A Comparative Study of Multilayer Perceptron and Support Vector Machine on Dermatological Disease Classification

Abilash Nair
abilash.nair.2@city.ac.uk

Abstract

Erythemato-squamous dermatological diseases present a challenge to physicians as diagnosis has proven to be difficult. The underlying symptoms for many of these diseases are common, making it difficult to differentiate and identify the disease responsible. This paper will attempt to investigate the effectiveness of two supervised neural network machine learning techniques, namely multilayer perceptron (MLP) and support vector machine (SVM), in classifying the disease accurately. Basic models will be built and hyperparameter tuning will establish the parameters needed to build effective models. Validation of the models will be established through k fold cross validation and the comparisons between models will be carried out by looking at various metrics, confusion matrices, training error plots, etc. The paper will show that the support vector machine (SVM) technique is better than the multilayer perceptron technique with regards accuracy.

1. Introduction

Erythemato-squamous diseases refer to the dermatological conditions that mainly affect the face and scalp. The disease presents itself in a variety of pathological conditions such as inflammation, itchiness, redness of the skin, hives, burning, irritation, etc. [7]. The main diseases that belong to this class are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris [8]. These types of diseases are quite challenging to diagnose as many of the underlying manifestations are common making it difficult to accurately differentiate the actual condition causing the illness.

The two neural network techniques that will be used to build a classification engine are the feedforward multilayer perceptron (MLP) and support vector machine (SVM). These two techniques were chosen as they are both supervised machine learning techniques. Furthermore, the effectiveness of these techniques can be contrasted with Ubeyli's paper in 2009 [1], where the author uses MLP as well as convolutional neural networks (CNN) to build a classification model. The MLP technique will serve as a benchmark and a comparison can be made with SVM. Ultimately, the goal of this paper will be to evaluate the effectiveness of MLP versus SVM in accurately classifying the disease.

2. Dataset Description and Analysis

The dermatology dataset was sourced from the UCI Machine Learning Repository [8]. The dataset contains 34 features, 366 observations, as well as a classification label. Most of the features are linearly scaled ranging from 0 to 3. Zero denotes absence of that feature and a three denotes severe presence. Values 1 and 2 denote intermediate severity. The family history of the patient is the only ordinal and binary feature in this dataset. Most of the dataset was intact apart from 8 instances of missing data in the age column. Due to the small size of the dataset, the mean value of the age column was used to impute the missing data. This method allows data to be preserved without disturbing the underlying distribution.

Scaling of the variables was required, and a standard scaling algorithm was used to scale the features. The distributions of the features were plotted using histograms. Since most

of the features are scaled from 0 to 3, the distributions are not continuous but discrete. The histograms reveal that features such as age are normally distributed. However, most features are skewed quite significantly indicating that some histopathological features are extremely common among all classes of disease. Interestingly, the distribution of the classes (labels) indicate that the dataset is severely imbalanced. The most common disease in the dataset appears to be psoriasis indicated by class 1 and the least common disease is pityriasis rubra pilaris with only around 20 occurrences.

Correlation plots were initially generated using the Pearson correlation factor. However, the Pearson correlation heatmap might be inappropriate as it is meant for continuous variables. A different correlation coefficient is required as technically most of the features are discrete and categorical in nature. The Cramer's V coefficient will be a better predictor of correlation between the features in this dataset. As it is prone to over-estimate the correlation, a correction factor was added [9]. The heatmap reveals that most features are not correlated, with only a few showing some weak correlation.

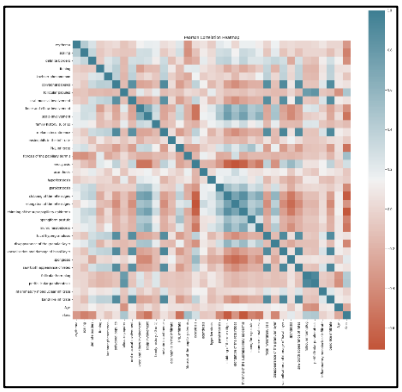

Figure 1. Pearson Correlation Heatmap

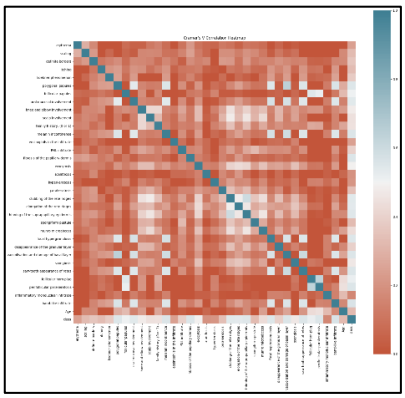

Figure 2. Cramer's V Correlation

3. Multilayer Perceptron (MLP) Model

Multilayer perceptron neural network is a feedforward neural network that resembles the neural structure of the human brain to solve very challenging problems. The basic building block of an MLP is the perceptron. There are usually three fully connected layers in an MLP, an input layer, a hidden layer, and an output layer. Each layer's output is defined by an activation function such as the sigmoid. Sometimes the MLP is defined by two hidden layers. The MLP is a supervised learning algorithm and each layer in the MLP has weighting coefficients which can be adjusted to optimise convergence. A popular training algorithm is the Stochastic Gradient Descent (SGD). The data is fed into the input layer and carried forward through the network, where each time the activation function processes and transforms the data. The most used technique for learning is known as backpropagation, which computes the error between the expected output and actual output. The error is then 'backpropagated' to the previous layers, and the weights are adjusted in each layer to minimise the error.

There are multiple advantages to using MLP models. It does not require any underlying probabilistic knowledge of the inputs especially probabilistic distributions and densities of the inputs. With just one or two hidden layers, it can be considered a universal approximator. The MLP is model is also very adaptive and good at generalisation. However, there are disadvantages to using an MLP model as well. It is computationally expensive to train a model. Furthermore, hyperparameter tuning is computational resource intensive as well as time

consuming. With large datasets or large networks, convergence and optimisation can become problematic.

4. Support Vector Machine (SVM)

SVM is another supervised learning neural network that classifies data by attempting to find a hyperplane in a multi-dimensional feature space. However, in a multidimensional feature space, there might be multiple hyperplanes that classifies the data. SVMs work by choosing the hyperplane that carries the most distance from clusters of data points in each class. SVM uses a loss function to find the maximal distance between clusters. Although, its primary function is as a linear classification model, SVMs can be made to classify in a non-linear fashion by a method known as kernel-trick. Kernels are essentially mathematical functions that transform the data. Kernels can be linear, polynomial, radial basis functions, etc.

Advantages of using SVMs are numerous. It is memory efficient as it only requires a subset of data points for training the network. It can be used in high dimensional feature spaces and in cases where the classes exceed the total number of features in a dataset. Compared to an MLP, optimisation is generally easily achieved. Disadvantages of using SVMs are that it is a lot easier to overfit the data and getting probability estimates from the model is harder.

5. Hypothesis

In multiple published papers, the consensus is that SVM outperforms MLP when it comes to classifying dermatological disease data. According to Ubeyli's paper in 2009 [1], SVM showed better performance than the Recurring Neural Network (RNN) and MLP networks, with SVM's total classification accuracy going as high as 98.32% as compared to 96.65% and 85.47% for other networks. In Abdi's and Giveki's paper published in 2013 [2], a comparative study between PSO-SVMs and AR-MLP was carried out with the PSO-SVM outperforming. Other papers [3][4], also show similar results. This paper proposes that the effectiveness of the SVM can be replicated, and a comparative analysis of both techniques will result in the SVM outperforming the MLP. Furthermore, the paper will attempt to improve the general effectiveness of both neural networks by implementing SMOTE [5] to balance the dataset.

6. Training and Evaluation Methodology

The dataset was split into the training and test datasets in a 75% to 25% split accordingly with a random seed state of 42. Since the classifier labels start with a zero in MLP neural networks, the class label of the dataset ranging from 1 to 6 had to be re-ranged. The classifier labels had one subtracted from it so that it ranges from 0 to 5. After splitting the dataset, the training and test dataset features were scaled separately to prevent leakage. To correct the imbalance, SMOTE technique was used to oversample the minority classes using the imbalanced-learn library. For MLP, a randomized grid search function from sklearn was used for hyperparameter tuning as it was less computationally expensive. The tuning was optimised for accuracy and 5-k fold cross validation was implemented for better generalisation. Early stopping was implemented to avoid overfitting the data. For SVM, the same randomised search function was used for hyperparameter tuning and the hyperparameters included the cost, gamma, and kernel function. Different kernels were compared and evaluated. Since SVM proved to be less computationally expensive, a 10-k fold cross validation was implemented.

Evaluation of the models was carried out on the test dataset. The metrics such as accuracy, recall, and f1 scores were used to compare the models. Furthermore, confusion

matrices were used to evaluate the performance of each model, with special emphasis on the misclassified data. In this paper, the accuracy score was the most important metric used to judge the performance of each model and as a metric to compare the models.

7. Choice of Parameters and Experimental Results

A preliminary MLP model was built using three layers. The input layer had 34 neurons corresponding to the 34 features in the dataset. The hidden layer was arbitrarily composed of 10 neurons. The output layer consisted of 6 neurons, one for each class. The activation function of the hidden layer was the rectified linear function (ReLU) and the output layer activation function was softmax as the model is attempting to solve a multi class classification problem. Softmax provides a probability of each class occurrence and the best probabilistic estimate becomes the final classification. Dropout was instituted in the neural net to help with regularisation. For the loss criterion, the cross-entropy loss function was chosen as it is more effective in multi-class classifiers. The stochastic gradient descent (SGD) was chosen as the optimiser. The hyperparameters included the learning rate, maximum number of epochs, hidden layer neurons, dropout rate, weight decay, and momentum. Similarly for SVM, hyperparameter tuning was used to establish the optimum kernel, cost, and gamma values. Compared to an MLP, the number of hyperparameters are low for an SVM and as a result convergence and speed of computation were better. A 10 k-fold cross validation strategy was used as a result. The cost was varied from 0.1 to 100, the gamma was varied from 0.001 to 1, and the RBF, polynomial, sigmoid, and linear kernels were evaluated.

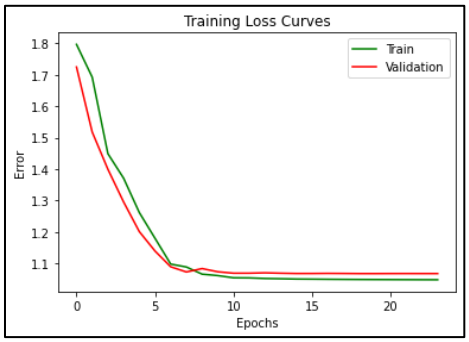

Figure 3. MLP Training Loss Curves

The top 10 ranked hyperparameter tunings is evaluated based on accuracy. For the MLP model, the best instance shows an accuracy rating of 98.94%. The weight decay for the best estimator is zero, momentum set at 0.8, learning rate set at 0.7, and dropout set to 0. The best estimator also suggests that the maximum number of epochs should be limited to 32, perhaps due to the early stopping mechanism making higher number of epochs redundant. The number of hidden layers was set to 22, which is more than twice the number of neurons in the preliminary model. For SVM, the best test score was 98.72%. The tuning recommends that the kernel should be linear, with a gamma of 0.01 and cost of 100. Cross validated results show the SVM outperforming the MLP. SVM accuracy approaches 99.25% as compared to MLP's 98.22%.

8. Analysis and Critical Evaluation of Results

The hyperparameters for the best estimator in a MLP model are quite interesting. The MLP model seems to quickly train itself in a few epochs as indicated by the maximum number of epochs, which was set to 32. Using SMOTE to balance the data could explain the small training

cycles and time needed to train the network. The number of neurons in the hidden dimension was set to 22. According to Hagen's neural network design textbook [10], the number of nodes in a hidden layer could be the mean of the input and output neurons. For instance, in this dataset 34 input neurons and 6 output neurons will give a mean of 20. The number suggested by tuning lies remarkably close to this number.

SVM hyperparameter tuning revealed, quite unsurprisingly, that the best kernel choice for the SVM is linear. This was expected as the dataset features were linearly scaled, and the classification does not require any non-linearities in the output. Although the majority of the ranked instances prefer linear kernels, two instances of RBF and sigmoid kernels are also seen. The cost was set to 100, which was also expected as the emphasis was on accuracy. A small cost could lead to a higher number of misclassifications. Again, for almost all instances, the gamma value was quite small. A higher gamma could lead to overfitting and the value must be carefully chosen. A gamma value of 0.01 for the best predictor is in line with expectations.

A comparative study on the effectiveness and performance of the models was carried out by looking at the accuracy score, recall score, f1 score, and confusion matrix plots. The accuracy score paints a clear picture with the SVM model outperforming the MLP model marginally. The 99.25% SVM accuracy slightly outperforms the MLP's 98.22%. This is quite surprising as a variety of papers [1][2][6], has the SVM outperforming the MLP by a comfortable margin. The SMOTE sampling technique to convert the dataset into a balanced dataset has vastly improved the effectiveness and accuracy of the MLP model. The models were used with the test data to look at classification parameters. The recall score indicates that the MLP model has some trouble with classes 0 (psoriasis) and 3 (pityriasis rosea). Similarly for SVM, the recall score is slightly lower for class 3 as well. The SVM just manages to outperform the MLP model by only having a lower recall score for one class. The f1 scores also suggest that the SVM model is slightly better than the MLP model.

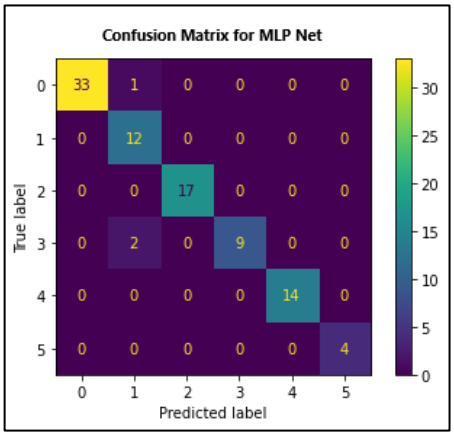

Figure 5. Confusion Matrix for MLP

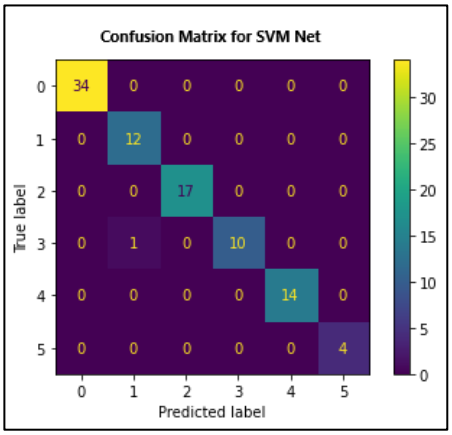

Figure 6. Confusion Matrix for SVM

Looking at the confusion matrices for both models reveals the actual performance of the models on the test data set. The MLP confusion matrix indicate that there are a total of three misclassified observations. As seen in the recall and f1 scores, the MLP model has some difficulty with classes 0 and 3 with 1 and 2 instances of misclassifications accordingly. The TPR for class zero is 97.05% and FNR is 2.95% whereas for class 3 the TPR is 81.8% and FNR is 18.2%. The MLP model misclassifies the class 3 instances as class 1 instead. For the SVM model, the class 3 instances are misclassified. The TPR for class 3 is 90.9% and FNR is 9.1% whereas for class 0 SVM achieves perfect classification. The SVM TPR for class 3 exceeds the

MLP TPR by a significant margin. The slightly worse performance of the MLP model could be due to a lack of weights penalising wrong classifications. The advantage of SVMs over MLPs is quite clearly documented [6], and therefore it should not be surprising that the SVM model is better. However, the paper assumed that the margin would be a lot wider. The results show that the MLP model closely matches the performance of the SVM model.

9. Conclusions, Lessons Learned, and Future Work

The analysis of both the models reveals that the MLP and SVM neural networks are highly effective in multi class classification problems. The accuracy scores of a fully trained and tuned neural network, be it MLP or SVM, ranges in the upper 90^{th} percentile. The paper hypothesized that the SVM model will outperform the MLP model when it comes to classifying erythemato-squamous dermatological diseases. The hypothesis was proven to be correct as the SVM did have better performance. However, a well-trained MLP model can match the performance of a SVM model. In this analysis, the SVM models were a lot easier to train and the computational resources needed to train and validate an SVM model was not high. However, the MLP proved to be quite difficult to train, and resource limitations inhibited the training effectiveness of the MLP model. In the future, the MLP network can be trained on GPUs and much more powerful machines so that the training will not be inhibited. SMOTE proved to be highly effective in balancing the dataset and improving the accuracy of the MLP model.

References

[1] Übeyli, E., 2008. Multiclass support vector machines for diagnosis of erythemato-squamous diseases. Expert Systems with Applications, 35(4), pp.1733-1740.
[2] Abdi, M. and Giveki, D., 2013. Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules. Engineering Applications of Artificial Intelligence, 26(1), pp.603-608.
[3] Kecman, V. & Kikec, M. "Erythemato-Squamous Diseases Diagnosis by Support Vector Machines and RBF NN" in Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 613-620.
[4] Karabatak, M. & Ince, M.C. 2009, "A new feature selection method based on association rules for diagnosis of erythemato-squamous diseases", Expert systems with applications, vol. 36, no. 10, pp. 12500-12505.
[5] Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P. 2002;2011;, "SMOTE: Synthetic Minority Over-sampling Technique", The Journal of artificial intelligence research, vol. 16, pp. 321-357.
[6] Caruana, R. & Niculescu-Mizil, A. 2006, "An empirical comparison of supervised learning algorithms", ACM, , pp. 161.
[7] Service, O., 2021. erythematosquamous dermatosis. [online] Ebi.ac.uk. Available at: <https://www.ebi.ac.uk/ols/ontologies/efo/terms?short_form=EFO_1000695> [Accessed 12 April 2021].
[8] Archive.ics.uci.edu. 2021. UCI Machine Learning Repository: Dermatology Data Set. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/Dermatology> [Accessed 12 April 2021].
[9] Bergsma, W. 2013, "A bias-correction for Cramér's V and Tschuprow's T", Journal of the Korean Statistical Society, vol. 42, no. 3, pp. 323-328.
[10] Hagan, M., Demuth, H., Beale, M. and De Jesús, O., 2016. Neural network design. [S. l.: s. n.].

Appendix 1 – Glossary

Synthetic Minority Oversampling Technique (SMOTE) – an oversampling technique which generates synthetic samples by randomly choosing k nearest neighbours, connecting datapoints lying close together in the feature space [5].

Multilayer Perceptron (MLP) – a feedforward neural network that is defined by an input, hidden, and output layer. Uses an optimising function and an error correcting algorithm such as backpropagation to train the network.

Support Vector Machine (SVM) – a neural network that attempts to find a hyperplane separating clusters of datapoints belonging to different classes by the largest margin in the feature space.

Perceptron – is a single layered neural network that is used as a binary classifier. It takes in an assortment of inputs and attempts to classify it through supervised learning.

Softmax – an exponential function that outputs a probability distribution. It outputs a vector which contains the probabilistic outcomes. Each element in the vector will add up to 1.

Regularisation – attempts to modify the data by either adding or omitting information to prevent overfitting.

Generalizability – is a concept in machine learning which refers to the ability of a machine learning model to predict accurately when it comes across new or unseen data. It refers to the model's adaptive capabilities when new data is fed into the model.

Cramer's V – a correlation coefficient used for categorical data. It tends to overestimate the correlation therefore correction factors need to be introduced.

Epoch – refers to a training cycle where the entire dataset passes forward through the network and backpropagates to complete one set of training. Usually a couple of epochs are needed to fully train a network.

Stochastic Gradient Descent (SGD) – an optimising function which attempts to smoothen the ML model. It iteratively tries to find the best fit by comparing the actual output and expected output.

Cross Entropy Loss – an error function also known as a log loss function which is used for backpropagation in a neural network. The loss increases when the actual output diverges from the expected output by a large margin. Usually used for classification problems.

Cross Fold Validation – refers to a technique in machine learning where the training dataset is partitioned into multiple segments known as k-folds. The portioned segment consists of the training dataset and validation dataset. The ML model is then trained and validated on multiple k folds.

Kernel – a mathematical function used to transform the data in a SVM neural network.

Activation function – a mathematical function associated with each layer in a MLP network that maps the data into a targeted output.

Appendix 2 – Implementation Details

MLP Training:

The MLP training is greatly improved by using a scaling algorithm and implementing SMOTE. For the scaling algorithm, a standard scaler function form sklearn was used. The scaler is fitted using the training data and then the train and test data is transformed. SMOTE was implemented using the imbalanced-learn library.

The accuracy improves when early stopping is used when training the data. Ultimately hyperparameter tuning is absolutely essential for the MLP network to find the best values for the hyperparameters. Also take care to define the seed state before training the model. The hyperparameter training results are shown below.

```
Rank: 1
Mean Test Score: 0.9893617021276595 (std: 0.011653671436280126)
Mean Fit Time: 0.32595372200012207 (std: 0.09701457190294462)
Network Parameters: {'optimizer__weight_decay': 0, 'optimizer__momentum': 0.8, 'optimizer__lr': 0.7, 'module__hidden_dim': 22, 'module__dropout': 0, 'max_epochs': 32}
Rank: 2
Mean Test Score: 0.9808281857698468 (std: 0.018292235769128612)
Mean Fit Time: 0.3456891059875488 (std: 0.05024799843852999)
Network Parameters: {'optimizer__weight_decay': 0.01, 'optimizer__momentum': 0.9, 'optimizer__lr': 0.3, 'module__hidden_dim': 16, 'module__dropout': 0.2, 'max_epochs': 64}
Rank: 3
Mean Test Score: 0.9807824296499656 (std: 0.014106694129002954)
Mean Fit Time: 1.9286970138549804 (std: 0.14454100546678916)
Network Parameters: {'optimizer__weight_decay': 0.02, 'optimizer__momentum': 0.2, 'optimizer__lr': 0.2, 'module__hidden_dim': 28, 'module__dropout': 0.3, 'max_epochs': 128}
Rank: 4
Mean Test Score: 0.9765728666209107 (std: 0.022704373876707224)
Mean Fit Time: 0.443543004989624 (std: 0.01652573611459005)
Network Parameters: {'optimizer__weight_decay': 0, 'optimizer__momentum': 0.6, 'optimizer__lr': 0.3, 'module__hidden_dim': 20, 'module__dropout': 0, 'max_epochs': 32}
Rank: 5
Mean Test Score: 0.9765728666209105 (std: 0.02168453920389013)
Mean Fit Time: 0.16990289688110352 (std: 0.021556472981998608)
Network Parameters: {'optimizer__weight_decay': 0.01, 'optimizer__momentum': 0.9, 'optimizer__lr': 0.9, 'module__hidden_dim': 22, 'module__dropout': 0.3, 'max_epochs': 64}
```

Figure Appendix 1. MLP Hyperparameter Tuning

Hyperparameter training is very resource intensive. Set n_iter to 10 and cv must be set between 3 and 5. If cv or n_iter values are beyond this range, model will not converge. If cv is set to 3, n_iter can go up to a maximum of 20. Cross validation must be used to evaluate the effectiveness of the MLP model.

SVM Training:

SVM training is a lot easier to train than MLP. Scaling and oversampling is technically not essential as the accuracy score for the preliminary basic SVM model is already around 98%. The methods described in the paper only improves the model by few tenths of a percentage point.

```
Rank: 1
Mean Test Score: 0.9872340425531915 (std: 0.021697955376990 56)
Mean Fit Time: 0.0027950286865234377 (std: 0.000397973671329911)
Network Parameters: {'kernel': 'linear', 'gamma': 0.01, 'C': 100}
Rank: 1
Mean Test Score: 0.9872340425531915 (std: 0.021697955376990 56)
Mean Fit Time: 0.003769516944885254 (std: 0.0051818164631948 61)
Network Parameters: {'kernel': 'linear', 'gamma': 0.1, 'C': 10}
Rank: 1
Mean Test Score: 0.9872340425531915 (std: 0.021697955376990 56)
Mean Fit Time: 0.0018636465072631836 (std: 0.004673903157466248)
Network Parameters: {'kernel': 'linear', 'gamma': 0.001, 'C': 100}
Rank: 1
Mean Test Score: 0.9872340425531915 (std: 0.021697955376990 56)
Mean Fit Time: 0.0028012752532958984 (std: 0.000401455801789394 37)
Network Parameters: {'kernel': 'linear', 'gamma': 0.1, 'C': 100}
Rank: 5
Mean Test Score: 0.9851063829787234 (std: 0.021382714087491247)
Mean Fit Time: 0.003394889831542969 (std: 0.000486841664056899 05)
Network Parameters: {'kernel': 'sigmoid', 'gamma': 0.001, 'C': 100}
Rank: 5
```

Figure Appendix 2. SVM Hyperparameter Tuning