Chapter 5

Experimental Results

This chapter shows the result of applying the proposed heart sound analysis algorithms discussed in the last chapter to the training set shown in Table 4.1. The training data had been modified by copying the normal heart sounds so that there roughly the same number of samples for each class. This was done to prevent the classifier(s) from favoring the majority class. Thus the training set $\mathbf{X}$ now consists of **97 feature vectors, 48 for normal and 45 for abnormal class**. The results are separated into three experiments, the first is cross-validation using a single neural network classifier with non-sampled training data $\mathbf{X}$. The number of hidden neurons was varied to determine the optimal number of hidden neuron. The second experiment was cross-validation using bagging classifiers and number of hidden neuron from obtained from experiment 1 and the number of bagging classifiers was varied. Finally in experiment 3, the number hidden neuron and the number of bagging classifiers were fixed using the results of the first two experiments and the decision threshold (T in Equation 4.9) was varied.

5.1 Experiment 1: Cross-validation using Single Neural Network

As explained in Chapter 2, classification results can be demonstrated by a set of three numbers: accuracy, sensitivity and specificity. These numbers in turn are calculated from a set of 4 numbers that provide the ”raw score” of a pattern recognition system. They are true positive (TP), false positive (FP), true negative (TN), and false negative (FN). They are defined as follows:

- TP: the ratio between samples which are actually positive over the number of samples classified as positive.
- FP: the ratio between samples which are actually negative over the number of samples classified as positive.
- TN: the ratio between samples which are actually negative over the number of samples classified as negative.
- FN: the ratio between samples which are actually positive over the number of samples classified as negative.

And accuracy, sensitivity and specificity are defined as the following,

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{N} \tag{5.1}$$

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5.3}$$

where N is the number of samples in the training set. The result of experiment 1 in shown in Table 5.1 where N denotes the number of hidden neurons. The result in Table 5.1 shows

Table 5.1: Varying Number of Hidden Neurons

N	TP	TN	FP	FN	Acc.	Sen.	Spec.
1	36	48	0	9	90.3	80	100
2	36	48	0	9	90.3	80	100
3	36	48	0	9	90.3	80	100
5	38	48	0	7	92.5	84.4	100
7	38	48	0	7	92.5	84.4	100
10	38	48	0	7	92.5	84.4	100
20	37	48	0	8	91.4	82.2	100

that the best performance of a single neural network occurs at 5 hidden neurons. However, considering that the performance of is not much lower (just over 2 %) for the case of one hidden neuron, it raises a question whether or not $\mathbf{X}$ is linearly separable. This is because if the hidden layer consists of only a single neuron the output neuron is just an identify map and can be ignored and the neural network is reduced to just two layers. A classification problem is said to be linearly separable if for a training set in R^n , there exist a hyperplane in R^{n-1} that can divide the space R^n, into two mutually exclusive regions each containing exclusively members of one class. A simple way to check whether or not $\mathbf{X}$ is indeed linearly separable, is a test using a simple perceptron model (a single neuron with a binary threshold function as activation function) that is,

$$y = \begin{cases} 1 & \text{if } \mathrm{wx} \geq \mathbf{0} \\ 0 & \text{otherwise,} \end{cases} \tag{5.4}$$

where y is the output of the perceptron, $\mathbf{x}$ is the input vector and $\mathbf{w}$ is the weight vector. The perceptron is the simplest neural network model that can only handle linearly separable classification problem. If the perceptron is successful, then $\mathbf{X}$ is linearly separable, otherwise, it is not. The perceptron was trained using $\mathbf{X}$ and also tested using $\mathbf{X}$ to give it the best possible chance. The result of perceptron trails is given in Table 5.2 The result in Table

Table 5.2: Perceptron Test

Trial	TP	TN	FP	FN	Acc.	Sen.	Spec.
1	24	48	0	21	77.4	53.3	100
2	39	20	28	6	63.4	86.7	41.7
3	40	36	12	4	81.7	90.1	75.0
4	40	28	20	5	73.1	88.9	58.3
5	36	36	12	9	77.4	80.0	75.0
6	25	48	0	20	78.5	55.6	100
7	41	28	20	4	74.2	91.1	58.3
8	37	44	4	8	87.1	82.2	91.7

5.2 shows that the perceptron fails to classify $\mathbf{X}$. In trial number 6, all the negative samples were correctly classified with about half of the positive samples were miss-classified. And in trial 7, most of the positive samples were correctly classified while about half of the negative samples were miss-classified. This shows that there are overlap of the data points such that trying to correctly classify one class will result in another being significantly miss-classified. Thus it can be concluded that $\mathbf{X}$ is NOT linear separable.

5.2 Experiment 2: Varying the Number of Classifier

In experiment 2, the cross-validation result using different number of bagging classifiers were tested. The number of hidden neuron was fixed at 5, which yielded maximum performance in experiment 1. The decision boundary is the median value. That is, if the number of classifiers is 4, then the decision threshold is 2 since all possible output are $\{0, 1, 2, 3, 4\}$. Therefore the number of classifier must be even because zero is also counted, so the number of possible output is odd and the median can be selected as the decision threshold. Table 5.3 shows the result of varying the of bagging classifiers. The result shows that there is little improvement as the number of classifiers is increased, so it could be set to a low number to save on the processing time.

5.3 Experiment 3: Varying the Number of Classifier and Decision Threshold

For experiment 3, the decision threshold is varied, and doing so may have some effect on the results of experiment 2 as well. In other words, experiment 2 and 3 may not be independent of each other, which means that to find the optimal values for number of classifiers and decision threshold, all possible combinations must be enumerated. This is done by following the same steps as in experiment 2, but for each number of classifier vary the decision threshold such that it include all values from the median down to 1 (the threshold was

Table 5.3: Varying Number of Classifiers Test

Num. Class.	TP	TN	FP	FN	Acc.	Sen.	Spec.
2	36	48	0	9	90.3	80.0	100
4	38	48	0	7	92.5	84.4	100
6	38	48	0	7	92.5	84.4	100
8	39	48	0	6	93.4	86.7	100
10	37	48	0	8	91.4	82.2	100

not increased since in all cases so far the specificity is already 100, increasing the decision threshold would only make the performance worse). The result is shown in Table 5.4, where the left column titled ”Combination” has the following format: 4.1 means four classifiers with decision threshold equal to 1, 6.2 means six classifiers with decision threshold equal to 2, etc. The case where number of classifiers equal to two was omitted since there is only one possible decision threshold. The result shows maximum performance at 6.1 and 8.1 (six and eight classifier respectively with decision threshold equal to 1), and 6.1 is more desirable since it runs faster. From experiment 1 through 3, it can be concluded that the optimal configuration for the bagging classifier are: **5 hidden neurons, 6 classifiers, and decision threshold equal to 1**.

Table 5.4: Varying both Number of Classifiers and Decision Threshold Test

Combination	TP	TN	FP	FN	Acc.	Sen.	Spec.
4.1	39	48	0	6	93.5	86.7	100
4.2	38	48	0	7	92.5	84.4	100
6.1	41	48	0	4	95.7	91.1	100
6.2	37	48	0	8	87.1	82.2	100
6.3	38	48	0	7	92.5	84.4	100
8.1	41	48	0	4	95.7	91.1	100
8.2	39	48	0	6	93.5	86.7	100
8.3	39	48	0	6	93.5	86.7	100
8.4	39	48	0	6	93.5	86.7	100