

# RESERVOIR INFLOW MODELING WITH ARTIFICIAL NEURAL NETWORKS: THE CASE OF KEMER DAM IN TURKEY

Umut Okkan<sup>1,\*</sup> and H. Yildirim Dalkilic<sup>2</sup>

<sup>1,\*</sup> Balıkesir University, Faculty of Engineering-Architecture, Department of Civil Engineering, Balıkesir, Turkey

<sup>2</sup> Dokuz Eylül University, Faculty of Engineering, Department of Civil Engineering, İzmir, Turkey

## ABSTRACT

It is very important to make both reservoir inflow modeling and operation studies on water resource engineering. In this paper, a comprehensive comparison on the application of two different artificial neural network algorithms in the monthly inflows of Kemer Dam, which is located in the Buyuk Menderes Basin/Turkey, was presented. Two types of neural networks, namely, feed-forward neural networks (FFNN) and generalized regression neural networks (GRNN), were examined. The best model combinations which require monthly areal precipitation, temperature and one-two months ahead areal precipitation values as the input data, was trained by using the monthly data depending on the records that spread to a time frame of 156 months, made between January 1980 - December 1992, and then tested by the 156 months of reservoir inflows, recorded between January 1993 - December 2005. When the long-term performances of the training and testing periods are compared, it was shown that GRNN approach has a better performance in the training period; on the other hand, FFNN proves itself to be more successful in the testing period. Seasonal comparisons were also examined by box-plot graphs and Mann Whitney U (M-W) non-parametric test statistics. In the results of seasonal comparing, it was shown that FFNN has the best performance in summer and autumn but GRNN in winter and spring. Besides, there were different drawbacks and advantages of these two approaches, which were also proven with this study. FFNN and GRNN algorithms are the successful black box techniques which are capable of reservoir inflow modeling without detailing the physical process.

**KEYWORDS:** Reservoir Inflow Modeling, Feed-forward Neural Networks, Generalized Regression Neural Networks, Kemer Dam

## 1. INTRODUCTION

There are numerous classifications that have been used in literature to describe the studies on river flow modeling which include system definitions, area-time scales and solution techniques [1, 2]. In general, there are three main approaches to represent a river basin system: white-box model (physically based distributed models), gray-box model (conceptual models) and the black-box model [3]. The white and gray-box models aim to simulate the physical mechanisms underlying each component in the transformation of precipitation into runoff, such as surface, subsurface and groundwater flow, infiltration, percolation, and evapotranspiration. The parameters relevant to these components for a certain river basin can be determined by different optimization techniques. Yet, due to their data requirements, uncertainties and complexities, these models may not be readily used in all applications.

A river basin can also be represented by black-box models which associate basin inputs and desired outputs without detailed considerations on the physical phenomena. In this context, conventional statistical models, such as regression analyses, curve fitting approaches and stochastic autoregressive models are commonly used. Recently, artificial neural networks (ANN), which are mathematical modeling tools inspired by the properties of the biological neural system and structured between basin inputs (precipitation, temperature, evaporation etc.) and flow data, have been used in the modeling studies.

There has been a number of ANN studies published in literature. In these studies, the rainfall-runoff relationship was modeled successfully by using ANN [4-11]. Moreover, ANN models were developed for river flow prediction [12, 13], and the performances of ANN models were compared to the other statistical methods (autoregressive modeling, regression analysis). The studies demonstrated that the results of the ANN were more precise than that of the conventional statistical methods. Furthermore, an autoregressive model was used for generating synthetic monthly flows and used as the training set of ANNs to forecast the Goksu River monthly mean flows in the East Mediterranean part of Turkey [14]. As well as

\* Corresponding author

modeling of river flows, reservoir inflows were also modeled by ANNs [15-18]. Other hydrological studies using ANN algorithms covered prediction of suspended sediment [19-23], regional flood frequency analysis [24, 25], short-term flow forecasting [26, 27], precipitation forecasting [28-30], temperature forecasting [31, 32], evaporation-evapotranspiration modeling [33-36], prediction of sanitary flows [37], groundwater applications [38-40], infiltration applications [41] and prediction of missing stream-flow data [42- 44].

ANN applications in hydrologic modeling studies generally include the feed forward neural networks. Since the feed forward neural network (FFNN) algorithms have some disadvantages relating to the presence of local minimums and the precision of assigned initial weights, generalized regression neural networks and radial basis neural networks, which are alternative techniques to FFNN, were improved and used successfully in modeling studies in order to overcome the shortcomings of FFNN algorithms [6, 11, 45-47].

In this study, a comprehensive study is presented on the application of two artificial neural network algorithms to model the monthly reservoir inflows of Kemer Dam's reservoir which is located in the Buyuk Menderes Ba-

sin/Turkey. Two types of neural networks, namely, feed-forward and generalized neural network, were examined.

## 2. MATERIALS AND METHODS

The study area covers the drainage basin of Kemer Dam, which is located in the Aegean region of Turkey (Fig. 1). The basin is fed by four rivers and stream-flow values are observed by four stream-flow gauging stations (Calikoy/EIE-730, Yemisendere/EIE-731, Degirmenalani/EIE-732, and Goktepe/EIE-733) located at the upstream of the dam (Fig. 1). These data were collected from the records of two institutes of Turkey: XXI. Regional Directorate of State Hydraulic Works, and Operational Directorate of Kemer Dam Power Plant which is a part of the Electrical Works Authority. Thus, the collected reservoir inflow data were prepared for the period between January 1980 and December 2005. In addition to inflow data, the monthly data of precipitation and temperature at Denizli and Mugla meteorological stations were obtained from the State Meteorological Organization of Turkey. Next, Thiessen weighted precipitation values and arithmetical mean temperature values were prepared for monthly time-scale, using records available at both stations.

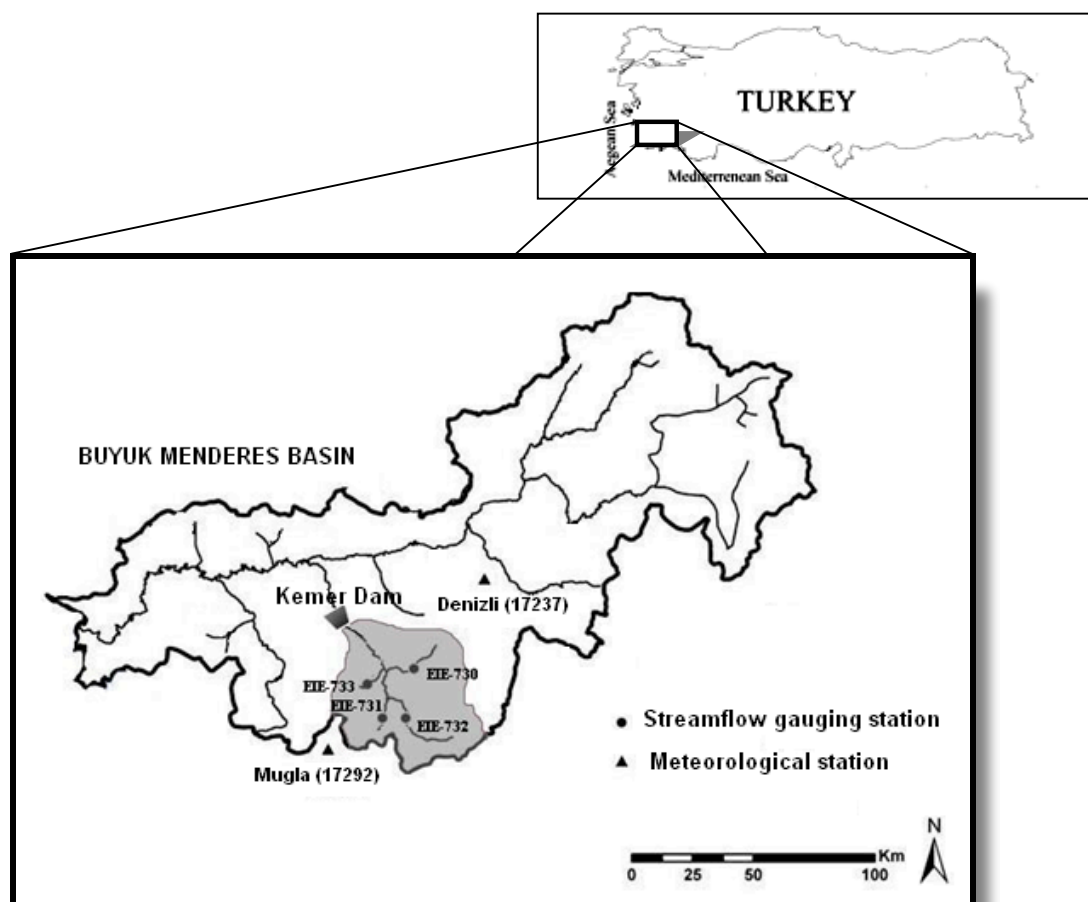


FIGURE 1 - Kemer Dam and the stream-flow gauging-meteorological stations within the study area.

**2.1. Feed-forward Neural Networks (FFNN)**

Artificial neural networks (ANN) are mathematical tools inspired by the properties of the biological neural system [48, 49]. The historical development of ANN involves the development of many different models and algorithms, but generally, the feed forward neural network (FFNN) models are used in applications. The basic concept of FFNN is that they are typically made up of single neurons which are organized in the form of layers (Fig. 2).

The first and last layers of FFNN are called the input and the output layers, respectively. The input layer does not perform any computations, but only serves to feed the input data to the hidden layer which is between the input and output layers. In general, there can be any number of hidden layers in the FFNN structures; however, only one or two hidden layers are used in applications. The number of hidden layers and also that of neurons of hidden layers can be determined by trial and error approaches [48, 49].

There are also three important components of a FFNN structure: weights, summing function and activation function. The importance and the functionality of the inputs on network are obtained with weights ( $W$ ), so the success of the model depends on the precise and correct determination of these weights. The summing function (*net*) acts to add all outputs; that is, each neuron input is multiplied by the weights and then summed. After computing the sum of weighted inputs for all neurons, the activation function  $f(.)$  serves to limit the amplitude of these values [49]. Various types of the activation function are possible but *sigmoid* function is preferred in this application.

$$f(.) \cong \frac{1}{1 + e^{-(.)}} \tag{1}$$

In addition to the structure and the components of FFNN, the running procedure of network is also important, which involves typically two phases; *forward computing* and *backward computing*.

In *forward computing*, each layer uses a weight matrix associated with all the connections made from the previous layer to the next layer (Fig. 1). The hidden layer has the weight matrix  $W_{ij}$  and activation function  $f^{(1)}$ ; the output layer has the weight matrix  $W_{jm}$  and activation function  $f^{(2)}$ . Given the network input vector  $x \in R^{n \times 1}$ , the output of the output layer, which is the response (output) of the network  $y \in R^{m \times 1}$ , can be written as follows:

$$y_m = f^{(2)} \left\{ \sum_{j=1}^m \left[ f^{(1)} \left( \sum_{i=1}^n x_i W_{ij} + b_j \right) \right] W_{jm} + b_m \right\} \tag{2}$$

After the phase of forward computing, *backward computing*, which depends on the algorithms to adjust weights, is used in FFNN. The process of adjusting these weights to minimize the differences between the actual and the desired output values is called training or learning of network. If these differences (errors) are higher than the desired values, the errors are passed backwards through the weights of the network. In ANN terminology, this phase is also called the back propagation algorithm. Once the comparison error is reduced to an acceptable level for the whole training set, the training period ends, and the net-

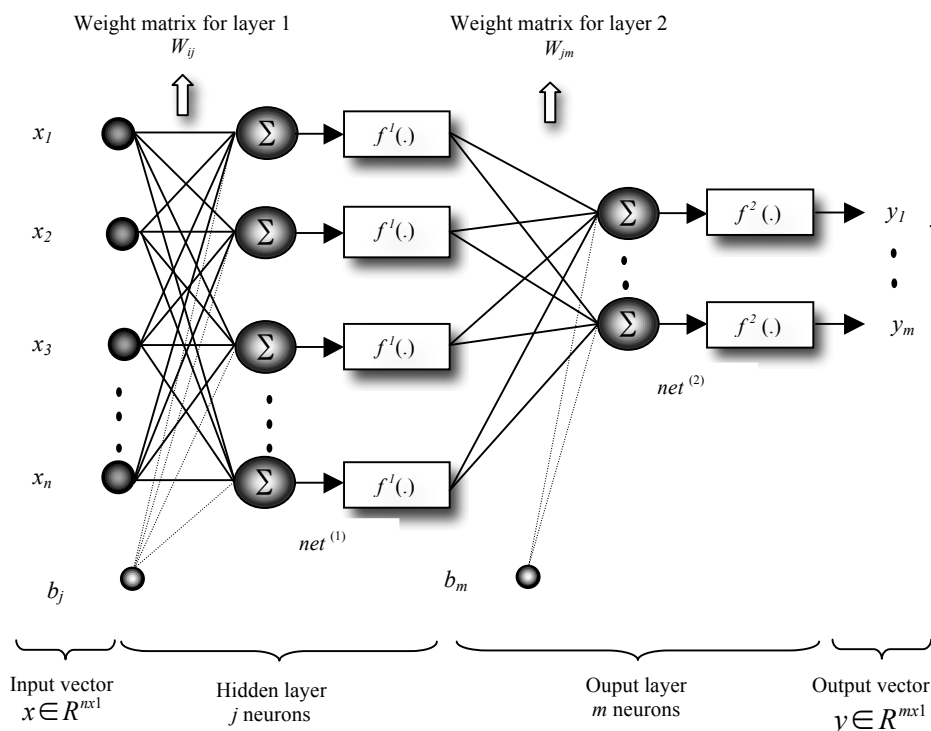


FIGURE 2 - FFNN structure

work is also tested for another known input and output data set in order to evaluate the generalization capability of the network [49].

Depending on the techniques to train FFNN models, different back propagation algorithms have been developed. In this study, the Levenberg-Marquardt back propagation algorithm was used for training of the FFNN. The Levenberg-Marquardt back propagation algorithm is a second-order nonlinear optimization technique that is usually faster and more reliable than any other back propagation techniques [21, 50].

The Levenberg-Marquardt optimization algorithm represents a simplified version of Newton’s method [51] applied to the training of FFNN [52, 53]. The training process can be viewed as finding a set of weights that minimize the error ( $e_p$ ) for all samples in the training set ( $Q$ ). The performance function is a sum of squares of the errors as follows:

$$E(W) = \frac{1}{2} \sum_{p=1}^P (d_p - y_p)^2 = \frac{1}{2} \sum_{p=1}^P (e_p)^2, P = mT \quad (3)$$

where,  $T$  is the total number of training samples,  $m$  is the number of output layer neurons,  $W$  represents the vector containing all the weights in the network,  $y_p$  is the actual network output, and  $d_p$  is the desired output.

When training with the Levenberg-Marquardt algorithm, the changing of weights  $\Delta W$  can be computed as follows:

$$\Delta W_k = - [J_k^T J_k + \mu_k I]^{-1} J_k^T e_k \quad (4)$$

Then, the update of the weights can be adjusted as follows:

$$W_{(k+1)} = W_k + \Delta W_k \quad (5)$$

where,  $J$  is the Jacobian matrix,  $I$  is the identify matrix,  $\mu$  is the Marquardt parameter which is to be updated using the decay rate  $\beta$  depending on the outcome. In particular,  $\mu$  is multiplied by the decay rate  $\beta$  ( $0 < \beta < 1$ ) whenever  $E(W)$  decreases, while  $\mu$  is divided by  $\beta$  whenever  $E(W)$  increases in a new step ( $k$ ).

### 2.2. Generalized Regression Neural Networks (GRNN)

The GRNN is a special four-layered neural network that imitates the regression process, and is used in the prediction of continuous variables [54]. This approach has been preferred in applications instead of the FFNN for the reason that the problem of local minimums was not faced in the GRNN so that it does not require an iterative procedure.

The GRNN, which is related to the normalized radial basis function, and is based on kernel regression, consists of four layers: input layer, pattern layer, summation layer and output layer. The typical structure of GRNN is shown in Fig. 3.

In the first layer, which does not perform any processing, an input vector is presented to the network. The number of neurons contained in this layer is equal to the number of elements,  $n$ , in the input vector. The input data are then passed onto the second layer, the pattern layer, where each training vector is represented [45, 54].

Thus, there are  $N$  pattern neurons running in parallel if the training data set consists of a total of  $i = 1, 2, \dots, N$  samples. Each neuron  $i$  generates an output  $\theta_i$  based on the input provided by the input layer:

$$\theta_i = \exp[-(x - u_i)^T (x - u_i) / 2\sigma^2] \quad (6)$$

where,  $x$  is the input vector,  $\sigma$  is the smoothing parameter, and  $u_i$  is the input portion of the  $i$ th training vector represented by the  $i$ th neuron in the pattern layer.

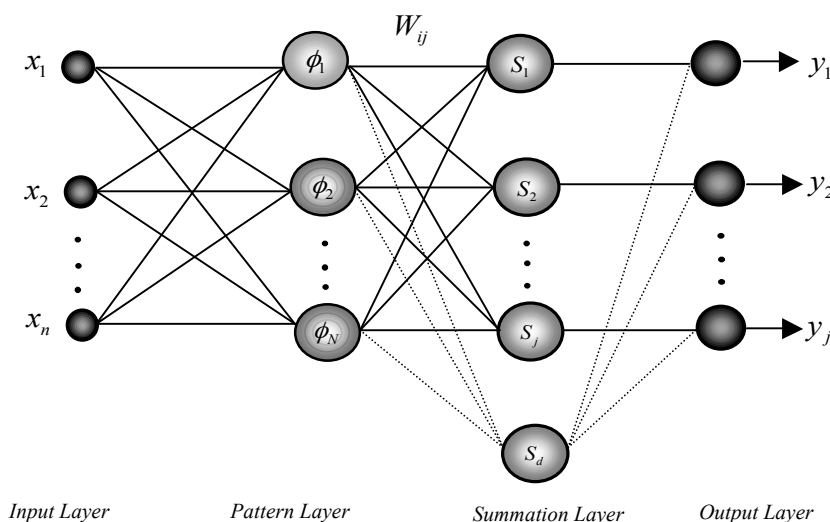


FIGURE 3 - GRNN structure

Then, every neuron in the pattern layer is connected to the summation layer which contains two groups of neurons, namely, numerator and denominator neurons. The group of numerator summation neurons is used for computing the weighted sum of the outputs from the pattern neurons. The transformation applied in the numerator neurons can be written as follows:

$$S_j = \sum_{i=1}^N W_{ij} \theta_i \tag{7}$$

where,  $S_j$  is the output from the  $j$ th numerator neuron,  $\theta_i$  is the output from the  $i$ th neuron in the pattern layer, and  $W_{ij}$  is the weight between the pattern layer and the summation layer.

The denominator group in the summation layer has only one neuron, which is computed by using the sum of the output from the pattern layer neurons, and can be defined as follows:

$$S_d = \sum_{i=1}^N \theta_i \tag{8}$$

where,  $S_d$  is the output from the denominator neuron, and  $\theta_i$  is the output from the  $i$ th neuron in the pattern layer.

The numbers of neurons in the output layer are equal to the number of numerator neurons. The outputs ( $y_j$ ) of GRNN can be computed as follows:

$$y_j = S_j / S_d \tag{9}$$

In application of the methodology, two MATLAB codes were written for Levenberg-Marquardt algorithm based FFNN and GRNN models. The application of the models to time series data consisted of two periods. 26 years (January 1980-December 2005) input-output data were used and divided into training and testing periods by proportions of 1/2 (January 1980-December 1992) and 1/2 (January 1993-December 2005), respectively.

Before presenting the input-output data to models, all data sets were scaled to the range 0-1 so that the different input signals had the same numerical range. The training and the testing subsets were scaled to the range of 0-1 using the equation  $z_t = (x_t - x_{\min}) / (x_{\max} - x_{\min})$ , where  $x_t$  is the unscaled data,  $z_t$  is scaled data, and  $x_{\max}$  and  $x_{\min}$  are the maximum and minimum values of the unscaled data, respectively. Then, the actual output values of the networks, which were in the range of 0–1, were converted to real-scaled values using the equation  $x_t = z_t (x_{\max} - x_{\min}) + x_{\min}$ .

In training, the number of the neurons in the hidden layer, the initial Marquardt parameter of Levenberg-Marquardt based FFNN model and the smoothing parameter of GRNN model were determined by trial-error. This network structure provided the best training result in terms of the minimum root mean square errors, RMSE (Eq. 10), and the maximum determination coefficients,  $R^2$  (Eq. 11) were also employed for the testing periods.

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (d_t - y_t)^2} \tag{10}$$

$$R^2 = \frac{\sum_{t=1}^T (d_t - d_{mean})^2 - \sum_{t=1}^T (d_t - y_t)^2}{\sum_{t=1}^T (d_t - d_{mean})^2} \tag{11}$$

where,  $T$  is the number of training or testing samples,  $y_t$  is the actual network output,  $d_t$  is the observed (desired) data in the  $t^{th}$  time period, and  $d_{mean}$  is the mean over the observed periods.

### 3. RESULTS

In this study, different neuron combinations of input and hidden layers for the FFNN approach have been tried. For the first combination, monthly precipitation and tem-

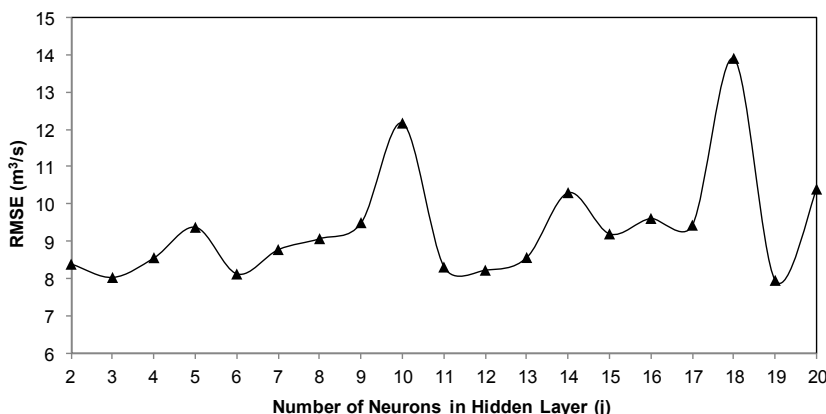


FIGURE 4 - Determination of number of neurons in hidden layer for the testing period of FFNN (4, 19, 1).

perature data were used as an input. With these concurrent monthly input data,  $R^2$  values of training and testing period are obtained as 0.726 and 0.616;  $RMSE$  values are 12.611 and 11.188  $m^3/sn$ , respectively. In order to increase the performance of the FFNN model, delaying process that rainfall transforms into runoff, was considered and previous monthly precipitation values were included in model. For each combination, the numbers of neurons in

the hidden layer were determined by the trial-error. All these trials are presented in Table 2.

At the end of the trials performed, FFNN (4, 19, 1) model with  $\mu=10^{-3}$ ;  $\beta=0.1$  values, and 25 iterations has the lowest  $RMSE$  and the highest  $R^2$  values, thus having the best performance. The scatter plot and hydrograph graphics of this model which has the best results of testing period are shown in Figs. 5a and 5b.

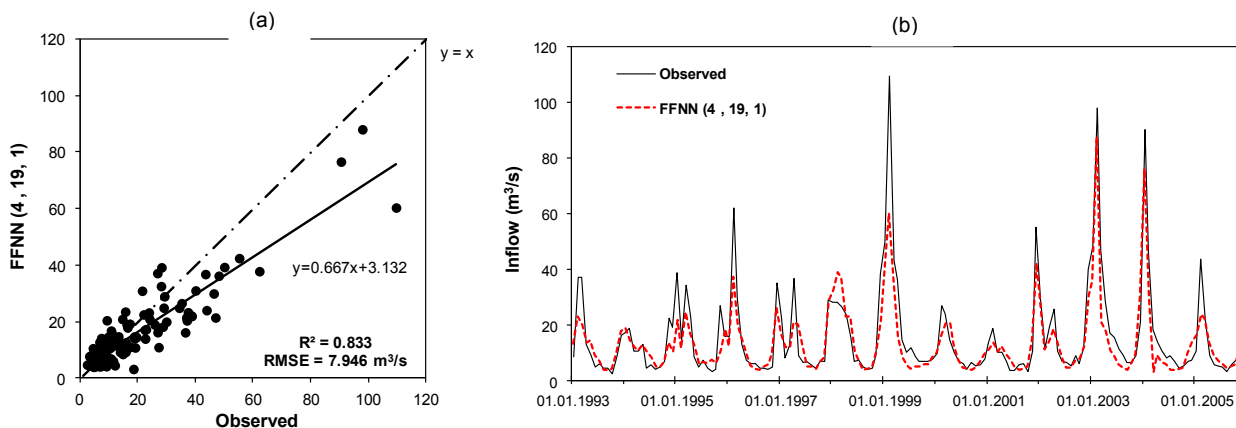


FIGURE 5 - Reservoir inflow estimations by FFNN (4, 19, 1) for the testing period.

TABLE 2 - FFNN performances for the training and testing periods.

Inputs	FFNN Structure			$R^2$		$RMSE (m^3/s)$	
	$n$	$j$	$m$	Training	Testing	Training	Testing
$P_t, T_t$	2	3	1	0.726	0.616	12.611	11.188
$P_t, T_t, P_{t-1}$	3	2	1	0.841	0.680	9.631	9.746
$P_t, T_t, P_{t-1}, P_{t-2}$	4	19	1	0.940	0.833	6.193	7.946

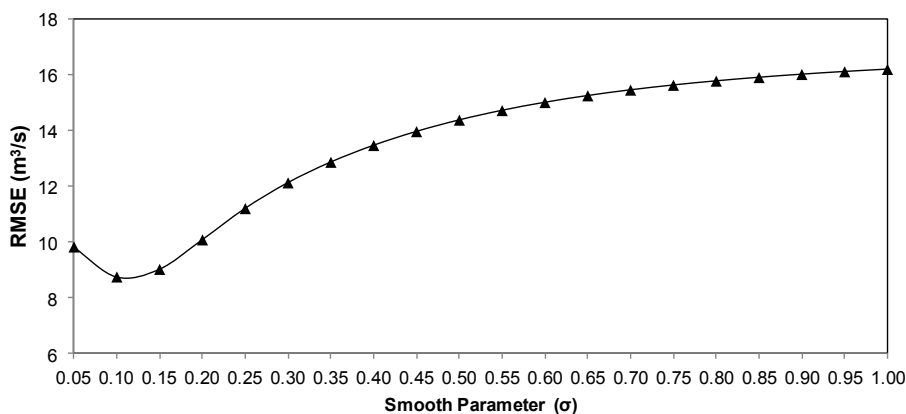


FIGURE 6 - Determination of smooth parameter for the testing period of GRNN model with 4 inputs.

TABLE 3 - GRNN performances for the training and testing periods.

Inputs	Smooth Parameter $\sigma$	$R^2$		$RMSE (m^3/s)$	
		Training	Testing	Training	Testing
$P_t, T_t$	0.05	0.798	0.564	10.855	11.394
$P_t, T_t, P_{t-1}$	0.10	0.891	0.669	8.067	10.056
$P_t, T_t, P_{t-1}, P_{t-2}$	0.10	0.966	0.756	4.448	8.743

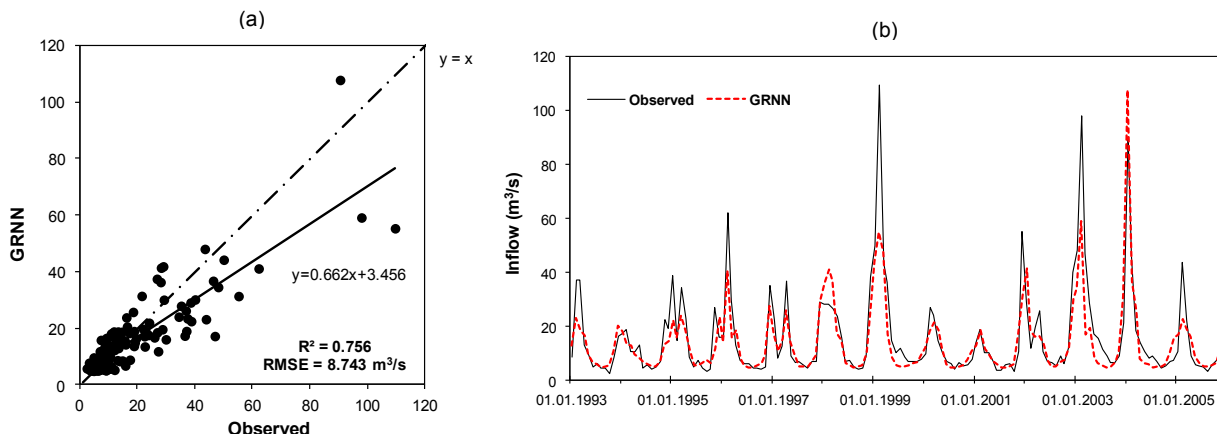


FIGURE 7 - Reservoir inflow estimations by GRNN model with 4 inputs for the testing period.

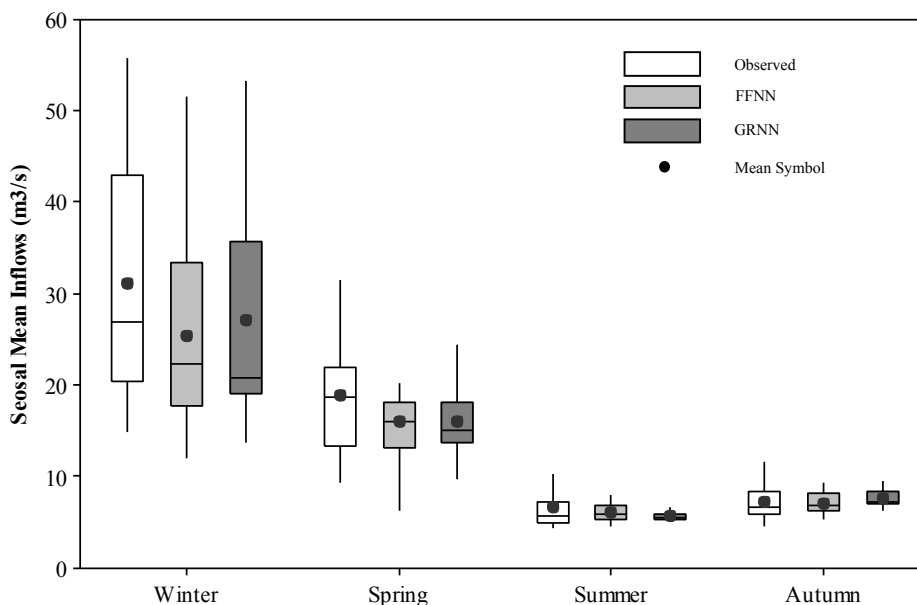


FIGURE 8 - Box-plots of the observed and the model estimations of seasonal mean inflows.

Similar combinations were also examined with GRNN model (Table 3), and in the training and testing period, with four inputs and a smoothing parameter having the value  $\sigma = 0.10$  (Fig. 6), the  $R^2$  values 0.966 and 0.756,

and RMSE values 4.448  $m^3/s$  and 8.743  $m^3/s$  were obtained, respectively, thus reaching the most efficient GRNN structure.

TABLE 4 - M-W statistics of (a) FFNN and (b) GRNN for testing periods.

(a)				
FFNN M-W Test Statistics	Winter	Spring	Summer	Autumn
Mann-Whitney U	60	62	81	73
z	1.26	1.15	0.18	0.59
Asymptotic. Sig. (2-tailed)	0.21	0.25	0.86	0.56
(b)				
GRNN M-W Test Statistics	Winter	Spring	Summer	Autumn
Mann-Whitney U	68	64	70	51
z	0.85	1.05	0.74	1.72
Asymptotic. Sig. (2-tailed)	0.40	0.29	0.46	0.09

In addition to long-term statistics of FFNN-GRNN models, seasonal box-plot presentations and homogeneities were examined.

Seasonal mean, minimum, maximum and, median statistics of the models and the observed data are shown in Fig. 8.

In the study, seasonal homogeneities were tested with Mann-Whitney U (M-W) test, a non-parametric statistical test used to analyze two comparison groups to identify whether they have the same distribution or not [55]. This test is based on bringing together and arranging of two groups. When the lining up of these group members is done, for each member, a line number is assigned. The membership status of these members (to which group they belong) is ignored. Then, all these line numbers are summed up. The sum of the members of the first group is  $R_1$  and sum of the members of the second group is  $R_2$ . Then, we can calculate the  $U$  values:

$$U_i = N_1 N_2 + \frac{N_i(N_i + 1)}{2} - R_i, \quad (i = 1, 2) \quad (12)$$

After the calculation for  $i=1$  and  $i=2$ , we get  $U_1$  and  $U_2$ , and the bigger is chosen ( $U^*$ ) to determine the test statistics.

$$z = \frac{U^* - \frac{N_1 N_2}{2}}{\sqrt{\frac{N_1 N_2 (N_1 + N_2 + 1)}{12}}} \quad (13)$$

Here,  $N_1$  and  $N_2$  are the numbers of data for the groups compared.

The  $z$  value is compared with 0.05 significance level ( $z_{cr} = 1.96$ ). If we get  $z < 1.96$ , it means that there is no significant difference between the observed data and model estimations. We can also use the asymptotic significance of  $z$  test statistics when making a comparison (Table 4).

#### 4. CONCLUSIONS

A comprehensive study was presented on the application of different artificial neural networks to model the monthly reservoir inflows. Two types of neural networks, FFNN and GRNN, were applied.

When the performances of the training and testing periods are compared, it is observed that GRNN approach has a better performance in the training period; on the other hand, FFNN proves itself to be more successful in the testing period.

When the testing period scatter graphs of the models are examined, it is observed that the standard deviations around the  $y=x$  line are far less in the FFNN models. In other words, when  $y=ax+b$  fitted lines in graphs are examined, it is observed that, in FFNN model, “ $a$ ” gets closer to the value 1, and “ $b$ ” gets closer to the value 0, compared to the GRNN model.

Seasonal comparisons are shown by box-plot graph and Mann Whitney U (M-W) non-parametric test. When the box-plot graph is examined, we see that summer and winter statistics are successful in both models. But when the median statistics are compared, FFNN model proves itself better than GRNN model for all seasons. When the extreme values of the seasons are examined, we see that both models are competent in predicting winter values.

When we examine M-W statistics, we see that both FFNN and GRNN predictions have homogeneities for all seasons. When  $z$  statistics are taken as a basis, FFNN has the best performance in summer and autumn. Moreover, GRNN has the best performance in winter and spring.

In addition to the input data used in the study, to increase the performance of the model, previous flow series ( $Q_{t-1}, Q_{t-2}, \dots, Q_{t-p}$ ) can be included into the models, considering the inner dependency effect, a concept that explains the interrelation between precipitation and flow values. But predicting the result with the fewest number of inputs (*lex parsimoniae*), although the data were relevant to our purpose, we have chosen not to include them. The data used in the study are regarded to be sufficient for the tools to enable them to function properly.

#### ACKNOWLEDGEMENTS

Authors feel responsible to thank English Instructor Cuneyt Guren (from Celal Bayar University, M.Sc.) for his valuable contribution to grammar revision of this study as well as workers of II. Regional Directorate of State Hydraulic Works and Turkish State Meteorological Service for their help with data collection.

#### REFERENCES

- [1] Singh, V.P. (1995) Computer Modelling of Watershed Hydrology, Water Resources Publications.
- [2] Singh, V.P. and Woolhiser D.A. (2002) Mathematical modeling of watershed hydrology. Journal of Hydrologic Engineering, V.7, 270–292.
- [3] Abbott, M.B. and Refsgaard, J.C. (1996) Distributed Hydrological Modeling, Kluwer Academic Publishers, Dordrecht. 308 pp.
- [4] Hsu, K.L.; Gupta, H.V. and Sorooshian, S. (1995) Artificial neural network modeling of rainfall-runoff process. Water Resources Research, V.31 (10), 2517–2530.
- [5] Minns, A.W. and Hall, M.J. (1996) Artificial neural networks as rainfall-runoff models, Hydrological Science Journal, V.41 (3), 399–417.
- [6] Fernando, D.A.K. and Jayawardena, A.W. (1998) Runoff forecasting using RBF networks with OLS algorithm. Journal of Hydrologic Engineering, V.3 (3), 203–209.
- [7] Dawson, C.W. and Wilby, R. (1998) An artificial neural network approach to rainfall-runoff modelling, Hydrological Science Journal, V.43, 47–66.



- [8] Campolo, M.; Andreussi, P. and Soldati, A. (1999) River flood forecasting with a neural network model, *Water Resources Research*, V.35, 1191-1197.
- [9] Tokar, A.S. and Johnson, P.A. (1999) Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, V.4 (3), 232-239.
- [10] Tokar, A.S. and Markus, M. (2000) Precipitation-runoff modeling using artificial neural network and conceptual models. *Journal of Hydrologic Engineering*, V.5 (2), 156-161.
- [11] Lin, G. and Chen, L. (2004) A non-linear rainfall-runoff model using radial basis function network. *Journal of Hydrology*, V.289, 1-8.
- [12] Kisi, O. (2005) Daily river flow forecasting using artificial neural networks and auto-regressive models. *Turkish Journal of Engineering and Environmental Sciences*, V.29, 9-20.
- [13] Diamantopoulou, M.J., Georgiou, P.E., and Papamichail, D.M. (2007). Performance of neural network models with kalman learning rule for flow routing in a river system. *Fresenius Environmental Bulletin*, V.16 (11b), 1474-1484.
- [14] Cigizoglu, H.K. (2003) Incorporation of ARMA models into flow forecasting by artificial neural networks. *Environmetrics*, V.14 (4), 417-427.
- [15] Jain, S.K. and Srivastava, D.K. (1999) Application of ANN for reservoir inflow prediction and operation. *Journal of Water Resources Planning and Management – ASCE*, V.125 (5), 263-271.
- [16] Coulibaly, P.; Anctil, F. and Bobee, B. (2000) Daily reservoir inflow forecasting using artificial neural Networks with stopped training approach. *Journal of Hydrology*, V.230, 244-257.
- [17] Mohammadi, K.; Eslami, H.R. and Dardashti, S.D. (2005) Comparison of Regression, ARIMA and ANN Models for Reservoir Inflow Forecasting using Snowmelt Equivalent (a Case study of Karaj). *Journal of Agricultural Sciences and Technology*, V.7, 17-30.
- [18] Razavi, S. and Araghinejad, S. (2009) Reservoir Inflow Modeling Using Temporal Neural Networks with Forgetting Factor Approach. *Water Resources Management*, V.23, 39-55.
- [19] Tayfur, G. (2002) Artificial neural networks for sheet sediment transport. *Hydrological Science Journal*, V.47 (6), 879-892.
- [20] Nagy, H.M.; Watanabe, K. and Hirano, M. (2002) Prediction of Sediment Load Concentration in Rivers using Artificial Neural Network Model. *Journal of Hydraulic Engineering ASCE*, V.128 6, 588-595.
- [21] Kisi, O. (2004) Multi-layer perceptrons with Levenberg-Marquardt optimization algorithm for suspended sediment concentration prediction and estimation. *Hydrological Sciences Journal*, V.49 (6), 1025-1040.
- [22] Cigizoglu, H.K. and Alp, M. (2006) Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software*, V.37, 63-68.
- [23] Agarwal, A.; Singh, R.D.; Mishra, S.K. and Bhunya, P.K. (2005) ANN-based sediment yield models for Vamsadhara river basin (India), *Water SA*, V.31 (1), 95-100.
- [24] Hall, M.J. and Minns, A.W. (1998) Regional flood frequency analysis using artificial neural networks. In: Babovic V, Larsen CL, editors *Hydroinformatics conference*. Proc. 2nd vol., third international conference on hydroinformatics, Copenhagen, Denmark. Rotterdam: A.A.Balkema 759-63.
- [25] Jingyi, Z. and Hall, M. J. (2004) Regional flood frequency analysis for the Gan-Ming River basin in China. *Journal of Hydrology*, V.296 (1-4), 98-117.
- [26] Zealand, C.M.; Burn, D.H. and Simonovic, S.P. (1999) Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology*, V.214, 32-48.
- [27] Xu, Z.X. and Li, J.Y. (2002) Short term inflow forecasting using an artificial neural network model. *Hydrological Process*, V.16, 2423-2439.
- [28] Hall, T. (1999) Precipitation forecasting using a neural network. *Weather and Forecasting*, V.14, 338-345.
- [29] Ramirez, M.C.V.; Velho, H.F.C. and Ferreira, N.J. (2005) Artificial neural network technique for rainfall forecasting applied to the São Paulo region. *Journal of Hydrology*, V.301 (1-4), 146-162.
- [30] Freiwan, M. and Cigizoglu, H.K. (2005) Prediction of total monthly rainfall in Jordan using feed forward backpropagation method. *Fresenius Environmental Bulletin*, V.14 (2), 142-51.
- [31] Ustaoglu, B.; Cigizoglu, H.K. and Karaca, M. (2008) Forecast of daily mean, maximum and minimum temperature time series by three artificial neural network methods. *Meteorological Applications*, V.15, 431-445.
- [32] Chattopadhyay, S.; Jhajharia, D. and Chattopadhyay, G. (2010). Univariate modelling of monthly maximum temperature time series over northeast India: neural network versus Yule-Walker equation based approach. *Meteorological Applications*, V.18 (1), 70-82.
- [33] Kumar, M.; Raghuwanshi, S.; Singh, R.; Wallender, W.W. and Pruitt, W.O. (2002) Estimating evapotranspiration using artificial neural network. *Journal of Irrigation and Drainage Engineering*, V.128, 224-233.
- [34] Sudheer, K.P.; Gosain, A.K.; Mohana, R.D. and Saheb, S.M. (2002) Modelling evaporation using an artificial neural network algorithm. *Hydrological Processes*, V.16, 3189-3202.
- [35] Keskin, M.E. and Terzi, O. (2006) Artificial Neural Network Models of Daily Pan Evaporation. *Journal of Hydrologic Engineering*, V.11 (1), 65-70.
- [36] Jain, S.; Nayak, P. and Sudheer, K. (2008) Models for estimating evapotranspiration using artificial neural networks, and their physical interpretation. *Hydrological Processes*, V.22, 2225-2234.
- [37] Djebbar, Y. and Alila, Y. (1998) Neural network estimation of sanitary flows. In: Babovic V, Larsen CL, editors. *Third International conference on hydroinformatics*, V.2, Denmark.
- [38] Shigidi, A. and Garcia, A.L. (2003) Parameter Estimation in Groundwater Hydrology Using Artificial Neural Networks. *Journal of Computing in Civil Engineering*, V.17 (4), 281-289.
- [39] Daliakopoulos, I.; Coulibaly, P. and Tsanis, I.K. (2004) Forecasting the Groundwater Level of an Aquifer with the use of Neural Networks. *Journal of Hydrology*, V.309, 229-240.
- [40] Ioannis, N.; Daliakopoulos, A.; Paulin, C.; Ioannis, K. and Tsanis, B. (2005) Groundwater level forecasting using artificial neural networks. *Journal of Hydrology*, V.309, 229-240.
- [41] Jain, A. and Kumar, A. (2006) An evaluation of artificial neural network technique for the determination of infiltration model parameters. *Applied Software Computing*, V.6, 272-282.

- [42] Panu, U.S.; Khalil, M. and Elshorbagy, A. (2000) Streamflow data Infilling Techniques Based on Concepts of Groups and Neural Networks. *Artificial Neural Networks in Hydrology*, Kluwer Academic Publishers, Netherlands, 235-258.
- [43] Khalil, M.; Panu, U.S. and Lennox, W.C. (2001) Group and neural networks based streamflow data infilling procedures. *Journal of Hydrology*, V.241, 153-176.
- [44] Illunga, M. and Stephenson, D. (2005) Infilling streamflow data using feed-forward back-propagation (BP) artificial neural networks: Application of standard BP and pseudo Mac Laurin power series BP techniques *Water SA*, V.31 (2), 171-176.
- [45] Cigizoglu, H.K. (2005a) Application of the Generalized Regression Neural Networks to Intermittent Flow Forecasting and Estimation. *Journal of Hydrologic Engineering*, V.10 (4), 336-341.
- [46] Cigizoglu, H.K. (2005b) Generalized regression neural networks in monthly flow forecasting. *Civil Engineering and Environmental Systems*, V.22 (2), 71-84.
- [47] Lin, G.; Wu, M.; Chen, L. and Tsai, F. (2009) An RBF-based model with an information processor for forecasting hourly reservoir inflow during typhoons. *Hydrological Process*, V.23, 3598–3609.
- [48] Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*. MacMillan New York. 768 pp.
- [49] Ham, F. and Kostanic, I. (2001) *Principles of Neurocomputing for Science and Engineering* Macgraw-Hill. USA. 672 pp.
- [50] Cigizoglu, H.K. and Kisi, O. (2005) Flow prediction by three back propagation techniques using k-fold partitioning of neural network training data. *Nordic Hydrology*, V.36 (1), 49–64.
- [51] Marquardt, D. (1963) An algorithm for least squares estimation of non-linear parameters. *Journal of the Society for Industrial and Applied Mathematics*, V.11 (2), 431–441.
- [52] Hagan, M.T. and Menhaj, M.B. (1994) Training feed forward techniques with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, V.5 (6), 989–993.
- [53] Hagan, M.T.; Demuth, H.P. and Beale, M. (1996) *Neural Network Design*. PWS Publishing, Boston. 712 pp.
- [54] Specht, D.F. (1991) A general regression neural network. *IEEE Transactions Neural Network*, V.2 (6), 568–76.
- [55] Mann, H.B., Whitney, D.R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, V.18, 50-60.

---

**Received:** March 21, 2011

**Revised:** July 18, 2011

**Accepted:** July 20, 2011

---

#### CORRESPONDING AUTHOR

**Umut Okkan**

Balikesir University  
Faculty of Engineering-Architecture  
Department of Civil Engineering  
Balikesir  
TURKEY

E-mail: [umutokkan@balikesir.edu.tr](mailto:umutokkan@balikesir.edu.tr)



# ADSORPTION OF PHOSPHINE IN OFF-GAS OF PHOSPHORUS SLUDGE UTILIZATION USING A MOLECULAR SIEVE 5A

Yanfu Wei, Xiaofeng Huang\*, Wenqing Li, Xiaoni Wang, Kunyang Gao, Yangsong Qin and Tao Zhou

Faculty of Environmental Science and Engineering, Kunming University of Science and Technology, Kunming 650093, P.R. China

## ABSTRACT

In order to utilize phosphorus sludge comprehensively, phosphine ( $\text{PH}_3$ ) in the off-gas of sodium hypophosphite production from phosphorus sludge was adsorbed with a molecular sieve 5A. Effects of adsorbent preparation and operating conditions were analyzed. The results showed that 5A molecular sieve could significantly enhance the adsorption ability after NaCl impregnation. The adsorbent modified by 0.3 M NaCl was applied in the adsorption of  $\text{PH}_3$ , and the best adsorption efficiency was obtained under the following conditions: calcination temperature (300 °C), drying temperature (110 °C), adsorption temperature (20 °C) and carrier gas flow-rate (10 ml/min). SEM and XPS analysis showed that adsorption of  $\text{PH}_3$  on modified 5A molecular sieve was mainly physisorption. In conclusion, modified 5A molecular sieve can be used in the adsorption of  $\text{PH}_3$  on phosphoric sludge utilization.

**KEYWORDS:**  $\text{PH}_3$ , adsorption, 5A molecular sieve, phosphorus sludge, utilization.

## 1. INTRODUCTION

Exposed phosphoric sludge would generate  $\text{P}_2\text{O}_5$ , which poses a great threat to the environment and human health. For example, 120 t phosphorus sludge stored in the local pond caused catastrophic water pollution at the drinking water sources of Panshidong Cave in Yunnan, China [1]. There is a necessity for processing the accumulated waste to create waste-free productions [2]. Within the limits of the previous technology, a process of phosphoric sludge to the sodium hypophosphite has been developed [3]; however, the process inevitably generates 20% (v/v)  $\text{PH}_3$  leaving from the reactor.

$\text{PH}_3$  is highly toxic by inhalation route, which can affect respiratory, neurological, and gastrointestinal system [4]. The hydride gas of  $\text{PH}_3$  is commonly used in the fabricating processes of the semiconductor and optoelectronic in-

dustries [5], and the field of flame retardants for cotton fabrics. Many traditional methods have been developed, such as combustion, dry adsorption, gas-solid catalytic oxidation reaction and gas-liquid oxidation reaction [6]. Most of the studies concerned with the removal of  $\text{PH}_3$  in synthesis gas, for instance, copper (Cu) loaded on the sol-gel-derived gamma-alumina ( $\text{Al}_2\text{O}_3$ ) adsorbents were tested to investigate the possibility of  $\text{PH}_3$  removal [7];  $\text{PH}_3$  toxic gas was absorbed by metal (Cu, Zn, or Mn) loaded ZSM-5 and Y zeolite [8]. Although a wide variety of adsorbents have been utilized to remove contaminants from process gas streams, there has been little research on the off-gas of sodium hypophosphite production from phosphorus sludge. Researchers concerned  $\text{PH}_3$  removal by chemical adsorption [7, 9-11], and obtained fruitful achievements. However, there has been less study aimed at physisorption of  $\text{PH}_3$ .

Based on the phosphoric sludge utilization to develop an environmentally friendly, resource-saving industrial process technology, the obtained pure  $\text{PH}_3$  is considered for producing high value products.

## 2. MATERIALS AND METHODS

### 2.1. Experimental materials

5A molecular sieve, 13X molecular sieve, activated alumina and granular activated carbon were used as adsorbents in this study. 5A molecular sieve (Sinopharm Chemical Reagent Co. Ltd) is a commercial product with a particle size of 3.0~5.0 mm. 13X molecular sieve (Shanghai BOJ Molecular Sieve Co., Shanghai, China), activated alumina (Shanghai Jiuzhou Chemicals Co., Ltd., Shanghai, China) and granular activated carbon (Nankai University catalyst Co., Ltd., Tianjin, China) were used in the experiments. Details of these materials are given in Table 1. Phosphoric sludge was from Yunnan Kunyang Phosphate Fertilizer Factory producing yellow phosphorus with electric furnace process.  $\text{N}_2$  used in the study was supplied by Kunming Messer Gas Products Co., Ltd.

Adsorbent preparation was as follows: A total of 30 g of each type of adsorbent was washed 3 times with 150 ml of distilled water to remove soluble impurities. Then, adsorbents were dried for 3 h at 110 °C, to further remove im-

\* Corresponding author