**ABSTRACT**

Breast cancer is the second most prevalent form of cancer worldwide. In this research paper, we have proposed a supervised artificial neural network capable of learning adaptively from the results of a fine needle aspirate (FNA) test on the breast mass of patients. The model was evaluated on the Wisconsin Breast Cancer Diagnosis dataset and achieved a median predictive accuracy of **97.54%**on the most extreme test condition. The model can potentially assist medical professionals in determining breast cancer and minimize human decisional errors.

**INTRODUCTION**

In 2018 alone, there were more than 2 million new cases of breast cancer reported worldwide (as per data from the World Cancer Research Fund). A fine needle aspirate (FNA) test on the breast mass of a patient, is one of the most critical examinations for determining breast cancer. A hollow needle attached to a syringe is used to withdraw the necessary amount of tissue from the area of suspicion and then, features are computed from a digitized image of the sample. These features describe various characteristics of the cell nuclei present in the image in a 3-Dimensional space. In this research paper, we had focused on developing a predictive model based on artificial neural network that is capable of analyzing those features and help predict malignancy with a high degree of accuracy.

**LITERATURE REVIEW**

A notable amount of research has been done to date on machine learning aided breast cancer detection and the associated literature consists of several astonishing pieces of work. **Guo, H. and Nandi, A.K. (2006)** in their paper had proposed a **Multilayer Perceptron (MLP)** model with retro propagation of error algorithm which attained an accuracy of **96.21%**. **Christobel, A. and Sivaprakasam, Y.** **(2011)** compared the performance of several classification algorithms and was able to achieve **96.99%**predictive accuracy with the **SVM (Support Vector Machine)** model. **Karabatak, M. and Ince, M.C. 2009** developed a diagnosis system based on Neural Networks (NNs) and Association Rules (AR) which was able to achieve a classification accuracy of **97.4%**. **Shahnaz, C., Hossain, J., Fattah, S.A., Ghosh, S. and Khan, A.I. 2017**was able to obtain an astonishing predictive accuracy of **98.06%** with their proposed Convolutional Neural Network model. In our research, **we were able to attain a median accuracy of 97.54%**(on a training and test split of 50% each)with a neural network equipped with adaptive learning capabilities.

**DATASET**

The dataset used for this research was obtained from [**UCI Machine Learning Repository**](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29). The dataset is composed of numerous features calculated from the digitized image obtained after a fine needle aspirate (FNA) test on 569 patients. Each record is labelled as either 'B' or 'M' standing for ‘Benign’ and ‘Malignant’ respectively.

**DATA ANALYSIS**

In the dataset, we found that there are 357 records with an outcome of ‘B’ (stands for ‘Benign’) whereas 212 records have ‘M’ (stands for ‘Malignant’) as an outcome.



Figure 1 : Distribution of output class

Thus, the distribution of the output data is slightly skewed in favour of 'Benign' outcomes.

We subsequently analysed the distribution of the mean of each independent variable in the dataset and it was visible that there were a couple of features that had a much higher range of values compared to the others.



Figure 2 : Mean of individual features

This necessitated a statistical normalization to be conducted on the set of independent variables. For which we used **min-max normalization** technique as our preferred method.

Equation 1 : Statistical Normalization

**FEATURE SELECTION**

**Principal Component Analysis** : After the normalization of the independent features, we conducted a Principal Component Analysis (PCA) on the feature set to understand the variance explained by each of them.

We conducted the test by steadily decreasing the training set concentration and simultaneously increasing the concentration of the test set. We started with an initial training set concentration of 90% (with a corresponding test set concentration of 10%) and then decreased it by 5% till we reached a training set concentration of 50% (with a corresponding test set concentration of 50%). In all of the instances, we found that around 99% of the variance in the data was explained by 15 of the 30 features in the feature set.

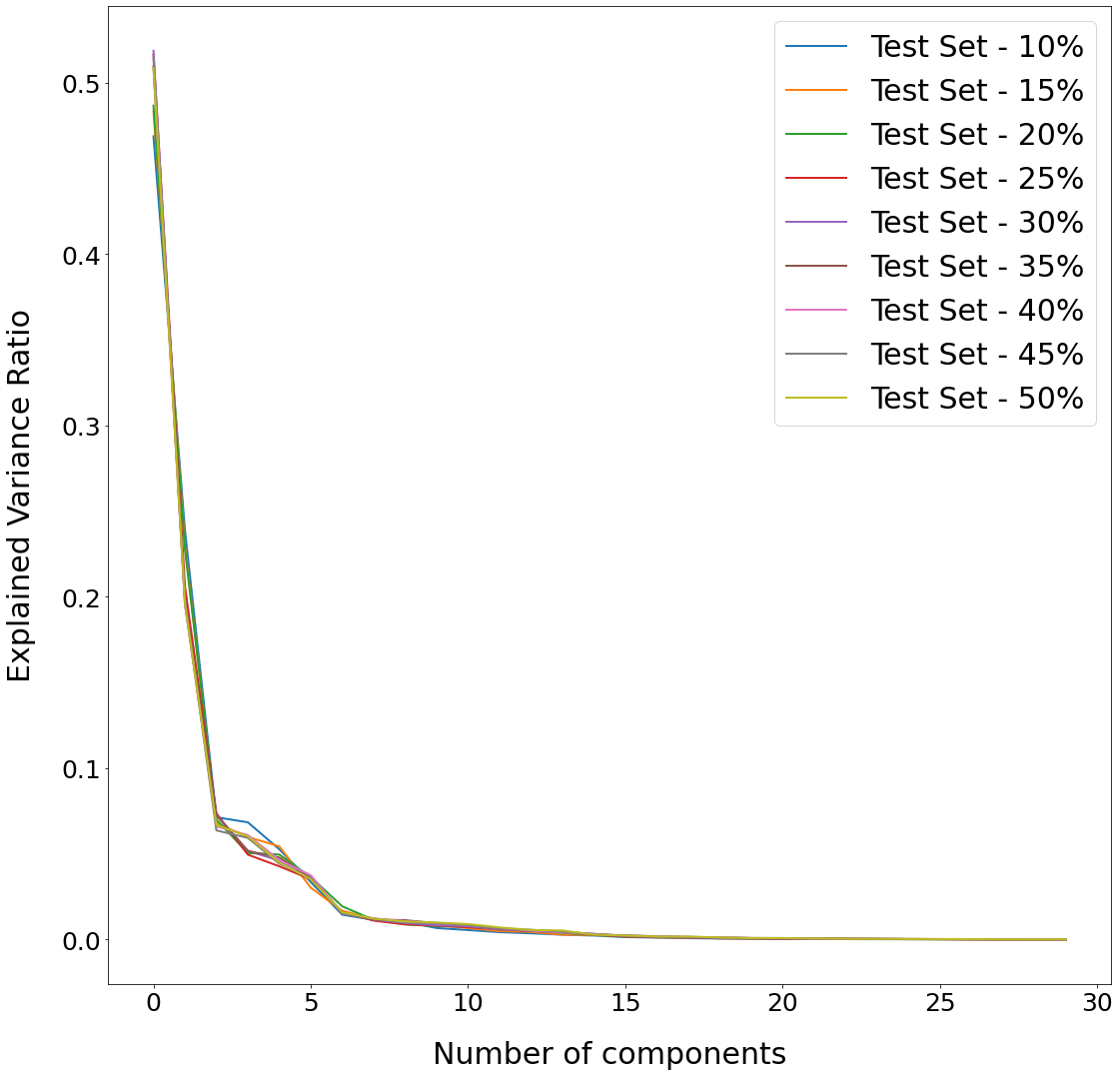


Figure 3 : Result of Principle Component Analysis

**ANOVA F-Statistics Test :** Despite the observation we made in our PCA test, we decided to conduct an Analysis of Variance (ANOVA) test on the feature set.

This decision was taken as Principal Component Analysis only measures the extent of variance explained by the independent variables, but it does not consider the interaction between the independent variables and the output variable.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Figure 4 : Result of ANOVA F-Statistics Test

We conducted the test in the same manner as before and steadily decreased the training set concentration from 90% to 50% with a 5% step in between (simultaneously increasing the concentration of the holdout set from 10% to 50%).

We observed that the ANOVA F-Statistics test found around 25 of the 30 features to be useful. However, the extent of their usefulness was quite different from each other as was evident via the scores obtained by the different variables. Thus we decided to continue using only those 25 features for training our models.

**MODELS :**

In this section, first, we have performed a comparative analysis of the prediction accuracies achieved by the various classification models on the Wisconsin Breast Cancer Diagnosis dataset. Then we tested a deep learning model on the same dataset and recorded the accuracy obtained by it.

**Model Training** : We trained our model in the same manner as in the PCA Test (and ANOVA F-Statistics Test). We consistently decreased the training set concentration from 90% to 50% while we increased the corresponding test set concentration from 10% to 50%, with a step of 5% in between.

In addition, for the deep learning models, we also trained them with decreasing batch sizes. Starting with an initial value of 32 and then halving it till we reached a minimum of 1.

Furthermore, in each of the outlined scenarios above, we recorded the median accuracy obtained by the models over 100 iterations. Since deep learning models are often susceptible to volatile performances and can yield varying predictive accuracies, even with the same set of circumstances.

**Classification Models** : The classification models we used for our analysis are Naive Bayes (NB), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Random Forest (RF). We used the default set hyperparameters for each of these models.

We also created an ensemble model (denoted by EN) consisting of the aforementioned classification models and recorded the prediction accuracy obtained by it.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | NB | LR | KNN | SVM | RF | EN |
| 90% | 10% | 89.47% | 98.25% | 96.49% | 100.00% | 94.74% | 98.25% |
| 85% | 15% | 91.86% | 97.67% | 97.67% | 100.00% | 96.51% | 98.84% |
| 80% | 20% | 91.23% | 96.49% | 97.37% | 97.37% | 96.49% | 96.49% |
| 75% | 25% | 91.61% | 96.50% | 95.80% | 96.50% | 96.50% | 95.80% |
| 70% | 30% | 90.64% | 95.32% | 96.49% | 97.66% | 97.66% | 96.49% |
| 65% | 35% | 91.00% | 95.00% | 96.50% | 97.50% | 95.50% | 97.00% |
| 60% | 40% | 91.23% | 94.30% | 96.05% | 96.93% | 94.30% | 96.93% |
| 55% | 45% | 92.22% | 94.16% | 95.33% | 96.50% | 94.16% | 97.28% |
| 50% | 50% | 93.33% | 95.09% | 95.79% | 97.19% | 95.09% | 96.84% |

Table 1 : Comparative Analysis of Classification Models

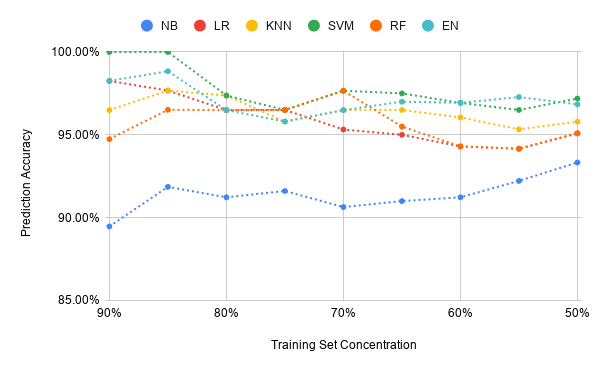


Figure 5 : Predictive Accuracy vs Training Set Concentration

We observed that the SVM model consistently outperformed all the other classification models. Delivering a **97.19%** predictive accuracy for the most extreme test condition (50% training and test set split). This observation is in line with the one made by **Christobel, A. and Sivaprakasam, Y.** (2011).

**Deep Learning Models** : In our analysis of the deep learning models, we first tested the neural network model we developed with a couple of well-known optimizers (**RMSProp** and**Adam**) from the Keras library. In both cases, we used the hyper-parameters enlisted in Table - 2.

|  |  |  |
| --- | --- | --- |
| Parameter | Description | |
| Kernel Initializer | Uniform | |
| Loss | Binary Crossentropy | |
| Activation Functions | 1st Hidden Layer | ReLU |
| Output Layer | Sigmoid |

Table 2 : List of Hyperparameters

In Tables – 3 and 4, we recorded the predictive accuracy obtained by those models. Whereas in Tables – 6 and 7, the number of epochs needed for the models to converge is recorded.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| 85% | 15% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 99.42% |
| 80% | 20% | 97.37% | 97.37% | 97.37% | 97.37% | 97.37% | 97.37% |
| 75% | 25% | 97.20% | 97.20% | 97.20% | 97.20% | 97.90% | 97.90% |
| 70% | 30% | 97.08% | 97.08% | 97.08% | 97.08% | 96.49% | 97.08% |
| 65% | 35% | 97.50% | 97.50% | 97.50% | 97.50% | 97.50% | 97.50% |
| 60% | 40% | 97.81% | 97.81% | 97.81% | 97.81% | 97.37% | 96.93% |
| 55% | 45% | 97.67% | 97.67% | 97.67% | 97.67% | 97.28% | 96.89% |
| 50% | 50% | 97.19% | 97.19% | 97.19% | 97.19% | 96.84% | 96.49% |

Table 3 : Median Accuracy with RMSProp Optimizer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| 85% | 15% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| 80% | 20% | 97.37% | 97.37% | 97.37% | 97.37% | 97.37% | 97.37% |
| 75% | 25% | 97.20% | 97.20% | 97.20% | 97.20% | 97.20% | 97.20% |
| 70% | 30% | 97.08% | 97.08% | 97.08% | 97.08% | 97.08% | 97.08% |
| 65% | 35% | 97.50% | 97.50% | 97.50% | 97.50% | 98.00% | 98.00% |
| 60% | 40% | 97.81% | 97.81% | 97.81% | 97.81% | 98.25% | 97.81% |
| 55% | 45% | 97.67% | 97.67% | 97.67% | 97.67% | 97.28% | 97.28% |
| 50% | 50% | 97.19% | 97.19% | 97.19% | 97.19% | 97.19% | 96.84% |

Table 4 : Median Accuracy with Adam Optimizer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 160 | 125 | 102 | 92 | 81 | 80 |
| 85% | 15% | 172 | 134 | 110 | 99 | 91 | 83 |
| 80% | 20% | 154 | 108 | 80 | 64 | 52 | 43 |
| 75% | 25% | 131 | 92 | 70 | 56 | 48 | 43 |
| 70% | 30% | 180 | 127 | 96 | 77 | 64 | 55 |
| 65% | 35% | 220 | 164 | 121 | 98 | 86 | 71 |
| 60% | 40% | 258 | 189 | 143 | 113 | 94 | 73 |
| 55% | 45% | 282 | 203 | 151 | 121 | 100 | 71 |
| 50% | 50% | 270 | 211 | 164 | 132 | 95 | 67 |

Table 5 : Median Epoch Count with RMSProp Optimizer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 124 | 105 | 87 | 82 | 74 | 76 |
| 85% | 15% | 120 | 111 | 90 | 83 | 78 | 75 |
| 80% | 20% | 102 | 81 | 67 | 56 | 47 | 39 |
| 75% | 25% | 76 | 60 | 49 | 42 | 37 | 32 |
| 70% | 30% | 118 | 93 | 75 | 61 | 52 | 45 |
| 65% | 35% | 145 | 140 | 94 | 77 | 61 | 53 |
| 60% | 40% | 178 | 142 | 112 | 91 | 72 | 61 |
| 55% | 45% | 190 | 151 | 119 | 98 | 81 | 65 |
| 50% | 50% | 185 | 149 | 124 | 108 | 86 | 72 |

Table 6 : Median Epoch Count with Adam Optimizer

**Step Decay Model** : In an attempt to obtain even better prediction accuracies, we developed a model that is capable of learning adaptively. Whereas it starts its training process with a relatively high learning rate of 0.1 and then as the training progresses, the learning rate continues to attenuate in a step-wise manner, halving after every 10 epochs.

This enables the model to learn the optimal weights relatively early in its training process as it can make comparatively larger adjustments to them. Then in the later phases, it primarily fine-tunes the weights till no further optimization can be done.

The model uses the same set of hyper-parameters enlisted in Table - 2. However, it uses is Stochastic Gradient Descent (SGD) as optimizer.

In Table – 7 we have recorded the median predictive accuracies obtained by the model in different test scenarios. In Table – 8 the median number of epochs needed by the model to converge is recorded.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 94.74% | 98.25% | 100.00% | 100.00% | 100.00% | 100.00% |
| 85% | 15% | 95.35% | 97.67% | 100.00% | 100.00% | 100.00% | 100.00% |
| 80% | 20% | 94.74% | 96.49% | 96.49% | 96.49% | 97.37% | 97.37% |
| 75% | 25% | 94.41% | 96.50% | 97.20% | 97.20% | 97.20% | 96.50% |
| 70% | 30% | 92.98% | 94.15% | 97.08% | 97.66% | 97.66% | 97.66% |
| 65% | 35% | 92.50% | 93.50% | 96.50% | 97.00% | 98.00% | 98.00% |
| 60% | 40% | 92.54% | 92.54% | 95.61% | 97.37% | 97.81% | 97.81% |
| 55% | 45% | 93.00% | 93.00% | 94.94% | 96.50% | 97.67% | 97.67% |
| 50% | 50% | 94.04% | 93.33% | 95.44% | 96.84% | 97.19% | 97.54% |

Table 7 : Step Decay Model

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | BS = 32 | BS = 16 | BS = 8 | BS = 4 | BS = 2 | BS = 1 |
| 90% | 10% | 203 | 209 | 179 | 47 | 41 | 39 |
| 85% | 15% | 205 | 211 | 179 | 49 | 44 | 44 |
| 80% | 20% | 203 | 208 | 189 | 169 | 36 | 27 |
| 75% | 25% | 205 | 200 | 154 | 52 | 27 | 22 |
| 70% | 30% | 206 | 203 | 180 | 59 | 35 | 26 |
| 65% | 35% | 206 | 208 | 192 | 170 | 65 | 31 |
| 60% | 40% | 208 | 209 | 215 | 220 | 180 | 50 |
| 55% | 45% | 208 | 208 | 215 | 224 | 228 | 68 |
| 50% | 50% | 209 | 206 | 198 | 178 | 66 | 57 |

Table 8 : Epoch Count Step Decay Model

|  |  |
| --- | --- |
| **C:\Users\Arnab\Downloads\chart.png** | **C:\Users\Arnab\Downloads\chart (1).png** |

Figure 6 : Comparison of Predictive Accuracy and Epochs

Our model achieved a median accuracy of **97.54%** for the most extreme test condition (50% training and test set split). Bettering the predictive accuracy of **97.19%** achieved by SVM and the other two Deep Learning models in our study itself.

**CONCLUSION**

In this study, our aim was to build an efficient predictive model capable of diagnosing breast cancer with high levels of accuracy and contribute to the noble research work conducted by scholars worldwide to combat this fatal disease. We found deep learning methodologies to be very effective in achieving the goal.

However, we did observe that the time taken by the deep learning models to converge were often greater than classification models. Furthermore, convergence time tended to increase with a decrease in batch size, even for the same training set concentration.

While this additional time cost incurred often resulted in greater predictive accuracy, in a real-life scenario, however, this can turn out to be a trade-off, especially if time is in limited availability. Thus, we believe that there is scope for future improvement in this area with the development of even more time-efficient predictive models.

**ACKNOWLEDGMENT**

We thank the donors and the personnel at the University of Wisconsin for helping build the Wisconsin Breast Cancer dataset. We also thank the peer reviewers whose comments were crucial in improving this manuscript.

Finally, I would like to dedicate this work to my father, who left us too soon at the young age of 54 owing to pancreatic cancer. He would have been proud to see me complete this work had he been alive today.

# **REFERENCES**

Shahnaz, C., Hossain, J., Fattah, S. A., Ghosh, S., & Khan, A. I. "Efficient approaches for accuracy improvement of breast cancer classification using wisconsin database." *IEEE Region 10 Humanitarian Technology Conference (R10-HTC)* (2017): 792-797. IEEE.