Two States Mapping Based Neural Network Model for Decreasing of Prediction Residual Error

Insung Jung, lockjo Koo, and Gi-Nam Wang

Abstract**—**The objective of this paper is to design a model of human vital sign prediction for decreasing prediction error by using two states mapping based time series neural network BP (back-propagation) model. Normally, lot of industries has been applying the neural network model by training them in a supervised manner with the error back-propagation algorithm for time series prediction systems. However, it still has a residual error between real value and prediction output. Therefore, we designed two states of neural network model for compensation of residual error which is possible to use in the prevention of sudden death and metabolic syndrome disease such as hypertension disease and obesity. We found that most of simulations cases were satisfied by the two states mapping based time series prediction model compared to normal BP. In particular, small sample size of times series were more accurate than the standard MLP model. We expect that this algorithm can be available to sudden death prevention and monitoring AGENT system in a ubiquitous homecare environment.

Keywords**—**Neural network, U-healthcare, Prediction, Time series, CAP (Computer Aided prediction).

I. INTRODUCTION

MANY people and industries are interested in the decision support system, and prediction systems for the better choice and reduction of risk based on intelligence method. Especially, artificial neural network based prediction systems. These methods are seemed to be successful to solve difficult and diverse problems by supervised training methods such as back-propagation algorithm. This algorithm is the most popular neural network architecture for supervised learning, because it is based on the weight error correction rules. Although back-propagation algorithm could correct weights, it still got a residual error. Therefore, standard neural network model based prediction system is slightly risky in the medical and clinical domain area. Cause of the prediction result may differ from the real value, the consequences may be seriousness of the disease or sudden death whereas the second state model learns about errors known as residual time series. Subsequently, second result can be used as a complementary support to reduce the residual error of the first state result. We compared the performance evaluation of each model using MAD (Mean absolute deviate) and MSE (mean square error). It shows two states mapping based prediction model result is better than standard prediction model with time series human bio signal data. Two states mapping based prediction model can use CAP (Computer Aided Prediction) system for prevention of sudden deaths and high-risk diseases. In addition, it could use personalized healthcare and health management system.

This paper has been arranged as per order, section 2 review related work, section 3 gives a detailed description of the material and methods used to design the Two states mapping based time series neural network model, section 4 describes about the result and discussion finally concluded with future perspective of the system with conclusion.

Insung Jung is with Department of Industrial Engineering Ajou University 442-749, Korea (e-mail: insung9908@gmail.com).

lockjo Koo is with Department of Industrial Engineering Ajou University 442-749, Korea (e-mail: lockjo9@gmail.com.).

Gi-Nam Wang is with Department of Industrial Engineering Ajou University 442-749, Korea (e-mail: gnwang@ajou.ac.kr).

II. RELATED WORK

Many people and industries are interested in the decision support system, and prediction systems for the better choice and reduction of risk based on intelligence method. Especially, ANN (artificial neural network) is considered as an effective approach to solve difficult and diverse problems using supervised training methods such as back-propagation algorithm [1]. Prediction or forecasting the future has been the goal of many research activities about an important problem for human, unknown phenomena, calamities and so on in a variety of disciplines that range from economics through physics to engineering[2].

Box-Jenkins methodology that was proposed by Box and Jenkins became highly popular among particular academics of empirical studies in 1970s [3]. Time series analysis provides a basis for identifying models that are used for economic and business planning. The Box-Jenkins time series methodology [4] has been one of the best off-line methods to model and forecast time series [5]. But when the time series data is not stationary, this methodology is inappropriate for forecasting and prediction, so it is imperative that user should transform to a stationary one [6]. Moreover, this methodology is difficult to predict a long term prediction. So neural network models have been used for modeling, forecasting, prediction of time series, because it is overcame the shortcut of Box-Jenkins methodology and applied for nonlinear models [6]. Hence, in this paper, time series prediction using neural network was examined as follows; Hashem et al. used multi-step ahead training algorithm for training neural network for use in long-term prediction. This algorithm was based on Box-Jenkins methodology in determining the training lead time and the network input. Output of this algorithm was the next value of time series which is similar to a single-step ahead predictor, but the difference was repeatedly fed back to the network for pre-determined number of cycles without changing the connection-weight. Experimental results that was tested on simulated time series data and Cario ozone data set, was constructed more efficient and accurate neural network. ; The training lead time is more accurate than Box-Jenkins time series [6]. Nie study was suggested a fuzzy-neural approach to the prediction of nonlinear time series by a fuzzy predictor on the basis of extracted rules [7]. GHOLIPOUR et al. examined several neural and neurofuzzy model with different learn algorithms for prediction of several benchmark chaotic system and time series. Locally Linear Model Tree (LoLiMoT) learn algorithm, Radial Basis Function (RBF) neural network, MultiLayer Perceptron and Adative network is compared by using cross validation technology. Moreover various several chaostic system and time series was examined as case studies [8]. The result of two studies was indicated that the efficiency of accuracy and learning speed was superior. Watanabe proposed two methods. First method was to predict the changing properties of non-stationary time series data with time varying parameters. This method was constructed by the hierarchical combination of each neural network for prediction of time series data and prediction of weight. Second method was to determine the length of the local stationary section. This method was used the addictive learning ability of multi-layered neural networks. The simulated experiments of prediction by AR model and single neural network model and of computational time were more effective [9]. Wakuya and Shida were proposed a bi-directional computation approach to improve an accuracy of time series prediction. This approach was worked by interacting between the forward–time direction and a separate backward-time processing system. This study is applied to "Data Set D" was time series and was artificially generated by a computer and are distinguished by their great length, consisting of 100,000 points. As a result, an improvement in dealing with untraining was same to training data and prediction was achieved more accurate than the conventional method [10]. In generally, a model with the smallest mean prediction error or An Information Criterion (AIC) was considered best one among a set of models. But Ishikawa and Moriyama proposed to use both a structural learning with forgetting and the mean prediction error or AIC to find a model with better generalization ability. The structural learning with forgetting and BP learning were applied. Result of simulation was indicated the structural learning with forgetting has better generalization ability than BP learning both in Jordan networks and buffer networks.[11-12]

III. Material & System Framework

A. Material

The major function of the knowledge database system is to provide and store information. The knowledge database system includes the values of the vital signs by medical standards, details of methods of prevention of diseases with disease-specific recommendations, methods for the evaluation of condition of emergency or need of medical intervention. Additionally, the vital signs of the person under consideration, the predictions made on him/her together with the medical recommendations are automatically stored in the knowledge database system.

Our simulation test materials were 2 types of data. First type of data was artificial data using AR (Auto correlation) models. That information describes into the Table I.

Example: The process models to be considered are auto-regressive models.

$$AR(2): X_t^1 = 1.49 X_{t-1}^1 - 0.653 X_{t-2}^1 + e_t$$
$$AR(3): X_t^2 = 2.146 X_{t-1}^2 - 1.598 X_{t-2}^2 + 0.409 X_{t-3}^2 + e_t$$
$$AR(4): X_t^3 = -1.876 X_{t-1}^3 - 1.781 X_{t-2}^3 - 1.201 X_{t-3}^3 - 0.373 X_{t-4}^3 + e_t$$
$$AR(5): X_t^4 = 1.840 X_{t-1}^4 - 0.893 X_{t-2}^4 - 0.613 X_{t-3}^4 - 0.879 X_{t-4}^4 - 0.310 X_{t-5}^4 + e_t$$

TABLE I
Sequences of Linear Stationary Series with Model Parameters (100 Sample Size)

Time	Model	E(y)	Model Parameters
1-100	AR(2)	10.0	(1.60, 1.49,-0.65)
100-200	AR(6)	10.4	(22.0,-0.77,-0.51, 0.31, 0.11,-0.03,-0.21)
201-300	AR(2)	10.5	(27.5,-1.07,-0.52)
301-400	AR(4)	10.1	(63.2,-1.87,-1.78,-1.20,-0.37)
401-500	AR(5)	9.80	(0.96, 1.84,-0.89,-0.61, 0.87,-0.31)
501-600	AR(3)	10.3	(0.445, 2.146,-1.598, 0.409)
601-700	AR(6)	10.5	(0.19, 2.40,-2.41, 1.88,-1.58, 0.90,-0.20)
701-800	AR(3)	15.0	(6.50, 0.92,-0.26,-0.26)
801-900	AR(4)	10.5	(6.4, 0.96,-0.41,-0.30, 0.14)
901-1000	AR(5)	10.1	(21.0,-0.37,-0.32, 0.10, 0.02,-0.50)

AR(2): $y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2}$

We generate synthetic 1000 series based on the given sequences of the parameters shown in Table I. The first 900 warm-up data points are deleted from each sequence and the remaining small series such as 50, 100, and 300 series are used for simulation. There are 10 sequences and each sequence has stationary small observations.

Second type of data was human vital signs data such as heart rate (HR), blood pressure (BP), breath rate (BR) and SPO2 was acquired by a watch type of bio-signal acquiring devices. Those data was stored in a knowledge database from ajou hospital data.

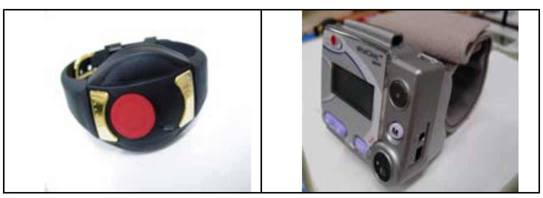

Fig. 1 Biological Signal Acquiring (source medic4all device)

B. *Computer Aided Prediction System Framework*

The Computer Aided Prediction (CAP) system predicts if a person's vital signs may result in the outbreak of an illness such as acute respiratory problem, hypertension etc. It also predicts the possible health condition of a person in normal health for the near future. In our framework, two-phase reverse neural network is used to enumerate what factors are posing serious problems in the person's life that leads to a person being diseased. This system acquires input vector about user's previous and current data ($Y_{t-1} \sim Y_{t-n}$) for the prediction of the risk-level. The steps followed by the CAP system are described in Fig. 2. First, CAP system checks the server for an updated data in order to monitor the user's status. Secondly, it makes a prediction and sets the new learning weights. Finally, it saves those vital data in the database.

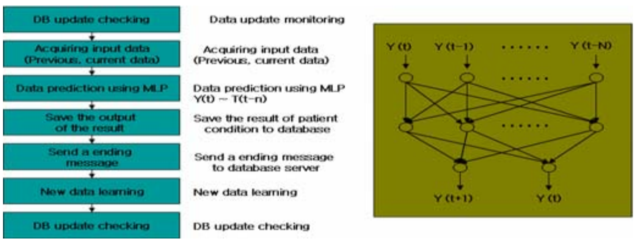

Fig. 2 CAP System framework

IV. Methodology

A. *Standard Neural Network Prediction Model*

Standard multilayer perceptron (MLP) architecture consists more than 2 layers; A MLP can have any number of layers, units per layer, network inputs, and network outputs such as fig 3models.

This network has 3 Layers; first layer is called input layer and last layer is called output layer; in between first and last layers which are called hidden layers. Finally, this network has three network inputs, one network output and hidden layer network.

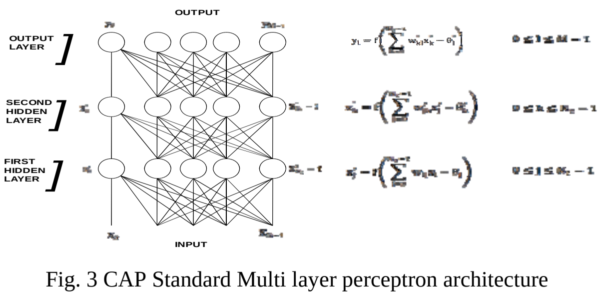

Fig. 3 CAP Standard Multi layer perceptron architecture

However, this research is compared with Back-propagation (BP) model. This model is the most popular in the supervised learning architecture because of the weight error correct rules. It is considered a generalization of the delta rule for nonlinear activation functions and multilayer networks.

The neural network prediction model is between input and output. Inputs are time-series data, and outputs are time-series estimate data with vital signs such as blood pressure, heart rate, SPO_2 and breath rate, and others.

This prediction model could be designed as follows. $\hat{y}_t$ is the estimated output, and $\hat{e}_t$ is the corresponding residual

$$\hat{y}_t = O_t = NN(X_t) + \hat{e}_t \quad (1)$$

According to the Richard P. Lippmann [13], he represents step of the back-propagation training algorithm and explanation.
The back-propagation training algorithm is an iterative gradient designed to minimize the mean square error between the actual output of multi-layer feed forward perceptron and the desired output. It requires continuous differentiable non-linearity. The following assumes a sigmoid logistic nonlinearity.

Step1: Initialize weights and offsets

Set all weights and node offsets to small random values.

Step2: Present input and desired outputs

Present a continuous valued input vector X_0, X_1…..X_{N-1} and specify the desired output d_0,d_1,….d_{M-1}. If the net is used as a classifier them all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until stabilize.

Step 3: Calculate Actual Output

Use the sigmoid non linearity from above and formulas as in fig 3 to calculate output y0,y1….yM-1.

Step 4: Adapt weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' \quad (3)$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t, w_j', is either the output of node i or is an input, η is a gain term, and δ_j, is an error term for node j, if node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j - y_j) \quad (4)$$

where d_j is the desired output of node j and y_j is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j'(1-x_j')\sum_k \delta_j^m w_{jk} \quad (5)$$

where k is over all nodes in the layers above node j.
Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight change are smoothed by

$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' + \alpha(w_{ij}(t) - w_{ij}(t-1))$,where0< α <1. (6)

Step 5: Repeat by going to step 2

B. Two States Mapping Based Neural Network for Decreasing Prediction Residual Error

The architecture of a back-propagation network is not completely constrained by the problem to be solved, although many in the industry utilize MLP for prediction systems, using time-series data. This means it still has residual errors in prediction cases. In addition, the standard multi layer perceptron (MLP) model (back propagation) has some problems, including local optimum, difficult to modify small sample size data, and one-direction learning (feed forward).

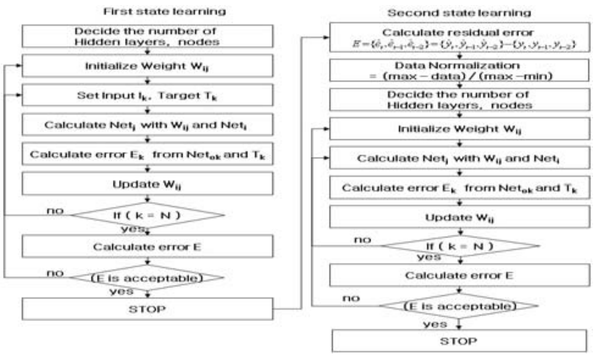

Fig. 4 Two state mapping based time series prediction model flowchart

1) The First State Time-Series Prediction Learning Model

The first state model equation (7) could be designed as follows; $\hat{y}_t$ is the estimated output, and $\hat{e}_t$ is the corresponding residual:

$$\hat{y}_t = O_t = NN\ (X_t) + \hat{e}_t \tag{7}$$

This model consists that inputs are time-series data ($X_i = y_{t-1}, y_{t-2}, y_{t-3} .... y_{t-n}$) of human vital signal, and outputs are time-series estimate data ($Y_t = \hat{y}_t, \hat{y}_{t-1}, \hat{y}_{t-2} ..... \hat{y}_{t-n+1}$) such as blood pressure, heart rate, SPO_2 and breath rate, and others.

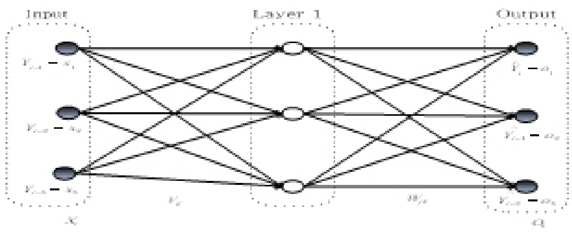

Fig. 5 First state time-series prediction learning model

Steps 1 to 6 explain the process of the time-series prediction learning model

Here we define as variables:

Input nodes set: $X_i = y_{t-1}, y_{t-2}, y_{t-3}$

Target set: $T = y_t, y_{t-1}, y_{t-2}$

Output set: $O = \hat{y}_t, \hat{y}_{t-1}, \hat{y}_{t-2}$

Weight vector between input layer and hidden layer: V

Weight vector between hidden layer and output layer: W

Sigmoid function: $f = \frac{1}{(1 + e^{-n^u})}$

Calculate error: δ, E

Step 1: Set up all node weight, Target, Input vector

Step 2: Compute input and hidden layer's node for calculation of output.

$$O_k^o = X_k^u \quad \text{for all} \tag{8}$$

$$n_j^{hu} = \sum x_i V_{ij} \tag{9}$$

$$h = 1, 2 \ldots hidden\_layer\_number$$

$$W_j^u = f(\sum x_i^u V_{ij})$$

$$O_i^m = f(n_k^u) = f(\sum_j V_{ij}^{mO\ m-1}{}_i W_j^u)$$

Step 3: Compare output and Target and then calculate error (δ)

$$\delta_i^m = [T_i^m - O_i^m] g'(g_i^M) \tag{10}$$

Step 4: Backward computing of error for mediation weight from output layer's node to input

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_i w_{ij} \delta_j^m \quad \text{(m= M, M-1 … 2)} \tag{11}$$

Step 5: Compute alternative value of weight

$$\Delta v_{ij}^m = n \delta_i^m O_j^{m-1} \tag{12}$$

$$\Delta \theta_i^m = -n \delta_i^m$$

$$v_{ij}^{new} = v_{ij}^{old} + \Delta v_{ij}, \theta_{ij}^{new} = \theta_{ij}^{old} + \Delta \theta_{ij} \tag{13}$$

Step 6: Repeat step 2 until accept error is less than setup.

2) The Second State of Learning Process

The model of forward learning has error correction mapping with estimated value ($\hat{y}_t$) with true value (y_t). It might calculate all time-series errors ($\hat{e}_t$).

This model is using the time-series error ($\hat{e}_{t-1}, \hat{e}_{t-2}, \hat{e}_{t-3}$)

$$\hat{e}_t = y_t - \hat{y}_t \tag{14}$$

This model shows the relationship between input and output. Inputs are residual errors ($X_i = e_{t-1}, e_{t-2}, e_{t-3}$) of estimate error between $\hat{y}_t$ and (y_t), and outputs are time-series error estimate ($\hat{e}_t$).

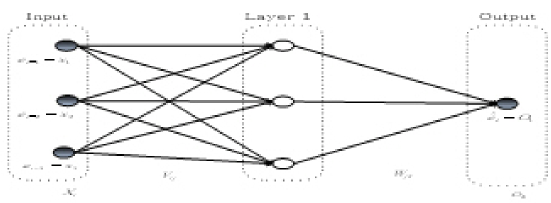

Fig. 6 Second state learning flowcharts with current residual error

Variables are defined as follows:

Input nodes set: $X_i = e_{t-1}, e_{t-2}, e_{t-3}$

Target set: $T = \hat{e}_t$, Output set: $O = \hat{e}_t$

Weight vector between input layer and hidden layer: W

Weight vector between hidden layer and output layer: V

Sigmoid function: $f = \frac{1}{(1+e^{-n^u})}$ (15)

Calculate error: δ, E

Step 1: Calculation of residual error $E_t = \hat{e}_t, \hat{e}_{t-1}, \hat{e}_{t-2} .... \hat{e}_{t-n+1}\} = \{\hat{y}_t, \hat{y}_{t-1}, \hat{y}_{t-2} .... \hat{y}_{t-n+1}\} - \{y_t, y_{t-1}, y_{t-2} .... y_{t-n+1}\}$ (16)

Step 2: Data normalization

$$Normalization\_data = \frac{(\max - data)}{(\max - \min)} \quad (17)$$

Step 3: Set up all node weights, Target, Input vector

Step 4: Compute between input and hidden layer's node for calculation of output.

$$O_k^o = X_k^u \text{ for all} \quad (18)$$

$$\begin{aligned} n_j^{hu} &= \sum x_i w_{ij} \\ h &= 1, 2 \ldots hidden\_layer\_number \\ V_j^u &= f(\sum x_i^u w_{ij}) \\ O_i^m &= f(n_k^u) = f(\sum_j w_{ij}^{mO} {}_j^{m-1} V_j^u) \end{aligned} \quad (19)$$

Step 5: Compare output and Target and then calculate error (δ)

$$\delta_i^m = [T_i^m - O_i^m] g'(g_i^M) \quad (20)$$

Step 6: Backward computing of error for mediation weight from output layer's node to input

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_i w_{ij} \delta_j^m \text{ (m= M, M-1 …… 2)} \quad (21)$$

Step 7: Compute alternative value of weight

$$\begin{aligned} \Delta w_{ij}^m &= n \delta_i^m O_j^{m-1} \\ \Delta \theta_i^m &= -n \delta_i^m \end{aligned} \quad (22)$$

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}, \theta_{ij}^{new} = \theta_{ij}^{old} + \Delta \theta_{ij} \quad (23)$$

Step 8: Repeat step 4 until accept error is less than setup.

V. Result

The prediction model performance test of adaptive hybrid two-phase reverse neural network model was compared to a normal MLP prediction neural network by matlab code. In this simulation, we set up sequential input data from previous (X_{t-n}) to current data (X_{t-1}) with several types of human biosignal data such as SpO2, heart rate, blood pressure, and breath rate.

TABLE II
Neural Network Layer Modeling Structure for Each Data

Heart Rate	Breath Rate	SpO2	High Blood Pressure	Low Blood Pressure
66	14	93	101	70
66	13	90	100	74
69	13	90	120	74

For the prediction model to operate, it must set up the period of input node ($X_t, X_{t-1} \cdots X_{t-n}$), target node ($Y_{t+1}, Y_t \cdots Y_{t-n-1}$), hidden layer number, hidden node number, and maximum iteration number. The adaptive hybrid two-phase reverse neural network model condition is exactly the same as a normal prediction model.

Each simulation has a different hidden layer number, hidden node number, and maximum iteration number setup. For the first hidden node, the number of hidden layers is 4. At this point, 500 iterations are carried out and then the iterations are incremented each time by 500 until it reaches 2,000. At the end of this iteration, the hidden layer is incremented by 4 and then a similar iteration is carried out. This process is continued until the hidden layers become 16. When the hidden layer number reaches 16, the whole cycle of iterations begins again for the second hidden node and is carried out until the hidden node reaches the third node.

The evaluation of both model performances was good for prediction but two-phased reverse neural network was better than normal back-propagation based on the prediction model. Results were checked for residual error using sum of mean square error and sum of mean absolute deviate. The best performances of a back-propagation prediction model (BPM) evaluation computed MSE and MAD as between 0.07 and 0.09 by sum of mean square error (MSE) and 2 and 5 by sum of mean absolute deviate (MAD).

In addition, setting parameters are normally one or two hidden layers and eight hidden nodes each. However, these were not optimal parameters in this simulation. In Tables III and IV the parameter settings and evaluation results for each bio-signal data table are presented.

TABLE III
PARAMETER SETTINGS AND BPM EVALUATION

	Input Node	Hidden Layer	Hidden Node	Output Node	Accept Error	MSE	MAD/N
SpO2	3	1	8	3	0.04	0.125	0.1258
BR	3	1	8	3	0.04	0.018	0.0775
HR	3	1	8	3	0.04	0.091	0.0916
HBP	3	1	8	3	0.04	0.071	0.0713
LBP	3	1	8	3	0.04	0.017	0.0853

TABLE IV
PARAMETER SETTINGS AND TSNN EVALUATION

	Input Node	Hidden Layer	Hidden Node	Output Node	Accept Error	MSE	MAD/N
SpO2	3	1	8	3	0.04	0.077	0.0770
BR	3	1	8	3	0.04	0.017	0.0742
HR	3	1	8	3	0.04	0.061	0.0614
HBP	3	1	8	3	0.04	0.069	0.0699
LBP	3	1	8	3	0.04	0.016	0.0734

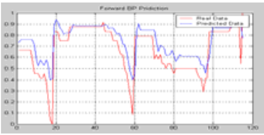
Fig. 7 BPM evaluation (SPO2)

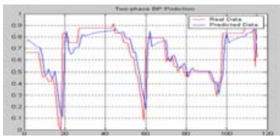
Fig. 8 TSNN evaluation (SPO2)

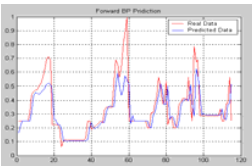
Fig. 9 BPM evaluation (HBP)

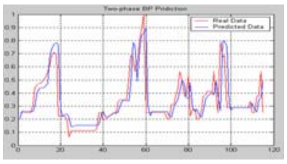
Fig. 10 TSNN evaluation (HBP)

Figs. 7–10 present prediction graphs of each bio signal data. Red lines are real data and blue lines are prediction data.

Comparing each model evaluation result, the two-phase reverse neural network evaluation result is much better than the normal back-propagation model result.

In this simulation, we determined that most of the simulation cases were satisfied by the two-phase reverse prediction neural network. In particular, small sample size of times series were more accurate than the standard back-propagation model.

VI. CONCLUSION

Modern medical service systems are lacking in terms of offering real-time patient monitoring, diagnosis, and early detection of disease symptoms and problems in patient's health state.

The primary role of two states mapping based neural network model is to generate time series of human vital signal prediction in the near future. This model is used to measure the problems of standard multi layer perceptron (MLP) model, which are local optimum, difficulty in modifying small sample size data, and one-direction learning (feed forward). Two states mapping based neural network model incorporates two training structures.

The model of forward learning is almost the same as the standard MLP model, between input vector of previous data ($X_i = y_{t-1}, y_{t-2}, y_{t-3}$) and the target vector ($Y_i = \hat{y}_t, \hat{y}_{t-1}, \hat{y}_{t-2}$). However, the forward learning model has error correction mapping with estimated value ($\hat{y}_t$) with true value (y_t). It might predict time-series error ($\hat{e}_t$). On the other hand, the backward learning model is the opposite way training from output of forward learning estimated value($Y_i = \hat{y}_t, \hat{y}_{t-1}, \hat{y}_{t-2}$) to input vector of forward learning ($X_i = y_{t-1}, y_{t-2}, y_{t-3}$).

It is found to be better than the previously used multi-layer perceptron model in that the residual error is highly reduced in the former. Especially, it might predict small sample size of data pattern. It is available to learn patient's personalized living body signal pattern. It can predict and preempt the symptoms of a disease.

The design of the adaptive hybrid clinical decision and prediction support system could use a CDSS AGENT system for personalized diagnosis, prevention, and recommendation. In addition, a Computer-Aided Prediction system explains how a person's lifestyle will affect his or her metabolic syndrome in the near future.

The research suggests the best prescription for prevention of diseases related to metabolic syndrome and high risk disease in the U-hospital, home healthcare system, PERS (Personal Emergency Response System), and silver town healthcare for elder people and patients.

ACKNOWLEDGMENT

This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

REFERENCES

[1] Lisboa PJ.,: "A review of evidence of health benefit from artificial neural networks in medical intervention", Neural Networks. Vol. 15, January 2002 pp. 11-39.

[2] Ali Gholipour, Babak N. Araabl, and Caro Lucas "Predicting Chaotic Time Series Using Neural and Neurofuzzy Models: A Comparative Study" Neural Processing Letters DOI 10.1007/s11063-006-9021-x, 24, 2006, pp. 217–239.

[3] Spyros Makridakis and Micheá Le Hibon"Arma Models and the Box & Jenkins Methodology"Journal of Forecasting Vol. 16, 1997, pp 147-163.

[4] Ashour, Z.: Artificial neural network models for forecasting ozone data, in Proceedings of The thirty annual conference ISSR, Cairo university, vol. 30, Part 3. 1995, .pp. 83-96.

[5] Box, G.E.P and G.M Jenkins, "Time series Analysis: Forecasting and Control, 2nd " ed., Oakland, CA: Holden-Day 1976.

[6] S. Hashem Z. H. Ashour E. F.Abdel Gawad A. Abdel Hakeem "A Novel approach for Training Neural Networks for Long-Term Prediction" IEEE Vol. 0-7803-5529-6 , 1999, pp.1594-1599.

[7] Junhong Nie "Nonlinear time-series forecasting: A fuzzy-neural approach" Neurocomputing v.16 no.1, 1997, pp. 63-76.

[8] Ali Gholipour, Babak N. Araabl, and Caro Lucas "Predicting Chaotic Time Series Using Neural and Neurofuzzy Models: A Comparative Study"Neural Processing Letters DOI 10.1007/s11063-006-9021-x, 24, 2006, pp 217–239.

[9] Eiji Watanabe "Time Series Prediction by a Modular Structured Neural Network" IEEE Vol. 0-7803-4859- 1, 1998, pp. 2501-2506 .

[10] Masumi Ishikawa, Teppei Moriyama, : "Prediction of time series by a structural learning of neural networks", Fuzzy Sets and Systems, V82 , 1996, 167-176.

[11] M. B. Priestly, Non-linear and Non-stationary Time Series Analysis, Academic Press, New York, 1989.
[12] R. A. Jacobs, M. I. Jordan, and A. G. Barto, Task Decomposition through Competition in a Modular Connectionist Architecture: the What and Where Vision Tasks, Cognitive Science 15, 1991, pp 219-250.
[13] Richard P. Lippmann,:" An introduction to computing with neural network", IEEE ASSP magazine, 1987, pp. 4-22.