MJFAS MALAYSIAN JOURNAL OF FUNDAMENTAL AND APPLIED SCIENCES
PRINT ISSN: 2289-5981 | ONLINE ISSN: 2289-599X

RESEARCH ARTICLE

Clinical pathway variance prediction using artificial neural network for acute decompensated heart failure clinical pathway

Mohamad Haider Abu Yazid [b,*], Muhammad Haikal Satria [a], Mohd Soperi Mohd Zahid [b], Mohamad Shukor Talib [b], Habibollah Haron [b], Azmee Abd Ghazi [c]

[a] *Faculty of Bioscience and Medical Engineering, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia*
[b] *Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia*
Department of Cardiology, Institut Jantung Negara (IJN), Malaysia

* *Corresponding author: mhaider04@gmail.com*

Article history
Submitted 19 December 2017
Revised 9 February 2018
Accepted 1 Mac 2018
Published Online 26 Mac 2018

Abstract

Patients in modern healthcare demand superior healthcare quality. Clinical pathways are introduced as the main tools to manage this quality. A clinical pathway is a task-oriented care plan that specifies steps to be taken for patient care. It follows the clinical course according to the specific clinical problem. During clinical pathway execution, variance or deviation from the specified care plan could occur, and may endanger the patient's life. In this paper, a proposed framework for artificial neural networks (ANNs) in clinical pathway variance predictions is presented. This proposed research method predicts the variance that may occur during Acute Decompensated Heart Failure Clinical Pathway. By using the Artificial Neural Network, 3 variances (Dialysis, PCI, and Cardiac Catherization) are predicted from 55 input. The results show that artificial neural networks with the Levenberg-Marquadt training algorithm with a 55-27-27-1 architecture achieve the best prediction rate, with an average prediction accuracy of 87.4425% for the training dataset and 85.255% for the test dataset.

Keywords: Artificial neural network, acute decompensated heart failure, variance prediction, clinical pathway

INTRODUCTION

A clinical pathway (CP) is set of tools which plays an important role in improving the quality of patient care and increasing healthcare organizations' efficiency by supporting the standardized treatment process. Clinical pathways were introduced in the mid-1980s by Zander and Bower, and can be described as the guidelines for clinical practice for specific groups of patients, based on a particular diagnosis. Clinical pathways specify the categories of care, activities, and procedures that need to be conducted for the patient until they are discharged from the hospital, displayed in a timeline. Deviation of actual care from the standardized care activity may happen anytime during an episode of care, which is called variance, and is also managed and handled by the clinical pathway.

Clinical pathways can bring benefits to the healthcare provider. Among their benefits are: 1) improve patient clinical outcomes; 2) help to reduce hospital costs and help hospital management to optimize resources in terms of equipment or personnel. Common practice of clinical pathways requires medical professionals to manually fill in predefined paper documents. This practice, known as a paper-based clinical pathway, is limited and not dynamic, which brings several problems. These problems include:

1) Limited to just the capacity of data collection and recording.
2) Separated from the hospital information system.
3) Lack of support for real time patient monitoring.
4) The complex logical and timed relationship of different activities cannot be described with the simple description in term of forms.
5) Unable to detect and handle variance dynamically.

To overcome some of these problems, an electronic clinical pathway is introduced. Efforts to develop computerized or electronic clinical pathways have already been going since the 1990s, when the linear sequential model of the electronic clinical pathway was developed in early 1990s. Since then, electronic clinical pathways evolved to state transition models of electronic clinical pathways in the late 1990s, the structural design was adapted in the 2000s, and was further developed to utilize an ontological design in 2007. Furthermore, the capability of electronic clinical pathways was further enhanced by embedding the electronic clinical pathway with electronic medical records and integrating the electronic clinical pathway with the nursing process.

Most of the current practice around electronic clinical pathways is not dynamic or adaptive. During the event of variance, most of the current electronic clinical pathways only provide the means to detect, record, or handle the variance occurrence. Most of the methods proposed for variance management in clinical pathways usually deal with one type of variance and rely on the fuzzy rules provided by the domain expert, which are difficult to obtain.

This research proposes a method to predict variance during clinical pathways in order to give better preparation for the treatment. Artificial Neural Networks (ANN) prove to be powerful tools for mapping nonlinear data, and are known to be useful in solving nonlinear problems where the rules to solve the problem are difficult to obtain or are unknown. This paper proposes the use of artificial neural networks to predict variance for acute decompensated heart failure (ADHF) clinical pathways. The objective of this paper is to show the framework of artificial neural networks for variance prediction in ADHF clinical pathways.

RELATED WORKS

Most of the research on clinical pathways mainly focuses on documenting, classifying, analyzing, and handling variances. Documenting variance deals with the aspect of recording its occurrences, while classifying variances will help the clinical pathway to identify what type of variance (system variance, staff variance, or patient variance) occurs and whether the variance is good (e.g. decreasing of patient's length of stay), or bad (e.g. patient complications during treatment). Table 1 shows a review on the research of clinical pathway variances.

S. Wakamiya and K. Yamauchi have proposed a system that is capable of managing clinical pathway variances. Low-cost implementation and portability are the main features of the proposed systems. Systems prior to the systems proposed by Wakamiya and Yamauchi needed specialized hardware and software, which are not easy to adapt for the use of other institutions. The proposed system could be implemented to various clinical pathways. Clinical pathways that have been implemented using the proposed systems are Gastroenteritis, Cardiac Catheterization, Bronchitis, Pneumonia, Cataracts, Acute Myocardial Infraction, Transurethral Ureterolithorispy and Infant Bruising.

Clinical variance management and analysis (CVMA) applications were proposed by Kate L. Hyet et al. in 2007. This application is designed to collect variance data for documentation, classification, and analysis of variance. The variance analysis application enabled the collection of variance data from clinical pathways and can readily be changed to accommodate new clinical pathways or additional variances. Unfortunately, this application is not integrated with electronic medical records or any electronic clinical pathway systems. The capabilities of this system are limited to managing reported variance data, and are unable to automatically redesign clinical pathways based on the reported variances. 15 clinical pathways are used in this study, which covers small rural hospitals and large regional hospitals.

A method for automated variance identification and analysis was proposed by Xiang Li et al. in 2014. The proposed method is able to automatically identify the deviation between actual patient traces in electronic medical records and a multistage clinical pathway. The clinical pathway variance analysis method proposed by Xian Li et al uses a clinical pathway and patient traces of cohort in Electronic Medical Records as inputs, and the variance analysis report as an output. The clinical pathway use in this study is the congestive heart failure clinical pathway. Even though the proposed method is able to identify deviation in the clinical pathway, it is still unable to define whether the deviation is positive (e.g. reduce length of stay) or negative outcomes (e.g. prolonged length of stay).

Several researchers have proposed methods for variance handling. Yan Ye et al. (2009) presented a knowledge-based variance management system which has been developed based on unified modeling language (UML) with the use of generalized fuzzy ECA (GFECA) rules and typed fuzzy petri net extended by process knowledge (TFPN-PK) models for analysis and handling of clinical pathway variances. The architecture of the proposed system consists of three levels, which are the client level, application level, and knowledge base levels. The client level consists of a user interface for different types of users and the clinical pathway workflow system. The application level consists of four modules, including a fuzzy reasoning and variance handling engine. The knowledge based level consists of medical knowledge which is represented using ontology. The variance handling engine is activated by the clinical pathway workflow system when the variance information and handling request is sent. The engine searches for the appropriate rules matching the variances in the GFECA rule base. If the rules are found, it performs rule reasoning; otherwise it activates the TFPN-PK based fuzzy reasoning engine. This system can be implemented to any clinical pathway.

Table 1 Review on selected clinical pathway variance studies.

Author, Year	Field	Clinical Pathway	Proposed Method
S. Wakamiya and K. Yamauchi (2006)	Variance documentation, classification, and analysis	Various	Electronic system for paper based clinical pathway management that is capable of documenting, classifying, and analyzing variances
Kate L. Hyet et al (2007)	Variance documentation, classification, and analysis	Various	Clinical variance management and analysis application are proposed
Xiang Li et al (2014)	Variance analysis	Congestive heart failure	Automated clinical pathway variance analysis for multistage clinical pathways is proposed
Yan Ye et al (2006)	Variance handling	General	Proposed generalized fuzzy ECA (GFECA) rules and typed fuzzy petri net extended by process knowledge (TFPN-PK) models for analysis and handling of clinical pathway variances
Gang Du et al (2012)	Variance Handling	Osteosarcoma preoperative chemotherapy	Takagi-Sugeno (T-S) fuzzy neural network with random cooperative decomposing particle swarm optimization algorithm (RCDPSO) and discrete binary version of PSO algorithm learning algorithm
Gang Du et al (2013)	Variance Handling	Osteosarcoma preoperative chemotherapy	Takagi-Sugeno (T-S) fuzzy neural network with random cooperative decomposing particle swarm optimization double mutation mechanism enhancement learning algorithm (RCDPSO_DM)

Gang Du et al proposed the use of the Takagi-Sugeno (T-S) fuzzy neural network with a novel hybrid learning algorithm for handling variances in liver poisoning of the osteosarcoma preoperative chemotherapy clinical pathway. The proposed method integrates the random cooperative decomposing particle swarm optimization algorithm (RCDPSO) and the discrete binary version of the PSO algorithm to optimize the structures and parameters of the T-S fuzzy neural network. The fuzzy neural network based variance handling method recommends the dosage of liver protection drugs based on two lab tests (Alanine Aminotransferase and Aspartate AminoTransferase) and the patient experience index. Even though the proposed method successfully improves the optimization performance, it still has a premature convergence and low precision that has not been solved completely. With the implementation of a double mutation in RCDPSO, the aforementioned problems are resolved.

Even though recent research has proposed the management of variance of clinical pathways, there is still a lack of a variance handling method that can detect and handle variances at the same time. It is important for clinical pathways to be able to detect and handle variances effectively to provide high quality care to the patient.

METHODS

This research proposed the application of a neural network to predict variance, deviation, or additional procedures for a heart failure clinical pathway. The case study used in this study is based on an Acute Decompensated Heart Failure (ADHF) clinical pathway. Data is taken from National Heart Institute (IJN) in Kuala Lumpur, Malaysia.

Based on the data collected, we have established the main treatment and variance that occurred during acute decompensated heart failure clinical pathway. Table 2 shows the overall number of procedures and treatments that were completed and recorded in the IJN Cardiology ADHF database.

Table 2 Procedure and treatment for heart failure patient taken from IJN databases.

Treatment/Procedure	Number of Treatments Recorded	Percentage
Diuretic Prescription	4138	92.00%
Aspirin Prescription	2671	59.42%
Digoxin	1925	42.82%
Clopidogrel Prescription	1780	39.59%
Dialysis	173	3.80%
PCI	64	1.42%
Angiography	278	6.18%
Cardiac Catheterization	44	0.90%

Based on Table 2, most of the patients have been treated using noninvasive treatment, where the most common treatment was using diuretic drugs. However, some of the patients required alternative or additional treatments and procedures for their condition. These additional procedures included Angiography and Cardiac Catheterization, and additional treatments included PCI and Dialysis treatment.

Dialysis is a process involving removing waste and excess water from the blood, and is used for those who have lost kidney function or for those with acute disturbance in kidney function. Percutaneous Coronary Intervention (PCI) is non-surgical revascularization technique used for treating obstructive coronary artery diseases, including unstable angina, multi vessel coronary disease (CAD), and acute myocardial infarction (MI). Angiography is a medical imaging technique used to visualize blood vessels, which is mainly used to visualize arteries, veins, or heart chambers. Cardiac catheterization (Card Cath) is a heart examination procedure to find out how well your heart is working. This research aims to predict these three additional treatments and procedures using an artificial neural network.

ANN framework for ADHF variance prediction

In this section, we briefly introduce the basic neural network concept for clinical pathway variance prediction. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. ANN has been implemented in various fields. In healthcare, ANN is implemented for clinical diagnosis, drug development, image analysis, and signal analysis. ANN has proven to be useful for modeling complex relationships between inputs and outputs, or to find patterns in data. Others advantages of ANN are:

1) Requires less formal statistical training to develop
2) Able to recognize complex non-linear relationships between independent and dependent variables
3) Capable of discovering all possible interactions between predictor values
4) Can be developed using different training algorithms.

Even though ANN is a powerful tool for prediction and has been widely used, it still depends on a trial and error process in order to obtain the successful model. Until now, there were no clear rules on how to obtain the best and most successful model of ANN. Since ANN is fully dependent on a trial and error process, there were 5 influencing factors that contributed to the best ANN model. These influencing factors are network structure, network algorithm, transfer function, training function, and performance function. These factors can be summarized as a neural network configuration.

Furthermore, data setup can also influence the effectiveness of the ANN model. In data setup, the collected raw data will be processed before being used for the ANN model. This process is also known as data preprocessing. After data processing is completed, the dataset will be divided into training and test datasets.

Data Setup

Data used for this research was collected from the Cardiology Medical Record (Acute Decompensated Heart Failure) Database in the National Heart Institute (IJN) of Malaysia. The data collected consists of 4495 patients between the period of 1st January 2009 – 22nd December 2015. Data is taken from the IJN Cardiology Medical Record Acute Decompensated Heart Failure (ADHF) Database. Based on the data analysis, we have determined 55 inputs and 3 outputs for the ANN model. Table 3 shows the inputs and outputs of the proposed neural network for ADHF variance prediction.

After the data is obtained, data preprocessing is conducted. Data cleaning and transformation are primarily used for data preprocessing in our dataset. Data cleaning involves filling in missing values in the dataset, smoothing the noisy data, and resolving inconsistencies in dataset. Missing values are the main issue in our dataset, where the dataset contains a lot of missing data resulting in misplaced data and human error. These missing values is filled using equation (1).

$$\mathbf{X} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{b_i} = \frac{1}{n}\left(\boldsymbol{b_1} + \boldsymbol{b_2} + \cdots + \boldsymbol{b_n}\right) \tag{1}$$

Where X is the value of missing data, n is the number of attributes in missing data classes, and b is the value of individual attributes in the missing data classes.

Data transformation techniques are applied after the data cleaning process is completed. Data transformation consolidated data into forms that are appropriate for the use of the neural network model. The raw data consisted of string and date/time data that the neural network cannot process. So these data need to be transformed to numerical or Boolean data types. Table 4 shows the examples of data transformation for the ANN dataset.

Table 3 Inputs and outputs of proposed variance prediction using ANN.

INPUT	OUTPUT
1. Gender	1. Dialysis Procedure
2. Age	2. PCI Procedure
3. Weight	3. Angiography Procedure
4. Height	
5. Smoking Habits	4. Cardiac Catheter Procedure
6. Previous Heart Failure Hospitalization	
7. Pre Hospital LVEF	
8. Coronary Artery Disease	
9. Previous PCI Procedure	
10. Previous CABG Procedure	
11. Previous MI	
12. Renal Insufficiency	
13. Creatinine more than 37	
14. Regular Dialysis	
15. Atrial Fibrillation	
16. Diabetes	
17. Hypertension	
18. Hyperlipidemia/Dyslipidemia	
19. Stroke	
20. COPD/Asthma	
21. Dyspnea	
22. Peripheral Edema	
23. Ascites	
24. Lung Crepitation	
25. Elevated JVP	
26. Hepatomegaly	
27. Admission Systolic BP	
28. Admission Diastolic BP	
29. Admission Heart Rate	
30. Admission Respiratory Rate	
31. ECG Procedure Done	
32. Rhythm ID	
33. Q Wave	
34. ST Segment Depression	
35. ST Segment Elevation	
36. No Infarction Evidence	
37. QRS Duration	
38. X-ray Procedure	
39. Cardiomegaly	
40. Pleural Effusion	
41. Congestion	
42. Ill-defined Opacity	
43. Acute Pulmonary Edema	
44. Urea Level	
45. Sodium Level	
46. Potassium Level	
47. Creatinine Level	
48. Uric Acid Level	
49. Random Blood Sugar Test	
50. Bilirubin Level	
51. Albumin Level	
52. CK Level	
53. Hemoglobin Test	
54. Admission LVEF	
55. LVEF not done	

Table 4 Data transformation examples.

Input Classes	Transformed Data
Gender	Male → 0, Female → 1
Previous MI (Myocardial Infarction)	Does not have → 0, Have → 1
Date of birth → Age	04-12-88 → 28

After data processing is completed, the dataset needs to be divided into training and test datasets. The data division process is an important part of evaluating neural network models. The training dataset is used by neural network models to discover potentially predictive relationships. The test dataset is used to evaluate the performance of the neural network model. This research used the k-fold cross validation technique. K-fold cross validation randomly partitioned into k equal size of subsamples. For this research, we use k value that used is 10 which means the dataset is partition into 10 samples. Of 10 partitions, a single partition of samples is used as test data to validate the neural model, while the other 9 samples are used as training data. This process is repeated 10 times with each of the partitions used exactly once as the test data.

NEURAL NETWORK CONFIGURATION

The feed forward neural network consists of three main layers, which are input layer, hidden layer, and output layer. Input and output usually consist of 1 layer, and the hidden layer could consist of a minimum of 1 layer. Fig. 1 shows examples of feed forward neural network architecture. The number of input nodes and output nodes depends on the collected data, while the numbers of hidden nodes for ANN are based on trial and error.

A guideline by Zhang, Patuwo and Hu (1998) recommended the number of hidden nodes according to “n/2”, “1n”, “2n”, and “2n+1” where *n* is the number of input nodes. Since the number of input nodes in this research is 61, the number of hidden nodes that will be used are 32, 61, 122, and 123, respectively. To limit the trial and error process, the number of hidden layers used for this research is 1. Table 5 shows the structures of neural networks used in this research.

Table 5 Proposed neural network structures.

Input	Hidden Layer	Hidden Nodes	Output	Network Structures
55	1	27	4	55-27-4
		55		55-55-4
		110		55-110-4
		111		55-111-4
	2	27		55-27-27-4
		55		55-55-55-4
		110		55-110-110-4
		111		55-111-111-4

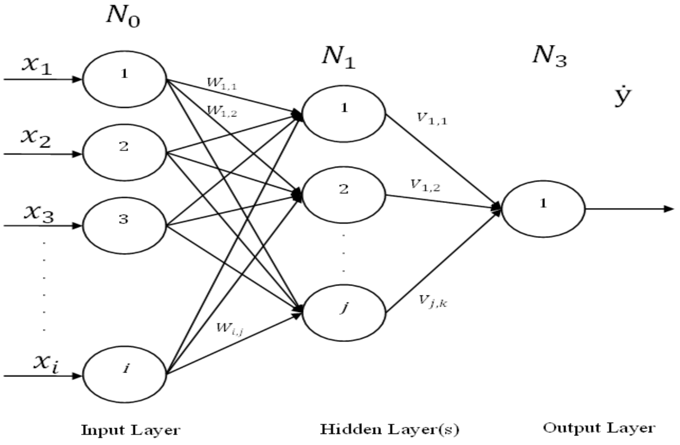

Fig. 1 Feed Forward Neural Network Architecture

Multi-layer feed forward neural networks are the most commonly used algorithms in medical diagnosis. The feed forward neural network can be described in equation (2).

$$\dot{y}_l = f\sum_i w_{i,j}\,x_i + b_i \qquad (2)$$

Where $\dot{y}_j$ is the output of the network, *f* is the transfer function, w_{ij} is the weight, x_i is the input, and b_i is the bias. From equation (2), the multi-layer feed forward neural network with 1 hidden layer can be further derived as:

$$\dot{y}_l(n) = f_k(\sum_{k=1}^{N_1} V_{j,k} f_j(\sum_{i=1}^{N_0} W_{i,j} x_i(n) + b_i) + b_j) \quad (3)$$

Where, ẏ is the output of the neural network, *f* is the transfer function of the neural network, *N* is the number of nodes in the respective layer (*N_0* is the number of nodes in the input layer and *N_1* is the number of nodes in the 1st hidden layer), *$V_{j,k}$* is the weight of the neural network between the hidden layer and output layer, *$W_{i,j}$* is the weight from the input layer to the hidden layer, $x_i(n)$ is the input of the neural network, and *b* is the bias of the neural network.

The most commonly used transfer function in neural networks is the sigmoid function. For this research we used the sigmoid transfer function. The sigmoid transfer function (log-sigmoid) can be represented as equation (4):

$$f = \frac{1}{1 + e^{-n}} \quad (4)$$

Where *n* is the output of the hidden layer.

The values of weight (*Wij*, *Vjk*, *Zkl*) and bias (*bi*, *bj*, *bk*) are iteratively changed by the training function in order to achieve the best prediction accuracy. Weight and bias are adjusted by the training function based on the error produced by the network. This error value could be obtained using the performance function. There were several performance functions that could be used in neural networks. We used MSE (Mean Squared Error) and Cross Entropy performance functions in this research. For multi-class classification problems, cross entropy is widely used as a neural network performance function. However, some of the training functions that use the Jacobian based matrix calculation (e.g. Lavenberg-Marquadt and Bayesian Regulization) used MSE or SSE (Sum Squared Error) for weight and bias calculations. Cross Entropy produces error values that heavily penalize outputs that are extremely inaccurate (*ẏ* near 1-*t*), with very little penalties to a fairly correct classification/prediction (*ẏ* near *t*). Minimizing cross entropy and MSE values leads to a good neural network model. Cross Entropy, using equation (5), produced error values and MSE using equation (6).

$$error = E = -t_l \bullet \log \dot{y}_l \quad (5)$$

$$error = E = \frac{1}{2}(t_l - \dot{y}_l)^2 \quad (6)$$

Where t_l is the output target and $\dot{y}_l$ is the neural network predicted output. After the error value is computed, the training algorithm will adjust the weight and bias of the neural network based on the error value. The training functions that are often used by researchers in the field of medical diagnosis are back propagation algorithms. Back propagation algorithm is a learning function and commonly used method for training neural networks, where it uses gradient descent algorithms that minimize squared error. Squared errors are minimized by using an iterative process of gradient descent. Gradient descent can be expressed in equation (7):

$$g_i = \frac{\partial E_i}{\partial W_i} \quad (7)$$

Where g_i is the gradient of the *i*th iteration, *E* is the error of the network for the *i*th iteration, and *W* is the weight and biases of the *i*th iteration. Weight and biases are updated in the direction of network error (performance function) decreases most rapidly (negative of gradient) using equation (8).

$$U_{i+1} = U_i - \eta g_i \quad (8)$$

Where U_i is a vector of current weight and biases, g_i is the current gradient, and η is the learning rate (proportional parameter which defines the step length of each iteration in the negative gradient descent). The learning rate value is defined by the user, where the small value of η could lead to a true approximation or prediction while slowing the learning process. However, choosing a larger value of η could speed up the neural network convergence, which may cause oscillation in the weight spaces. Basically, back propagation works as follows:

1. The neural network is given input x and the error of the network network is calculated,
2. Sensitivities (ΔW_i and Δb_i) are propagated from the output layer to the first layer, and
3. The weight w and biases b of the neural network are updated.

Back propagation uses the chain rule in order to compute derivatives of the squared error with respect to the weights and biases in the hidden layers.

This algorithm is called back propagation because the derivatives are computed first in the last layer of the network and then propagated backwards through the network to compute the derivatives in the hidden layer. However, there are several variations of the back propagation algorithm. The variations of the back propagation training algorithm include:

1. Lavenberg-Marquadt
2. Bayesian Regulation back propagation.
3. BFGS quasi-Newton backpropagation
4. Resilient Back propagation
5. Scaled Conjugate Gradient
6. Conjugate Gradient with Powell/Beale Restarts
7. Fletcher-Reeves Conjugate Gradient
8. Polak-Ribiére Conjugate Gradient
9. One Step Secant
10. Gradient descent back propagation
11. Gradient descent with adaptive learning rate back propagation
12. Gradient descent with momentum
13. Gradient descent with momentum and adaptive learning rate back propagation

This research used several variations of back propagation algorithms, including the gradient descent method (Gradient Descent with Momentum, Gradient Descent with Momentum and adaptive learning rate), the Conjugate gradient method (Scaled Conjugate), and the Quasi-Newton method (Lavenberg-Marquadt).

Gradient descent with momentum

As mentioned previously, the low value of η produces a better prediction/approximation but will slow the neural network convergences, while a larger value of η will cause the neural network to converge faster but will result in oscillation in the weight spaces. These problems can be resolved by introducing a momentum constant. The gradient descent with momentum constant (*m*) can written as:

$$U_{i+1} = U_i - \eta g_i + m \quad (9)$$

Gradient descent with momentum and adaptive learning rate

Gradient Descent with momentum and adaptive learning rate is the extension of the gradient descent with momentum algorithm, where the learning rate is adaptively changed by the neural network instead of being specified by the users. The learning rates are changed according to these 3 rules:

If the squared error of the training set is increased by more than the specified percentage ζ (which is usually between 1-5 percent) after the weight is updated, then the weight will be discarded. The learning rate then will be multiplied by some factors p (usually between the values of 0-1) and the momentum constant will be set to zero.

If the squared error is decreased after the weight update, then the weight is accepted. The learning rate then will be multiplied by some factors $\eta > 1$ and if the momentum constant is previously set to zero, it is reset to its original value.

If the square error increases by less than ζ, then the weight update is accepted. The learning rate and momentum constant are unchanged.

Levenberg-Marquadt

The Levenberg-Marquadt was designed to minimize functions that are sums of squares of other nonlinear functions, and is suited for neural network trainings that use the mean squared error as the performance index. When the performance function uses the mean squared error or sum squared error, the Hessian matrix can be approximated as:

$$H = J^T J \tag{10}$$

And the gradient can be computed as:

$$g = J^T E \tag{11}$$

Where J is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases, and E is a vector of network error. The vector of current weights and biases can be updated using this approximation:

$$U_{i+1} = U_i - [J^T J + \mu I]^{-1} J^T E \tag{12}$$

Parameter μ is adaptively changed until it reduces the network error. The μ value will decrease after each successful iteration that results in a reduction of network performance function and is only increased when a tentative step would increase the value of the performance function.

The scaled conjugate gradient algorithm is based on conjugate directions instead of a local gradient. Conjugate gradient back propagation typically involves 4 steps:

1. Select the first search direction p_0 to be the negative of the gradient:

$$p_0 = -g_0 \tag{13}$$

2. The line search is then performed using equation (9) to determine the optimal distance to move along the current search direction, where α_k is used as a positive scalar which determines the scale of the step size taken by the function

$$L_{k+1} = L_k + \alpha_k p_k \tag{14}$$

3. Select the next search direction using:

$$p_k = -g_k + \beta_k p_{k-1} \tag{15}$$

The value of β can be calculated using several functions:

$$\beta_k = \frac{\Delta g_{k-1}^T g_k}{\Delta g_{k-1}^T p_{k-1}} \; Or \; \beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \; Or \; \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T p_{k-1}} \tag{16}$$

4. If the algorithm is not converged, continue from Step 2.

Development of neural network model

This research proposed a method for ADHF clinical pathway variance prediction using an ANN model. The method used to develop ANN based variance prediction of ADHF clinical pathways can be summarized in Fig. 2. The data collection process was conducted at the National Heart Institute (IJN). After the data collection process, data analysis was conducted to identify variance and input data. Then, data preprocessing was conducted, along with the determination of neural network structures. The neural network experiment is conducted after the datasets and neural network structures are finalized. The experiment was conducted using the MATLAB R2014b on the computer with the following specifications: Intel Core i-7 with 2.60 GHz, RAM 12.0 GB, and a 64-bit processing system. The selection of the transfer function, training function, and performance function has been discussed in the Neural Network Configuration section. After the experiment has been conducted, the results will be compared to find the best ANN model for ADHF clinical pathway variance prediction. The main criteria to determine the best ANN model are the lowest fitness function and prediction accuracy. The fitness function, used for evaluation, is the same as the performance function used by the network to update the weights, which are Cross Entropy and MSE. The summary of the neural network configuration used in our research is shown Table 6.

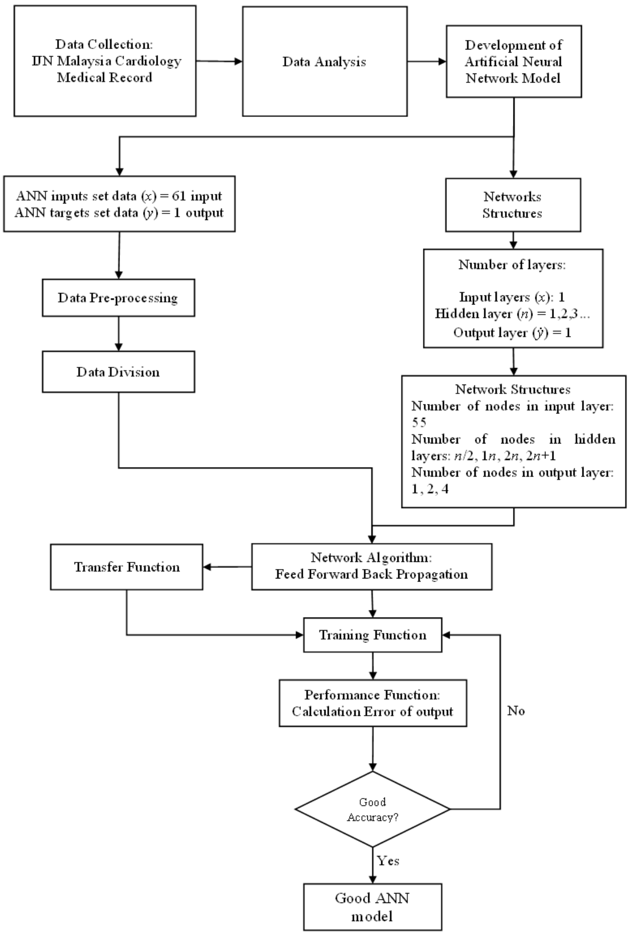
Data Collection:
IJN Malaysia Cardiology Medical Record
Data Analysis
Development of Artificial Neural Network Model
ANN inputs set data (x) = 61 input
ANN targets set data (y) = 1 output
Networks Structures
Number of layers:
Input layers (x): 1
Hidden layer (n) = 1,2,3...
Output layer (ŷ) = 1
Data Pre-processing
Data Division
Network Structures
Number of nodes in input layer: 55
Number of nodes in hidden layers: n/2, 1n, 2n, 2n+1
Number of nodes in output layer: 1, 2, 4
Transfer Function
Network Algorithm:
Feed Forward Back Propagation
Training Function
Performance Function:
Calculation Error of output
No
Good Accuracy?
Yes
Good ANN model

Fig. 2 ANN-based model development flowchart for ADHF clinical pathway variance prediction.

Table 6 ANN structures used for ADHF variance prediction.

<table>
<tr><th colspan="6">NETWORK STRUCTURES</th></tr>
<tr><td colspan="6">55-27-4</td></tr>
<tr><td colspan="6">55-55-4</td></tr>
<tr><td colspan="6">55-110-4</td></tr>
<tr><td colspan="6">55-111-4</td></tr>
<tr><td colspan="6">55-27-27-4</td></tr>
<tr><td colspan="6">55-55-27-4</td></tr>
<tr><td colspan="6">55-110-27-4</td></tr>
<tr><td colspan="6">55-111-27-4</td></tr>
<tr><th colspan="6">TRAINING FUNCTION</th></tr>
<tr><td rowspan="3">Lavenberg-Marquadt</td><td rowspan="3">Scaled Conjugate</td><td colspan="2">Gradient Descent with momentum</td><td colspan="2">Gradient Descent with momentum and adaptive learning rate</td></tr>
<tr><td>Learning Rate (η)</td><td>Momentum (α)</td><td>Initial Learning Rate (η)</td><td>Momentum (α)</td></tr>
<tr><td>0.01 - 0.9</td><td>0.1 - 0.9</td><td>0.01 - 0.9</td><td>0.1 - 0.9</td></tr>
<tr><th colspan="6">TRANSFER FUNCTION</th></tr>
<tr><td colspan="3">Sigmoid – Logsig (Hidden Layer)</td><td colspan="3">Softmax (Output Layer)</td></tr>
<tr><th colspan="6">PERFORMANCE FUNCTION</th></tr>
<tr><td colspan="3">Mean Squared Error (MSE)</td><td colspan="3">Cross Entropy</td></tr>
</table>

Neural network evaluation

The mean squared error (MSE) and prediction accuracy are used to evaluate the performance of the proposed neural network model. A good neural network model can be achieved when the MSE is low and the prediction accuracy is high. MSE can be calculated using equation (4). Prediction accuracy can be computed using equation (17).

$$Accuracy = \left(1 - \frac{m}{s}\right) x\ 100\% \qquad (17)$$

Where m is the number of misclassified samples and s is the number of data samples.

EXPERIMENTAL RESULTS

Experimental result 1:

Table 7 Training function: Scaled conjugate BP (trainscg).

Architecture	Training (%)	Test (%)	Cross Entropy (Training)	Cross Entropy (Test)
55-27-4	52.38	47.24	0.831	0.865
55-55-4	53.81	45.79	0.833	0.894
55-110-4	52.19	44.99	0.833	0.898
55-111-4	55.59	48.03	0.84	0.91
55-27-27-4	**55.73**	**50.8**	**0.839**	**0.868**
55-55-55-4	52.46	46.04	0.836	0.872
55-110-110-4	50.52	44.35	0.836	0.889
55-111-111-4	53.35	44.88	0.83	0.9

Table 8 Training Function: Gradient descent with momentum BP (traingdm).

Architecture	Training (%)	Test (%)	Cross Entropy (Training)	Cross Entropy (Test)
55-27-4	54.6	47.62	0.828	0.87
55-55-4	**55.07**	**43.95**	**0.821**	**0.903**
55-110-4	51.7	43.03	0.834	0.912
55-111-4	44.07	40.39	1.018	1.031
55-27-27-4	53.78	48.42	0.84	0.868
55-55-55-4	53.39	47.11	0.835	0.88
55-110-110-4	41.182	45.8	0.872	0.891
55-111-111-4	56.84	46.99	0.874	0.931

Table 9 Training function: Gradient descent with momentum and adaptive learning rate BP (traingdx).

Architecture	Training (%)	Test (%)	Cross Entropy (Training)	Cross Entropy (Test)
55-27-4	54.14	47.76	0.832	0.87
55-55-4	47.94	39.47	0.871	0.931
55-110-4	49.02	40.65	0.863	0.922
55-111-4	48.89	43.41	0.943	0.988
55-27-27-4	**53.64**	**48.94**	**0.841**	**0.867**
55-55-55-4	53.54	46.04	0.843	0.884
55-110-110-4	43.3	40.37	0.888	0.91
55-111-111-4	43.91	40.93	0.942	0.977

Table 10 Training function: Levenberg-Marquardt (trainlm).

Architecture	Training (%)	Test (%)	MSE (Training)	MSE (Test)
55-27-4	55.02	48.94	0.303	0.319
55-55-4	53.97	43.93	0.313	0.318
55-110-4	58.6	44.74	0.306	0.315
55-111-4	57.28	46.19	0.299	0.322
55-27-27-4	**57.78**	**50.92**	**0.30**	**0.32**
55-55-55-4	55.55	49.2	0.308	0.325
55-110-110-4	57.74	45.93	0.302	0.322
55-111-111-4	54.95	45.39	0.304	0.326

The objective of this research was to find the best neural network model for ADHF clinical pathway variance prediction. There were 32 different models of neural networks compared, which differ in training functions and network structures. Initial results show that the neural network with a 55-27-27-4 architecture produces the best results in the 3 of 4 training functions used. The Levenberg-Marquadt training function produces the best prediction results among the other 4 training functions.

However, initial results show the best overall prediction result (Lavenberg-Marquadt) only achieved 57.78% for the training dataset and 50.92% for the test dataset. The low prediction rate may be caused by the fact that several patients may have several variances in their treatment. In order to increase the neural network prediction rate, we changed the neural network structures in our experimental setup. We using the best neural network model obtained in our initial experiment and changed the output nodes to 1 output, where each output class is trained and tested separately. The network configuration used in the second experimental setup was:

1. Network Structures = 55-27-27-1
2. Training Function = Levenberg-Marquadt (trainlm)
3. Transfer Function = Sigmoid – Logsig (Hidden Layer) and Satlins (Output Layer)
4. Performance Function = MSE

Experimental result 2:

Table 11 Dialysis prediction results.

X Fold	Training (%)	Test (%)	MSE (Training)	MSE (Test)
1	93.2	86.6	0.66	0.11
2	91.5	84.2	0.07	0.11
3	90	76.3	0.07	0.17
4	96.4	88.2	0.03	0.09
5	91.9	85.5	0.07	0.12
6	92.2	86.8	0.06	0.1
7	93.5	94.7	0.06	0.07
8	92.2	94.7	0.06	0.05
9	92.2	94.7	0.06	0.05
10	91.5	94.7	0.06	0.03
Average	**92.46**	**88.64**	**0.12**	**0.09**

Table 12 PCI prediction results.

X Fold	Training (%)	Test (%)	MSE (Training)	MSE (Test)
1	92.4	92.1	0.07	0.07
2	93.2	84.2	0.07	0.15
3	92.1	94.7	0.07	0.05
4	91.5	97.4	0.08	0.25
5	92.6	89.5	0.07	0.1
6	92.2	93.4	0.06	0.06
7	92.4	92.1	0.07	0.08
8	92.4	92.1	0.08	0.08
9	92.4	92.1	0.07	0.07
10	91.9	96.1	0.08	0.03
Average	**92.31**	**92.37**	**0.072**	**0.094**

Table 13 Angiography prediction results.

X Fold	Training (%)	Test (%)	MSE (Training)	MSE (Test)
1	69.1	69.7	0.20	0.21
2	70.6	72.4	0.22	0.24
3	65.9	60.5	0.22	0.23
4	74.6	65.8	0.16	0.23
5	65.7	64.5	0.22	0.22
6	70.6	65.8	0.20	0.21
7	74.7	60.5	0.17	0.25
8	77.3	65.8	0.16	0.20
9	66.8	57.9	0.16	0.30
10	64.6	73.7	0.30	0.26
Average	**69.99**	**65.66**	**0.201**	**0.235**

Table 14 Cardiac catheter prediction results.

X Fold	Training (%)	Test (%)	MSE (Training)	MSE (Test)
1	94.7	96.1	0.04	0.03
2	96.4	98.7	0.03	0.01
3	94.5	93.4	0.04	0.03
4	94.5	97.4	0.04	0.02
5	94.7	96.1	0.04	0.04
6	94.8	94.7	0.04	0.05
7	94.5	97.4	0.04	0.03
8	95.4	89.5	0.03	0.09
9	94.9	93.4	0.04	0.07
10	95.7	86.8	0.03	0.11
Average	**95.01**	**94.35**	**0.037**	**0.048**

Table 15 Overall results.

Variances	Training (%)	Test (%)	MSE (Training)	MSE (Test)
Angiography	69.99	65.66	0.201	0.235
Cardiac Catheter	95.01	94.35	0.037	0.048
Dialysis	92.46	88.64	0.12	0.09
PCI	92.31	92.37	0.072	0.094
Average	**87.4425**	**85.255**	**0.1075**	**0.11675**

Discussion

The accuracy of variance prediction improved significantly by changing its structure. The overall prediction rate increased to 87.445% from 57.78% for the training dataset, and increased from 50.92% to 85.255% for the test dataset. It is shown that network structures play a significant role in improving the prediction accuracy. Generally, artificial neural networks performed well in predicting 3 of 4 cases of variance in the ADHF clinical pathway. The artificial neural network can predict variance cases of Cardiac Cathether, PCI, and Dialysis with high accuracy (around 90% accuracy). However, for the case of angiography, the prediction results still did not achieve the desired performances.

CONCLUSION

In this paper, the methodology of variance prediction for ADHF clinical pathway using an artificial neural network is presented. The best model of neural network has been obtained and presented. Future works will involve the combination of fuzzy logic with proposed neural network models for the improvement of classified results. Furthermore, the application of feature selection techniques could be used to increase prediction accuracy.

ACKNOWLEDGEMENT

This research was funded with a Research University Grant (RUG) with vote no (PY/2017/00781) with a collaborative effort between the University Technology of Malaysia (UTM) and the National Heart Institute (IJN) Malaysia.

REFERENCES

Beale, Hudson, M., Hagan, M. T., Demuth, H. B. 2012. Neural network toolbox™ user's guide. In R2012a, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098,, www. mathworks. com. 2012.

Buerkle, T., Baur, T., Hoess, N. 2006. Clinical pathways development and computer support in the EPR: Lessons learned. In Hasman, A., Haux, R., Van Der Lei, J. (Eds.) Ubiquity: Technologies for Better Health in Aging Societies *Proceedings of MIE 2006*. 27–30 August. Bathmen, NDL: IOS Press, Amsterdam, 1025–1030.

Chu, S., Cesnik, B. 1998. Modelling computerized clinical pathways, In. Cesnik, B., Alexa, T., McCray, J., Scherrer, R. (Eds.). *Proceedings of Medinfo'98.* 17–22 August, Seoul, Korea: IOS Press: 559–563.

Chu, S., Cesnik, B. 1998. Improving clinical pathway design: Lessons learned from a computerized prototype. *International Journal of Medical Informatics 51*, 1–11.

Chu, S. 2001.Computerized clinical pathway management systems and the implications. *Collegian, 8*(2), 19–24.

DiJerome, L. 1992. The nursing case management computerized system: Meeting the challenge of health care delivery through technology. *Computers in Nursing*, *10*(6), 250-258.

Du, G., Jiang, Z., Diao, X., Yao, Y. 2013. Intelligent ensemble T–S fuzzy neural networks with RCDPSO_DM optimization for effective handling of complex clinical pathway variances. *Computers in Biology and Medicine, 43*(6), 613-634.

Filippo, A., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., Havel, J. 2013. Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*, *11*(2), 47-58.

Gang, D., Jiang, Z., Diao, X., Ye, Y., Yang, Y. 2012. Variances handling method of clinical pathways based on ts fuzzy neural networks with novel hybrid learning algorithm. *Journal of Medical Systems 36*(3), 1283-1300.

Gang, D., Jiang, Z., Diao, X., Yao, Y. Knowledge extraction algorithm for variances handling of CP using integrated hybrid genetic double multi-group cooperative PSO and DPSO. *Journal of Medical Systems, 36*(2), 979-994.

Guoqiang, Z., Patuwo, B. E., Hu, M. Y. 1998. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting 14*(1), 35-62.

Hagan, M. T., Demuth, H. B., Beale, M. 1997. *Neural Network Design*. Boston, MA: PWS Publishing Co.

Hao, A.T., Hsu, C.Y., Li-Fang, H., Jian, W.S., Wu, L. B., Kao, C.C., *et al.* 2006. Apply creative thinking of decision support in electrical nursing record. *Studies in Health Technology and Informatics, 124*, 313–319.

Hao, A. T., Huang, L. F., Wu, L. B., Kao, C.C., Lu, M. S., Jian, W.S., Chang, H. K., Hsu, C. Y. 2006. Building an innovation electronic nursing record pilot structure with nursing clinical pathway. *Studies in Health Technology and Informatics 122*, 279–283.

Hofmann, P. A. 1993. Critical path methods. An important tool for coordinating clinical care. *Journal on Quality Improvement*, *19*, 235-246.

Hurley. K. F., Abidi, S. S. R. 2007. Ontology engineering to model clinical pathways towards the computerization and execution of clinical pathways. *20th IEEE International Symposium on Computer-Based Medical System*. 20-22 June. Maribor: IEEE, 536-541.

Hyett, K. L., Podosky, M., Santamaria, N., Ham, J. C. 2007. Valuing variance: The importance of variance analysis in clinical pathways utilisation. *Australian Health Review, 31*(4), 565-570.

Møller, Fodslette, M. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks, 6*(4), 525-533.

Okada, O., Ohboshi, N., Uroda, T., Nagase, K. Yoshihara, H. 2005. Electronic clinical path system based seistructured data model using personal digital assistant for onsite access. *Journal of Medical Systems, 29*(4), 379–389.

Remzi, M., Anagnostou, T., Ravery, V., Zlotta, A., Stephan, C., Marberger, M., Djavan, B. 2003. An artificial neural network to predict the outcome of repeat prostate biopsies. *Urology*, *62*(3), 456-460.

Tu, J. V. 1996. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiol*, *49*, 1225-1231.

Wakamiya, Shunji, Yamauchi, K. 2006. A new approach to systematization of the management of paper-based clinical pathways. *Computer Methods and Programs in Biomedicine, 82*(2), 169-176.

Watarai, Y., Kano, N., Fukuma, E., Kameda, T., Bushi, A., Kuroki, M., *et al.* 2001. Effectiveness of navigation care map. *Geka Chiryo 85*(3), 271–277. (in Japanese).

Xiang, L., Liu, H., Zhang, S., Mei, J., Xie, G., Yu, Y., Li, J., Lakshmanan, G. T. 2013. Automatic variance analysis of multistage care pathways. *Studies in Health Technology and Informatics, 205*,715-719.

Yan, Y., Jiang, Z., Diao, X., Du, G. 2009. A knowledge-based variance management system for supporting the implementation of clinical pathways. *International Conference on Management and Service Science*. 20-22 September. Wuhan, 1-4.

Ye, Y., Jiang, Z., Diao, X., Du, G. 2008. A semantics-based clinical pathway workflow and variance management framework. *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*. 12-15 October. Beijing: IEEE, 758-763.

Zhen, H., Jing-Song, L., Hai-Yan, Y., Xiao-Guang, Z. 2009. Modeling of clinical pathways based on ontology. 14-16 August. *IEEE International Symposium on IT in Medicine & Education (ITME)*. Jinan: IEEE, 1170-1174, August 2009.