A PRACTICAL COMPARISON AMONG NEURAL NETWORKS, BAYESIAN NETWORKS AND COLLABORATIVE FILTERING IN CLASSIFYING DIABETES MELLITUS PATIENTS

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.

SIGNIFICANCE OF THE STUDY

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.

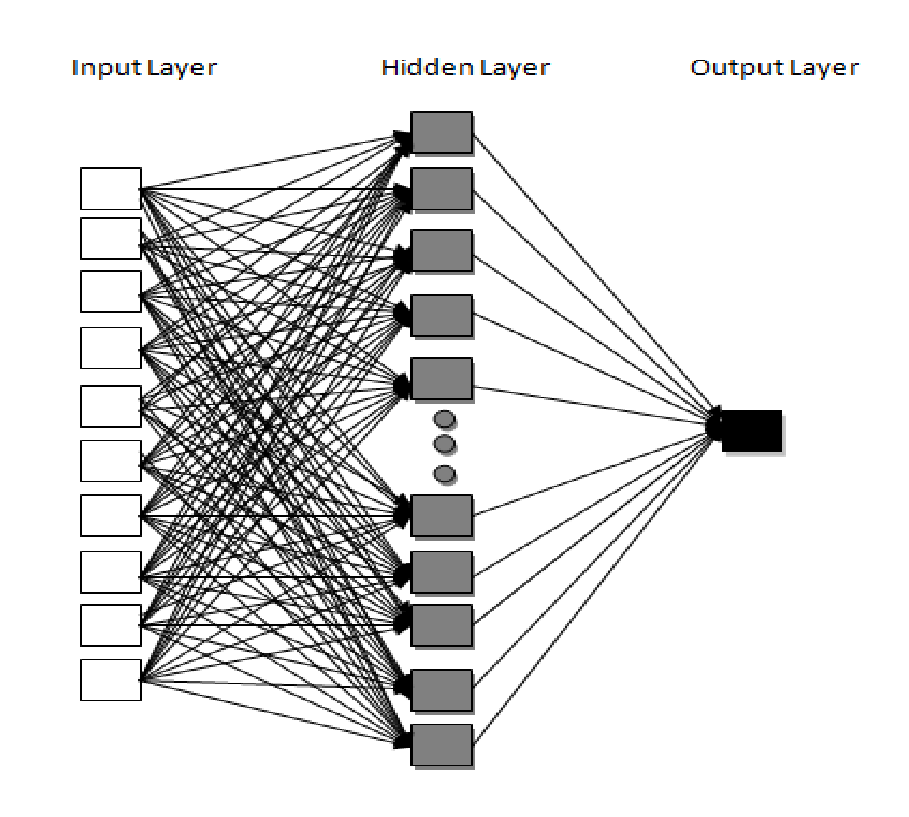
METHODOLOGY

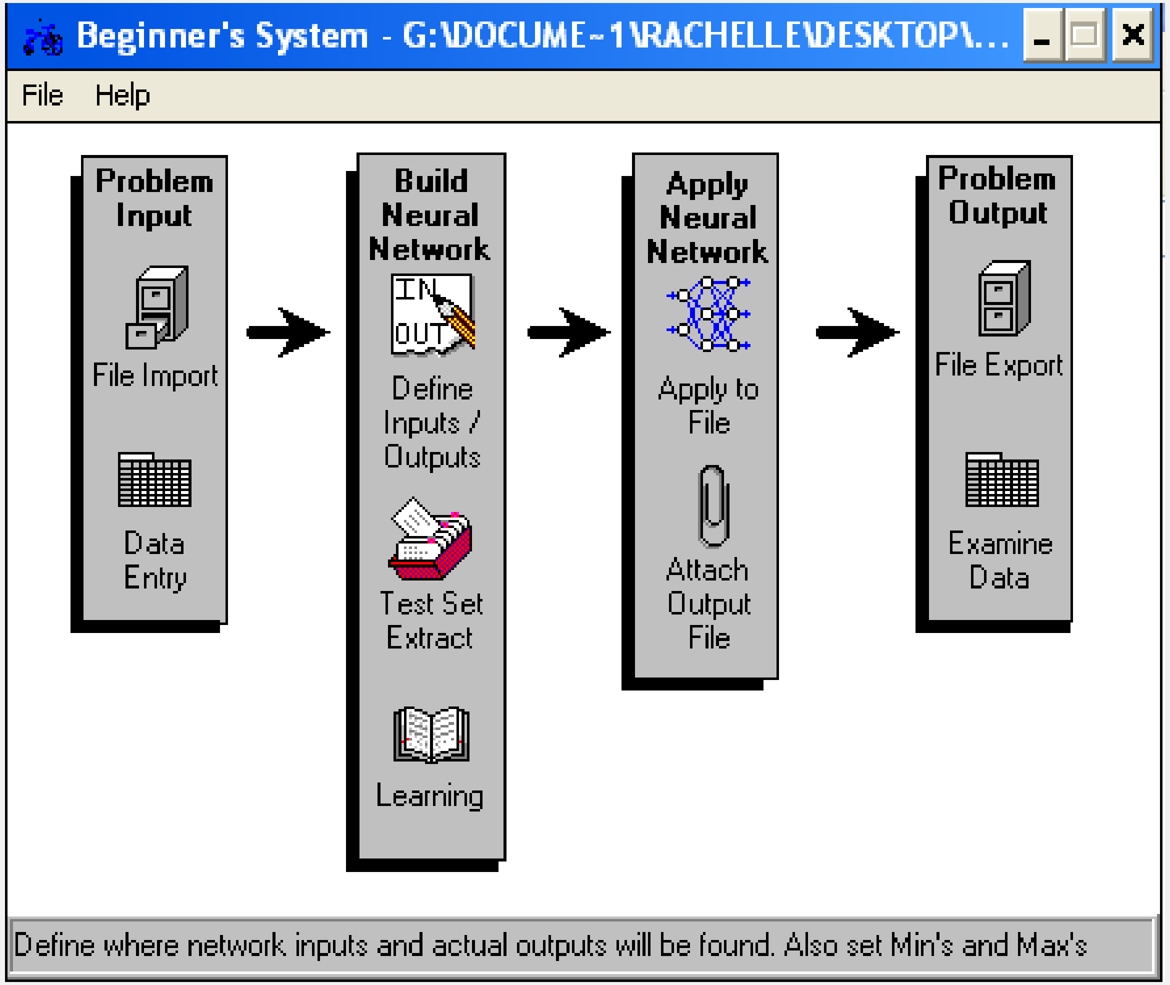
The authentic patient health records served as the data set for the study. This data set was obtained from the Practical Fusion, a free Web-based Electronic Health Records (EHR). It originally consists of 86144 patient health records; however, not all of these were used in the implementation. The features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

NEURAL NETWORKS

There are two kinds of learning in ANN: **Supervised Learning** and **Unsupervised Learning**. Supervised Learning is used when data set already includes the target output, unsupervised learning if otherwise. We used Supervised Learning in this study since data set includes target output which is DMI (Diabetes Mellitus Indicator).

The number of hidden neurons was incremented to observe its effect in the fitting of data. With 10 input neurons, an output neuron and a hidden layer that increments with the number of hidden neurons, all connected by weights, the network structure looks like this:





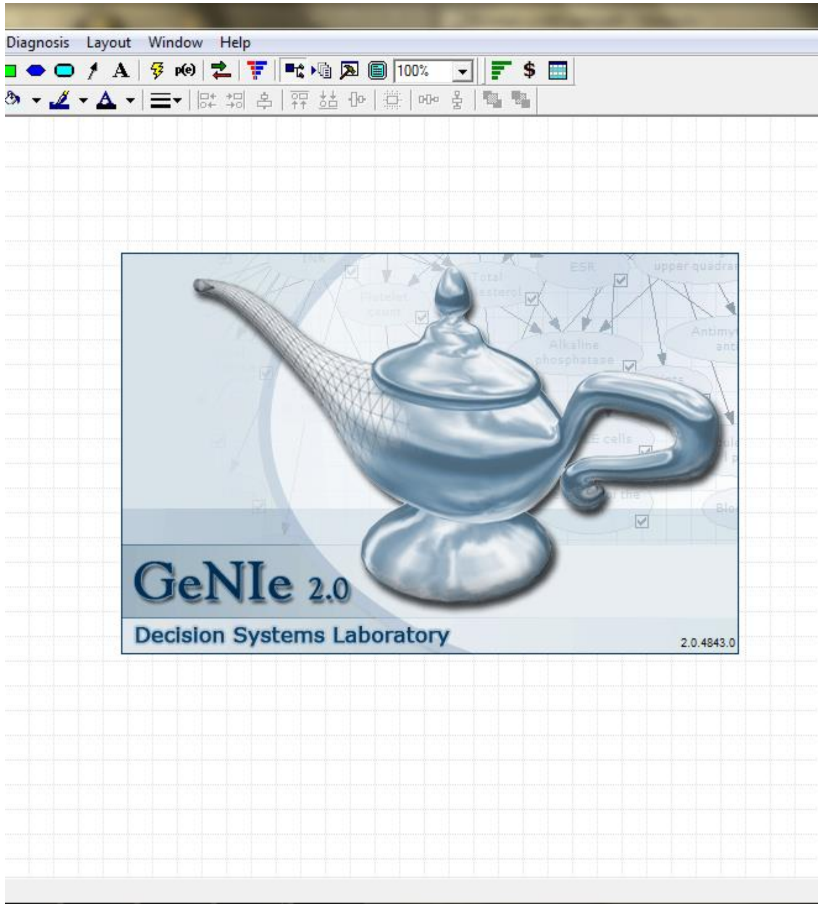
Define where network inputs and actual outputs will be found. Also set Min's and Max's

- Two implementations were made – one which we manually coded and the other one using the NeuroShell software.

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.

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.



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. Famous example of this is the commercial system Amazon.com

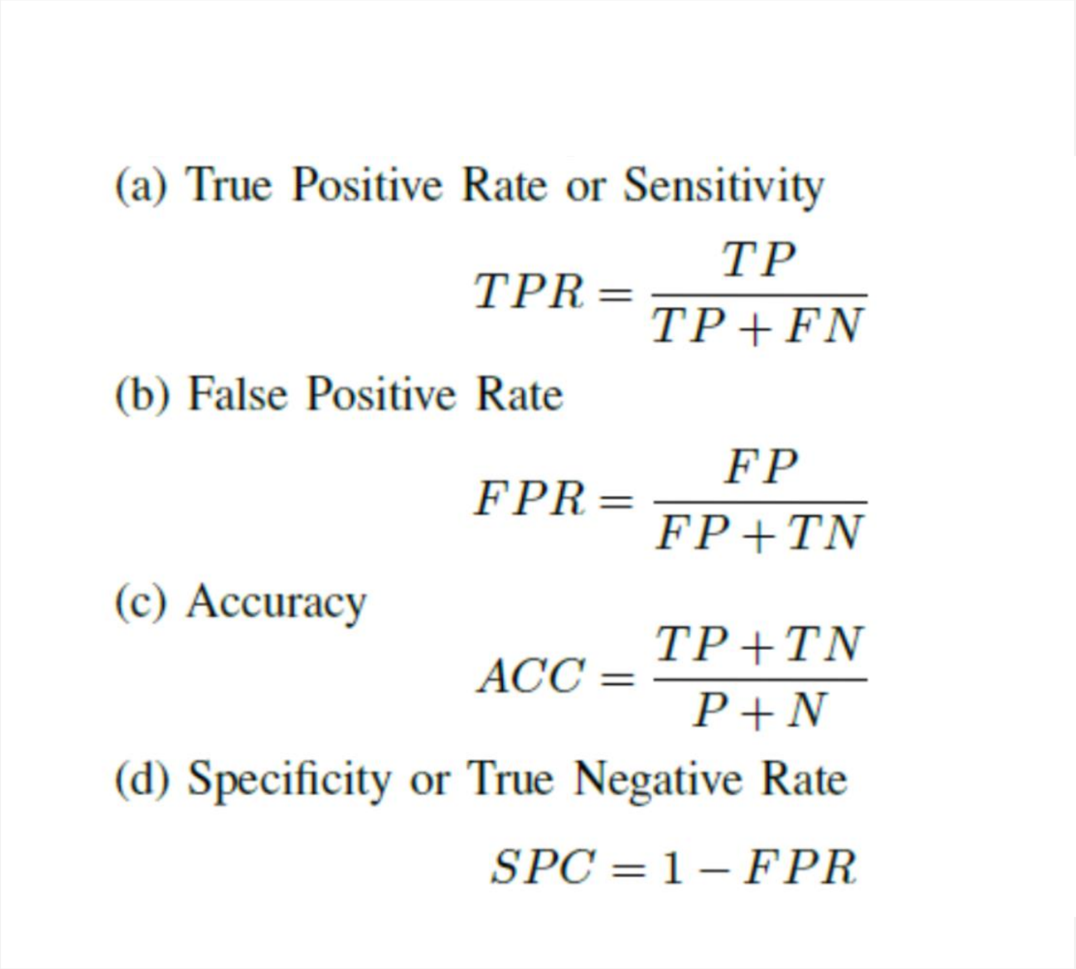
Prediction $P(a, i)$ on the active patient a (test) for feature *i,* is computed based on the similarity between patient a and other patient u who has previously provided a value for that feature.

Three techniques under Neighborhood-based of Memory-basedCF were implemented, namely: User-based CF using Pearson Correlation, User-Based CF using Vector Cosine Correlation and Item-based CF using Pearson Correlation.

ROC CURVE

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.

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).*



(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

$$SPC = 1 - FPR$$

RESULTS AND DISCUSSION

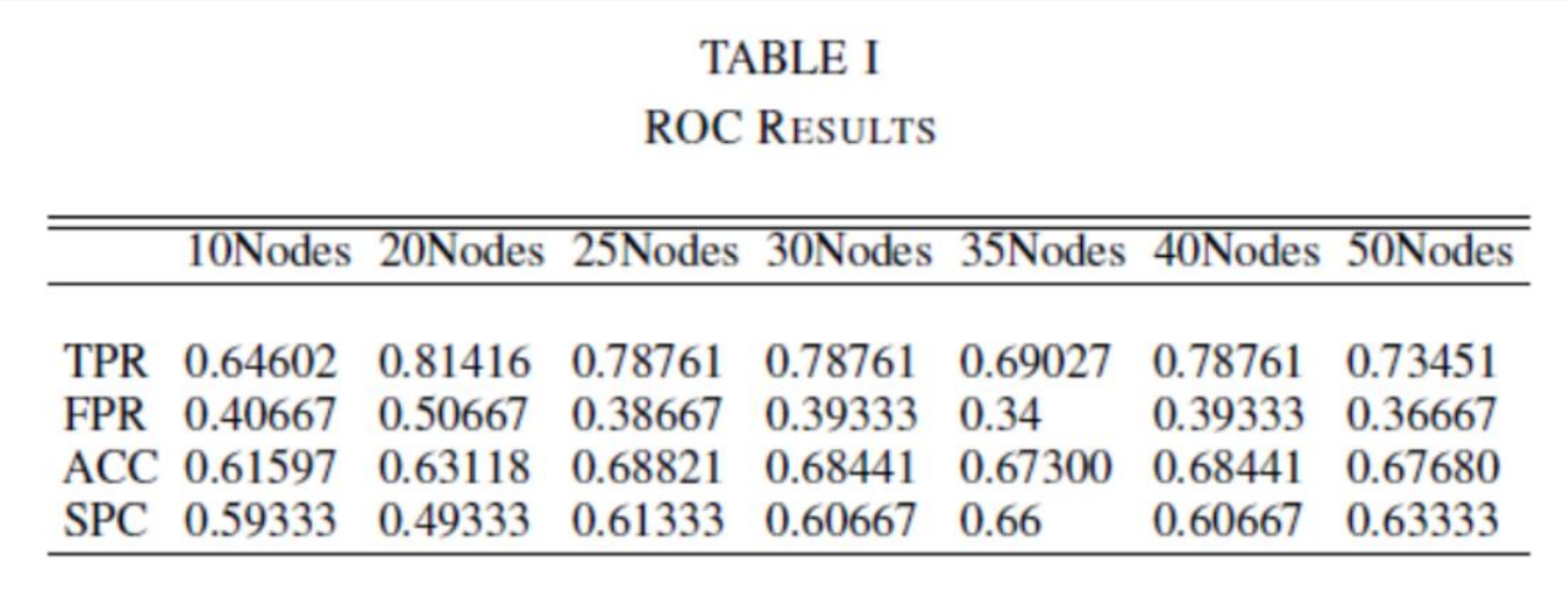


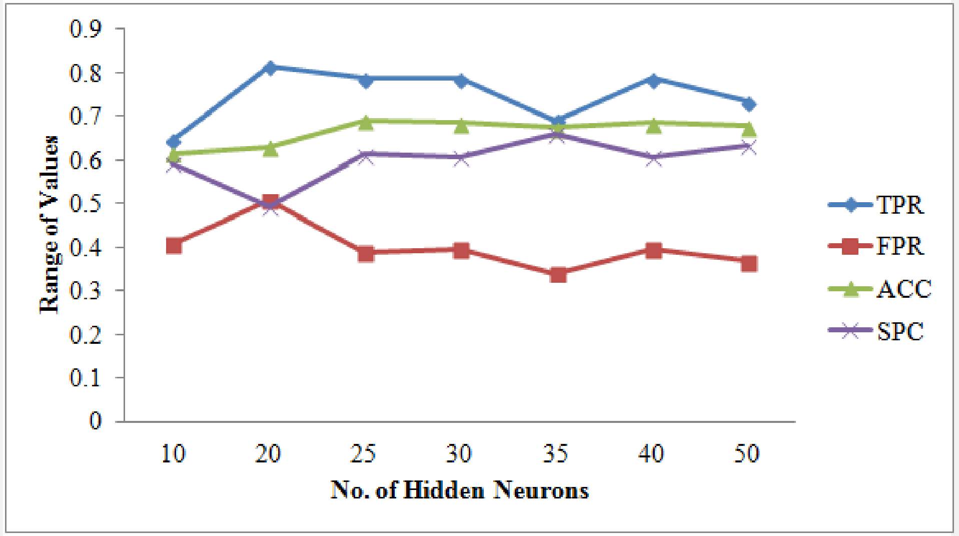
TABLE I

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 evaluating the test results of NN we manually coded, we arrived at this table of results.

NEURAL NETWORKS



It 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.

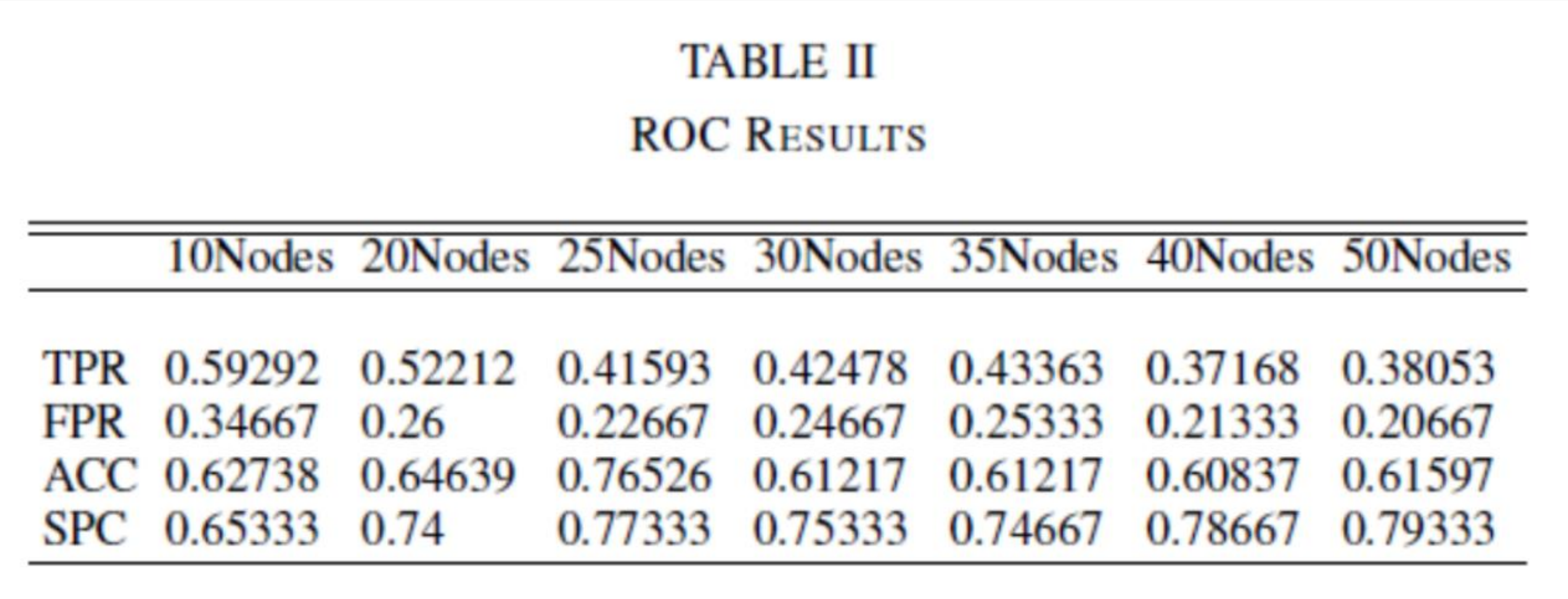


TABLE II

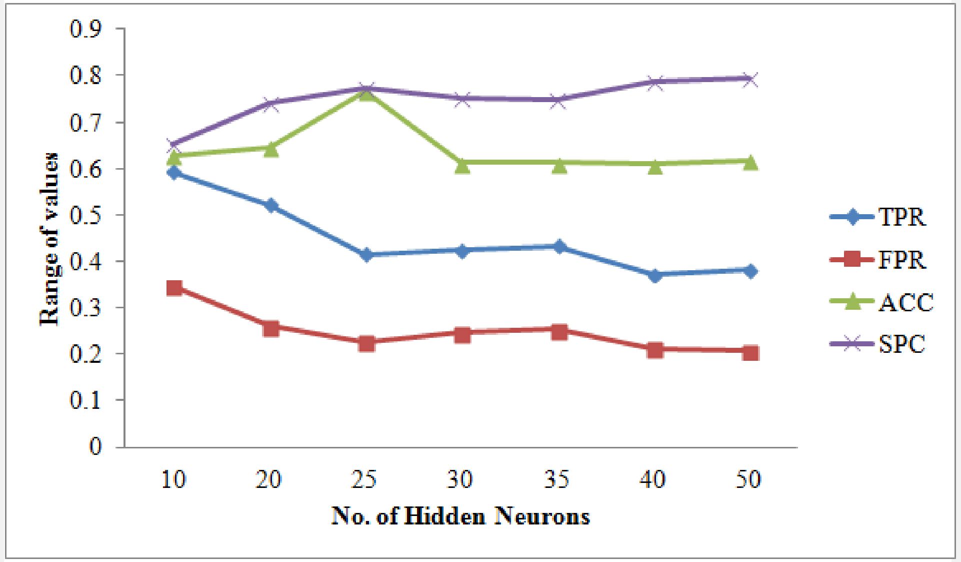
ROC RESULTS

	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

Evaluating the test results of NN implementation using NeuroShell, we arrived at this table of results.

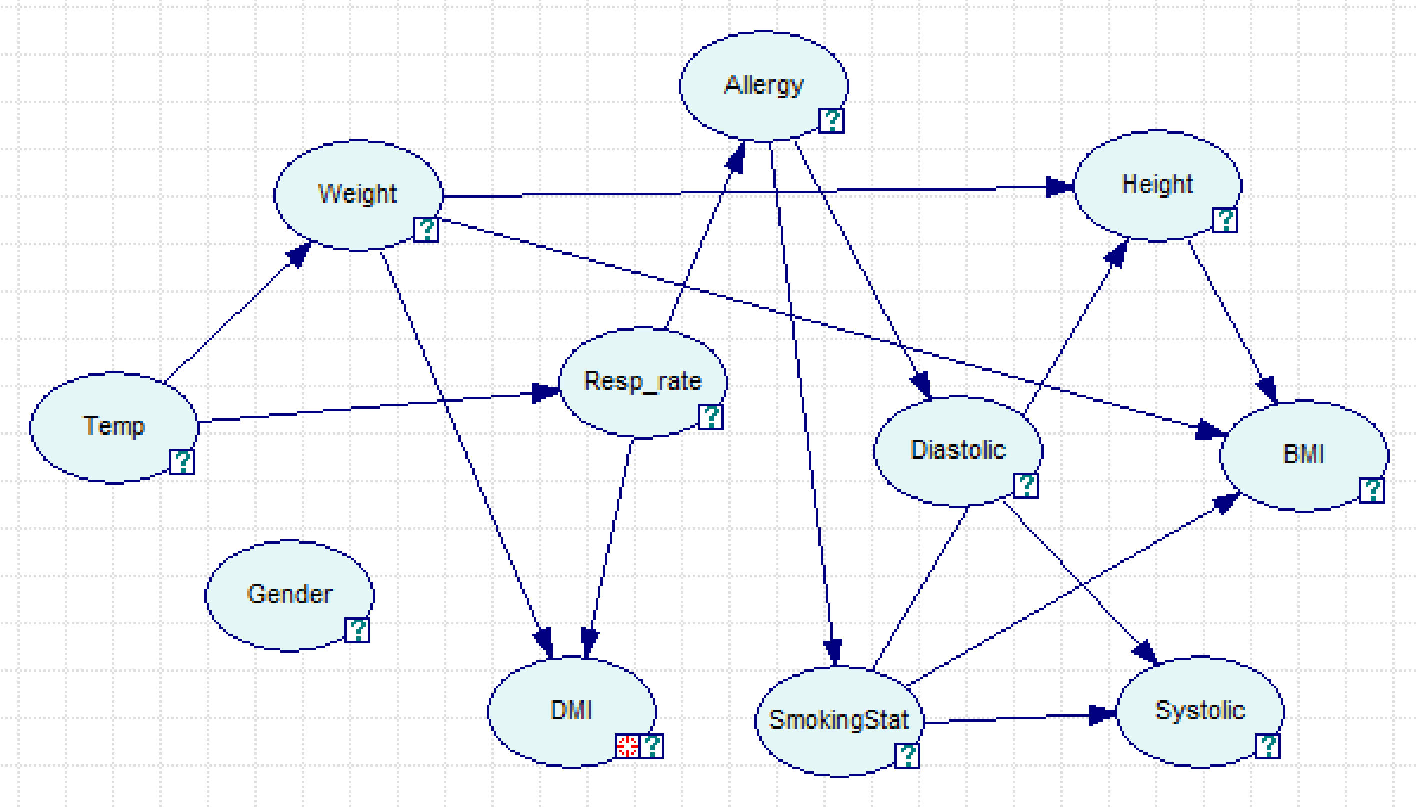
NEURAL NETWORKS

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.



For both implementations of Neural Networks, NN with 25 Hidden Neurons got the highest Accuracy Rate. TPR and FPR of the manually coded NN implementation are higher as compared to NeuroShell Implementation yet its SPC rate is higher than the latter.

BAYESIAN NETWORKS



This is the DAG produced by Genie software.

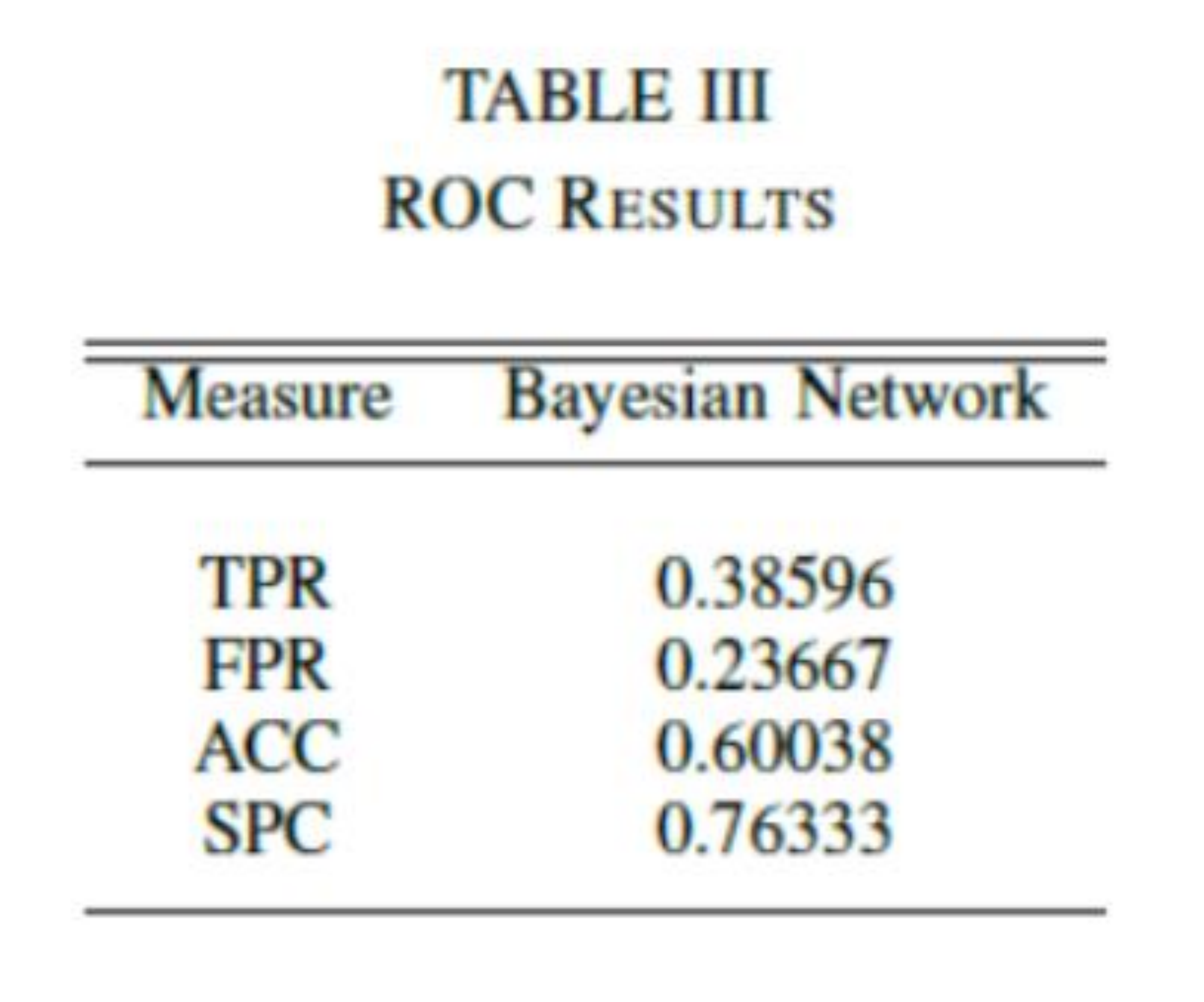


TABLE III
ROC RESULTS

Measure	Bayesian Network
TPR	0.38596
FPR	0.23667
ACC	0.60038
SPC	0.76333

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.

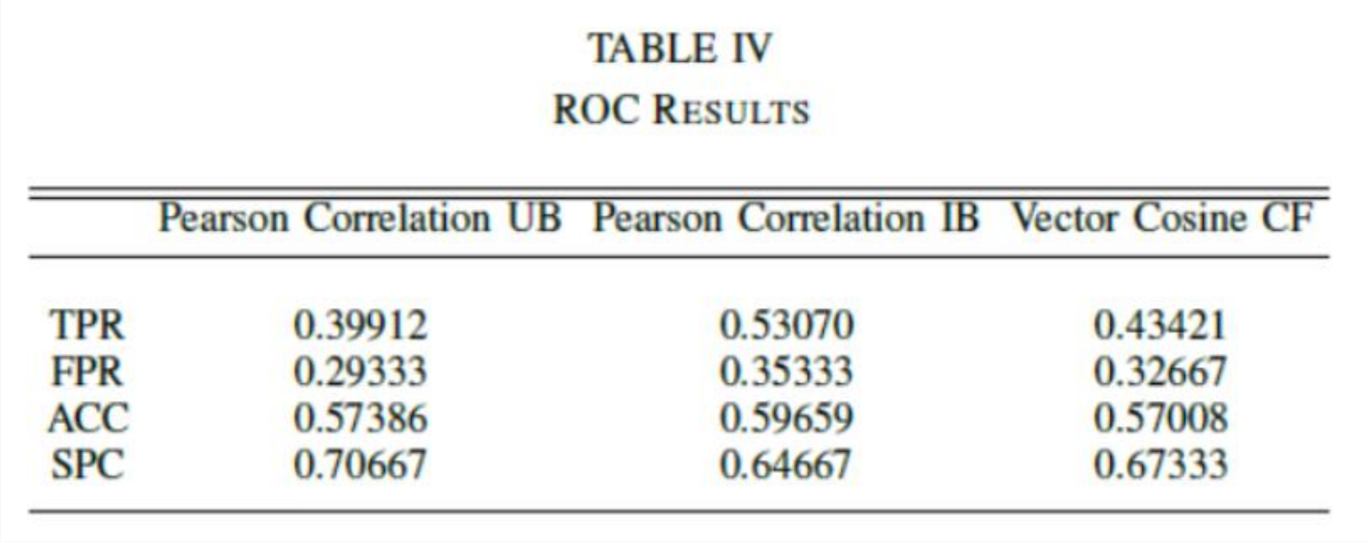


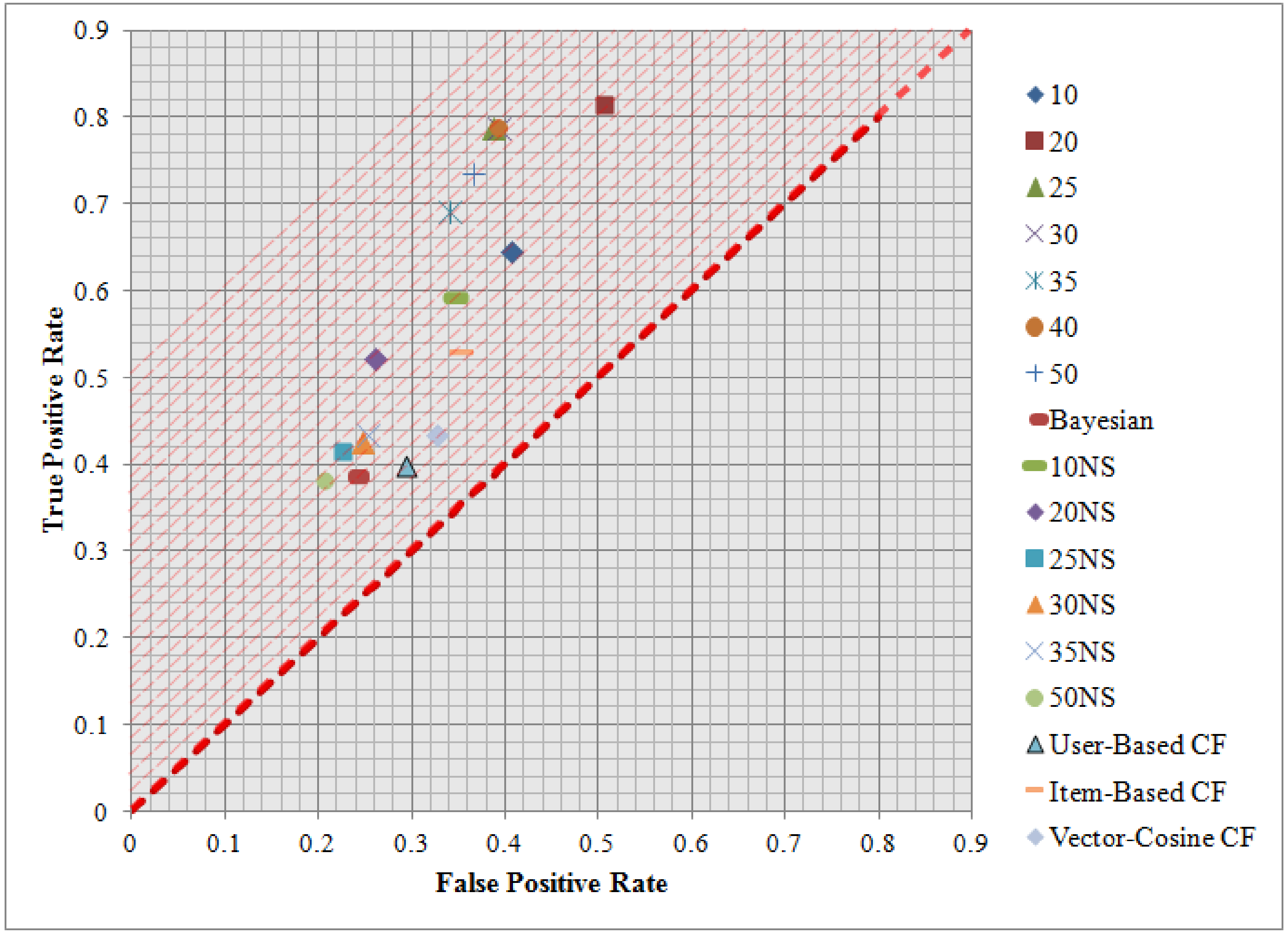
TABLE IV
ROC RESULTS

	Pearson Correlation UB	Pearson Correlation IB	Vector Cosine CF
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

COLLABORATIVE FILTERING

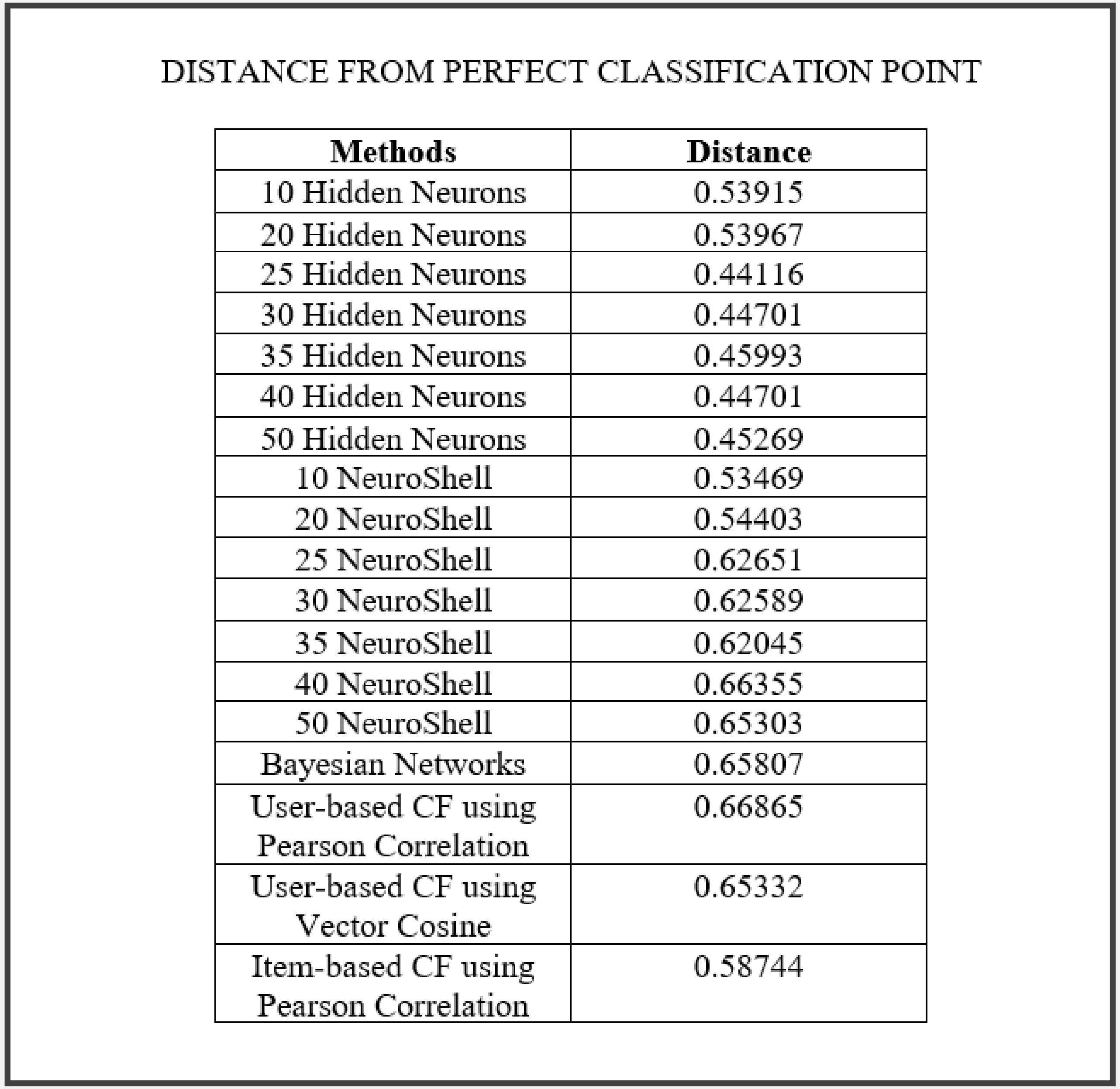
Among the three CF techniques, Item-based CF using Pearson correlation had the highest *Sensitivity* or hit rate, while the CF technique that got the lowest FPR resulting to a low Type I error is User-based CF using Pearson correlation. The table also showed that the most *accurate* algorithm among the 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.

COMPARISON OF ALL ALGORITHMS



ROC Space and plot of all the algorithms

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 is by computing the distance of plot from **perfect classification point.**



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

CONCLUSION

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.