

Non-linear Estimation Methods for Hematocrit Density based on Changing Pattern of Transduced Anodic Current Curve

HIEU TRUNG HUYNH¹, JUNG-JA KIM² and YONGGWAN WON¹

¹Department of Computer Engineering
Chonnam National University
300 Yongbong-dong, Buk-gu, Gwangju 500-757
REPUBLIC OF KOREA

hthieu@hcmut.edu.vn, ykwon@chonnam.ac.kr

²Division of Bionics and Bioinformatics
Chonbuk National University
664-14 St. #1 Dukjin-dong, Dukjin-gu, Chonbuk 561-756
REPUBLIC OF KOREA
jungjakim@chonbuk.ac.kr

Abstract: - Many studies reported that the hematocrit (HCT) is the most highly influencing factor affecting the accuracy of the glucose measurements by portable/handheld devices. It is also known as an important factor for clinical decision-making situations. Therefore, estimation of HCT plays a crucial role for enhancing accuracy of glucose measurements and performance of therapy. In this paper, we present novel methods for hematocrit estimation from the transduced current curve which is produced by glucose-oxidase reaction in strip-type electrochemical biosensors. The proposed methods are nonlinear, including neural networks and support vector machine. Input features are composed of two parts: the sampled points of the time-varying current curve and extended extra features computed from those sampled points.

Key-Words: - Hematocrit, hematocrit estimation, nonlinear methods, biosensors, transduced current curve

1 Introduction

Hematocrit is expressed as the proportion of blood volume that is occupied by red blood cells (RBCs). It is one of the primary characteristics in the whole blood and a useful clinical indicator in surgical and hemodialysis. A low hematocrit is referred to *anemic* which reduces the capacity of blood to carry oxygen and high hematocrit can be *polycythaemia* which may be a warning signal for serious circulatory failure. In addition, many studies reported that hematocrit is a factor that significantly affects the accuracy of glucose measurements by handheld devices [1,2,3]. The glucose results are underestimated at higher hematocrit levels, while overestimated at lower hematocrit levels. Therefore, estimating this factor is one of the most important steps for improving performance of glucose measurements by handheld devices.

Estimation of hematocrit can be done manually by centrifugation method, in which a capillary tube called micro-hematocrit tube is filled with blood. After being centrifuged at 10,000RPM for five minutes, the blood in the tube is separated into layers. The top layer is liquid plasma, the white

blood cells (WBCs) and platelets form a thin layer between the plasma and the RBCs that is the buffy coat, and the bottom of the tube is the red blood cells (RBCs) with the greatest weight. The hematocrit is measured as the percent of the RBC column to the total blood column.

Another approach for hematocrit measurement applied in modern lab equipment is using automated analyzer which can make several other measurements at the same time, and the hematocrit is indirectly calculated by multiplying the red cell count by the mean cell volume. Some reports also indicated that hematocrit can be determined from *impedance*, in which Hanai's model to blood is used with assumption of red blood cells to be non-conducting. This method requires an initial hematocrit measurement by a classical method. It can be used only for continuous hematocrit monitoring. In addition, the *dielectric spectroscopy* which is called as *impedance spectroscopy* is also applied to estimate the hematocrit [4,5]. Hematocrit is estimated based on the interaction of an external field with the electric dipole moment of sample which is often expressed as permittivity. A nonlinear model for hematocrit estimation from the

permittivity β -dispersion change was proposed by Foster and Schwan [5]. However, the best result on correlation was obtained from a linear model proposed by Treo [4]. All of the above methods for hematocrit estimation are quite complicated or require individual devices which can not be used to reduce the effects of hematocrit in glucose measurement by handheld devices.

In our studies, we investigated approaches for estimating hematocrit density using electrochemical glucose biosensors which is originally designed for glucose measurement with handheld devices. These biosensors use an enzyme to break the blood glucose down and produce ions. These ions are transferred to an electrode to produce a current which is called the transduced current. Since the ionization activity is varying along time, the current curve is presented as time-varying curve. We tried to estimate the HCT density from this calibration curve, based on the assumption that the changing pattern of the current curve includes all information about the characteristics of the blood samples, including the hematocrit density. In this paper, applying nonlinear methods for hematocrit estimation not only from current points sampled from the transduced current curve but also from extra features of current curve which can offer better performance.

The rest of this paper is organized as follows. Section 2 describes glucose measurements by electrochemical biosensors and the transduced current curve. A method for extract extra features is shown in section 3. Nonlinear methods for hematocrit estimation are shown in section 4. In section 5, we present experimental results on root mean squares error (RMSE). Finally, we make conclusion in section 6.

2 Transduced Anodic Current Curve

Glucose oxidase has been used widely in several different industries ranging from a glucose biosensor for diabetes monitoring to food preservative. In industry of electrochemical glucose biosensor, the glucose oxidase (GOD) enzyme is used to catalyze the oxidation of glucose by oxygen to produce gluconic acid and hydrogen peroxide:

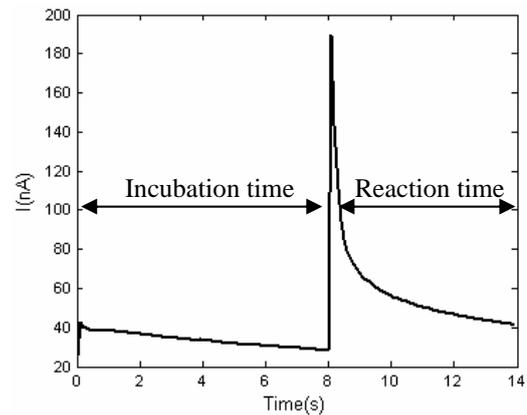
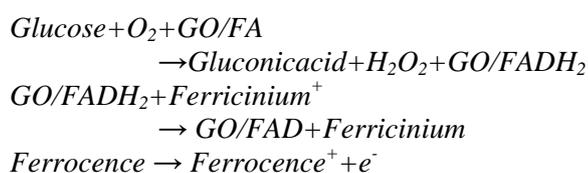


Figure 1. An example of the transduced current curve

The reduced form of the enzyme (GO/FADH₂) is oxidized to its original state by an electron mediator (ferrocence). The resulting reduced mediator is then oxidized by the active electrode to produce the transduced anodic current (e⁻). An instance of the transduced anodic current curve obtained through time for 14 seconds using a biochemical glucose biosensor is shown in Fig. 1. In this curve, the first 8 seconds can be called as *incubation* time which is time for waiting chemical reaction producing electric signal with high-enough level. We only concern the second part of the current curve during the last six seconds. During this time, the anodic current curve was sampled at the frequency of 10Hz to produce current points. There are 59 current points sampled from the second part of current curve considered as the *input pattern vector* for hematocrit estimation.

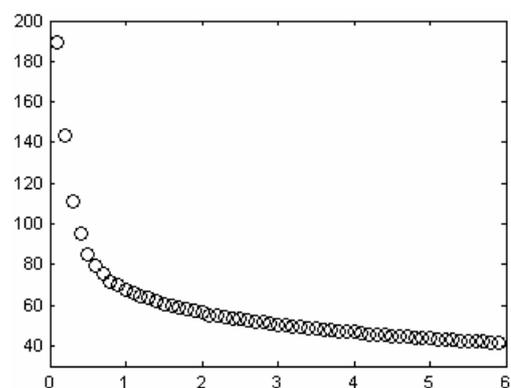


Figure 2. Transduced anodic current points used in estimation of hematocrit

3 Extra features from transduced current curve

It appears that the current points sampled from the transduced current curve may be an exponential function of time, as depicted in Fig. 3. Hence, a

reasonable model of these current points can be formulated by

$$x_n = ap_n^b, n=1, 2, \dots, d. \quad (1)$$

where a, b are the parameters of approximated model, d is the number of sampled points, and x_n is values of current response at time point p_n .

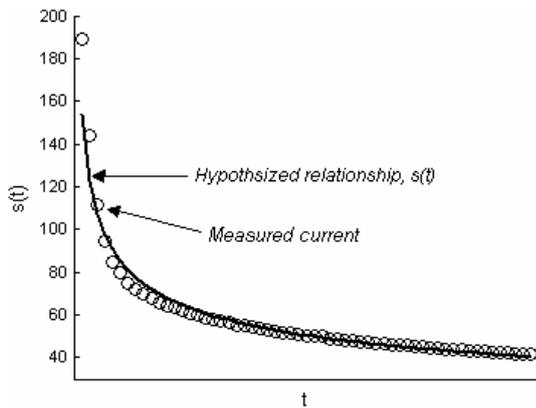


Figure 3. Current sampled points used in estimation of hematocrit

It is obvious that for this model we have to estimate the parameters a and b . We can change it to a linear model which can be easily handled. Taking logarithm on both sides of Eq. (1), we have

$$\log(x_n) = \log(a) + b \log(p_n), n=1, \dots, d. \quad (2)$$

In the matrix form, we have

$$\mathbf{A} = \mathbf{L}\mathbf{G}, \quad (3)$$

where

$$\mathbf{A} = [\log(x_1), \log(x_2), \dots, \log(x_d)]^T,$$

$$\mathbf{G} = [\log(a) \ b]^T$$

and

$$\mathbf{L} = \begin{bmatrix} 1 & \log(p_1) \\ 1 & \log(p_2) \\ \vdots & \vdots \\ 1 & \log(p_d) \end{bmatrix}$$

The least mean square solution for $\mathbf{G} = [\theta_1 \ \theta_2]^T$ from the linear model (3) is

$$\hat{\mathbf{G}} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T \mathbf{A}. \quad (4)$$

Let $\mathbf{U} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$, then the Eq. (4) can be rewritten by

$$\hat{\mathbf{G}} = \mathbf{U}\mathbf{A}. \quad (5)$$

Finally, the estimator for a and b is $a = \exp(\theta_1)$, $b = \theta_2$. Thus, from the sampled points of the current curve,

it becomes simple to estimate parameters of the approximated model. These extra parameters together with the sampled points are used as the input features for the nonlinear methods to estimate the hematocrit as described in the following section.

Sampled current points

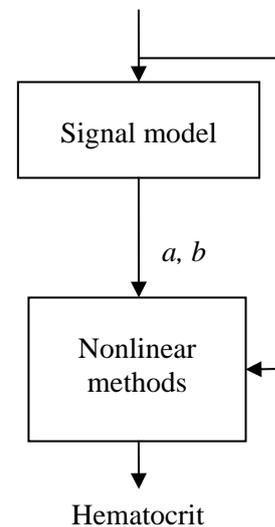


Figure 4. Hematocrit estimation using transduced current curve and its extra features

4 Nonlinear methods for Hematocrit Estimation

4.1 Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised learning method used for classification and regression. It is based on the framework of statistical learning theory which has been developed by Vapnik et. al [6]. It has been gaining popularity due to many attractive features and promising empirical performance. It can be also used widely in medical diagnosis [7]. In SVM, the loss functions must be introduced to measure distance. Four possible loss functions can be used that are quadratic, Laplace, Huber and ϵ -insensitive. The quadratic loss function corresponding to the conventional least-square error criterion allows easy estimation of parameters. However, it is very sensitive to outliers. Two loss functions that are less sensitive to outliers are quadratic and Laplacian ones. However, these loss functions do not produce sparseness in the support vectors. Vapnik took into account this issue by introducing ϵ -insensitive loss function which is an approximation to Huber's loss function but allows obtaining a

sparse set of support vectors. The ε -insensitive loss function is defined by

$$L_\varepsilon(t) = \begin{cases} 0 & \text{for } |g(\mathbf{x}, \mathbf{w}) - t| < \varepsilon \\ |g(\mathbf{x}, \mathbf{w}) - t| - \varepsilon & \text{otherwise} \end{cases} \quad (6)$$

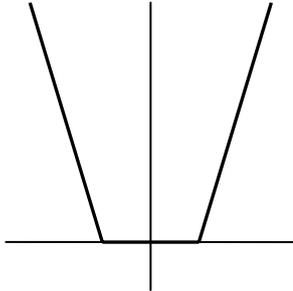


Figure 5. ε -insensitive loss function

In function approximation, a SVM using ε -insensitive loss function can be called as ε -support vector regression or ε -SVR.

Given $(\mathbf{x}_j, t_j), j=1, 2, \dots, N$ be training patterns, ε -SVR tries to find parameters so that the approximated function $g(\mathbf{x}, \mathbf{w})$ has most ε deviation from the actually obtained targets t_j for all training data, and as flat as possible. It does not care about errors as long as they are less than ε , but will not accept any deviation larger than this. In the cases where $g(\mathbf{x}, \mathbf{w})$ is a linear function, the approximation function is given by

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b, \quad (7)$$

and the problem for finding \mathbf{w} in ε -SV regression can be written formally as:

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad (8)$$

$$\text{subject to } \begin{cases} t_j - \langle \mathbf{w}, \mathbf{x}_j \rangle - b \leq \varepsilon \\ \langle \mathbf{w}, \mathbf{x}_j \rangle + b - t_j \leq \varepsilon \end{cases}$$

Sometimes, the actual function g can not approximate all pairs (\mathbf{x}_j, t_j) , or we should allow for some errors. A method based on “soft margin” was proposed [6]. One can introduce slack variables ξ_j, ξ_j^* to cope with otherwise infeasible constraints of the optimization which is described in the equation (8). Therefore, we now arrive at a problem stated as:

$$\text{minimization: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{j=1}^N (\xi_j + \xi_j^*) \quad (9)$$

$$\text{subject to } \begin{cases} t_j - \langle \mathbf{w}, \mathbf{x}_j \rangle - b \leq \varepsilon + \xi_j \\ \langle \mathbf{w}, \mathbf{x}_j \rangle + b - t_j \leq \varepsilon + \xi_j^* \\ \xi_j, \xi_j^* \geq 0 \end{cases}$$

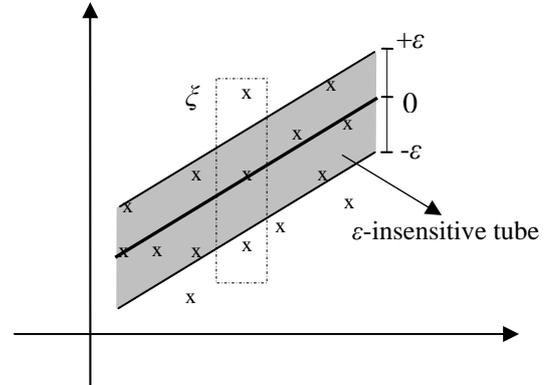


Figure 6. Soft margin loss setting corresponds for a linear SV machine.

where $C > 0$ is a constant. This problem can be converted to the dual optimization problem that maximizes

$$\begin{cases} -\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ -\varepsilon \sum_{j=1}^N (\alpha_j + \alpha_j^*) + \sum_{j=1}^N t_j (\alpha_j - \alpha_j^*) \end{cases}$$

$$\text{subject to } \begin{cases} \sum_j (\alpha_j - \alpha_j^*) = 0 \\ 0 \leq \alpha_j, \alpha_j^* \leq C \end{cases}$$

where $\alpha_j, \alpha_j^*, \eta_j, \eta_j^* \geq 0$ are dual variables. The solution for \mathbf{w} is given by

$$\mathbf{w} = \sum_{j=1}^N (\alpha_j - \alpha_j^*) \mathbf{x}_j \quad (10)$$

and b can be computed by:

$$\begin{aligned} b &= t_j - \langle \mathbf{w}, \mathbf{x}_j \rangle - \varepsilon \quad \text{for } \alpha_j \in (0, C) \\ b &= t_j - \langle \mathbf{w}, \mathbf{x}_j \rangle + \varepsilon \quad \text{for } \alpha_j^* \in (0, C) \end{aligned} \quad (11)$$

In the cases which are not possible to have a linear function on the training data, a nonlinear mapping can be applied in order to map the data into other feature space where the linear model can be used. In addition, we can use an approach via kernels defined by:

$$K(\mathbf{x}_1, \mathbf{x}_2) := \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle. \quad (12)$$

The solution for \mathbf{w} in this case is given by

$$\mathbf{w} = \sum_{j=1}^N (\alpha_j - \alpha_j^*) \Phi(\mathbf{x}_j) \quad (13)$$

Another approach for SVM in regression proposed by Schölkopf *et al.* [8] is called ν -SVR which incorporates a change with ν . The optimization problem is given by minimization of

$$\frac{1}{2} \|\mathbf{w}\|^2 - \varepsilon \nu + \frac{1}{N} \sum_{j=1}^N \xi_j \quad (14)$$

$$\text{subject to } \begin{cases} t_j - \langle \mathbf{w}, \mathbf{x}_j \rangle - b \leq \varepsilon + \xi_j \\ \langle \mathbf{w}, \mathbf{x}_j \rangle + b - t_j \leq \varepsilon + \xi_j^* \\ \xi_j, \xi_j^*, \varepsilon \geq 0 \end{cases}$$

Thus, SVM finds a globally optimal solution. It uses structural risk minimization and can generalize well to unseen data. The computational complexity of SVMs does not depend on the dimensionality of the input space.

4.2 ELM algorithm

Neural network is one of the nonlinear methods widely used in learning machine and function approximation [9,10,11]. It can efficiently resolve problems which are difficult to handle by parametric methods. A feedforward neural network may consist of one input layer, one output layer and one or multiple hidden layers. However, in function approximation, it was shown that a single hidden layer feedforward neural network (SLFN) can approximate any function with arbitrary small error if activation function is chosen properly.

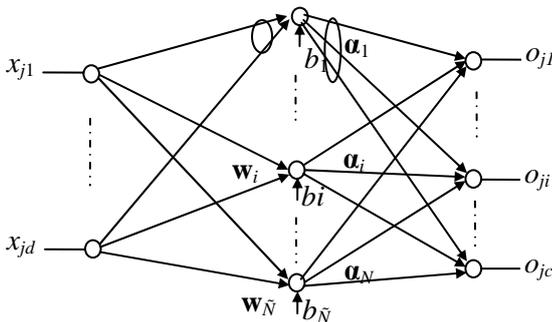


Figure 7. The architecture of the SLFN

A typical architecture of SLFN with \tilde{N} hidden units and C output units is given in Fig. 7. If the hidden units adopt the activation function $f(\cdot)$ then the SLFN is mathematically modeled by:

$$\mathbf{o} = \sum_{i=1}^{\tilde{N}} \alpha_i f(\mathbf{w}_i \cdot \mathbf{x} + b_i), \mathbf{x} \in \mathbb{R}^d \quad (15)$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{id}]^T$ is the weight vector connecting from the input units to the i -th hidden unit, b_i is the bias of the i -th hidden unit, and $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iC}]^T$ is the weight vector connecting from the i -th hidden unit to the output units. Note that $\mathbf{w}_i \cdot \mathbf{x} = \langle \mathbf{w}_i, \mathbf{x} \rangle$ is the inner product of \mathbf{w}_i and \mathbf{x} .

For a training set consisting of pairs of vectors $(\mathbf{x}_j, \mathbf{t}_j)$, $j=1, 2, \dots, N$, where \mathbf{x}_j and \mathbf{t}_j are the j -th input pattern and its target, respectively. The main goal of training process is to adjust the network weights \mathbf{w}_i , α_i , and b_i so that they minimize the error function defined by:

$$E = \sum_{j=1}^N (\mathbf{o}_j - \mathbf{t}_j)^2 = \sum_{j=1}^N \left(\sum_{i=1}^{\tilde{N}} \alpha_i f(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right)^2 \quad (16)$$

Traditionally, this minimization process can be done by gradient-descent based algorithms, in which parameter vector \mathbf{W} consisting of weights (\mathbf{w}_i, α_i) and biases b_i is iteratively adjusted as follows

$$\mathbf{W} = \mathbf{W} - \eta \frac{\partial E}{\partial \mathbf{W}} \quad (17)$$

where η is a learning rate determining the speed at which the network obtains a minimum in the criterion function $E(\mathbf{w})$. In the feedforward neural networks, a popular learning algorithm based on gradient descent is back-propagation (BP) learning algorithm that gradients can be calculated and the vectors \mathbf{W} can be determined by error propagation from the output layer to the input layer.

However, this algorithm often meets problems such as:

- It can converge very slowly if the learning rate is too small. However, for a large learning rate, it can be unstable or divergent.
- It can stop at local minima instead of global minima.
- It may be over-trained and obtain worse generalization performance.

These problems have been overcome by many improvements proposed by many researchers [12,13,14]. However, up to now, most of the training algorithms based on gradient descent are still slow due to many learning steps which may be required in the learning process.

Recently, Huang et al. [15] proposed an effective training algorithm for SLFNs that is extreme learning machine (ELM). This algorithm can overcome problems in gradient-descent based algorithms.

Given $f(\cdot)$ be the activation function of hidden units. ELM tries to find the network parameters that minimize error of equation defined by

$$\mathbf{H}\mathbf{A}=\mathbf{T}, \quad (18)$$

where \mathbf{A} is output weight matrix, $\mathbf{T}=[\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_N]^T$ is the desired target vector and \mathbf{H} is hidden layer output matrix defined by

$$\mathbf{H}=\begin{bmatrix} f(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & f(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & f(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}. \quad (19)$$

A salient feature of ELM algorithm is that the biases and input weights are randomly assigned, output weights are determined by pseudo-inverse operation of hidden layer output matrix as follows:

$$\hat{\mathbf{A}}=\mathbf{H}^\dagger\mathbf{T}, \quad (20)$$

where \mathbf{H}^\dagger is Moore-Penrose (MP) generalized inverse of \mathbf{H} . This is the minimum norm least-squares solution of (18). In summary, the ELM algorithm can be described as following:

Algorithm ELM: Given a training set $S=\{(\mathbf{x}_j, \mathbf{t}_j) \mid j=1, \dots, N\}$, activation function $f(\mathbf{x})$, and number of hidden node \tilde{N} .

- Randomly assign the input weights \mathbf{w}_m and biases $b_m, m=1, 2, \dots, \tilde{N}$.
- Determine the output matrix \mathbf{H} of the hidden layer by using Eq. (19).
- Determine the output weight matrix \mathbf{A} by using Eq. (20).

Thus, this algorithm can determine the network parameters by non-iterative steps. It can offer good performance at high learning speed in many applications.

When the whole training set is not available, a development of ELM called online sequential extreme learning machine (OS-ELM) was proposed by N.Y. Liang et al. [16]. It is an online sequential learning algorithm for SLFNs based on the ELM and can learn one-by-one or block-by-block of data. In OS-ELM, the input weights and hidden layer biases are also randomly chosen and the output weights can be updated by arriving data.

However, because of randomly selecting, the biases and input weights of ELM and OS-ELM are non-optimal which results in a large number of

hidden units are required. Hence, a large memory is needed to save network parameters in devices, and the trained network responds slowly to new input patterns.

4.3 RLS-ELM algorithm

An improvement of ELM to obtain compact trained networks was shown in our previous study called as regularized least squares ELM (RLS-ELM) [17]. This algorithm determines network weights based on linear models. From (18), matrix \mathbf{H} can be approximated by

$$\mathbf{H}=\mathbf{T}\mathbf{A}^\dagger, \quad (21)$$

where \mathbf{A}^\dagger is MP generalized inverse of \mathbf{A} . We can express equation (21) in the form of

$$f[\mathbf{X}\mathbf{W}]=\mathbf{T}\mathbf{A}^\dagger, \quad (22)$$

where $f[\mathbf{X}\mathbf{W}]_{ij}=f([\mathbf{X}\mathbf{W}]_{ij})=f(\mathbf{w}_j \cdot \mathbf{x}_i + b_j)$, matrices \mathbf{X} and \mathbf{W} are defined by:

$$\mathbf{X}=\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \\ 1 & 1 & \dots & 1 \end{bmatrix}^T,$$

$$\mathbf{W}=\begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_{\tilde{N}} \\ b_1 & b_2 & \dots & b_{\tilde{N}} \end{bmatrix}.$$

In most cases, the activation function f is invertible. So, Equation (22) can be rewritten as follows:

$$\mathbf{X}\mathbf{W}=f^{-1}[\mathbf{T}\mathbf{A}^\dagger], \quad (23)$$

where $f^{-1}[\mathbf{T}\mathbf{A}^\dagger]_{ij}=f^{-1}([\mathbf{T}\mathbf{A}^\dagger]_{ij})$. If we define a matrix \mathbf{P} by

$$\mathbf{P}=\mathbf{T}^\dagger f^{-1}[\mathbf{T}\mathbf{A}^\dagger], \quad (24)$$

where \mathbf{T}^\dagger is the MP generalized inverse of \mathbf{T} , then Equation (23) becomes

$$\mathbf{X}\mathbf{W}=\mathbf{TP} \quad (25)$$

A method used for the regularization of ill-posed problems is Tikhonov regularization [18], in which the solution for \mathbf{W} of (25) can be found by minimizing

$$\|\mathbf{X}\mathbf{W}-\mathbf{TP}\|^2+\lambda\|\mathbf{W}\|^2, \quad (26)$$

where λ is a positive constant. The solution for (26) is given by

$$\hat{\mathbf{W}}=(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{TP}. \quad (27)$$

This is the direct solution for \mathbf{W} . An indirect solution is given by

$$\hat{\mathbf{W}}=\mathbf{X}^T\mathbf{Y}, \quad (28)$$

where

$$\mathbf{Y} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{TP}. \quad (29)$$

Eq. (24) shows the determining method of matrix \mathbf{P} . However, we do not know \mathbf{A} at beginning of training process, so the matrix \mathbf{P} should be randomly assigned and then it can be used to estimate input weights and hidden layer biases by Eq. (27) or Eq. (28). After determining these parameters, the output weights are computed by the MP generalized inverse as given by Eq. (20).

In summary, the RLS-ELM algorithm for compact SLFNs can be described as follows:

Algorithm RLS-ELM: *Given a training set $S = \{(\mathbf{x}_j, t_j) \mid j=1, \dots, N\}$, activation function $f(\mathbf{x})$, and number of hidden node \tilde{N} .*

- Randomly assign the values for the matrix \mathbf{P} .
- Estimate the input weights \mathbf{w}_m and biases b_m by using Eq. (27) or Eq. (28).
- Determine the output matrix \mathbf{H} of the hidden layer by using Eq. (19).
- Determine the output weight matrix \mathbf{A} by using Eq. (20).

Thus, the network parameters of SLFNs can be determined by the non-iterative procedures. They are simple and have low computational complexity, which results in extremely high speed for both training and testing.

In many real-world applications, the number of outputs C is much smaller than the number of inputs d . Hence, though we still assign random values for matrix \mathbf{P} , the number of randomly chosen values for matrix \mathbf{P} ($C \times \tilde{N}$) is much smaller than $(d+1) \times \tilde{N}$ that is used for the original ELM. Especially, in hematocrit estimation, the number of sampled current points is $d=59$ while $C=1$. Therefore, if we use the whole of current points as input features then RLS-ELM assigns only \tilde{N} random values for matrix \mathbf{P} which is reduced sixty times in comparison with the original ELM.

In addition, Bartlett claimed that the networks tend to produce better generalization performance if the network parameters have small norm [19]. The solution for \mathbf{W} by using (27) or (28) is found with small norm which tend to offer better generalization performance of trained SLFNs. The number of hidden units is also reduced significantly by using RLS-ELM. In hematocrit estimation with the whole current points, it is 5 while that for the original ELM is 15. Therefore, the memory needed to saving network parameters is reduced three times by using RLS-ELM, this is important in hardware implementation.

4.3 ELS-ELM algorithm

This section briefly introduces another improvement of ELM for training SLFNs with compact networks called as evolutionary least squares ELM (ELS-ELM) algorithm [20]. It is combination of ELM and differential evolution (DE) [21], in which the input weights and hidden layer biases are determined by using least-squares and three steps of DE process. Suppose that input weights and hidden layer biases form an individual in population:

$$\theta = \{ \mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_{\tilde{N}}^T, b_1, b_2, \dots, b_{\tilde{N}} \}.$$

Each individual in the initial generation is generated by [22]:

$$\hat{\mathbf{W}} = \mathbf{X}^\dagger \mathbf{TP}, \quad (26)$$

where \mathbf{X}^\dagger is MP generalized inverse of \mathbf{X} .

The output weights corresponding to each individual are computed by MP generalized inverse. Three steps of DE process are used and individuals with better fitness values are retained for the next generation. The fitness function is chosen as the root mean squares error (RMSE) on the whole training set or the validation set and defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (o_j - t_j)^2}, \quad (27)$$

where o and t are actual output and desired output corresponding to the j -th input pattern. In summary, the ELS-ELM for SLFNs can be described as follows:

a) Initialization: *Generate the initial generation being composed of parameter vectors $\{ \theta_{i,G} \mid i=1, 2, \dots, NP \}$ as the population, where NP is the population size.*

For each individual θ in the population, we do:

- i) Randomly assigning the values for the matrix \mathbf{P} .
- ii) Estimating input weights and hidden layer biases of θ by using Eq. (26).
- iii) Calculating the hidden-layer output matrix \mathbf{H} by Eq. (19).
- iv) Determining the output weights \mathbf{A} by using Eq. (20).
- v) Calculate the fitness value.

b) Training process:

At each generation G , we do:

- i) *Mutation:* the mutant vector is generated as $\mathbf{v}_{i,G+1} = \theta_{r1,G} + F(\theta_{r2,G} - \theta_{r3,G})$, where $r1, r2, r3$ are different random indices and F is a constant factor used to control the amplification of the differential variation.

ii) *Crossover*: the trial vector is formed so that

$$\mu_{ji,G+1} = \begin{cases} \mathbf{v}_{ji,G+1} & \text{if } \text{rand } b(j) \leq CR \\ & \text{or } j = \text{rnbr}(i) \\ \boldsymbol{\theta}_{ji,G+1} & \text{if } \text{rand } b(j) > CR \\ & \text{and } j \neq \text{rnbr}(i), \end{cases}$$

where $\text{rand } b(j)$ is the j -th evaluation of a uniform random number generator, CR is the crossover constant and $\text{rnbr}(i)$ is a randomly chosen index which ensures at least one parameter from $\mathbf{v}_{ji,G+1}$.

iii) Determine the output weights by Eq.(20).

iv) Evaluate the fitness for each individual.

v) *Selection*: The new generation is determined by:

$$\boldsymbol{\theta}_{i,G+1} = \begin{cases} \mu_{i,G} & \text{if } \varphi(\boldsymbol{\theta}_{i,G}) - \varphi(\mu_{i,G}) > \varepsilon\varphi(\boldsymbol{\theta}_{i,G}), \\ \mu_{i,G} & \text{if } |\varphi(\boldsymbol{\theta}_{i,G}) - \varphi(\mu_{i,G})| < \varepsilon\varphi(\boldsymbol{\theta}_{i,G}) \\ & \text{and } \|\mathbf{A}^{\mu_{i,G}}\| < \|\mathbf{A}^{\boldsymbol{\theta}_{i,G}}\|, \\ \boldsymbol{\theta}_{i,G} & \text{otherwise,} \end{cases}$$

where $\varphi(\cdot)$ is the fitness function and ε is a predefined tolerance rate. Three steps of DE process are repeated until the goal is met or a maximum learning epochs is completed. This algorithm can obtain compact networks and good performance for function approximation. Therefore, the performance for hematocrit estimation can be improved.

5 Experimental Results

In this section, experimental results of nonlinear methods for hematocrit estimation are presented. Blood samples in dataset were collected from volunteers that were randomly selected. During data collection for experiments, there were outliers which affect the final results. We can remove these outliers by using distance based or area-descent based methods [23]. After removing outliers, we have a dataset of 199 blood samples. Every sample was divided into two parts. The first part was used in determining the transduced current curve. The second part was used to determine the desired hematocrit value which is measured from accurate machine in hospital. The distribution of accurate hematocrit values is shown in Fig. 8 with mean value of 36.02 and deviation of 6.39. This distribution is fairly representing the trend of hematocrit values for human beings. The dataset was divided into two subsets: 40 percent of the data set was used for training and the remaining 60 percent was used for blind test for the methods used for this study which are introduced in the Section 4.

The input features were normalized into the range [0, 1].

In the experimental study for SVM, we tried different combination of cost parameters C and parameters ν : $C=[2^{11}, 2^{10}, \dots, 2^{-1}, 2^{-2}]$ and $\nu=[2^0, 2^{-1}, \dots, 2^{-10}]$. The average results of 50 trials for each combination of (C, ν) were computed and the best performance was obtained when $C=2^9$ and $\nu=2^{-2}$. In ELS-ELM algorithm, the population size NP was set to twice the number of parameters of the networks; F and CR were set to 1 and 0.8, respectively. The number of hidden units used for ELM was 15, and that for both RLS-ELM and ELS-ELM was 5.

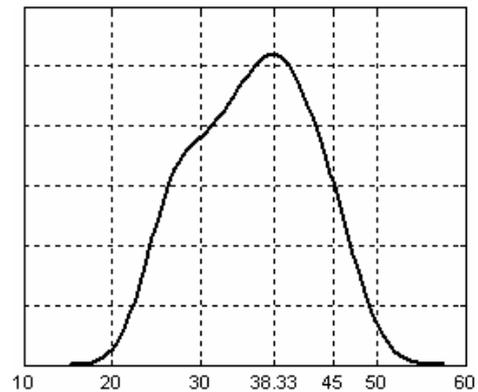


Figure 8. The distribution of collected hematocrit

Method	Training		Testing	
	RMSE	Mean	RMSE	Mean
SVM	3.73	0.25	4.58	-0.09
ELS-ELM	4.25	10^{-5}	4.63	-0.18
RLS-ELM	4.30	10^{-7}	4.90	-0.05
ELM	4.27	10^{-4}	4.90	-0.26

Table 1. Comparison of root mean square errors (RMSE) without using extra features

Method	Training		Testing	
	RMSE	Mean	RMSE	Mean
SVM	3.73	0.23	4.12	-0.19
ELS-ELM	3.72	10^{-8}	3.97	0.10
RLS-ELM	3.69	10^{-8}	4.05	-0.15
ELM	3.81	10^{-2}	4.20	0.10

Table 2. Comparison of root mean square errors (RMSE) with using extra features

The average results of fifty trials without using extra features are shown in Table 1, and those with using

extra features are shown in Table 2. In Table 1, we can see that the ELS-ELM can obtain RMSE of 4.63 for testing set which is smallest in comparison with the mentioned ELM algorithms, and the RMSE for SVM approach was 4.58 which is little smaller than the ELM based methods. In Table 2, we can see that ELS-ELM can obtain RMSE of 3.97 which is smallest in comparison with other mentioned nonlinear methods. Furthermore, results obtained by using extra features are better than those without using extra features.

Although the nonlinear methods that we used for estimating hematocrit density from the whole blood using a glucose biosensor do not give considerable difference in performance, these results can be very important fact for the next studies in reducing effects of hematocrit in glucose measurements by portable devices. In other words, one can select a method depending upon the computational complexity of the methods while considering the specification and computational capability of the hand-held measurement system.

6 Conclusions

As generally accepted in clinical laboratory, the hematocrit is an important factor for clinical decision marking and significantly affects the accuracy of glucose measurements from the whole blood. Estimating hematocrit density by using the transduced current can be an important research step for enhancing performance of glucose measurements by handheld devices.

This paper introduces application of nonlinear methods for hematocrit estimation from the transduced current curve obtained by using portable devices. The experimental results show that the nonlinear models achieve acceptable performance. In comparison with ELM, RLS-ELM and ELS-ELM can obtain better performance with compact networks which can save memory in hardware implementation and make the trained networks responding fast to the new input patterns. This fact is very important for application of a method to hand-held device which has limited computational specification in both hardware and software. Experimental results also show that two extra features extracted from the current curve contribute significantly in enhancing performance of hematocrit estimation.

References:

[1] R. F. Louie, Z. Tang, D. V. Sutton, J. H. Lee, G. J. Kost, "Point of Care Glucose Testing: effects of critical variables, influence of reference instruments, and a modular glucose meter

design", *Arch Pathol Lab Med*, vol. 124, 2000, pp. 257-266.

- [2] Z. Tang, J. H. Lee, R. F. Louie, G. J. Kost, D. V. Sutton, "Effects of Different Hematocrit Levels on Glucose Measurements with HandHeld Meters for Point of Care Testing", *Arch Pathol Lab Med*, vol. 124, 2000, pp. 1135-1140.
- [3] E. S. Kilpatrick, A. G. Rumley, H. Myin, "The effect of variations in hematocrit, mean cell volume and red blood count on reagent strip tests for glucose", *Ann Clin Biochem*, vol. 30, 1993, pp. 485-487.
- [4] E. F. Treo, C. J. Felice, M. C. Tirado, M. E. Valentinuzzi, and D. O. Cervantes, "Hematocrit Measurement by Dielectric Spectroscopy", *IEEE Transactions on Biomedical Engineering*, vol. 25, no. 1, 2005, pp. 124-127.
- [5] K.R. Foster and H.P. Schwan, "Dielectric properties of tissues and biological materials: A critical review", *CRC Crit. Rev. Biomed. Eng.*, vol. 17, no. 1, 1989, pp. 25-104.
- [6] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.
- [7] Sudhir D. Sawarkar, Ashok A. Ghatol "Breast cancer malignancy identification Using Support Vector Machine", *WSEAS transaction on computers*, Issue 8, Volume 5, August 2006, pp. 1707-1712.
- [8] Schölkopf B., A. Smola, R. C. Williamson, and P. L. Berlett, "New Support Vector Algorithms", *Neural Computation*, vol. 12, 2000, pp. 1207-1245.
- [9] Ng, A. W. M., Tran, D. H. & Osman, Y. N., 2006, "Modeling Serviceability Deterioration of Concrete Stormwater Pipes Using Genetic Algorithm Trained Neural Network", *WSEAS Transaction on Systems*, Vol.5, No. 7, pp. 1662-1670.
- [10] Benvenuti M., Colantonio S., and Di Bono, "Tracking of Moving Targets in Video Sequences", *WSEAS Transaction on Systems*, Vol. 4 no. 4 (2005), pp. 359-364.
- [11] Yu-Ju Lin, Chin-Sheng Huang, Che-Chern Lin, "Determination of Insurance Policy Using Neural Networks and Simplified Models with Factor Analysis Technique", *WSEAS Transaction on Information science and Applications*, vol. 5, no. 10, (October 2008), pp.1405-1415.
- [12] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights", *Int'l Joint Conf. Neural Neural Networks*, vol. 3, San Diego, CA, (1990) pp.21-26.

- [13] Jim Y. F. Yam and Tommy W. S. Chow, "Feedforward Networks Training Speed Enhancement by Optimal Initialization of the Synaptic Coefficients", *IEEE Trans. On Neural Networks*, vol. 12, no. 2, (2001) pp.430-434.
- [14] L. S. Nghia, J. Sjöberg and M. Viberg, "Adaptive neural nets filter using a recursive levenberg-marquardt search direction", *Proc. Asilomar Conf. Signals, Syst., Comput.*, vol. 1-4, (Nov. 1998) pp.697-701.
- [15] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, (2006) pp. 489-501.
- [16] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward neural networks", *IEEE Trans. on Neural Networks*, vol. 17, no. 6, (2006) pp. 1411-1423.
- [17] H.T. Huynh, J. Kim and Y.Won, "An Improvement of Extreme Learning for Compact Single Hidden Layer Feedforward Neural Networks", *International Journal of Neural System*, vo.18, no.5 (2008) pp.1-9.
- [18] A. N. Tikhonov and A. V. Arsenin, "Solution of Ill-posed Problems", *Winston & Son, Washington*, 1977.
- [19] P. L. Bartlett, "The Sample Complexity of Pattern Classification with Neural networks: The Size of the Weights is More Important than the Size of the Network", *IEEE Trans. On Inform. Theory*, vol. 44, no. 2, (1998) 525-536.
- [20] H.T. Huynh and Y. Won, "Evolutionary Algorithm for Training Compact Single Hidden Layer Feedforward Neural Networks", *Proc. of the IEEE World Congress on Computational Intelligence (WCCI2008)*, (June 2008) pp. 3027-3032.
- [21] R. Storn, K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *Journal of Global Optimization*, vol. 11, (1997) pp. 341-359.
- [22] H. T. Huynh and Y. Won, "Small Number of Hidden Units for ELM with Two-Stage Linear Model", *IEICE Trans. on Information and Systems*, vol. E91-D, no. 4, (2008) pp. 1042-1049.
- [23] Hieu Trung Huynh, M.-T.T. Hoang, N. H. Vo, and Yonggwan Won, "An improvement of Outlier Detection in Linear regression based on Area-Descent", *WSEAS Transactions on Computer Research*, vol. 1, no. 2, (December 2006) pp.174-180.