Neural Comput & Applic (2013) 23:1019–1034
DOI 10.1007/s00521-012-1026-y

ORIGINAL ARTICLE

Fast neural network learning algorithms for medical applications

Ahmad Taher Azar

Received: 16 January 2012 / Accepted: 21 June 2012 / Published online: 14 July 2012

Abstract Measuring the blood urea nitrogen concentration is crucial to evaluate dialysis dose (*Kt*/*V*) in patients with renal failure. Although frequent measurement is needed to avoid inadequate dialysis efficiency, artificial intelligence can repeatedly perform the forecasting tasks and may be a satisfactory substitute for laboratory tests. Artificial neural networks represent a promising alternative to classical statistical and mathematical methods to solve multidimensional nonlinear problems. It also represents a promising forecasting application in nephrology. In this study, multilayer perceptron (MLP) neural network with fast learning algorithms is used for the accurate prediction of the post-dialysis blood urea concentration. The capabilities of eight different learning algorithms are studied, and their performances are compared. These algorithms are Levenberg–Marquardt, resilient backpropagation, scaled conjugate gradient, conjugate gradient with Powell–Beale restarts, Polak–Ribiere conjugate gradient and Fletcher–Reeves conjugate gradient algorithms, BFGS quasi-Newton, and one-step secant. The results indicated that BFGS quasi-Newton and Levenberg–Marquardt algorithm produced the best results. Levenberg–Marquardt algorithm outperformed clearly all the other algorithms in the verification phase and was a very robust algorithm in terms of mean absolute error (MAE), root mean square error (RMSE), Pearson's correlation coefficient (R_p^2) and concordance coefficient (R_C). The percentage of MAE and RMSE for Levenberg–Marquardt is 0.27 and 0.32 %, respectively, compared to 0.38 and 0.41 % for BFGS quasi-Newton and 0.44 and 0.48 % for resilient backpropagation. MLP-based systems can achieve satisfying results for predicting post-dialysis blood urea concentration and single-pool dialysis dose ${}_{\mathrm{sp}}Kt/V$ without the need of a detailed description or formulation of the underlying process in contrast to most of the urea kinetic modeling techniques.

Keywords Multilayer perceptron (MLP) · Optimum topology · Levenberg–Marquardt (LM) · Conjugate gradient algorithms · Quasi-Newton algorithms

1 Introduction

The mathematical modeling of dialysis starts with the development of the kinetic of solutes [137]. Methods for quantitating dialysis using urea kinetics were already developing in the San Francisco offices of Frank Gotch and John Sargent, who had expanded Wolf's equations and developed more precise quantitative expressions for single-pass dialysis after a prospective and multicenter study [National Cooperative Dialysis Study (NCDS); 65, 90]. The results of the National Cooperative Dialysis Study (NCDS) showed the relationship between the urea kinetics and the clinical outcome of patients on hemodialysis [90]. This was the creation of the *Kt*/*V* formula and urea kinetic modeling (UKM) for measuring delivered dose and adequacy of dialysis. The NCDS design was based on the assumption that blood urea nitrogen (BUN) could be used as a surrogate for uremic toxins and that the dose of dialysis could be defined by the level of BUN achieved [113]. The calculation of *Kt*/*V* is now widely used to quantify HD treatment [2, 11, 23, 31, 40–44, 63, 75]. Many mathematical models of hemodialysis have been referred to, starting from single-pool urea kinetic models (spUKM) [30, 33, 34,

A. T. Azar (✉)
Computer and Software Engineering Department,
Faculty of Engineering, Misr University for Science
and Technology (MUST), 6th of October City, Egypt
e-mail: ahmad_T_azar@ieee.org

66, 112, 125, 64] and following with either double-pool urea kinetic models (dpUKM; [118], [117]) [25] or more compartments [46] where the uremic toxins are distributed (urea, creatinine and uric acid). The single-pool model (denoted as $_{\mathrm{sp}}Kt/V$) assumes that urea equilibrates instantaneously among all body compartments, while double-pool model behaves as if urea is distributed between two distinct compartments, intracellular and extracellular. Urea once removed by dialysis from extracellular space will be refilled by an osmotic gradient across the cell membrane. This is a non-instantaneous process. Typically, it takes 30 min to equilibrate 95 % of an initial urea gradient over the cell membrane if the gradient is not perpetuated by further dialysis. This effect, usually referred as *rebound*, is the measurable trace of the delayed urea mass transfer across the cell membrane [4, 59, 79, 80, 88, 106, 119, 126, 139]. Double-pool kinetic models are more complex than spUKM and much more than “bedside” formulae. A trade-off solution led to the definition of the equilibrated Kt/V (eKt/V). Kt/V values of less than 1.0 have been associated with higher rates of morbidity and mortality than values greater than 1.0 [1, 12, 32, 70, 74, 92]. However, other data suggest that values greater than 1.0 has been an underestimate and that a Kt/V greater than 1.2 or 1.3 is ideal [28]. In 1993, Renal Physicians Association (RPA) developed practice guidelines for hemodialysis and recommended that delivered Kt/V should be at least 1.2 for $_{\mathrm{sp}}Kt/V$, and when the Kt/V falls below this level, corrective action should be undertaken [108]. Because many end-stage renal disease (ESRD) patients do not receive an adequate dose of hemodialysis, the National Kidney Foundation (NKF) Dialysis Outcomes Quality Initiative (DOQI) decided to reevaluate the issue of hemodialysis adequacy. The NKF-DOQI clinical practice guidelines, published in 1997, have set the minimum of delivered dialysis dose [100]. Recognizing that some dialysis care teams may prefer to follow hemodialysis dosing using a double-pool model for Kt/V, the HD Adequacy Work Group recommends that the minimum prescribed dose in a double-pool model be considered to be 1.05 for eKt/V for patients dialyzing three times per week [101]. Due to this variability of measuring Kt/V, DOQI guidelines recommended a target dose at least 15 % higher than the listed minimum (1.4 for $_{\mathrm{sp}}(Kt/V)$ and 1.15 for $_{\mathrm{eq}}(Kt/V)$) [102].

1.1 Problem statement

However, because of the complexity of the UKM, several methods and nomograms have been developed for calculating delivered dialysis dose. The estimation of Kt/V by various formulae and these shortcut methods always require pre-dialysis and post-dialysis blood urea nitrogen (BUN) to calculate the fractional index of urea removal (Kt/V). The methods of sampling pre- and post-BUN samples during a dialysis treatment are explained in the NKF/DOQI Guidelines for Hemodialysis Adequacy [101]. Pre-BUN measurements must be taken before the treatment begins, and the sample should not be tainted with normal saline or heparin. The critical measurement is post-BUN sampling and is usually underestimated because of access recirculation, cardiac recirculation and urea rebound [37, 38]. It is important to sample after recirculation has been resolved but before urea rebound has started [102]. Not waiting long enough leads to the overestimation of Kt/V value because only dialyzed blood is being sampled; waiting too long time allows urea rebound to start and causes underestimation of Kt/V value [87]. Thus, the estimation of the post-dialysis blood urea concentration (C_{post}) has become an important subject of research in nephrology because of its influence on the calculation of the single-pool and double-pool dialysis dose (Kt/V). To overcome these sampling errors, artificial intelligence can repeatedly perform the forecasting tasks and may be a satisfactory substitute for laboratory tests. Artificial neural network (ANN) possesses a variety of alternative features such as massive parallelism, distributed representation and computation, generalization ability, adaptability and inherent contextual information processing [68, 73, 93, 98]. The ANN is an information processing system that roughly replicates the behavior of a human brain by emulating the operations and connectivity of biological neurons [131]. In many fields of clinical medicine, ANN have been used successfully to solve complex and chaotic problems without the need of mathematical models and a precise understanding of the mechanisms involved [29, 104, 107, 130, 132]. ANN applications have been used in many areas of nephrology including pharmacologic studies [27, 124, 134, 135], primary renal diseases [53, 55, 60, 84, 122, 123], transplantation [54, 77, 120], peritoneal dialysis [128] and hemodialysis [3, 5, 6, 47–50, 56–58, 62, 67]. In this study, a multilayer perceptron NN is used to evaluate and compare the validity of ANN as a useful tool to predict post-dialysis blood urea concentration (C_{post}) and single-pool dialysis dose ($_{\mathrm{sp}}Kt/V$). The neural network is trained with the adjusted-error backpropagation algorithm. The back-propagation training is an iterative gradient algorithm designed to minimize the root mean square error between the actual output of a feedforward NN and a desired output.

2 Subjects and methods

The study was carried out at four dialysis centers. Blood urea nitrogen (BUN) was measured in all serum samples at a central laboratory. The overall study period was 6 months. The study subjects consisted of 310 hemodialysis patients collected from four dialysis centers that gave their informed consent to participate. They are 165 male and 145 female patients, with ages ranging 14–75 years (48.97 ± 12.77, mean and SD) and dialysis therapy duration ranging 6–138 months (50.56 ± 34.67). The etiology of renal failure was chronic glomerulonephritis (65 patients), diabetic nephropathy (60 patients), vascular nephropathy (55 patients), hypertension (51 patients), interstitial chronic nephropathy (45 patients), other etiologies (18 patients) and unknown cause (16 patients). The vascular access was through a native arteriovenous fistula (285 patients) and a permanent jugular catheter (25 patients). Patients had dialysis three times a week, in 3- to 4-h sessions, with a pump arterial blood flow of 200–350 ml/min and flow of the dialysis bath of 500–800 ml/min. The dialysate consisted of the following constituents: sodium 141 mmol/l, potassium 2.0 mmol/l, calcium 1.3 mmol/l, magnesium 0.2 mmol/l, chloride 108.0 mmol/l, acetate 3.0 mmol/l and bicarbonate 35.0 mmol/l. Special attention was paid to the real dialysis time, so that time-counters were fitted to all machines for all sessions, to record effective dialysis duration (excluding any unwanted interruptions, for example, due to dialysis hypotensive episodes). All patients were dialyzed with 1.0 m^2 polyethersulfone low-flux dialyzer, 1.2 m^2 cellulose-synthetic low-flux dialyzer (hemophane), 1.3 m^2 polyethersulfone low-flux dialyzer, 1.3 m^2 low-flux polysulfone dialyzer, 1.6 m^2 low-flux polysulfone dialyzer and 1.3 m^2 high-flux polysulfone dialyzer. The dialysis technique was conventional hemodialysis, no patient being treated with hemodiafiltration. A Fresenius model 4008B and 4008S dialysis machine equipped with a volumetric ultrafiltration control system was used in each dialysis. Fluid removal was calculated as the difference between the patients' weight before dialysis and their target dry weight. Pre-dialysis body weight, blood pressure, pulse rate and axillary temperature were measured before ingestion of food and drink. Pre-dialysis BUN (C_{pre}) was sampled from the arterial port before the blood pump was started. Post-dialysis BUN (C_{post}) was obtained from the arterial port at the end of HD after slowing the blood flow rate (Q_B) to 50 ml/min, and the blood extraction was made 15 s later [102].

3 Model description

3.1 Multilayer perceptron

Multilayer perceptron (MLP) is one of the most frequently used neural network architectures in biomedical applications, and it belongs to the class of supervised neural networks [24]. It performs a humanlike reasoning, learns the attitude and stores the relationship of the processes on the basis of a representative data set that already exists. The attraction of MLP can be explained by the ability of the network to learn complex relationships between input and output patterns, which would be difficult to model with conventional methods. It consists of a network of nodes arranged in layers. A typical MLP network consists of three or more layers of processing nodes: an input layer, one or more hidden layers and an output layer (see Fig. 1). The input layer distributes the inputs to subsequent layers. Input nodes have liner activation functions (AFs) and no thresholds. Each hidden unit node and each output node have thresholds associated with them in addition to the weights. The hidden unit nodes have nonlinear AFs, and the outputs have linear AFs. Hence, each signal feeding into a node in a subsequent layer has the original input multiplied by a weight with a threshold added and then is passed through the AFs that may be linear or nonlinear (hidden units). Note that unlike other layers, no computation is involved in the input layer. The principle of the network is that when data are presented at the input layer, the network nodes perform calculations in the successive layers until an output value is obtained at each of the output nodes. This output signal should be able to indicate the appropriate class for the input data. As shown in Fig. 1, the input to the Jth hidden-layer neuron is given by

$$\text{net}_j = \sum_{i=1}^{N_i} w_{ji}x_i - \theta_j \quad 1 \le j \le N_H \tag{1}$$

where net_j is the weighted sum of inputs $x_1, x_2, \ldots, x_p$, θ_j is the bias, w_{ji} is the connection weight between the input x_i and the hidden neuron j. The output of the jth hidden neuron is expressed by

$$y_j = f(\text{net}_j). \tag{2}$$

The logistic (sigmoid) function is a common choice of the AFs in the hidden layers, as defined in (3).

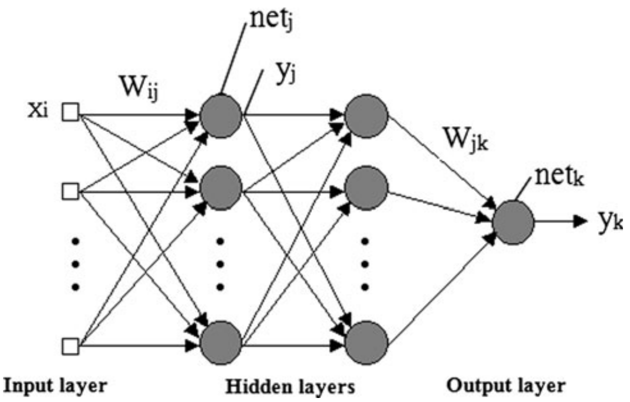


Fig. 1 Multilayer perceptron (MLP) neural network

$$f(\mathrm{net}_j) = \frac{1}{1 + e^{-s\mathrm{net}_j}} \tag{3}$$

where s is the slope of the sigmoid function and by adjusting the value of s to less than unity makes the slope shallower with the effect that the output will be less clear. The bias term θ_j contributes to the left or right shift of the sigmoid AFs, depending on whether θ_j takes a positive or negative value. This function approaches 1, for large positive values of net_j, and 0 for large negative values of net_j [83, 115]. The input to the kth output neuron is given by

$$\mathrm{net}_k = \sum_{j=1}^{N_H} y_j w_{jk} \tag{4}$$

The output of‘ whole neural network is given by:

$$y_k = f(\mathrm{net}_k). \tag{5}$$

The overall performance of the MLP is measured by the mean square error (MSE) expressed as

$$E = \frac{1}{N}\sum_{p=1}^{N_v} E_p = \frac{1}{N}\sum_{p=1}^{N_v}\sum_{i=1}^{M} (t_k(i) - y_k(i))^2 \tag{6}$$

where E_p corresponds to the error for the Pth pattern, N is the number of training patterns, t_k is the desired target of the kth output neuron for the Pth pattern and y_k is the output of the kth output neuron from the trained network.

3.2 Feature selection

The purpose of feature selection can be broadly categorized into visualization, understanding of data, data cleaning, redundancy and/or irrelevancy removal, and performance (e.g., prediction and comprehensibility) enhancement. The inclusion of irrelevant and redundant features increases the computational complexity of the classifier/predictor. Consequently, feature selection has been the one area of much research efforts [21, 69, 133]. Different sets of input parameters were tested in order to see which of them gave the best prediction of the post-dialysis urea concentration (C_{post}). Since one of the drawbacks of ANNs is the interpretation of the final model in terms of the importance of variables, DTREG predictive modeling software was used to compute the importance of variable [121]. The calculation is performed using sensitivity analysis where the values of each variable are randomized and the effect on the quality of the model is measured. Sensitivity analysis is based on the measurement of the effect that is observed in the output y_k due to the change that is produced in the input x_i. Thus, the greater the effect observed in the output, the greater the sensitivity present with respect to the input [97]. Theodore, six important variables are selected as the data set for ANN-C_{post} predictor. They are urea pre-dialysis (C_{pre}, mg/dL) at the beginning of the procedure, ultrafiltration volume (UFV, dL), pre-dialysis weight (Pre-Wt, kg), effective blood-side urea clearance (KeUB, dL/min), desired dialysis time (T_d, min) and urea generation rate (G, mg/min). The generation rate of urea is linked to the protein nitrogen appearance rate because most protein nitrogen is excreted as urea. A low pre-dialysis or time-averaged plasma urea level may be found in patients in whom urea removal is inadequate but in whom the urea generation rate is also low (e.g., due to poor protein intake).

3.3 Training methodology and ANN configuration

During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function (see 6). Backpropagation (BP) is widely used for predictive modeling by using the concept of MLP training and testing [111]. The backpropagation algorithm is a gradient-descent method to minimize the squared-error cost function [116]. Assume that for each layer, the error in the output of the layer is known. If the error of the output is known, then it is not hard to calculate changes for the weights, so as to reduce that error. Backpropagation gives a way to determine the error in the output of a prior layer by giving the output of a current layer as feedback. The process is therefore iterative: starting at the last layer and calculating the changes in the weight of the last layer. Then, calculate the error in the output of the prior layer and repeat. In basic backpropagation learning algorithms, the synaptic weights W are updated as follows:

$$W_{k+1} = W_k + \Delta W_k \tag{7}$$

where k is the iteration in discrete time and ΔW_k is the current weight adaptation that can be expressed as follows [73]:

$$\Delta W_k = -\eta \frac{\partial e_k}{\partial W_k} \tag{8}$$

where η is the learning rate, usually its vale is between zero and one, and $\partial e_k / \partial W_k$ is the gradient of the error function in (6) to be minimized.

However, the major disadvantages of standard BP are its relatively slow convergence rate and being trapped at the local minima because it solves (8) iteratively until the error function reaches the minimum [140]. Backpropagation using gradient descent often converges very slowly or not at all. On large-scale problems, its success depends on user-specified learning rate and momentum parameters. There is no automatic way to select these parameters, and if incorrect values are specified, the convergence may be exceedingly slow, or it may not converge at all. While backpropagation with gradient descent is still used in many neural network programs, it is no longer considered to be the best or fastest algorithm [121]. Therefore, many powerful optimization algorithms have been devised for faster convergence than steepest descent direction methods. The first category of these algorithms is based on heuristic techniques, which were developed from the performance analysis of the standard steepest descent algorithm such as variable learning rate BP and resilient BP [16]. The second category is based on standard numerical optimization techniques such as conjugate gradient algorithms [17, 61, 76, 82], quasi-Newton algorithms [14] and Levenberg–Marquardt [94]. The conjugate gradient algorithms are usually much faster than variable-learning-rate backpropagation algorithm, although the results will vary from one problem to another. However, conjugate gradient algorithms require a line search at each iteration. This line search is computationally expensive, because it requires that the network response to all training inputs be computed several times for each search. Therefore, the scaled conjugate gradient algorithm (SCG), developed by Moller [96], was designed to avoid the time-consuming line search per learning iteration, which makes the algorithm faster than other second-order algorithms. Newton's method is an alternative to the conjugate gradient methods for fast optimization and convergence [16]. Quasi-Newton method is recommended for most networks with a small number of weights (less than a couple of hundred). It is a batch update algorithm, whereas backpropagation adjusts the network weights after each case. Quasi-Newton works out the average gradient of the error surface across all cases before updating the weights once at the end of the epoch. There is also no need to select learning or momentum rates for quasi-Newton, so it can be much easier to use than backpropagation. Levenberg–Marquardt (LM) training is considered the most efficient algorithm for training median-sized artificial neural networks [72]. It is an intermediate optimization algorithm between the Gauss–Newton method and gradient-descent algorithm addressing the shortcomings of each of those techniques. LM uses gradient descent to improve on an initial guess for its parameters and transforms to the Gauss–Newton method as it approaches the minimum value of the cost function. Once it approaches the minimum, it transforms back to the gradient-descent algorithm to improve the accuracy [72]. This means that when the current solution is far from the correct one, the algorithm behaves like a gradient-descent method, and when the current solution is close to the correct solution, it becomes a Gauss–Newton method. In this study, eight fast learning algorithms are tested; which are scaled conjugate gradient (SCG), conjugate gradient with Powell–Beale restarts (CGB) [105], Polak–Ribiere conjugate gradient (CGP) [71], Fletcher–Reeves conjugate gradient (CGF) algorithm [51, 52], Levenberg–Marquardt (LM) [94], resilient backpropagation (RP) [109], BFGS quasi-Newton [14, 39] and one-step secant [15]. These algorithms are compared based on their generalization and prediction performances during training and testing phases.

The modeling process starts by obtaining a data set (input–output data pairs) and dividing it into a training set and validation data set. Training data constitute a set of input and output vectors. There are 310 training data used for training, and to avoid overfitting problems during predictive modeling process, random subsampling cross-validation is used, which is also known as Monte Carlo cross-validation (MCCV) and based on randomly splitting the data into subsets, whereby the size of the subsets is defined by the modeler [9, 103, 110, 134]. Therefore, data are divided randomly into 60, 20 and 20 % for training, validation and testing purposes, respectively. Random subsampling cross-validation improved neural network generalization using early stopping technique. The training set is used for computing the gradient and updating the network weights and biases, whereas the validation set was used to validate and test the ANN model. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error in the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned [16]. A threshold and tolerance value for the error between the actual and desired output is determined. Then, an error for each data pair is found. Successful prediction during the training and test phases was defined as a performance within a tolerance or less.

To choose the best architecture of MLP neural network, it was trained and tested for different configurations. In fact 3-layer model (one input, one hidden and one output) is recommended for MLP. To simplify the study, only three types of AFs were considered: linear (L), log-sigmoid (*logsig*), and tan-sigmoid (*tansig*). Therefore, different topologies with six inputs and three layers (6-X-1) were sequentially trained, using either the logsig or *tansig* AFs in the hidden layer with exception of the input layer for which linear AF was used at all times. It is also recommended that the output layer has liner AFs because if the last layer of a multilayer network has sigmoid neurons, then the outputs of the network are limited to a small range, but if linear output neurons are used, the network outputs can take on any value [16]. The number of neurons used in the hidden layer was varied systematically between 4 and 13 with one step while monitoring the prediction error. In the 6-X-1 topologies, only two possible combinations were created and trained, namely L-logsig-L and L-tansig-L. All networks were trained in batch mode. The weights of the network with the minimum training error were saved as the final network structure. We subsequently used the testing data set to evaluate the predictive ability of the trained network. Each network was trained two different times to account for different random initialization weights. After training, the two possible topologies were compared with each other in terms of performance on the validation set.

3.4 Statistical analysis

Statistical analysis was performed using NCSS 2007 and MedCalc 12.1.0.0 [95, 99]. The results were shown as the mean ± standard deviation (SD) of the mean. Confidence intervals on the discrepancy between different estimates of the same parameter are calculated as Bland–Altman 95 % limits of agreement. Comparison of means between the two groups was made using paired-sample *T* test. The *P* level reported with a *t* test represents the probability of error involved in accepting the hypothesis about the existence of a difference between the means of two groups, and it is the significance level for the statistical test. The performance of NN predictors was evaluated and agreement comparisons between actual and predicted data was made using mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), normalized mean square error (NRMSE), Pearson's correlation coefficient (R_p), concordance statistic (R_C) and median algebraic difference. If there is no systematic bias, the concordance statistic is equal to the Pearson's correlation; otherwise, the magnitude of the concordance statistic is smaller, as it reflects both association and systematic bias between methods. One-way analysis of variance (ANOVA) is used with the number of hidden-layer neurons as factors to determine whether changing the NN structure produced significant changes in network performance (MSE). Statistical significance is tested for one-way ANOVA by comparing the *F* test statistic.

4 Results and discussion

4.1 Testing and validation process of the ANN

Once the model structure and parameters have been identified, it is necessary to validate the quality of the resulting model. In principle, the model validation should not only validate the accuracy of the model but also verify whether the model can be easily interpreted to give a better understanding of the modeled process. It is therefore important to combine data-driven validation, aiming at checking the accuracy and robustness of the model, with more subjective validation, concerning the interpretability of the model. There will usually be a challenge between flexibility and interpretability, the outcome of which will depend on their relative importance for a given application. While it is evident that numerous cross-validation methods exist, the choice of the suitable cross-validation method to be employed in the ANN is based on a trade-off between maximizing method accuracy, stability and minimizing the operation time. In this research, random subsampling cross-validation method is adopted for ANN because of its accuracy and possible implementation. Prediction accuracy is calculated by comparing the difference of predicted and measured values. If the difference is within tolerance, as in $|C_{\text{post-predicted}} - C_{\text{post-measured}}| \leq \varepsilon$, accurate prediction is achieved. The tolerance ε is defined based on the recommendations of an expert in the hemodialysis field. Prediction accuracy of about 1.5 % is allowed for C_{post} prediction and can be defined as follows:

$$\text{Accuracy} = \frac{\left| C_{\text{post}}^{\text{predicted}} - C_{\text{post}}^{\text{measured}} \right| \leq \varepsilon}{|\text{predicted set}|} \quad (9)$$

In the study by Blum [20], Swingler [127], Boger and Guterman [22], Berry and Linoff [18], different rules for selecting the number of hidden units are proposed. These studies concluded that the convenience of adding more hidden units to a multilayer perceptron to find a better local minimum do not always apply because of drawbacks related to random selection of the initial weights of the ANN. Intuitively, increasing the number of identical hidden neurons (HN) represents adding more degrees of freedom to the backpropagation algorithm in a multilayer perceptron trained with early stopping. One rule of thumb is that HN should never be more than twice as large as the input layer [18]. Therefore, the number of hidden neurons was opted between 4 and 13 to check the optimum topology for each model. With this assumption, different trials were performed to find the best ANN in each method using different AFs. The ANN that achieved the lowest MSE was selected as the model. Table 1 shows the results with different hidden node neurons. Bold values are the minimum MSEs achieved for each learning algorithm and were selected as the optimum topology. To investigate the effect of adding hidden neurons, all topologies between 4 and 13 hidden neurons with the two possible combinations of AFs were initialized with random weights and trained with early stopping. The results in Table 1 revealed that adding more hidden neurons does not considerably help reduce the mean squared error of NN model. If the starting weights were kept the same, the error would certainly reduce in trend. However, there is no way to know in advance which initial set of weights will yield the lowest error, and as a conclusion, it is possible for a topology with less hidden units to achieve lower errors than one with more hidden units, if the starting point is different. This

Table 1 Performance metrics of ANN learning method (expressed as MSE) based on the number of hidden node neurons and type of AF

Method	HLAF	Number of hidden neurons									
		6-4-1	6-5-1	6-6-1	6-7-1	6-8-1	6-9-1	6-10-1	6-11-1	6-12-1	6-13-1
RP	AF1	1.1537	1.9848	0.14742	4.4563	1.5137	8.2459	8.8104	13.1864	10.099	7.104
	AF2	1.5865	0.10718	**0.08666**	1.9219	1.3822	11.9275	4.6989	2.7088	8.6728	14.845
LM	AF1	2.0162	0.18659	0.11664	2.1637	0.18206	0.28088	0.07459	**0.04905**	0.0901	0.16217
	AF2	0.35265	1.5525	0.19539	0.20083	0.18254	0.17855	0.07613	0.38169	0.2000	0.25324
SCG	AF1	1.8074	2.2535	18.7721	5.2289	5.2756	11.9003	14.411	20.5877	17.202	22.4559
	AF2	**1.3277**	3.3063	1.8823	2.2688	2.3105	7.0617	12.4238	9.9666	4.3495	13.4992
CGB	AF1	1.16076	**0.7153**	1.1438	1.6965	2.2306	3.1112	3.1968	13.6501	10.6135	15.3126
	AF2	0.73625	1.4091	4.335	0.98929	2.1787	3.0312	1.599	3.5175	8.8405	20.8702
CGP	AF1	3.3188	12.6725	2.6273	5.3287	8.921	7.1729	17.5686	11.9626	28.4648	23.8713
	AF2	**0.77875**	10.1264	1.699	9.6155	1.4256	4.2158	17.6596	13.4107	16.5769	14.894
CGF	AF1	4.6197	9.3199	1.8317	4.8644	3.1679	5.0739	8.3746	**1.0825**	3.7454	7.4189
	AF2	5.9678	8.7034	1.8485	1.7745	2.3877	5.2627	12.4727	2.5998	14.9534	7.189
BFG	AF1	0.11489	1.0696	0.12365	0.29434	1.0877	**0.03955**	0.23222	14.3982	0.48536	19.1641
	AF2	0.31063	0.52206	0.10186	0.48481	0.17788	0.33133	0.078564	17.8813	22.044	10.5629
OSS	AF1	21.0727	7.1308	11.9804	2.2735	8.0358	10.8333	17.0811	15.0792	19.0454	13.7257
	AF2	4.1632	1.9442	1.8478	1.916	4.7367	**1.5364**	14.1921	15.6096	7.9985	13.0616

HLAF hidden-layer activation function, *AF1* logsig AF, *AF2* tansig AF, *RP* resilient backpropagation, *SCG* scaled conjugate gradient, *CGB* conjugate gradient with Powell–Beale restarts, *CGP* Polak–Ribiére conjugate gradient, *CGF* Fletcher–Powell conjugate gradient, *BFG* BFGS quasi-Newton, *OSS* one-step secant

proves the conclusions made by Barron [8], DeVore et al. [45], Lawrence et al. [87], Sarle [114], Tetko et al. [129] and Weigend [136] regarding the convenience of adding more HN to MLP for finding a better local minimum, which does not always apply because of drawbacks related to random selection of the initial weights of the ANN.

The ANN's training convergence speeds of eight learning algorithms are also compared in Table 2. Bold values are the optimum number of epochs and convergence time in sec. The optimum topology for each learning algorithm is shown in Fig. 2. The results in Table 1 indicate that BFGS quasi-Newton and Levenberg–Marquardt algorithm produced the best results. They outperformed clearly all the other methods and were very robust algorithms. For the BFGS quasi-Newton algorithm, an MSE error of 0.03955 is obtained using 9 HN, while an MSE error of 0.04905 was obtained using 11 HN when applying the Levenberg–Marquardt algorithm. The Levenberg–Marquardt algorithm training ended with a gradient value of 0.1339 with an optimum adaptive μ value of 0.1, which it acquired at epoch 92. Quasi-Newton techniques are superior to simple 'batch' gradient descent and lead to significantly better solutions requiring fewer training epochs. Therefore, they constitute serious alternatives to gradient-descent methods. In addition, these methods do not suffer from the problem related to the specification of the learning rate parameter, which is crucial for the performance of the gradient-descent method. The results also indicate that the Marquardt algorithm is very efficient when training networks that have up to a few hundred weights. Resilient BP algorithm was able to generalize well and achieved an MSE of 0.08666 using 6 HN. However, SCG has the smallest convergence speed (epochs = 55 as shown in Table 2), the optimum number of HN is achieved with high MSE (1.3277).

Generally, as shown in Table 3, increasing the number of hidden-layer neurons resulted in decreasing the accuracy of prediction of post-dialysis urea concentration for both types of AFs using resilient backpropagation (RP), scaled conjugate gradient (SCG), conjugate gradient with Powell–Beale restarts (CGB) and Polak-Ribiere conjugate gradient (CGP) quantified by statistically significant changes in MSE and R^2 ($P < 0.05$). There is also statistically significant decrease in MSE by increasing the number of hidden-layer neurons using *tansig* AF for BFGS quasi-Newton and one-step secant methods. For other learning algorithms, the interaction between the number of hidden-layer neurons and the NN performance was not statistically significant ($P > 0.05$).

4.2 Verification of system output

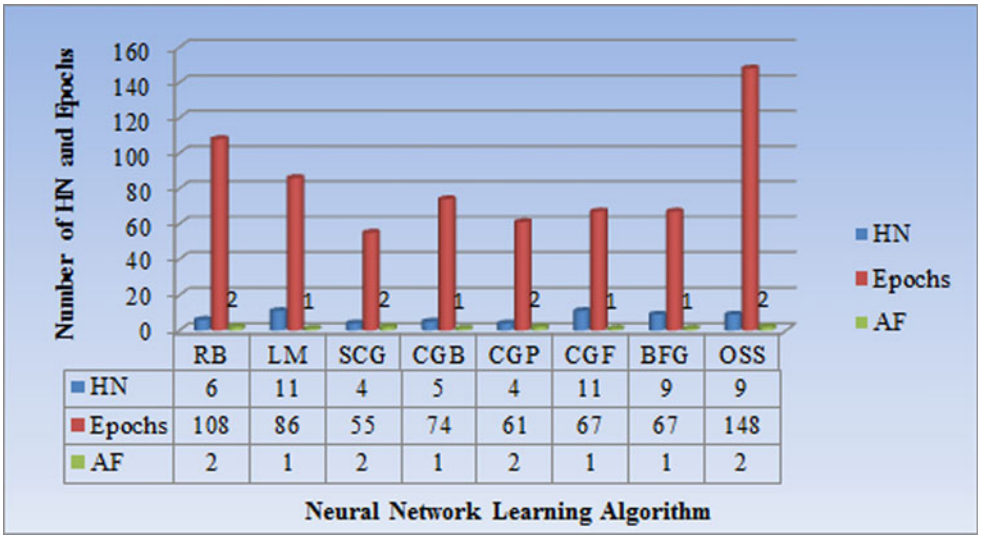
Testing, validation and verification is inevitable part of any modeling process. These processes help modelers and other stakeholders build confidence in the appropriateness and usefulness of the model. Whatever modeling paradigm or solution technique is being used, the performance measures extracted from a model will only have some bearing on the

Table 2 Comparison of convergence speeds [epochs (time in sec)] of different learning methods for ANN

Learning method	HLAF	Number of hidden neurons									
		6-4-1	6-5-1	6-6-1	6-7-1	6-8-1	6-9-1	6-10-1	6-11-1	6-12-1	6-13-1
RB	AF1	10 (0s)	17 (0s)	43 (0s)	9 (0s)	190 (1s)	119 (1s)	29 (0s)	22 (0s)	52 (0s)	54 (0s)
	AF2	5 (0s)	44 (0s)	**108 (1s)**	218 (2s)	164 (1s)	24 (0s)	93 (0s)	163 (0s)	140 (1s)	36 (0s)
LM	AF1	21 (0s)	13 (0s)	22 (0s)	8 (0s)	106 (1s)	82 (1s)	22 (0s)	**86 (1s)**	51 (1s)	42 (0s)
	AF2	168 (2s)	19 (0s)	53 (0s)	49 (0s)	38 (0s)	36 (0s)	61 (1s)	17 (0s)	26 (0s)	12 (0s)
SCG	AF1	110 (1s)	51 (0s)	47 (0s)	43 (0s)	86 (1s)	41 (0s)	24 (0s)	18 (0s)	21 (0s)	28 (0s)
	AF2	**55 (0s)**	33 (0s)	81 (1s)	77 (1s)	92 (1s)	36 (0s)	23 (0s)	27 (0s)	73 (1s)	18 (0s)
CGB	AF1	53 (2s)	**74 (2s)**	54 (1s)	93 (3s)	44 (1s)	72 (2s)	26 (0s)	33 (1s)	37 (1s)	17 (0s)
	AF2	53 (1s)	43 (1s)	36 (1s)	87 (2s)	40 (1s)	79 (2s)	91 (2s)	21 (0s)	40 (1s)	27 (0s)
CGP	AF1	99 (2s)	27 (0s)	72 (2s)	33 (1s)	44 (1s)	27 (0s)	26 (1s)	15 (0s)	20 (0s)	11 (0s)
	AF2	**61 (1s)**	19 (0s)	77 (2s)	24 (0s)	111 (2s)	26 (0s)	24 (0s)	11 (0s)	18 (0s)	32 (0s)
CGF	AF1	61 (1s)	13 (0s)	50 (1s)	48 (1s)	39 (1s)	44 (1s)	39 (1s)	**67 (2s)**	54 (1s)	32 (0s)
	AF2	25 (0s)	21 (0s)	87 (2s)	78 (2s)	111 (3s)	40 (1s)	22 (0s)	71 (2s)	23 (0s)	46 (1s)
BFG	AF1	163 (3s)	34 (1s)	62 (1s)	61 (1s)	34 (1s)	**67 (2s)**	56 (2s)	51 (1s)	44 (1s)	28 (0s)
	AF2	59 (1s)	57 (1s)	104 (2s)	61 (1s)	105 (3s)	114 (3s)	113 (3s)	139 (4s)	826 (28s)	141 (7s)
OSS	AF1	26 (0s)	32 (1s)	35 (1s)	49 (1s)	26 (0s)	42/0.01	16 (0s)	27 (1s)	14 (0s)	25 (0s)
	AF2	40 (1s)	82 (2s)	61 (1s)	88 (2s)	35 (1s)	**148/(4s)**	16 (0s)	17 (0s)	55 (1s)	35 (1s)

HLAF hidden-layer activation function, *AF1* logsig AF, *AF2* tansig AF, *RP* resilient backpropagation, *SCG* scaled conjugate gradient, *CGB* conjugate gradient with Powell–Beale restarts, *CGP* Polak–Ribiére conjugate gradient, *CGF* Fletcher–Powell conjugate gradient, *BFG* BFGS quasi-Newton, *OSS* one-step secant

Fig. 2 Optimum topology for NN learning algorithms



	RB	LM	SCG	CGB	CGP	CGF	BFG	OSS
HN	6	11	4	5	4	11	9	9
Epochs	108	86	55	74	61	67	67	148
AF	2	1	2	1	2	1	1	2

real system represented if the model is a good representation of the system. Of course, what constitutes a good model is subjective, but from a performance modeling point of view, our criteria for judging the goodness of models will be based on how accurately measures extracted from the model correspond to the measures that would be obtained from the represented system. Verification is like debugging; it is intended to ensure that the model does what it is intended to do. It is important to remember that validation does not imply verification, nor verification implies validation. However, in practice, validation is often blended with verification, especially when measurement data are available for the system being modeled. If a comparison of system measurements and model results suggests that the results produced by the model are close to those obtained from the system, then the implemented model is assumed to be both a verified implementation of the assumptions and a valid representation of the system. Verification of the proposed system is divided into two phases to examine the output behavior of the model.

Table 3 Effect of changing the number of hidden neurons on NN performance with different learning algorithms

Learning method	AF1			AF2		
	R^2	*F* ratio	*P* value	R^2	*F* ratio	*P* value
RP	0.6500	14.8580	0.0048*	0.5507	9.8057	0.0140*
LM	0.28	3.1453	0.1141	0.176	1.7082	0.2275
SCG	0.6117	12.6003	0.0075*	0.5896	11.4952	0.0095*
CGB	0.7436	23.1983	0.0013*	0.4801	7.3878	0.0263*
CGP	0.6306	13.6557	0.0061*	0.5167	8.5533	0.0192*
CGF	0.0012	0.0097	0.9241	0.3732	1.2949	0.2881
BFG	0.3882	5.0751	0.0543	0.5160	8.5297	0.0193*
OSS	0.0643	0.5495	0.4797	0.5403	9.4040	0.0154*

* Statistically significant ($P < 0.05$)

- The first phase is a comparison between measured C_{post}, calculated one by spUKM and predicted values using the best topologies of eight NN models.
- The second phase is a comparison between ${}_{\text{sp}}Kt/V$ calculated by the NN models and other traditional methods.

4.2.1 *Comparison of measured post-dialysis urea concentration, calculated value by spUKM and predicted values using the best topologies of NN models*

In order to verify the accuracy of the proposed system, the estimated post-dialysis urea concentration by the model was compared with the measured value and the calculated one by single-pool variable volume (SPVV) urea kinetic model as proposed by Gotch and Sargent [65] to determine how frequently these methods would lead to a different prescription management. Post-dialysis urea concentration can be calculated by SPVV using the following formula [63]:

$$C(t) = C_o\left(\frac{V_o - Q_f t}{V_0}\right)^{\frac{K-Q_f}{Q_f}} + \frac{G}{K - Q_f}\left[1 - \left(\frac{V_o - Q_f t}{V_0}\right)^{\frac{K-Q_f}{Q_f}}\right] \tag{10}$$

In Table 4, the predicted C_{post} by ANN models and the calculated values by SPVV urea kinetic model are compared to the reference measured sample. The result of the table clearly demonstrates the superiority of ANN models over UKM. The mean value of measured post-dialysis urea concentration sample ($C_{\text{post-measured}}$) was 49.2774 ± 12.4092 mg/dL (median = 49 mg/dL) with a coefficient of variation (CV) of 25.2 %. The estimated value of C_{post} by SPVV-UKM was 54.5935 ± 14.1303 mg/dL. The mean value of ANN_{RP}, ANN_{LM}, ANN_{SCG}, ANN_{CGB}, ANN_{CGP}, ANN_{CGF}, ANN_{BFGS} and ANN_{OSS} are 49.2703 ± 12.3906 mg/dL (median = 49.1364 mg/dL), 49.2699 ± 12.4175 mg/dL (median = 49.1428 mg/dL), 49.2455 ± 12.3569 mg/dL (median = 48.8908 mg/dL), 49.2986 ± 12.338 mg/dL (median = 49.1254 mg/dL), 49.2920 ± 12.2014 mg/dL (median = 48.8287 mg/dL), 49.1963 ± 12.3866 mg/dL (median = 49.0743 mg/dL), 49.2995 ± 12.3942 mg/dL (median = 49.1554 mg/dL) and 49.31421 ± 12.3337 (median = 49.3978).

Statistically significant difference was found between measured C_{post} and $C_{\text{post-SPVV-UKM}}$ ($P < 0.0001$), but no statistically significant differences were found between measured C_{post} and ANN models ($P > 0.05$). The SPVV-UKM overestimated the post-dialysis urea concentration (median difference $\Delta = -5$), leading to an underestimation of ${}_{\text{sp}}Kt/V$. The SPVV-UKM method resulted in a relatively large absolute error of C_{post} than ANN models (5.6839). In terms of accuracy, the results revealed that Levenberg–Marquardt, BFGS quasi-Newton algorithm and resilient backpropagation are the acceptable accurate NN algorithms. The accuracy of other NN learning algorithms is out of acceptable clinical range as shown in Table 4. However, BFGS quasi-Newton algorithm achieved the smallest MSE in the training phase; Levenberg–Marquardt algorithm outperformed clearly all the other algorithms in the verification phase and was a very robust algorithm in terms of MAE, RMSE, R_p^2 and concordance coefficient. The percentage of MAE and RMSE for Levenberg–Marquardt is 0.27 and 0.32 %, respectively, compared to 0.38 and 0.41 % for BFGS quasi-Newton and 0.44 and 0.48 % for resilient backpropagation. Therefore, Levenberg–Marquardt NN model was selected in subsequent analysis for single-pool dialysis dose estimation. The results also show that one-step secant algorithm is the slowest network, and it has the worst results in the verification phase.

Measurement biases of 0.0075, −0.0221 and 0.0071 were observed for ANN_{LM}, ANN_{BFGS} and ANN_{RP}, respectively. The limits of agreement using Bland and Altman agreement analysis [19] indicate a possible error of +0.43 to −0.41 mg/dL when using $C_{\text{post\_ANNLM}}$ instead of the measured C_{post}, which is acceptable from a routine point of view as shown in Fig. 3. When $C_{\text{post-ANNBFGS}}$ is used instead of the measured C_{post}, error would fall between

Table 4 Comparisons of C_{post} predicted by ANN models and calculated by SPVV-UKM with the reference measured sample

Agreement comparison	Comparisons with the reference measured C_{post} sample								
	SPVV-UKM	ANN_{RP}	ANN_{LM}	ANN_{SCG}	ANN_{CGB}	ANN_{CGP}	ANN_{CGF}	ANN_{BFGS}	ANN_{OSS}
Mean bias	−5.3161	0.0071	0.0075	0.0789	−0.0212	−0.0146	0.0812	−0.0221	−0.0368
MAE	5.6839	0.2106	0.1274	0.9854	0.7768	0.7786	0.8543	0.1822	0.9398
MAPE	0.1171	0.0044	0.0027	0.0215	0.0172	0.0176	0.0185	0.0038	0.0200
RMSE	6.4119	0.3142	0.2131	1.4059	1.0653	1.0998	1.0985	0.2715	1.2659
NRMSE	0.0972	0.0048	0.0032	0.0213	0.0161	0.0167	0.0166	0.0041	0.0192
P value of *t* test	<0.0001	0.9943	0.9939	0.9744	0.9830	0.9882	0.9351	0.9823	0.9705
Median difference	−5	−0.1364	−0.1428	0.1092	−0.1254	0.1713	−0.0743	−0.1554	−0.3978
Pearson's R_p^2	0.9442	0.9994	0.9997	0.9843	0.9926	0.9923	0.9922	0.9995	0.9896
Concordance R_c	0.8920	0.9997	0.9999	0.9935	0.9963	0.9960	0.9961	0.9998	0.9947
Accuracy (%)[a]	10.3276	0.4389	0.2690	2.1633	1.7148	1.7265	1.8537	0.3782	2.002

[a] For the current study, accurate model must be less than or equal to 1.5 % as defined by (9)

+0.51 and −0.55 mg/dL as shown in Fig. 4, which falls into acceptable margins for routine estimation. As shown in Fig. 5, a possible error of +0.62 to −0.61 mg/dL was obtained when using C_{post_ANNRP} instead of the measured C_{post}.

4.2.2 *Comparison of estimated $_{sp}Kt/V$ by ANN_{LM} and other traditional methods*

Several different methods are in use for calculating $_{SP}Kt/V$, giving results that vary significantly. To obtain meaningful comparisons, single-pool dialysis dose *Kt*/*V* was calculated according to the best selected NN models (ANN_{LM}) and also according to Lowrie and Teehan [91], Jindal et al. [81], Keshaviah et al. [86], Barth [10], Calzavara et al. [26], Daugirdas [35], Basile et al. [13], Ijely and Raja [78],

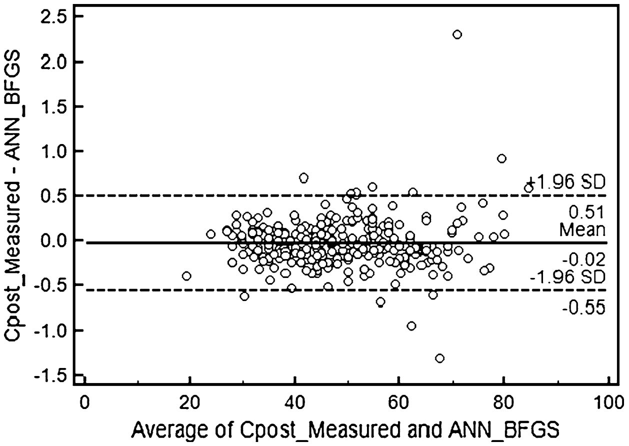


Fig. 4 Bland–Altman plot showing the mean of ($C_{post\text{-}measured}$ and $C_{post\text{-}ANNBFGS}$) compared with the difference between means

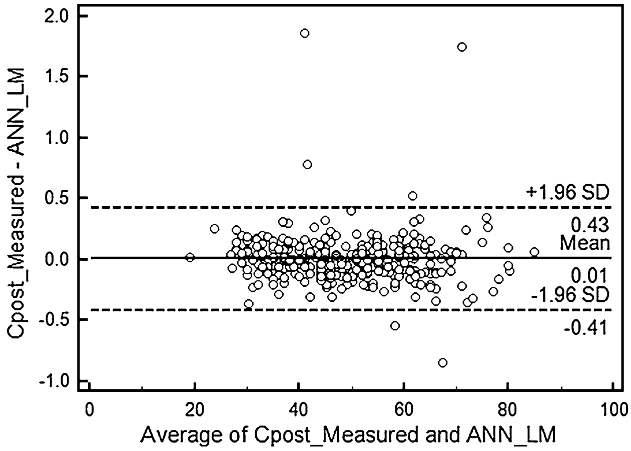


Fig. 3 Bland–Altman plot showing the mean of ($C_{post\text{-}measured}$ and C_{post_ANNLM}) compared with the difference between means

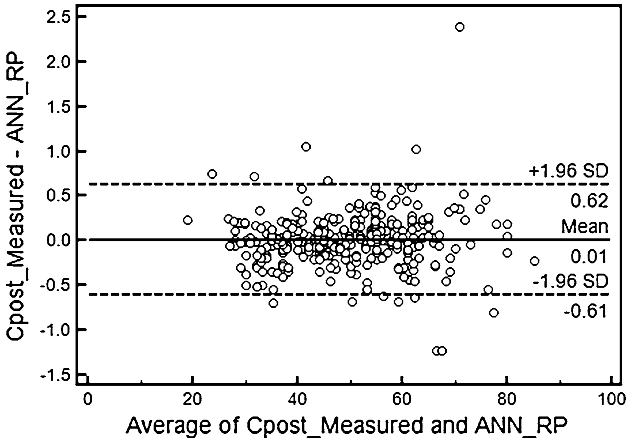


Fig. 5 Bland–Altman plot showing the mean of ($C_{post\text{-}measured}$ and $C_{post\text{-}ANFDIAC}$) compared with the difference between means

Daugirdas [36], Kerr et al. [85] and Azar [7] models. Table 5 summarizes the simplified models used for comparison.

The single-pool *Kt*/*V* estimated by $\mathrm{ANN}_{\mathrm{LM}}$ can be calculated using Daugirdas [36] formula by replacing the ratio R by the following ratio:

$$R_{\mathrm{ANN}} = \frac{C_{\mathrm{post-ANN_{LM}}}}{C_{\mathrm{pre}}}. \qquad (11)$$

The descriptions of the ${}_{\mathrm{sp}}Kt/V$ results obtained from the $\mathrm{ANN}_{\mathrm{LM}}$ and from the simplified models are shown in Table 6. Mean, standard deviation, variance, median, minimum and maximum ranges of all methods are summarized in this Table. The different ${}_{\mathrm{sp}}Kt/V$ calculations were compared with *Kt*/*V* calculated using $C_{\mathrm{post\_ANNLM}}$, which was considered to be the reference method. Single-pool *Kt*/*V* estimated by $\mathrm{ANN}_{\mathrm{LM}}$ was 1.3710 ± 0.1786 and had been compared with (1) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Lowrie}}$, 1.1486 ± 0.1458; (2) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Jindal}}$, 1.5177 ± 0.1982; (3) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Keshaviah}}$, 1.3347 ± 0.1694; (4) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Barth}}$, 1.4462 ± 0.1536; (5) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Calzavara}}$, 1.0331 ± 0.1094; (6) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Basile}}$, 1.2787 ± 0.1139; (7) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Ijely}}$, 1.2230 ± 0.08917; (8) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Daugirdas1}}$, 1.4134 ± 0.2046; (9) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Daugirdas2}}$, 1.3708 ± 0.1789; (10) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Kerr}}$, 1.3736 ± 0.2081 and (11) ${}_{\mathrm{sp}}Kt/V_{\mathrm{Azar}}$, 1.2739 ± 0.1560.

Table 5 Different models for simplified calculation of ${}_{\mathrm{sp}}Kt/V$ values

Model	Formula
Lowrie and Teehan [89]	$Kt/V = \ln\left(C_{\mathrm{pre}}/C_{\mathrm{post}}\right)$
Jindal et al. [81]	$Kt/V = \left[\frac{0.04\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{C_{\mathrm{post}}} \times 100\right] - 1.2$
Keshaviah et al. [84]	$Kt/V = 1.62\ln\left(C_{\mathrm{pre}}/C_{\mathrm{post}}\right)$
Barth [10]	$Kt/V = \left[\frac{0.031\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{C_{\mathrm{post}}} \times 100\right] - 0.66$
Calzavara et al. [26]	$Kt/V = \left[\frac{2\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{\left(C_{\mathrm{pre}}+C_{\mathrm{post}}\right)}\right]$
Daugirdas first generation [35]	$Kt/V = -\ln(R - 0.008 \times t - UF/Wt)$
Basile et al. [13]	$Kt/V = \left[\frac{0.023\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{C_{\mathrm{post}}} \times 100\right] - 0.284$
Ijely and Raja [77]	$Kt/V = \left[\frac{0.018\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{C_{\mathrm{post}}} \times 100\right]$
Daugirdas second generation [36]	$Kt/V = -\ln(R - 0.008 \times t) + (4 - 3.5 \times R) \times \mathrm{UF}/Wt$
Kerr et al. [83]	$Kt/V = \left[\frac{0.042\left(C_{\mathrm{pre}}-C_{\mathrm{post}}\right)}{C_{\mathrm{post}}} \times 100\right] - 1.48$
Azar [7]	$Kt/V = -0.081 + 1.082\ln\left(C_{\mathrm{pre}}\right) - 1.053\ln\left(C_{\mathrm{post}}\right)$

C_{pre} pre-dialysis blood urea concentration, C_{post} post-dialysis blood urea concentration, *R* $C_{\mathrm{post}}/C_{\mathrm{pre}}$, *t* dialysis duration (hours), *UF* ultrafiltration volume per dialysis (L), *W* post-dialysis body mass of the patient (kg)

The differences between calculated ${}_{\mathrm{sp}}Kt/V$ by traditional methods and ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ values comparisons are shown in Table 7. The estimated ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ values were statistically different ($P < 0.01$) from the calculated ${}_{\mathrm{sp}}Kt/V$ values from other models, except for those ${}_{\mathrm{sp}}Kt/V$ values calculated according to the Daugirdas second-generation formula ($P = 0.9855$, $t = 0.0182$) and Kerr ($P = 0.8683$, $t = -0.1659$). The biggest absolute difference from ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ mean values was obtained using Calzavara's, Lowrie's, Jindal's and Ijely's models [delta of 0.3379, 0.2224, −0.1467 and 0.1481, respectively ($P < 0.05$)]. The least absolute difference from ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ mean values was found to be 2.65 % using Keshaviah's model (delta of 0.0363).

The Bland and Altman agreement analysis shows that one can expect a minimum difference from ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ values of +0.01 to −0.01 using Daugirdas second-generation method as shown in Fig. 6 and +0.14 to −0.14 using Kerr's model as shown in Fig. 7. The mean value of ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ was almost 0.019 % higher than ${}_{\mathrm{sp}}Kt/V$ values calculated according to the Daugirdas second-generation formula, and this difference was not statistically significant.

As shown in Fig. 7, the mean value of ${}_{\mathrm{sp}}Kt/V_{\mathrm{ANNLM}}$ was almost 0.188 % lower than ${}_{\mathrm{sp}}Kt/V$ values calculated according to the Kerr's model, and this difference was also not statistically significant.

5 Conclusion

To attain the best solution in a specific problem, several NN learning algorithms must be tested. The choice of the ANN topology is critical for delineating, and it is clear that there is not an algorithm that outperforms all the other NN algorithms in all problems. Each algorithm should be executed several times with different weights' initialization, and the best ANN can be chosen. For an optimal assessment of network-based prediction, further validation would be required in the form of a prospective comparison with clinical practice and conventional statistical analysis. Therefore, a comparison analysis for the generalization capability of ANN using eight different learning algorithms in training has been studied. All tested artificial neural network (ANN) were feedforward networks with nonlinear AFs in the hidden layer and a linear output layer that predicted the target value. This research was done after collecting a total of 310 medical records from four dialysis centers during 6 months. This study demonstrated the feasibility of predicting the post-dialysis blood urea

Table 6 ${}_{sp}Kt/V$ results obtained with ANN_{LM} model and other traditional methods

Model	Mean	SD	Variance	Median	Min	Max
ANN_{LM}	1.3710	0.1786	0.0319	1.3845	0.7129	1.7214
Lowrie and Teehan [89]	1.1486	0.1458	0.02125	1.1618	0.6627	1.4083
Jindal et al. [81]	1.5177	0.1982	0.03926	1.5483	0.7381	1.8217
Keshaviah et al. [84]	1.3347	0.1694	0.02869	1.3500	0.7700	1.6364
Barth [10]	1.4462	0.1536	0.02358	1.4699	0.8421	1.6818
Calzavara et al. [26]	1.0331	0.1094	0.01198	1.0466	0.6395	1.2140
Basile et al. [13]	1.2787	0.1139	0.01298	1.2963	0.8304	1.4535
Ijely and Raja [77]	1.2230	0.08917	0.007951	1.2367	0.8722	1.3598
Daugirdas first generation [35]	1.4134	0.2046	0.04186	1.4205	0.7164	1.8791
Daugirdas second generation [36]	1.3708	0.1789	0.03202	1.3819	0.7168	1.7234
Kerr et al. [83]	1.3736	0.2081	0.04329	1.4057	0.5551	1.6928
Azar [7]	1.2739	0.1560	0.02433	1.2910	0.7495	1.5531

SD standard deviation

Table 7 Differences between ${}_{sp}Kt/V$ obtained with ANN_{LM} and other traditional models

Model	Mean difference	SE of difference	95 % CI	
Lowrie and Teehan [91]	0.2224 ± 0.06246	0.0131	0.2154	0.2294
Jindal et al. [81]	−0.1467 ± 0.0660	0.0152	−0.1541	−0.1393
Keshaviah et al. [86]	0.0363 ± 0.0580	0.0139	0.0298	0.0428
Barth [10]	−0.0752 ± 0.0609	0.0134	−0.0820	−0.0684
Calzavara et al. [26]	0.3379 ± 0.0830	0.0119	0.3287	0.3472
Basile et al. [13]	0.0923 ± 0.0804	0.0120	0.0834	0.1013
Ijely and Raja [78]	0.1481 ± 0.09892	0.0113	0.1370	0.1591
Daugirdas first generation [35]	−0.04241 ± 0.0329	0.0154	−0.0461	−0.0387
Daugirdas second generation [36]	0.00026 ± 0.00522	0.0143	0.00032	0.00084
Kerr et al. [85]	−0.00258 ± 0.0710	0.0156	−0.0105	0.0054
Azar [7]	0.09714 ± 0.05756	0.0135	0.0907	0.1036

SE standard error, *95 % CI* 95 % confidence interval of the differences

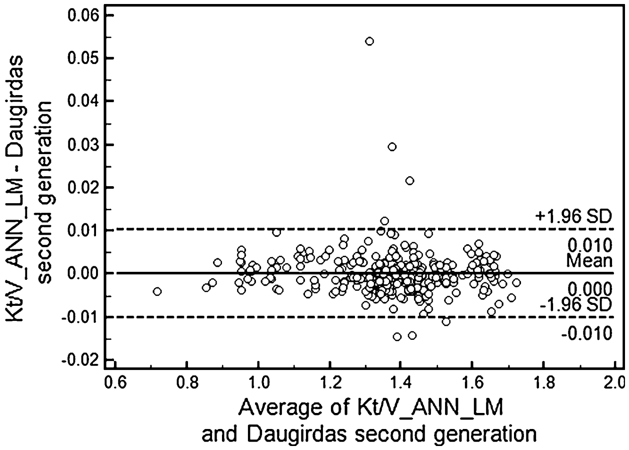


Fig. 6 Bland–Altman plot showing the mean of ${}_{sp}Kt/V_{ANNLM}$ and ${}_{sp}Kt/V_{Daugirdas\ second\ generation}$ compared with the difference between means

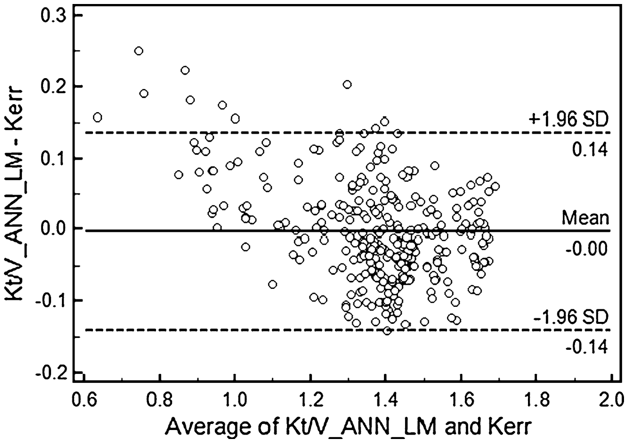


Fig. 7 Bland–Altman plot showing the mean of ${}_{sp}Kt/V_{ANNLM}$ and ${}_{sp}Kt/V_{Kerr}$ compared with the difference between means

concentration (C_{post}) using Levenberg–Marquardt NN learning algorithm with an RMSE of 0.2131 mg/dL and an overall predictive accuracy of 0.2690 %. These results were obtained with a 6-11-1 network topology with linear AFs in the input layer, log-sigmoid AFs in the hidden layer and linear AFs in the output layer. In general, all the algorithms were able to generalize and provide a good fit of the data set, but in terms of accuracy, the results revealed that Levenberg–Marquardt, BFGS quasi-Newton algorithm and resilient backpropagation are the acceptable accurate NN algorithms because their predicted values fell into acceptable margins for routine estimation. LM algorithm can provide better generalization performance compared to the other algorithms. It is robust and fast. The OSS learning algorithm did not attain good results, required lots of computing power and was very slow. The accuracy of the ANN was prospectively compared with other traditional methods for predicting single-pool dialysis dose ($_{sp}Kt/V$). The results are highly promising, and a comparative analysis suggested that NN approach outperformed other traditional models. Although neural networks never replace the human experts, they can helpful for decision making, prediction, classifying and screening and also can be used by domain experts to cross-check their diagnosis. Further work is required to develop these concepts for other dialysis variables in order to minimize blood sampling techniques and translate them into rigorous outcome-based adequacy targets suitable for clinical usage and searching of the optimum dialysis treatment patterns for each individual needs of the patient. The comparative analysis can also be applied in other medical applications in order to select the best learning neural network algorithm.

Acknowledgments I would like to highly appreciate and gratefully acknowledge Phillip H. Sherrod, software developer and consultant on predictive modeling, for his support and consultation during modeling process. The author thanks all medical staff at the nephrology Department in Ahmad Maher Teaching Hospital, Cairo, Egypt, for their invaluable support during the course of this study.

References

1. Acchiardo SR, Hatten KW, Ruvinsky MJ, Dyson B, Fuller J, Moore LW (1992) Inadequate dialysis increases gross mortality rate. ASAIO J 38(3):M282–M285
2. Ahrenholz P, Taborsky P, Bohling M et al (2011) Determination of dialysis dose: a clinical comparison of methods. Blood Purif 32(4):271–277
3. Akl AI, Sobh MA, Enab YM, Tattersall J (2001) Artificial intelligence: a new approach for prescription and monitoring of hemodialysis therapy. Am J Kidney Dis 38(6):1277–1283
4. Alloatti S, Molino A, Manes M, Bosticardo GM (1998) Urea rebound and effectively delivered dialysis dose. Nephrol Dial Transplant 13(Suppl 6):25–30
5. Azar AT, Wahba KM (2011) Artificial neural network for prediction of equilibrated dialysis dose without intradialytic sample. Saudi J Kidney Dis Transpl 22(4):705–711
6. Azar AT, Balas VE, Olariu T (2010) Artificial neural network for accurate prediction of post-dialysis urea rebound. doi:10.1109/SOFA.2010.5565606
7. Azar AT (2008) Estimation of accurate and new method for hemodialysis dose calculation. Clin Med Insights Urol Issue 1:15–21
8. Barron AR (1993) Universal approximation bounds for superpositions of a sigmoid function. IEEE Trans Inf Theory 39(3):930–945
9. Baroni M, Clementi S, Cruciani G, Costantino G, Rignanelli D, Oberrauch E (1992) Predictive ability of regression models: part II. Selection of the best predictive PLS model. J Chemom 6(6):347–356
10. Barth RH (1988) Direct calculation of Kt/V: a simplified approach to monitoring of haemodialysis. Nephron 50(3):191–195
11. Basile C, Vernaglione L, Lomonte C et al (2010) A new index of hemodialysis adequacy: clearance × dialysis time/bioelectrical resistance. J Nephrol 23(5):575–586
12. Basile C, Vernaglione L, Lomonte C et al (2010) Comparison of alternative methods for scaling dialysis dose. Nephrol Dial Transplant 25(4):1232–1239
13. Basile C, Casino F, Lopez T (1990) Percent reduction in blood urea concentration during dialysis estimates Kt/V in a simple and accurate way. Am J Kidney Dis 15(1):40–45
14. Battiti R, Masulli F (1990) BFGS optimization for faster and automated supervised learning, INCC 90 Paris, international neural network conference, pp 757–760
15. Battiti R (1992) First and second order methods for learning: between steepest descent and Newton's method. Neural Comput 4(2):141–166
16. Beale MH, Hagan MT, Demuth HB (2011) Neural network toolbox^{TM} 7 user's guide. The MathWorks, Inc., Natick
17. Beale EML (1972) A derivation of conjugate gradients. In: Lootsma FA (ed) Numerical methods for nonlinear optimization. Academic Press, London
18. Berry MJA, Linoff G (1997) Data mining techniques. Wiley, New York
19. Bland JM, Altman DG (1986) Statistical methods for assessing agreement between two methods of clinical measurement. Lancet 8:307–310
20. Blum A (1992) Neural networks in C++. Wiley, New York
21. Blum A, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97(1–2):245–271
22. Boger Z, Guterman H (1997) Knowledge extraction from artificial neural network models, IEEE systems, man, and cybernetics conference, Orlando, FL
23. Breitsameter G, Figueiredo AE, Kochhann DS (2012) Calculation of Kt/V in haemodialysis: a comparison between the formulas. J Bras Nefrol 34(1):22–26
24. Bridle JS (1989) Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Fougelman-Soulie F (ed) Neurocomputing: algorithms, architectures and applications. Springer, Berlin, pp 227–236
25. Burgelman M, Vanholder R, Fostier H, Ringoir S (1997) Estimation of parameters in a two-pool urea kinetic model for hemodialysis. Med Eng Phys 19(1):69–76
26. Calzavara P, Vianello A, da Porto A, Gatti PL, Bertolone G, Caenaro G, Dalla Rosa C (1988) Comparison between three mathematical models Of Kt/V. Int J Artif Organs 11(2):107–110
27. Camps-Valls G, Porta-Oltra B, Soria-Olivas E, Martin-Guerrero JD, Serrano-Lopez AJ, Perez-Ruixo JJ, Jimenez-Torres NV (2003) Prediction of cyclosporine dosage in patients after kidney transplantation using neural networks. IEEE Trans Biomed Eng 50(4):442–448
28. Collins AJ, Ma JZ, Umen A, Keshaviah P (1994) Urea index and other predictors of hemodialysis patient survival. Am J Kidney Dis 23(2):272–282
29. Cross SS, Harrison RF, Kennedy RL (1995) Introduction to neural networks. Lancet 346(8982):1075–1079
30. Daugirdas JT, Leypoldt JK, Akonur A et al. (2012) Improved equation for estimating single-pool Kt/V at higher dialysis frequencies. Nephrol Dial Transplant (in print). doi:10.1093/ndt/gfs115

31. Daugirdas JT (2011) Is there a minimal amount of time a patient should be dialyzed regardless of measured KT/V? Semin Dial 24(4):423–425
32. Daugirdas JT, Greene T, Depner TA, Levin NW, Chertow GM (2011) Modeled urea distribution volume and mortality in the HEMO study. Clin J Am Soc Nephrol 6(5):1129–1138
33. Daugirdas JT, Greene T, Chertow GM, Depner TA (2010) Can rescaling dose of dialysis to body surface area in the HEMO study explain the different responses to dose in women versus men? Clin J Am Soc Nephrol 5(9):1628–1636
34. Daugirdas JT, Depner TA, Greene T et al (2010) Standard Kt/Vurea: a method of calculation that includes effects of fluid removal and residual kidney clearance. Kidney Int 77(7):637–644
35. Daugirdas JT (1989) The post: pre dialysis plasma urea nitrogen ratio to estimate Kt/V and nPCR: validation. Int J Artif Organs 12(7):420–427
36. Daugirdas JT (1993) Second generation logarithmic estimates of single-pool variable volume Kt/V: an analysis of error. J Am Soc Nephrol 4(5):1205–1213
37. Daugirdas JT, Schneditz D, Leehey DJ (1996) Effect of access recirculation on the modeled urea distribution volume. Am J Kidney Dis 27(4):512–518
38. Daugirdas JT, Greene T, Depner TA, Leypoldt J, Gotch F, Schulman G, Star R (2004) Factors that affect postdialysis rebound in serum urea concentration, including the rate of dialysis: results from the HEMO Study. J Am Soc Nephrol 15(1):194–203
39. Dennis JE, Schnabel RB (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall, Englewood Cliffs
40. Depner TA (1990) Prescribing hemodialysis: a guide to urea modeling, 2nd edn. Springer, Berlin
41. Depner TA (1994) Assessing adequacy of hemodialysis urea modeling. Kidney Int 45(5):1522–1535
42. Depner TA (1996) Quantifying hemodialysis. Am J Nephrol 16(1):17–28
43. Depner TA (1999) History of dialysis quantitation. Semin Dial 12(1):S14–S19
44. Depner TA (2005) Hemodialysis adequacy: basic essentials and practical points for the nephrologist in training. Hemodial Int 9(3):241–254
45. DeVore RA, Howard R, Micchelli CA (1989) Optimal nonlinear approximation. Manuscripta Math 63(4):469–478
46. Fernandez de Canete J, Del Saz Huang P (2010) First-principles modeling of fluid and solute exchange in the human during normal and hemodialysis conditions. Comput Biol Med 40(9):740–750
47. Fernandez EA, Valtuille R, Willshaw P, Perazzo CA (2001) Using artificial intelligence to predict the equilibrated post-dialysis blood urea concentration. Blood Purif 19(3):271–285
48. Fernandez EA, Valtuille R, Willshaw P, Perazzo CA (2003) Dialysate-side urea kinetics. Neural network predicts dialysis dose during dialysis. Med Biol Eng Comput 41(4):392–396
49. Fernandez EA, Valtuille R, Presedo J, Willshaw P (2005) Comparison of different methods for hemodialysis evaluation by means of ROC curves: from artificial intelligence to current methods. Clin Nephrol 64(3):205–213
50. Fernandez EA, Valtuille R, Rodriguez Presedo J, Willshaw P (2005) Comparison of standard and artificial neural network estimators of hemodialysis adequacy. Artif Organs 29(2):159–165
51. Fletcher R, Reeves CM (1964) Function minimization by conjugate gradients. Comput J 7(2):149–154
52. Fletcher R (2000) Practical methods of optimization, 2nd edn. Wiley, New York
53. Furness PN, Kazi J, Levesley J, Taub N, Nicholson M (1999) A neural network approach to the diagnosis of early acute allograft rejection. Transplant Proc 31(8):3151
54. Furness PN, Levesley J, Luo Z, Taub N, Kazi JI, Bates WD, Nicholson ML (1999) A neural network approach to the biopsy diagnosis of early acute renal transplant rejection. Histopathology 35(5):461–467
55. Gabutti L, Ferrari N, Mombelli G, Marone C (2004) Does cystatin C improve the precision of Cockcroft and Gault's creatinine clearance estimation? J Nephrol 17(5):673–678
56. Gabutti L, Burnier M, Mombelli G, Malé F, Pellegrini L, Marone C (2004) Usefulness of artificial neural networks to predict follow-up dietary protein intake in hemodialysis patients. Kidney Int 66(1):399–407
57. Gabutti L, Vadilonga D, Mombelli G, Burnier M, Marone C (2004) Artificial neural networks improve the prediction of Kt/V, follow-up dietary protein intake and hypotension risk in haemodialysis patients. Nephrol Dial Transplant 19(5):1204–1211
58. Gabutti L, Machacek M, Marone C, Ferrari P (2005) Predicting intradialytic hypotension from experience, statistical models and artificial neural networks. J Nephrol 18(4):409–416
59. Garred LJ, Canaud B, Bosc JY, Tetta C (1997) Urea rebound and delivered Kt/V determination with a continuous urea sensor. Nephrol Dial Transplant 12(3):535–542
60. Geddes CC, Fox JG, Allison ME, Boulton-Jones JM, Simpson K (1998) An artificial neural network can select patients at high risk of developing progressive IgA nephropathy more accurately than experienced nephrologists. Nephrol Dial Transplant 13(1):67–71
61. Gill PE, Murray W, Wright MH (1982) Practical optimization. Emerald Group Publishing Limited, UK
62. Goldfarb-Rumyantzev A, Schwenk MH, Liu S, Charytan C, Spinowitz BS (2003) Prediction of single-pool Kt/v based on clinical and hemodialysis variables using multilinear regression, tree-based modeling, and artificial neural networks. Artif Organs 27(6):544–554
63. Gotch FA (1990) Kinetic modeling in hemodialysis. In: Nissenson AR, Fine RN, Gentile DE (eds) Clinical dialysis, 2nd edn. Appleton & Lange, Norwalk, pp 118–146
64. Gotch FA (2001) Evolution of the single-pool urea kinetic model. Semin Dial 14(4):252–256
65. Gotch FA, Sargent JA (1985) A mechanistic analysis of the national cooperative dialysis study. Kidney Int 28(3):526–538
66. Grzegorzewska AE, Banachowicz W (2008) Evaluation of hemodialysis adequacy using online Kt/V and single-pool variable-volume urea Kt/V. Int Urol Nephrol 40(3):771–778
67. Guh JY, Yang CY, Yang JM, Chen LM, Lai YH (1998) Prediction of equilibrated postdialysis BUN by an artificial neural network in high-efficiency hemodialysis. Am J Kidney Dis 31(4):638–646
68. Gurney K, Wright MJ (1997) An introduction to neural networks. UCL Press (Taylor & Francis group), London
69. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Leran Res 3:1157–1182
70. Hakim RM, Breyer J, Ismail N, Schulman G (1994) Effects of dose of dialysis on morbidity and mortality. Am J Kidney Dis 23(5):661–669
71. Hagan MT, Demuth HB, Beale MH (2002) Neural network design. PWS Publishing, Boston
72. Hagan MT, Menhaj M (1994) Training feed-forward networks with the Marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993
73. Haykin S (1999) Neural networks, 2nd edn. Prentice Hall, New Jersey

74. Held PJ, Port FK, Wolfe RA, Stannard DC, Carroll CE, Daugirdas JT, Bloembergen WE, Greer JW, Hakim RM (1996) The dose of haemodialysis and patient mortality. Kidney Int 50(2):550–556
75. Henning MR (2007) Affecting Kt/V: an analysis of staff interventions. Dial Transplant 36(11):584–601
76. Hestenes M (1980) Conjugate direction methods in optimization. Springer, New York
77. Heston TF, Norman DJ, Barry JM, Bennett WM, Wilson RA (1997) Cardiac risk stratification in renal transplantation using a form of artificial intelligence. Am J Cardiol 79(4):415–417
78. Ijely GK, Raja RM (1991) Simplified calculation of PCR and Kt/V. Abstract 24th Annual JASN Meeting, p 329
79. Jean G, Chazot C, Charra B, Terrat JC, Vanel T, Calemard E, Laurent G (1998) Is post-dialysis urea rebound significant with long slow hemodialysis? Blood Purif 16(4):187–196
80. Jean G, Charra B, Chazot C, Laurent G (1999) Quest for post-dialysis urea rebound-equilibrated Kt/V with only intradialytic urea samples. Kidney Int 56(3):1149–1153
81. Jindal KK, Manuel A, Goldstein MB (1987) Percent reduction in blood urea concentration during hemodialysis (PRU). A simple and accurate method to estimate Kt/V urea. ASAIO Trans 33(3): 286–288
82. Johansson EM, Dowla FU, Goodman DM (1990) Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method. Int J Neural Syst (IJNS) 2(4):291–301
83. Jordan MI (1995) Why the logistic function? A tutorial discussion on probabilities and neural networks. MIT computational cognitive science report 9503. http://www.cs.berkeley.edu/*jordan
84. Kazi JI, Furness PN, Nicholson M, Ahmed E, Akhter F, Naqvi A, Rizvi A (1999) Interinstitutional variation in the performance of Baysian belief network for the diagnosis of acute renal graft rejection. Transplant Proc 31(8):3152
85. Kerr PG, Argiles A, Canaud B, Flavier JL, Mion CM (1993) Accuracy of Kt/V estimations in high-flux hemodiafiltration using percent reduction of urea: incorporation of urea rebound. Nephrol Dial Transplant 8(2):140–153
86. Keshaviah PR, Hanson GI, Berkseth RO, Collins AJ (1988) A simplified approach to monitoring in vivo therapy prescription. Trans Am Soc Artif Organs 34(3):620–622
87. Lai YH, Guh JY, Chen HC, Tsai JH (1995) Effects of different sampling methods for measurement of post dialysis blood urea nitrogen on urea kinetic modeling derived parameters in patients undergoing long-term hemodialysis. ASAIO J 41(2):211–215
88. Leblanc M, Charbonneau R, Lalumiere G, Cartier P, Deziel C (1996) Postdialysis urea rebound: determinants and influence on dialysis delivery in chronic hemodialysis patients. Am J Kidney Dis 27(2):253–261
89. Lawrence S, Giles CL, Tsoi AC (1997) Lessons in neural network training: overfitting may be harder than expected. In: Proceedings of the fourteenth national conference on artificial intelligence, AAAI-97. AAAI Press, Menlo Park, pp 540–545
90. Lowrie EG, Laird NM, Parker TF et al (1981) Effect of the hemodialysis prescription on patient morbidity: report from the national cooperative dialysis study. N Engl J Med 305(20): 1176–1181
91. Lowrie EG, Teehan BP (1983) Principles of prescribing dialysis therapy: implementing recommendations from the national co-operative dialysis study. Kidney Int 23(Suppl 13):S113–S122
92. Lowrie EG, Zhu X, Lew NL (1998) Primary associates or mortality among dialysis patients: trends and reassessment of Kt/V and urea reduction ratio as outcome-based measures of dialysis dose. Am J Kidney Dis 32(Suppl 4):S16–S31
93. Malmgren H, Borga M, Niklasson L (2000) Artificial neural networks in medicine and biology, perspectives in neural computing. Springer, Goteborg
94. Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. SIAM J Appl Math 11(2):431–441
95. MedCalc (2012) Medical calculator statistical software. http://www.medcalc.be. Accessed on Jan 2012
96. Moller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6:525–533
97. Montano JJ, Palmer A (2003) Numeric sensitivity analysis applied to feedforward neural networks. Neural Comput Appl 2(2):119–125
98. Mitchell T (1997) Machine learning. The McGraw-Hill Companies, Inc., New York
99. NCSS (2012) Statistical and power analysis software. http://www.ncss.com. Accessed on Jan 2012
100. NKF-K/DOQI (1997) Clinical practice guidelines for hemodialysis adequacy. Am J Kidney Dis 30(Suppl 2):S15–S57
101. NKF-K/DOQI (2001) Clinical practice guidelines for hemodialysis adequacy: update. Am J Kidney Dis 37(Suppl 1):S7–S64
102. NKF-K/DOQI (2006) Clinical practice guidelines for hemodialysis adequacy, update, 2006. Hemodialysis adequacy 2006 work group. Am J Kidney Dis 48(Suppl 1):S2–S90
103. Picard RR, Cook RD (1984) Cross-validation of regression models. J Am Stat Assoc 79(387):575–583
104. Penny W, Frost D (1996) Neural networks in clinical medicine. Med Decis Mak 16(4):386–398
105. Powell M (1977) Restart procedures for the conjugate gradient method. Math Program 12(1):241–254
106. Prado M, Roa L, Palma A, Milán JA (2004) A novel mathematical method based on urea kinetic modeling for computing the dialysis dose. Comput Methods Programs Biomed 74(2):109–128
107. Reggia JA (1993) Neural computation in medicine. Artif Intell Med 5(2):143–157
108. Renal Physicians Association (1993) Clinical practice guideline: adequacy of hemodialyses. Renal Physicians Association, Washington
109. Riedmiller M, Braun H (1993) A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the IEEE international conference on neural networks, vol 1, pp 586–591, San Francisco, CA. doi:10.1109/ICNN.1993.298623
110. Robert CP, Casella G (2004) Monte Carlo statistical methods, 2nd edn. Springer, New York. ISBN 0-387-21239-6
111. Rumelhart DE, McClelland JL (1986) Parallel distributed processing: exploration in the microstructure of cognition. MIT Press, Cambridge
112. Sargent JA, Gotch FA (1980) Mathematic modeling of dialysis therapy. Kidney Int Suppl 10:S2–S10
113. Sargent J (1983) Control of dialysis by a single-pool urea model: the national cooperative dialysis study. Kidney Int 23(suppl 13):S19–S26
114. Sarle WS (1995) Stopped training and other remedies for overfitting. In: Proceedings of the 27th symposium on the interface of computing science and statistics, pp 352–360
115. Sarle WS (1997) Neural network FAQ. Periodic posting to the Usenet newsgroup comp.ai.neural-nets. ftp://ftp.sas.com/pub/neural/FAQ.html
116. Schalkoff RJ (1997) Artificial neural networks. McGraw-Hill International Editions
117. Schneditz D, Daugirdas JT (2001) Compartment effects in hemodialysis. Semin Dial 14(4):271–277
118. Schneditz D, VanStone J, Daugirdas JT (1993) A regional blood circulation alternative to in-series two compartment urea kinetic modeling. ASAIO J 39(3):M573–M577
119. Schneditz D, Fariyike B, Osheroff R, Levin NW (1995) Is intercompartmental urea clearance during hemodialysis a perfusion term? A comparison of two pool urea kinetic models. J Am Soc Nephrol 6(5):1360–1370

120. Sheppard D, McPhee D, Darke C, Shrethra B, Moore R, Jurewitz A, Gray A (1999) Predicting cytomegalovirus disease after renal transplantation: an artificial neural network approach. Int J Med Inform 54(1):55–76

121. Sherrod PH (2012) DTREG predictive modeling software. www.dtreg.com. Accessed on Jan 2012

122. Shoskes DA, Ty R, Barba L, Sender M (1998) Prediction of early graft function in renal transplantation using a computer neural network. Transplant Proc 30(4):1316–1317

123. Simic-Ogrizovic S, Furuncic D, Lezaic V, Radivojevic D, Blagojevic R, Djukanovic L (1999) Using ANN in selection of the most important variables in prediction of chronic renal allograft rejection progression. Transplant Proc 31(1–2):368

124. Smith BP, Ward RA, Brier ME (1998) Prediction of anticoagulation during hemodialysis by population kinetics and an artificial neural network. Artif Organs 22(9):731–739

125. Smye SW, Hydon PE, Will E (1993) An analysis of the single-pool urea kinetic model and estimation of errors. Phys Med Biol 38(1):115–122

126. Spiegel DM, Baker PL, Babcock S, Cantiguglia R, Klein M (1995) Hemodialysis urea rebound: the effect of increasing dialysis efficiency. Am J Kidney Dis 25(1):26–29

127. Swingler K (1996) Applying neural networks: a practical guide. Academic Press, London

128. Tangri N, Ansell D, Naimark D (2008) Predicting technique survival in peritoneal dialysis patients: comparing artificial neural networks and logistic regression. Nephrol Dial Transplant 23(9):2972–2981

129. Tetko IV, Livingstone DJ, Luik AI (1995) Neural network studies 1, comparison of overfitting and overtraining. J Chem Info Comp Sci 35(5):826–833

130. Trujillano J, March J, Sorribas A (2004) Methodological approach to the use of artificial neural networks for predicting results in medicine. Med Clin (Barc) 122(Suppl 1):59–67

131. Tsoukalas LH, Uhrig RE (1997) Fuzzy and neural approaches in engineering. Wiley, NY

132. Tu JV (1996) Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol 49(11):1225–1231

133. Vapnik VN (1998) Statistical learning theory. Wiley-Interscience, USA

134. Veng-Pedersen P, Modi NB (1992) Neural networks in pharmacodynamic modeling. Is current modelling practice of complex kinetic systems at a dead end? J Pharmacokinet Biopharm 20(4):397–412

135. Veng-Pedersen P, Modi NB (1993) Application of neural networks to pharmacodynamics. J Pharm Sci 82(9):918–926

136. Weigend A (1994) On overfitting and the effective number of hidden units. In: Proceedings of the 1993 connectionist models summer school, pp 335–342

137. Wolf AV, Remp DG, Kiley JE, Currie GD (1951) Artificial kidney function; kinetics of hemodialysis. J Clin Invest 30(10):1062–1070

138. Xu QS, Liang YZ (2001) Monte Carlo cross validation. Chemom Intell Lab Syst 56(1):1–11

139. Yashiro M, Watanabe H, Muso E (2004) Simulation of post-dialysis urea rebound using regional flow model. Clin Exp Nephrol 8(2):139–145

140. Zweiri YH, Whidborne JF, Seviratne LD (2002) A three-term backpropagation algorithm. Neurocomputing 50:305–318