic (Eds.), Sixth International Conference on Hydroinformatics. Published on CD.

Ahmad, S., Simonovic, S.P., 2005. An artificial neural network model for generating hydrograph from hydro-meteorological parameters. Journal of Hydrology 315, 236—251.

Amenu, G.G., Markus, M., Kumar, P., Demissie, M., 2007. Hydrologic applications of MRAN algorithm. Journal of Hydrologic Engineering, ASCE 12 (1), 124—129.

Aqil, M., Kita, I., Yano, A., Nishiyama, S., 2007. A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. Journal of Hydrology 337, 22—34.

Asefa, T., Kemblowski, M., McKee, M., Khalil, A., 2006. Multi-time scale stream flow predictions: the support vector machines approach. Journal of Hydrology 318, 7—16.

Chau, K.W., 2006. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. Journal of Hydrology 329, 363—367.

Chen, J., Adams, B.J., 2006a. Integration of artificial neural networks with conceptual models in rainfall—runoff modeling. Journal of Hydrology 318, 232—249.

Chen, J., Adams, B.J., 2006b. Semidistributed form of the tank model coupled with artificial neural networks. Journal of Hydrologic Engineering, ASCE 11 (5), 408—417.

Chen, S.T., Yu, P.-S., 2007. Real-time probabilistic forecasting of flood stages. Journal of Hydrology 340, 63—77.

Chiang, Y.-M., Chang, L.-C., Chang, F.-J., 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall—runoff modeling. Journal of Hydrology 290, 297—311.

Chiang, Y.-M., Hsu, K.-L., Chang, F.-J., Yang Hong, Y., Sorooshian, S., 2007. Merging multiple precipitation sources for flash flood forecasting. Journal of Hydrology 340, 183—196.

Coulibaly, P., Anctil, F., Bobeé, B., 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. Journal of Hydrology 230, 244—257.

Coulibaly, P., Evora, N.D., 2007. Comparison of neural network methods for infilling missing daily weather records. Journal of Hydrology 341, 27—41.

Dawsona, C.W., Abrahartb, R.J., Shamseldinc, A.Y., Wilbyd, R.L., 2006. Flood estimation at ungauged sites using artificial neural networks. Journal of Hydrology 319, 391—409.

DHV MED, 2001. <http://www.dhvmed.com/GISweb2.htm>. (accessed 05.09.07).

Dorado, J., Puertas, J., Santos, A., Rivero, D., Pazos, A., Rabunal, J.R., 2003. Prediction and modeling of the rainfall—runoff transformation of a typical urban basin using ANN and GP. Applied Artificial Intelligence 17 (4), 329—343.

Filho, A.J.P., dos Santos, C.C., 2006. Modeling a densely urbanized watershed with an artificial neural network, weather radar and telemetric data. Journal of Hydrology 317, 31—48.

Garbrecht, J.D., 2006. Comparison of three alternative ANN designs for monthly rainfall—runoff simulation. Journal of Hydrologic Engineering, ASCE 11 (5), 502—505.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor.

Hsieh, B.B., Fong, M.T., Jorgeson, J.D., Skahill, B.E., 2004. Watershed similarity analysis using integration of GIS and unsupervised—supervised artificial neural networks. In: Liong, Phoon, Babovic (Eds.), Sixth International Conference on Hydroinformatics. Published on CD.

Imrie, C.E., Durucan, S., Korre, A., 2000. River flow prediction using artificial neural networks: generalisation beyond the calibration range. Journal of Hydrology 233, 138—153.

Iorgulescu, I., Beven, K.J., 2004. Nonparametric direct mapping of rainfall—runoff relationships: an alternative approach to data analysis and modeling? Water Resources Research 40, W08403. doi:10.1029/2004WR00309.

Jain, A., Srinivasulu, S., 2004. Development of effective and efficient rainfall—runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network

techniques. Water Resources Research 40, W04302. doi:10.1029/2003WR00235.

Jia, Y., Culver, T.B., 2006. Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. Journal of Hydrology 331, 580—590.

Khadam, I.M., Kaluarachchi, J.J., 2004. Use of soft information to describe the relative uncertainty of calibration data in hydrologic models. Water Resources Research 40, W11505. doi:10.1029/2003WR00293.

Kingston, G.B., Maier, H.R., Lambert, M.F., 2005. Calibration and validation of neural networks to ensure physically plausible hydrological modeling. Journal of Hydrology 314, 158—176.

Lin, G.-F., Wang, C.-M., 2007. A nonlinear rainfall—runoff model embedded with an automated calibration method — Part 1: The model. Journal of Hydrology 341, 186—195.

Markel, D., Shamir, U., 2002. Monitoring Lake Kinneret and its watershed: forming the basis for management of a water supply lake. In: Rubin, H., Nachtnebel, P., Fuerst, J., Shamir, U. (Eds.), Water Resources Quality Preserving the Quality of our Water Resources. Springer-Verlag, pp. 177—190.

Muttil, N., Liong, S.Y., 2004. Physically interpretable rainfall—runoff models using genetic programming. In: Liong, Phoon, Babovic (Eds.), Sixth International Conference on Hydroinformatics. Published on CD.

Nilsson, P., Uvo, C.B., Berndtsson, R., 2006. Monthly runoff simulation: Comparing and combining conceptual and neural network models. Journal of Hydrology 321, 344—363.

Pang, B., Guo, S., Xiong, L., Li, C., 2007. A nonlinear perturbation model based on artificial neural network. Journal of Hydrology 333, 504—516.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

Sahoo, G.B., Ray, C., De Carlo, E.H., 2006. Use of neural network to predict flash flood and attendant water qualities of a mountainous stream on Oahu, Hawaii. Journal of Hydrology 327, 525—538.

Schulla, J., Jasper, K., 2001. Model Description WaSiM-ETH. Internal Report. ETH-Zurich, Switzerland.

Sechi, G.M., Triverio, A., 2004. NNRRT: a software for artificial neural networks modeling of rainfall—runoff process. In: Liong, Phoon, Babovic (Eds.), Sixth International Conference on Hydroinformatics. Published on CD.

Shrestha, R.R., Bárdossy, A., Michael, R., 2007. A hybrid deterministic—fuzzy rule based model for catchment scale nitrate dynamics. Journal of Hydrology 342, 143—156.

Sivakumar, B., Jayawardena, A.W., Fernando, T.M.K.G., 2002. River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. Journal of Hydrology 265, 225—245.

Solomatine, D.P., Dulal, K.N., 2003. Model trees as an alternative to neural networks in rainfall—runoff modeling. Hydrological Science Journal 48 (3), 399—411.

Solomatine, D.P., Xue, Y., 2004. M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai river in China. Journal of Hydrologic Engineering, ASCE 9 (6), 491—500.

Sugawara, M., 1995. Tank model. In: Singh, V.P. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, Littleton, Colorado.

Teegavarapu, R.S.V., Chandramoulia, V., 2005. Improved weighting methods, deterministic and stochastic data-driven models for estimation of missing precipitation records. Journal of Hydrology 312, 191—206.

Tokar, A.S., Markus, M., 2000. Precipitation-runoff modeling using artificial neural networks and conceptual models. Journal of Hydrologic Engineering, ASCE 5 (2), 156—161.

Villafana, G.C., Cluckie, I.D., Han, D., 2004. An approach on evaluating the pre-processing of rainfall and runoff data in an artificial neural network model for real-time forecasting. In: Liong, Phoon, Babovic (Eds.), Sixth International Conference on Hydroinformatics. Published on CD.

Wang, W., Van Gelder, P.H.A.J.M., Vrijling, J.K., Ma, J., 2006. Forecasting daily streamflow using hybrid ANN models. Journal of Hydrology 324, 383—399.

Wu, J.S., Han, J., Annambhotla, S., Bryant, S., 2005. Artificial neural networks for forecasting watershed runoff and stream flows. Journal of Hydrologic Engineering, ASCE 10 (3), 216—222.

Zealand, C.M., Burn, D.H., Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks. Journal of Hydrology 214, 32—48.

Zhang, B., Govindaraju, R.S., 2003. Geomorphology-based artificial neural networks (GANNs) for estimation of direct runoff over watersheds. Journal of Hydrology 273, 18—34.