A Practical Comparison among Neural Networks, Bayesian Networks, and Collaborative Filtering in Classifying Diabetes Mellitus Patients

A Special Problem

Presented to the Faculty of

the Institute of Computer Science

University of the Philippines Los Baños

In partial fulfillment

of the requirements for the Degree of

Bachelor of Science in Computer Science

By

Rachelle G. Bondad

October 2013

ACKNOWLEDGEMENT

First of all, I would like to thank God Almighty for always being there for me; for giving me light during times I feel like I can't accomplish the tasks, and for giving me the strength to push and believe that I can finish this research.

Secondly, I'd like to thank my family especially my parents, Gilda and Francisco E. Bondad for bringing me up in this world and for giving me the opportunity to study in this prestigious school, though it meant they have to work harder to fulfill all my student needs.

I would also like to acknowledge my sister from another mother, Michelle Caramay, for giving me just the escape I needed when things in my SP got stressful, and for all the happy moments you and your family shared/ would have shared to me. Thank you.

To my friends who have always been there from the start until now, encouraging and supporting me, I offer to you my sincerest gratitude.

Lastly, I would like to thank my adviser, Prof. Jaderick Pabico, for sharing to me his every knowledge about my topic, and for always helping me solve the problems I encountered as I go along with my SP.

ABSTRACT

Machine Learning is a study of systems that allows learning and prediction based from a data. Implementing Machine Learning for medical purposes is one of its useful and important applications. Diabetes Mellitus is a major health concern worldwide. This paper presents a way to improve data evaluation on Diabetes Mellitus by using different machine learning approaches, namely: Neural Networks, Bayesian Networks, and Collaborative Filtering.

TABLE OF CONTENTS

I. INTRODUCTION

Medical Informatics, a study of structure and properties of medical data, is a very important field since it deals with human health. It is a discipline that aims to improve communication, understanding and management of medical information by combining the acquired data and knowledge, using different tools to arrive with a decision [1]. In some cases, there are vast amount of medical data but only few theories available. These cases can be subject to extensive application of computational intelligence methods such as statistical learning systems, fuzzy systems, neural networking and many other machine learning approaches.

Diabetes Mellitus (DM) is a common and worldwide medical concern, thus making the prediction of its occurrence as an interest for many. The disease is characterized by having abnormally high levels of blood sugar (glucose) either because of low production of insulin or the body's cell does not respond properly to insulin, or both [2]. It causes serious complications such as cardiovascular diseases, blindness, kidney failure and others. DM affects approximately 350 million people worldwide and is projected to double between year 2005 and 2030 according to the World Health Organization (WHO).

Before DM can be suspected, a physician performs several medical examinations on the patient. These suspicions will only be confirmed by conducting laboratory observation in patient’s blood sample. This is the traditional way of predicting DM patients. The patient had lost a lot of money just in the process of knowing what ailment he has gotten. With the kind of technology nowadays, people are in search for a more convenient way to classify Diabetes Mellitus patients.

Previous attempts to classify Diabetes Mellitus patients commonly focused on a certain technique only, like the works of Adeyemo and Akinwonmi [3], and Shanker [4] which

concentrated on ANN in predicting Diabetes Mellitus patients. However, this study will be a comparison of different intelligence methods such as Neural Networks, Bayesian Networks, and Collaborative Filtering.

The study will be useful in deriving improvements in the areas of data evaluation on the characteristics and prevention of Diabetes Mellitus. Collected data from patients suffering from DM over the years can now be analyzed for rapid diagnosis of the disease. It also intends to offer a new standard trend in medicine by diagnosing future patients with the same disease state. It will assist the physician in the diagnosis process by evaluating the symptoms. Early detection of Diabetes prevents or delays the onset of the disease and its long-term complications, reducing casualties since patients are to receive proper and adequate treatment.

II. REVIEW OF RELATED LITERATURE

Classification can be achieved in many ways. The focus of this study is to classify Diabetes Mellitus patients using Neural Networks, Bayesian Networks and Collaborative Filtering.

A. Neural Networks

Neural networks, also known as Artificial Neural Networks (ANN), are inspired by biological functions having the ability to learn and provide solution for complex problems. ANN works by 'learning' the data and comparing it to the actual record. It is a directed graph with weighted connections. Classification of networks is based on how these connections were established and how weights are adjusted to minimize the error measure on the response of the network to a stimulant [5].

Neural network functions through a learning process. There are two types of learning: Supervised Learning and Unsupervised Learning. Supervised Learning is performed when training data consists of inputs and a corresponding output. The network adjusts its weight parameter so that it produces minimal error in predicting the output of a sample data [6]. Unsupervised learning has no distinctions between inputs and outputs.

B. Bayesian Networks

Bayesian networks (BN), also known as Belief Networks, belong to the Probabilistic Graphical Models (GM) represented by Directed Acyclic Graph (DAG) of $G = (V;A)$, where V is the node and A is the arc [7]. For example, node X with an arrow going to node Y indicates that node Y depends on the value of node X, making node X as the parent and node Y as the child. A node is made to represent the variables while the edges (arcs) drawn connecting the nodes serve as the probabilistic dependencies of one variable to another. These dependencies are estimated

usually using statistical and computational methods. A BN also reflects conditional independence statement, namely each variable is independent from its non-descendants in the graph given the state of its parents.

C. Collaborative Filtering

Collaborative Filtering (CF) is designed to make recommendations or predictions of the unknown preference of a person based from the known preferences of a group of person [8]. This technique uses intuitive assumption that people will have the same or at least some common preference with his similar peer. Collaborative Filtering is usually used as a marketing strategy. An example is the commercial system Amazon.com which is an online retailer of books, music, games and others. They use Collaborative Filtering for the user-rating data and make predictions or recommendations for the next product for their user to buy.

D. Attempts in comparing different algorithms

Some people focus on a certain classification algorithm while some people put their effort in comparing different algorithms. Endo et al. [9] evaluated seven common algorithms namely Artificial Neural Networks, Naive Bayes, Bayesian Net, Logistic Regression Model, Decision Trees with Naive Bayes, Decision Trees (ID3) and Decision trees (J48) in predicting Breast Cancer survival. They have found out that Logistic Regression Model has the highest accuracy, Decision Trees (J48) has the highest sensitivity and ANN showed the highest specificity. They have also observed that Decision Trees model is more sensitive to survival prediction, as Bayesian Model is to death prediction.

Mantzaris et al. [10] examined a variety of Artificial Neural Network (ANN) models in classifying an orthopedic disease, namely osteoporosis. Multi-layer Perceptrons (MLP) and Probability Neural Networks (PNN) were used to predict the osteoporosis risk factors. Transfer

functions and learning algorithms were considered after modifying the number of nodes in the hidden layer. PNNs were implemented with spread values ranging from 0.1 to 50, and 4 or 2 neurons in the output layer, according to the coding of osteoporosis desired outcome. As a result, PNNs outperformed the MLPs, and are proven as appropriate intelligence method to use in osteoporosis risk factor prediction. Also, it was observed that over fitting problem was more frequent in MLPs as their spread value increased.

E. Previous Attempts to enhance data evaluation on Diabetes

Sapon et al. [11] predicted the occurrence of Diabetes Mellitus patients by using different supervised learning algorithms of ANN. They examined the performance of algorithms such as BFGS Quasi-Newton algorithm, Bayesian Regularization and Levenberg-Marquardt. Based on the analysis, Bayesian regulation gives the most accurate prediction. It produces a regression value of 0.99576 and a prediction accuracy of 88.8% which indicates a good correlation and confirms that this is the most suitable way to predict Diabetes among the three methods.

Adeyemo and Akinwonmi [3] conducted also a study in ANN for the rapid diagnosis of Diabetes Mellitus. They modeled their work using both Classification and Predictive Neural networks. They also modeled it using Multi-Layered Perceptron (MLP), Radial Basis Function (RBF) and Generalized Regression Neural Networks/Probabilistic Neural Netwoks (GRNN/PNN). The performance of ANN was evaluated using Mean Squared Errors (MSE) and the Learning Curve. The classification results show that 88.8% were classified correctly. They also highlighted the importance of each variable as a contributor in classification process. The GRNN/PNN gave the best result while the other two architectures failed to model the problem. The diagnosis test data set achieved an MSE of 0.573684961 while the treatment data set got an MSE of 49.09971153.

Shanker [4] performed a study about ANN in predicting the onset of DM among Pima Indian women. He then compared it with the logistic regression. He used eight variables/inputs and MSE as a criterion. First, he experimented with the variables by dropping them one by one from the model using F-ratio as criterion. Until, in the final model, only three inputs were left giving him the test classification percentage of 80.21%. On the other hand, test classification percentage using logistic regression is 80.21% respectively. In the second procedure, they applied the backward elimination procedure in selecting variables. They sequentially deleted variables that are least significantly (at 0.05 levels) needed in the training sample. The end model has an overall test classification rate of 80.21% while the logistic regression produced a test rate of 78.65%.

III. OBJECTIVES OF THE STUDY

The general objective of the study is to classify Diabetes Mellitus patients. Specifically, the study aims:

1. To implement Neural Networks, Bayesian Networks and Collaborative Filtering in classifying Diabetes Mellitus Patients; and
2. To compare the performances of Neural Networks, Bayesian Networks, and Collaborative Filtering in classifying Diabetes Mellitus patients.

IV. METHODOLOGY

A. Implementing ANN, Bayesian Networks and Collaborative Filtering

The authentic patient health records served as the data set for the study. This data set was obtained from the Practical Fusion, a free Web-based Electronic Health Records (EHR). It originally consists of 86144 patient health records; however, not all of these were used in the implementation. The features present in the data set are Gender, Height, Weight, BMI, Systolic, Diastolic, Respiratory Rate, Temperature, Smoking Status, Allergy and Diabetes Mellitus Indicator (DMI).

1. NEURAL NETWORKS

A typical Neural Network is composed of an input layer, hidden layer and the output layer which are connected by weights. Ten features served as the neurons for the input layer, namely: Gender, Height, Weight, BMI, Systolic, Diastolic, Respiratory Rate, Temperature, Smoking Status and Allergy. Each feature as an input in the classification system can increase the cost and running time of a recognition system [12]. The number of neurons in the input and output layer are usually given in the problem, however, the number of hidden neurons has been far from clear. The number of neurons/nodes in hidden layer was incremented in this study. The output layer consisted of a neuron that served as indicator if patient has Diabetes. Supervised Learning was implemented since the data set already provides the target output which is the DMI.

In this ANN implementation, three data sets, taken from the original data set, were used. Earlier, it is said that there are a total of 86144 records, but only 755 out of those records have a positive DMI, and the rest were already negative. So for the training to be more efficient, we randomly chose 1000 records that have negative DMI from the original data set, and get all

records that have positive DMI. We did this again twice more, getting another set of 1000 records different from the first 1000 that we got and again getting all those that have positive DMI, giving us a total of three data sets with 1755 records each. Then these data sets were subdivided into training set, validation set and test set with a proportion of 70-15-15.

Once the data sets are finalized, and neurons for each layer are identified, the basic feed-forward algorithm can now be implemented using batch training [13].

1. Initialize the network weights with random values between 0 and 1.
2. Present the first training data pattern.
3. Apply the sigmoid function in the weighted sum of inputs for the hidden layer using this formula:

$$z_h = sigmoid\left(\sum_{j=0}^{d} w_{hj}x_j\right) = \frac{1}{1 + e^{-\sum_{j=0}^{d} w_{hj}x_j}}$$

   where w_{hj} is the initialized connection weight to each hidden neuron h going from 1 to H where H is the number of hidden neurons. Also x_j is the input neuron and x_0 is the bias with a weight of w_0. Summation goes from 0 to d where d is the number of input neurons.
4. Using the computed sigmoid function per hidden neuron, we now compute for the network's output using this formula:

$$o = sigmoid\left(\sum_{h=0}^{H} v_h z_h\right) = \frac{1}{1 + e^{-\sum_{h=0}^{H} v_h z_h}}$$

   where v_h is the initialized connection weight to output layer
5. Compare the network output with the target output and get the Mean Squared Error (MSE).

$$err = \frac{(o - O)^2}{2} + err_{t-1}$$

where O is the target output and err_{t-1} is the error of the previous training pattern.

6. Compute the gradient per training data pattern [14].

   (a) Formula in getting the gradient of connection weights to hidden layer

$$\frac{\partial E}{\partial w_t} = (\sum_{h=0}^{H} x_j z_h (1 - z_h) v_h (o - O) o (1 - o)) + \frac{\partial E}{\partial w_{t-1}}$$

   where $\frac{\partial E}{\partial w_{t-1}}$ is the previous gradient.

   (b) Formula in getting the gradient of connection weights to output layer

$$\frac{\partial E}{\partial w_t} = (\sum_{h=0}^{H} z_h (o - O) o (1 - o)) + \frac{\partial E}{\partial w_{t-1}}$$

7. Repeat step3 to step6 until all the training patterns have been inputted to the network.
8. Before correcting the weights, we present first the validation set to the same network and get the validation error. The weights of the epoch with the lowest validation error will be used to train the test set.
9. Compute for the delta using this formula [15].

$$\Delta w_t = \eta(\frac{\partial E}{\partial w_t}) + \alpha(\Delta w_{t-1})$$

   where η is the *learning rate*, α is the *momentum* and Δw_{t-1} is the previous delta. Add the computed delta to the previous weights.
10. Training continues until 1000 epochs, unless the trend of validation error constantly gets higher for hundred epochs, then the training will be stopped.
11. After the end of the training process, the final weights will be applied to the test set and the test error will be computed. Using just the same process as above, these final weights

will be applied to the second data set that would result to another set of final weights, which will again be applied to the third and final data set.

NeuroShell2, software that creates beginners and advanced neural networks, was also used in this study for additional comparison. We performed the beginner's which allowed us to train and apply a simple net structure. This was achieved by following the steps here.

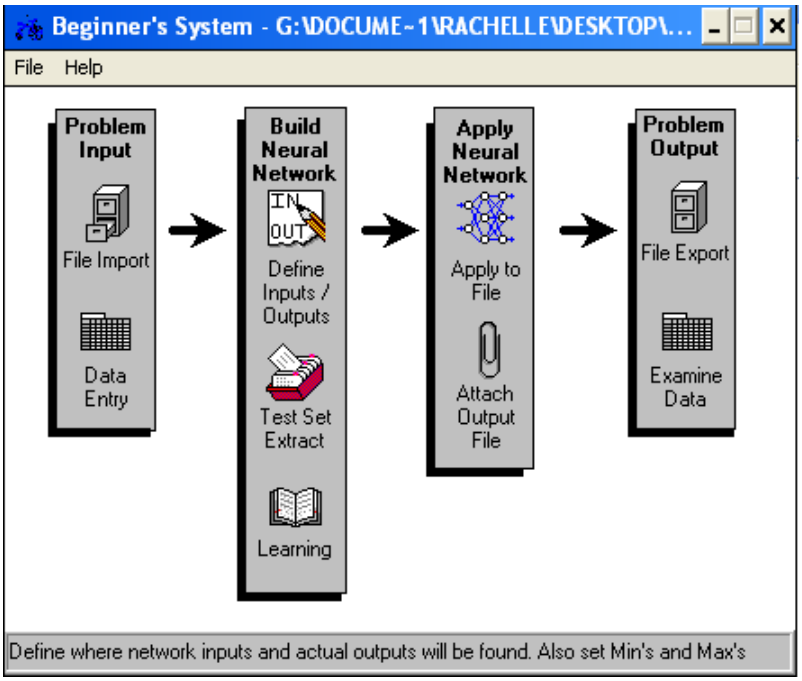

Fig.1 This is how to create a Neural Network using NeuroShell.

2. BAYESIAN NETWORKS

With the kind of problem we have, we cannot tell which feature affects which. So, in this Bayesian Networks implementation, we used Genie, user friendly software for determining graphical decision theoretic models to determine the Direct Acyclic Graph (DAG). Continuous values like Weight, Height, BMI, etc. were discretized first while discrete values like Gender, Allergy and Smoking Status remained the same. Users are free to set the parameters such as Background knowledge, Max Parent Count, Iterations, Seed, Max Time and many more. Also,

this software allows computation of the probability of the occurrence of a state given different states directly/indirectly affecting it.

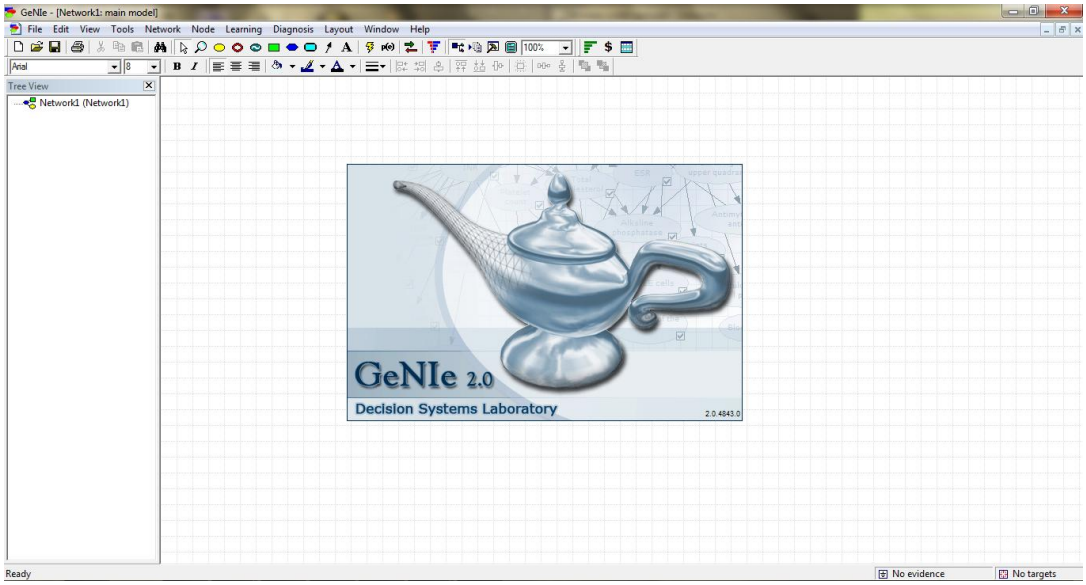

Fig. 2 Genie software was used to implement Bayesian Networks.

The data set in Neural Networks implementation was used in this implementation, except instead of three, we only get one data set. This data set was subdivided into Training Set and Test Set with a 70-30 proportion. Training Set was entered into the Genie software to produce a DAG that best fitted the inputted data. The test set was then inputted manually into the network using the produced DAG.

3. COLLABORATIVE FILTERING

Before each application of collaborative filtering, relevant training patients must be grouped first according to diagnoses they are similar [8]. This is to remove the influence of patients who have little or no similarity with the patient for whom predictions are made. Training patients with no similarity with active patient has weight of 0 and does not contribute to the prediction. Also, by removing them, the runtime is reduced.

Collaborative Filtering is used to make prediction $P(a, i)$ on the active patient *a* (testing) for feature *i,* based on the similarity between patient *a* and other patient *u* who has previously provided a value for that feature.

The training and test set used were exactly the same sets used in Bayesian Networks implementation. Also, three collaborative filtering algorithms were implemented, namely: User-based CF using Pearson Correlation, User-Based CF using Vector Cosine Correlation and Item-based CF using Pearson Correlation. These algorithms are all under Neighborhood-based CF, one of the many techniques in Memory-based CF category.

In User-based CF using Pearson Correlation, the *Pearson Correlation* measures the similarity or how the two variables linearly relate each other [8]. This similarity $w(u, v)$ between patient *u* and *v* is calculated using this equation:

$$w_{u,v} = \frac{\sum_{i \epsilon I}\left(r_{u,i} - \overline{r_u}\right)(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \epsilon I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i \epsilon I}(r_{v,i} - \overline{r_v})^2}}$$

where *I* is the set of features and $\overline{r_u}$ is the average observed value of the co-rated feature of the *u*th patient.

In User-based CF using Vector Cosine Correlation, similarity between two documents can be measured by considering a document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors (Macskassy, 2008). This similarity between vector of patient $\vec{U}$ and $\vec{V}$ is given by [8]:

$$w_{u,v} = \cos\left(\vec{U}, \vec{V}\right) = \frac{\vec{U} \bullet \vec{V}}{\left\|\vec{U}\right\| * \left\|\vec{V}\right\|}$$

where "•" denotes the dot-product of the vectors and $\left\|\vec{U}\right\|$ indicates the length of vector *U*.

The general collaborative filtering equation for User-Based CF in predicting value for active patient *a* with feature *i* is [8]:

$$P(a,i) = \overline{r_a} + \frac{\sum_{u \epsilon U}\left(r_{u,i} - \overline{r_u}\right) * w_{a,u}}{\sum_{u \epsilon U}\left|w_{a,u}\right|}$$

where *U* is the set of patients, $\overline{r_a}$ and $\overline{r_u}$ are the averages for patient *a* and patient *u* on all other features, and $w_{a,u}$ is the similarity between the patient *a* and patient *u.* This formula was used for both Pearson Correlation and Vector Cosine algorithm of User-based CF.

In Item-based CF using Pearson Correlation, this formula was used to compute for the similarity between features *i* and *j* [17]:

$$w_{i,j} = \frac{\sum_{u \epsilon U}\left(r_{u,i} - \overline{r_i}\right)\left(r_{u,j} - \overline{r_j}\right)}{\sqrt{\sum_{u \epsilon U}\left(r_{u,i} - \overline{r_i}\right)^2}\sqrt{\sum_{u \epsilon U}\left(r_{u,j} - \overline{r_j}\right)^2}}$$

where $r_{u,i}$ is the observed value of patient *u* on feature *i* and $\overline{r_i}$ is the average of the *i*th feature by those patients. The *simple weighted average* was used to predict $P(u,i)$ for patient *u* on feature *i.*

$$P(u,i) = \frac{\sum_{n \epsilon N} r_{u,n} w_{i,n}}{\sum_{n \epsilon N}\left|w_{i,n}\right|}$$

where summations are all over the features for patient *u*, $w_{i,n}$ is the weight between features *i* and *n*, and $r_{u,n}$ is the observed value for patient *u* on feature *n.*

B. Comparing the performances of ANN, Bayesian Network, and Collaborative Filtering in classifying Diabetes Mellitus patients

To determine the accuracy of classification, we had a test set where we applied the network learned in training. The accuracy of classification should be noted because a network may produce a good classification in the training set but fails on the test data set because of its complex and unrepresented data.

Receiver Operating Characteristic, also known as ROC Curve, was used to further analyze the classification done by the algorithms. It is created by plotting the fraction of *True Positive Rate* (TPR) versus the fraction of *False Positive Rate* (FPR) for different cut-off points of a parameter [18].

Before plotting the curve, there were some derivations performed on each of the predicted outputs per algorithm. A predicted positive output is considered *True Positive (TP)* if the target output is also positive, however it is considered a *False Positive (FP)* if target output is negative. A predicted negative output is then considered as *False Negative (FN)* if target output is positive, else if target output is also negative, then it is considered as *True Negative (TN)*. This can better be understood in this illustration:

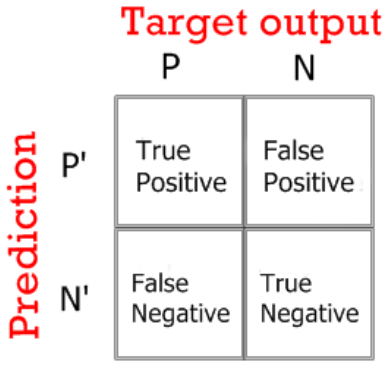

Fig. 3 This is the illustration for ROC Derivations.

After classifying to what group the output belongs, we now compute for the following measures:

(a) True Positive Rate or Sensitivity

$$TPR = \frac{TP}{TP + FN}$$

(b) False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

(c) Accuracy

$$ACC = \frac{TP + TN}{P + N}$$

(d) Specificity or True Negative Rate

$$SPC = 1 - FPR$$

V. RESULTS AND DISCUSSION

A. Implementing ANN, Bayesian Networks and Collaborative Filtering

1. NEURAL NETWORKS

The number of neurons in the hidden layer was incremented to observe its effect in the fitting of data. Setting too few hidden neurons can lead to high training and generalization errors due to under-fitting, while too many hidden units may result to low training errors but still high generalization errors due to over-fitting [5]. In addition, the ideal number of hidden neurons will never be less than the number of inputs neurons. Having 10 input neurons, we started at 10 as number of hidden neurons for the network, incremented it by either 5 or 10, and ended with 50 hidden neurons. We arrived at this neural network structure with 10 input neurons, a hidden layer that varies with the number of hidden neurons and the output layer with only a neuron, all fully connected by weights. The same network structure was implemented using NeuroShell.

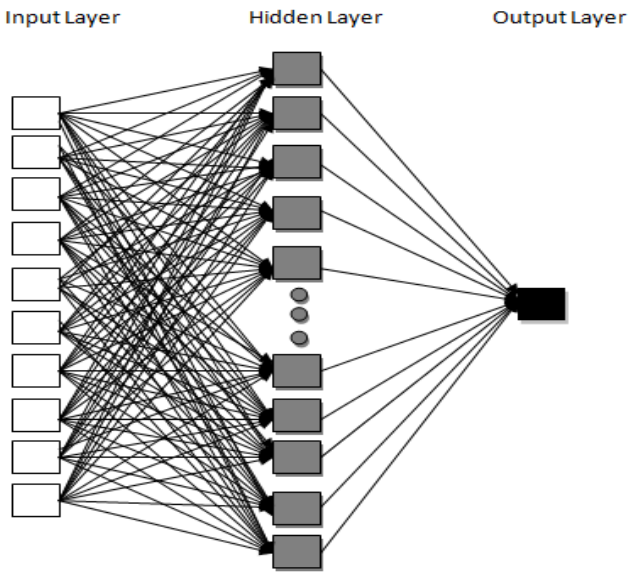

Fig. 4 This is the model for classifying patients using ANN.

Predicted outputs for the test set range from 0 to 1. Therefore, for patients with predicted output closer to 1, they are considered as *positive* DMI, and patients with predicted output closer to 0 is considered as *negative* DMI.

2. BAYESIAN NETWORKS

Genie allows its users to force or forbid arcs based from initial knowledge. In this problem, we know from the start that the rest of the features do not define or contribute to the occurrence of Gender. Therefore, we forbid arcs that lead to Gender. However, we did not restrict arcs going out of Gender because we don't have idea if it affects other features or not. Also, we forced arcs from features Height and Weight to BMI because we know that Height and Weight predicts the BMI. Based from the training set inputted and knowledge added, Genie program derived this DAG.

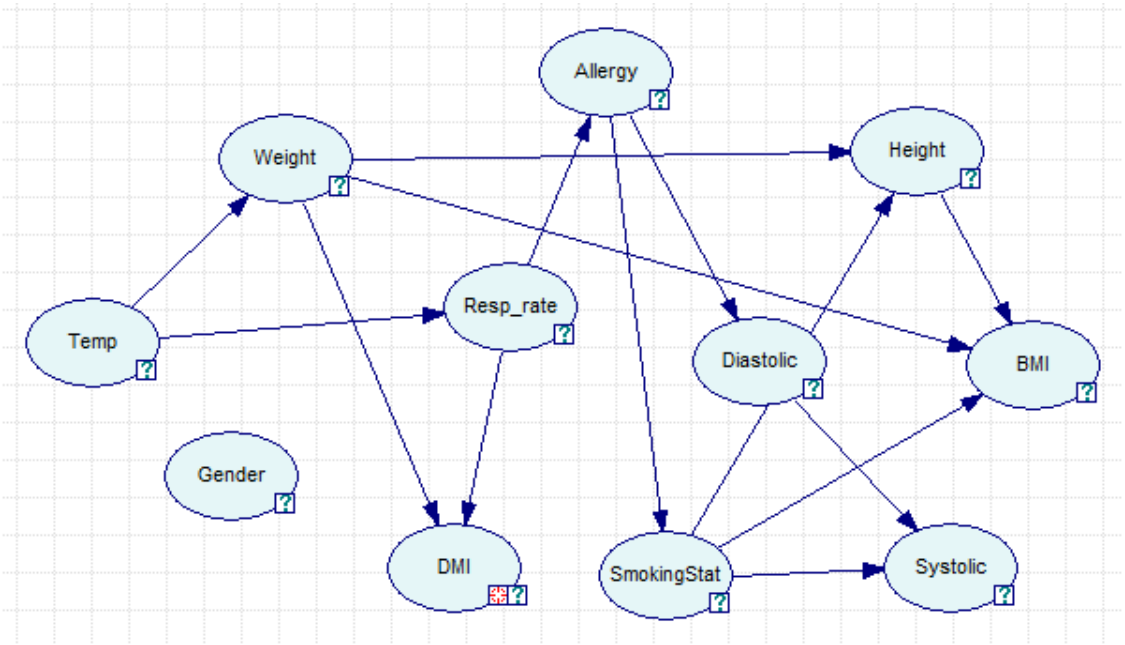

Fig. 5 This is the DAG produced by Genie.

Gender, as seen in Fig. 5, has no connections to the other features; therefore we can now say that it does not affect the state or occurrence of any feature. DMI is the target output, so we only focused on the features directly affecting it, namely, Weight, Respiratory Rate and Temp.

Weight and Respiratory Rate are the parents of DMI, making the occurrence of Diabetes Mellitus dependent on those two. Meanwhile, Temp is the parent of Weight and Respiratory Rate, making them dependent on the state of Temp.

After a DAG was produced, we manually inputted each of the pattern in test set to the network, which gave back an output value that we only considered as either *positive* DMI if closer to 1 and *negative* DMI if closer to 0.

3. COLLABORATIVE FILTERING

Collaborative Filtering techniques used in this study are under Neighbor-based of Memory-Based CF which depend its prediction from the similarity of patient from its neighbors (patient), or features from its co-features. A representation for this is a table or matrix of data set where missing fields with question marks are the ones to be predicted. Its output also ranges from 0 to 1, considering the output closer to 0 as *positive* DMI and *negative* if closer 0.

B. Comparing the performances of ANN, Bayesian Network, and Collaborative Filtering in classifying Diabetes Mellitus patients

1. NEURAL NETWORKS

Two implementations of neural networks were made in this study. The first was manually coded with some intended improvements in the basic neural network implementation, and the second using NeuroShell software. The test scores for each ROC measure of the first implementation are as follows:

TABLE I

THIS IS THE TABLE OF ROC RESULTS.

	10Nodes	**20Nodes**	**25Nodes**	**30Nodes**	**35Nodes**	**40Nodes**	**50Nodes**
TPR	0.64602	0.81416	0.78761	0.78761	0.69027	0.78761	0.73451

FPR	0.40667	0.50667	0.38667	0.39333	0.34	0.39333	0.36667
ACC	0.61597	0.63118	0.68821	0.68441	0.67300	0.68441	0.67680
SPC	0.59333	0.49333	0.61333	0.60667	0.66	0.60667	0.63333

By interpreting the table of results, we arrived at this chart that shows the trend for each measure.

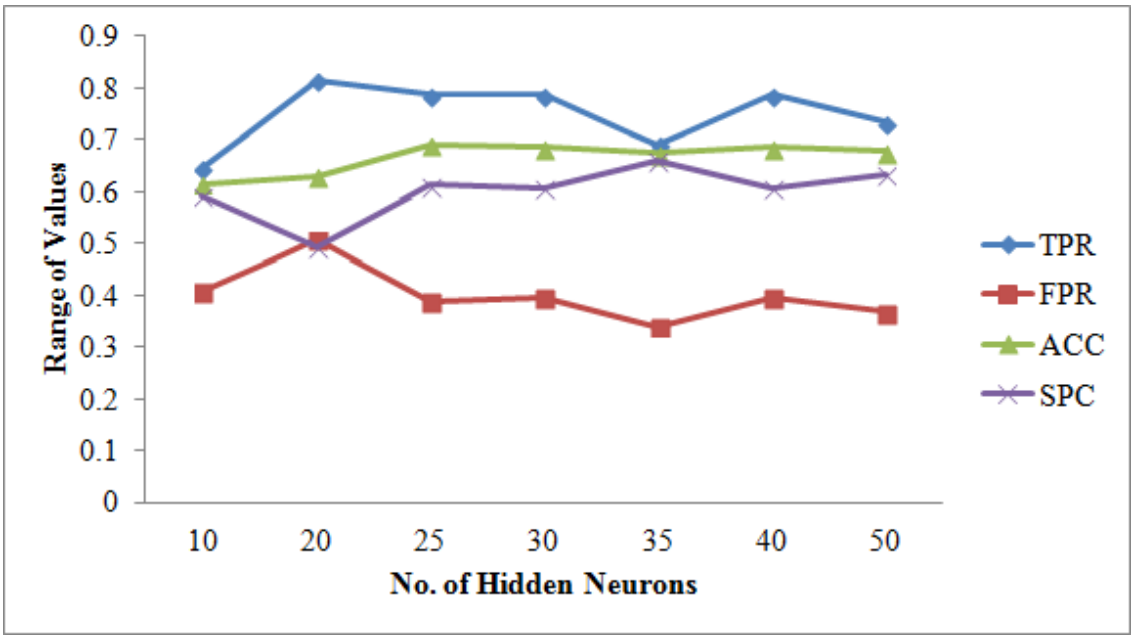

Fig. 6 Here is the interpretation of results for manually coded Neural Networks.

The above table and diagram shows that NN with 20 Hidden neurons got the highest rate for TP, but also got the highest rate for FP. It means that among the other NN structures, this structure is the most *sensitive* and has the highest hit rate. It also got the lowest Type II error rate yet also has the highest Type I error rate. In terms of *Accuracy* rate, NN with 25 Hidden neurons got the highest, meaning this structure has the closest prediction value to the actual value. The NN structure that got the highest *Specificity* rate, proportion of negatives correctly identified is the NN with 35 Hidden nodes. A high specificity rate indicates low Type I error.

For the second implementation which is using the NeuroShell software, we had this table of test scores for each measure.

TABLE II

THIS IS THE TABLE OF ROC RESULTS.

	10Nodes	20Nodes	25Nodes	30Nodes	35Nodes	40Nodes	50Nodes
TPR	0.59292	0.52212	0.41593	0.42478	0.43363	0.37168	0.38053
FPR	0.34667	0.26	0.22667	0.24667	0.25333	0.21333	0.20667
ACC	0.62738	0.64639	0.76526	0.61217	0.61217	0.60837	0.61597
SPC	0.65333	0.74	0.77333	0.75333	0.74667	0.78667	0.79333

Based from the table of results above, we arrived at this chart that shows the trend for each measure.

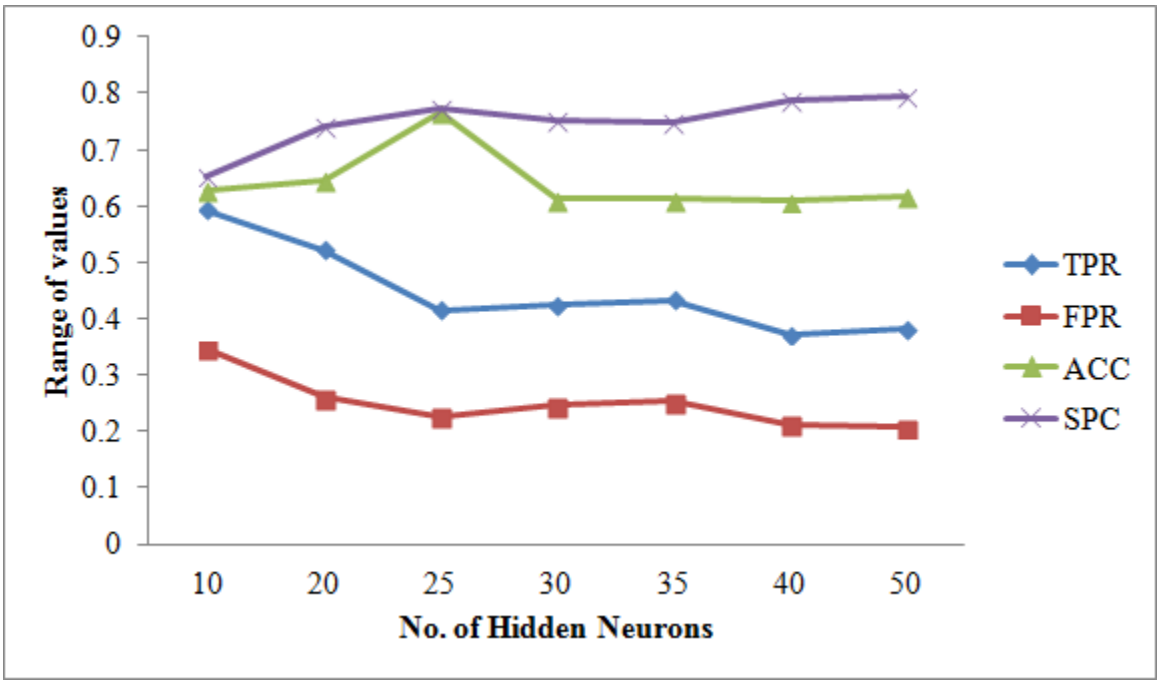

Fig. 7 This is the interpretation of results for Neural Networks using NeuroShell.

It shows that NN with 10 Hidden neurons got the highest TP rate being the most *sensitive* and has the highest FP rate among other NN structures, while NN with 25 Hidden nodes got the

highest *Accuracy* rate. The network structure that got the highest *Specificity* rate is NN with 50 Hidden nodes.

Looking at the interpretations of results for both methods, we can notice that they vary in outcome as to which network structure performed well in every measure used, but they arrived at the same result in terms of *Accuracy* rate which led us to NN with 25 Hidden neurons as the most accurate. Also, we can see that Specificity Rates are higher in NN implementation using NeuroShell software because it had lower FPR. In terms of the test scores for TPR and FPR, the neural networks we manually coded got a higher score than the NeuroShell software probably because we had greater data set and the approach is different.

2. BAYESIAN NETWORKS

After classifying the test output of Bayesian Networks into positives and negatives, and by following the ROC derivations, we arrived at this table of results.

TABLE III

THIS IS THE TABLE OF ROC RESULTS FOR BAYESIAN NETWORKS.

Measure	Bayesian
TPR	0.38596
FPR	0.23667
ACC	0.60038
SPC	0.76333

We can observe that the test scores for TPR and FPR of Bayesian Networks are relatively lower as compared to the test scores in Neural Networks implementations. Lower TPR only means that the network had a lower hit rate while low FPR indicates low Type I error. Test score for ACC is not that far from the ACC score of Neural Networks implementations. The SPC rate was impressive since it reached the 70% mark which only meant it had a high true negative rate.

3. COLLABORATIVE FILTERING

Three Collaborative Filtering techniques were implemented in this study all belonging to Memory-based Collaborative Filtering. The algorithms vary only in terms of computation of *similarity* and/or whether it is *user-based* or *item-based.* A CF is *user-based* if similarity is computed between the users/patients while *item-based* if it is computed between the items. Using Pearson Correlation and Vector Cosine correlation as similarity equation in predicting occurrence of DM, we arrived at this table of results.

TABLE IV

THIS IS THE TABLE OF ROC RESULTS FOR COLLABORATIVE FILTERING.

	Pearson Correlation UB	**Pearson Correlation IB**	**Vector Cosine UB**
TPR	0.39912	0.53070	0.43421
FPR	0.29333	0.35333	0.32667
ACC	0.57386	0.59659	0.57008
SPC	0.70667	0.64667	0.67333

Among the three CF techniques, Item-based CF using Pearson correlation had the highest *Sensitivity* or hit rate, while the CF technique that got the lowest FPR resulting to a low Type I error is User-based CF using Pearson correlation. The table also showed that the most *accurate* algorithm among the three in predicting DM patients is Item-based using Pearson Correlation, though the test scores aren't quite that far from each other. User-based CF using Pearson correlation got the highest *Specificity* rate.

ROC curve allows us to analyze and compare all the intelligence methods used in the study. It helps us in understanding which algorithm performed best by plotting it in a space that

shows clearly if implementations made resulted to a better or worse algorithm to use in predicting.

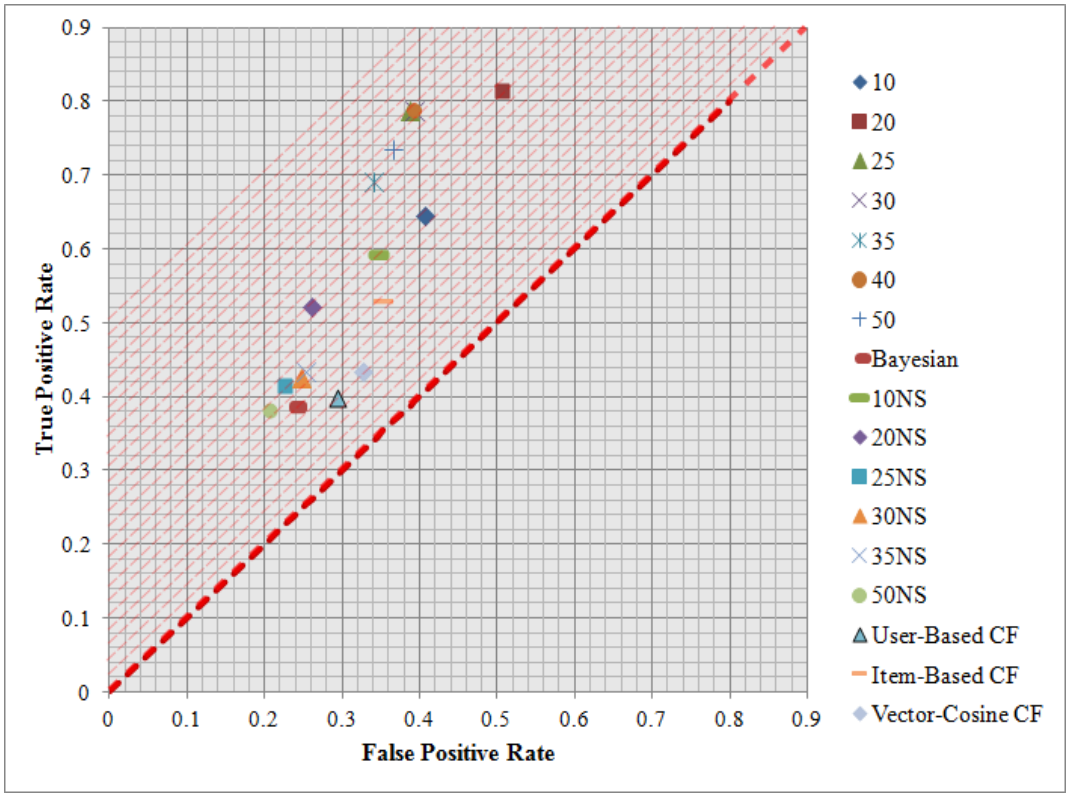

Fig. 8 This is the ROC Space and plot of all the algorithms used in the study.

The dashed red line or the *line of no-discrimination* in an ROC space and plot shows that all plots above the line result to better classification while those that are below it are worse for classification. In Fig. 8, all plots lie above the line, therefore we can say that all methods implemented in the study resulted to a better classification. The best classification method can be determined by extending the diagonal line upwards and looking at the point that would last touch the line. However, in the graph we have (See Fig. 8), it is unclear who last touched it because the three plots, namely 25 Hidden Neurons, 30 Hidden Neurons and 40 Hidden neurons likely occupy the same spot.

Another way of finding the best classification method is by looking at the plot that has the highest sensitivity rate (100% TPR) and the highest specificity rate (100% SPC) or lowest

FPR. This plot is called *perfect classification* which lies in the point (0, 1). By looking at the ROC space, we can notice that NN with 20 hidden neurons we manually coded had the highest TPR yet also had the highest FPR, making it hard for us to say that this is the best algorithm for the problem. We therefore determine the best classification method by choosing the plot that has the shortest distance from the *perfect classification point* using distance formula. The table below shows the distance of each method from the said point.

TABLE V

DISTANCE FROM PERFECT CLASSIFICATION POINT

Methods	**Distance**
10 Hidden Neurons	0.53915
20 Hidden Neurons	0.53967
25 Hidden Neurons	0.44116
30 Hidden Neurons	0.44701
35 Hidden Neurons	0.45993
40 Hidden Neurons	0.44701
50 Hidden Neurons	0.45269
10 NeuroShell	0.53469
20 NeuroShell	0.54403
25 NeuroShell	0.62651
30 NeuroShell	0.62589
35 NeuroShell	0.62045
40 NeuroShell	0.66355
50 NeuroShell	0.65303
Bayesian Networks	0.65807
User-based CF using Pearson Correlation	0.66865
User-based CF using Vector Cosine	0.65332
Item-based CF using Pearson Correlation	0.58744

VI. CONCLUSION AND FUTURE WORK

Among Neural Networks, Bayesian Networks and Collaborative Filtering techniques implemented to classify Diabetes Mellitus patients, Neural Networks with 25 Hidden neurons got the shortest distance from the perfect classification point. From one of the tables presented above, this method also got the highest Accuracy rate among neural network structures. In this problem only, we can therefore say, that by using ROC Curve as evaluation method, Neural Networks with 25 Hidden neurons is the best and the most accurate method to use in classifying Diabetes Mellitus patients.

The study can be further improved by adding more input features or more records with almost similar quantities for positives and negatives. Additional machine learning methods or techniques for more comparison is also one to look forward to.

REFERENCES

[1] “Medical Informatics FAQ,” http://www.faqs.org/faqs/medicalinformatics-faq/, 2012

[2] “What is Diabetes? What causes Diabetes?”, http://www.medicalnewstoday.com/info/diabetes/, 2013.

[3] A. Adeyemo and A. Akinwonmi, “On the Diagnosis of Diabetes Mellitus using Artificial Neural Network Models,” African *Journal Of Computing and ICT*, vol. 4, no. 2, pp. 1-8, September 2011.

[4] M. Shanker, “Using Neural Networks to Predict the Onset of Diabetes Mellitus,” *Journal of Chemical Information and Computer Sciences*, vol. 36, 1996.

[5] S.Xu and L.Chen, “A Novel Approach for Determining the Optimal Number of Hidden Layer neurons for FNNs and Its Application in Data Mining,” in *Proceedings The 5th International Conference on Information Technology and Applications*, June 2008, pp. 683–686.

[6] N. S. Philip, “Studies in artificial neural network modeling,” Ph.D. dissertation, Cochin University of Science and Technology, 2001.

[7] M. Scutari, “Learning Bayesian Networks with the bnlearn R Package,” *Journal of Statistical Software*, vol. 35, pp. 1-22, July 2010.

[8] X. Su and T. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Advances in Artificial Intelligence*, vol. 2009, no. 4, January 2009.

[9] A.Endo, T. Shibata, and H. Tanaka, “Comparison of Seven Algorithms to Predict Breast Cancer Survival,” *Biomedical Soft Computing and Human Sciences*, vol. 13, no. 2, pp. 11–16, 2008.

[10] D.Mantzaris, G.Anastassopoulos, and D.Lymberopoulos, "Medical Disease Prediction using Artificial Neural Networks," in *Proceedings of the 8th IEEE International Conference on Bioinformatics and Bioengineering*, Athens, Greece, October 2008.

[11] M.Sapon, K.Ismail, and S.Zainudin, "Prediction of diabetes by using artificial neural network," in *2011 International Conference on Circuits, System and Simulation IPCSIT*, vol. 7, IACSIT Press, Singapore, 2011.

[12] "Model Extremely Complex Functions, Neural Networks," http://www.statsoft.com/textbook/neural-networks/intro , 2013.

[13] L. E. Parker, "*Notes on Multilayer, Feedforward Neural Networks*," 2006.

[14] J. Heaton, "Neural network training (part 3): Gradient calculation," August 2011. [Online]. Available: http://www.youtube.com/watch?v=p1-FiWjThs8

[15] J. Heaton, "Neural network training (part 4): Backpropagation," August 2011. [Online]. Available: http://www.youtube.com/watch?v=p1FiWjThs8

[16] S. Macskassy, "*Machine Learning Slide(CS 567),*" 2008.

[17] G. Salton and M.McGill, *Introduction to Modern Information Retrieval*. NY, USA: McGraw-Hill, New York, 1983.

[18] "ROC Curve analysis in MedCalc," 2013. [Online]. Available: http://www.medcalc.org/manual/roc-curves.php

ABOUT THE AUTHOR

Rachelle G. Bondad is a senior BS Computer Science student in the Institute of Computer Science, University of the Philippines, Los Baños. She is the second child among the three children of Gilda and Francisco Bondad. Her interests include tweeting, reading ebooks and hanging out with friends.