A Comparative Study of Support Vector Machines & Multilayer Perceptron Using the CTG Dataset

Abstract

This paper attempts to present and critically evaluate two supervised machine learning approaches using Cardiotocography (CTG) data. CTG monitoring is a widely used technique to monitor fetal well-being during pregnancy. We attempted to develop two classification systems, based on the Multilayer Perceptron (MLP) and Support Vector Machine (SVM) approaches. The aim is to identify which system might be more effective in correctly categorising CTG scans into three classes: normal (‘Class 0’), which illustrates normal fetal conditions; suspect (‘Class 1’), where abnormalities might be present, and pathologic (‘Class 2’), where abnormalities have been detected. Given the critical need to avoid False Negative (Type II) errors, our evaluation approach focused on the Recall metric in the comparative evaluation of these approaches. On the basis of our empirical results, the SVM approach outperformed MLP in our model test stage, achieving an overall Recall result of 0.824 (SVM) versus 0.801 (MLP). SVM also outperformed when comparing the Recall test results of the individual classes, with SVM achieving a superior Recall score in the critical ‘Class 2’ test, which is key to avoiding pathological false negatives. Here the SVM approach achieved 0.840 versus 0.800 using MLP. The outperformance of SVM versus MLP was consistent with our Hypothesis that we expected SVM to deliver a superior performance, given the mathematical power of this classification technique.

Introduction & Motivation

Cardiotocography (CTG) monitoring is one of the popular ways to monitor fetal well-being during pregnancy. The primary purpose of fetal surveillance using CTG is to prevent adverse fetal outcomes [1]. Obstetricians use the data such as fetal heartbeat and uterine contractions to determine if a fetus is pathologic or not. Earlier the data derived from CTG was analysed by the obstetricians. This resulted in a waste of time and medical resources. The automatic classification of fetal state can help save obstetrician’s time and improve classification performance [1]. Additionally, it will provide doctors information that can help prevent unnecessary C-sections, and be more efficient in monitoring fetal well-being.

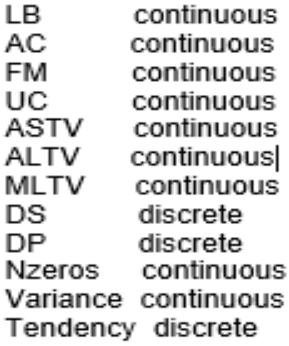
This paper aims to critically evaluate Multilayer Perceptron (MLP) and Support Vector Machine (SVM)). In this case they are specifically devised to automatically classify the Fetal state classes. There are three fetal state classes classified by obstetricians, mainly Normal, Suspect and Pathologic.

Description of the Dataset

The dataset based on processed fetal cardiotocograms with the respective diagnostic features was retrieved from the UCI website [2]. It consisted of 2126 CTG records with 40 attributes. The classification had been undertaken by expert obstetricians, who have classified 176 cases as pathologic, 1655 cases as normal, 295 cases as suspicious.

Data Wrangling and Exploratory Analysis

Features that may be associated with significant fetal compromise and require further action are baseline variability, complicated variable decelerations and late decelerations. File details, dates & timings were considered irrelevant for analysis. Features (i.e., suspect pattern, vigilance, sleep, vagal simulation, shift patterns) which were not medically necessary to directly determine if the fetus was pathologic had to be removed. The features, histogram Median, histogram Mode, histogram Mean had to be taken out in order to get rid of Multicollinearity in the dataset. We took care of the few missing values and duplicate records by dropping them. The outliers in our final dataset are values which can be informative so they weren’t removed. There was a class imbalance present in the dataset.
Class 0 is classified as a normal reading;
Class 1 suspect (requiring further investigation) and
Class 2 is pathologic (where abnormality is present).
This class imbalance was managed using a borderline SMOTE technique. The observation mix, pre the application of borderline SMOTE had 1488 total observations, of which 78% were class 0 (normal), 13% class 1 (suspect) and 9% class 2 (pathologic). The Borderline SMOTE process created 1165 observations in each of the three classes (with all matching the number of the original normal ‘0’ class). Our final dataset datatypes are as represented in the figure-1.



LB	continuous
AC	continuous
FM	continuous
UC	continuous
ASTV	continuous
ALTV	continuous|
MLTV	continuous
DS	discrete
DP	discrete
Nzeros	continuous
Variance	continuous
Tendency	discrete

Fig.1.Dataset- datatypes

Summary of Algorithms Used (MLP vs SVM)

The Multi-Layer Perceptron (MLP) is a commonly used machine learning technique which evolved from earlier studies on the single neuron neural networks. It is a supervised learning approach. MLPs can be used to solve both regression and classification problems by changing the activation function of the output layer and the number of output neurons, allowing the output layer in the network architecture to produce the necessary output for each task. An MLP network is usually trained using the gradient descent process, so that a cost function is minimised. This is achieved

via a process of Backpropagation, where the weights and biases of the model are recalibrated with the aim of minimising the differences (i.e., costs), between actual and predicted values.
Support Vector Machines (SVM) originated out of a binary classification approach. The nature of the approach is that the algorithm finds a dividing hyperplane between different classes of variable. This dividing hyperplane maximises the distance between the nearest points of each of the different classes. To do this, it focuses on the nearest class vectors, namely the 'support' vectors, and ignores the points of each class which are further away from this dividing plane.

The approach has been generalised to handle multi class classification problems. Where the core approach uses a 'one vs one' approach for binary classification, this has been adapted to a 'one vs all' approach to deal with multi class classification, therefore 'all' other classes are handled as a single class. SVMs, like MLPs, can be utilised in classification or regression by using different loss functions.

SVM's take on an approach called the Kernel trick, which allows SVM to solve data separation tasks which are non-linear in nature. Starting from a low dimensional space, data is mapped into a high dimensional space using the Kernel trick. This allows the separating hyperplane to be discovered, before being projected back into the initial lower dimensional space. Common kernels, which takes data as an input and transforms it as instructed, are linear (which finds the best straight-line divider), polynomial (for more complex classification) and radial basis function (rbf), with the last two more suited for complex data distributions. Key parameters for SVMs include C, the cost of regularisation, which determines how much misclassification can be tolerated (low C means more misclassification is allowed). Another key variable is Gamma, where high Gamma means that only training set values beside the class boundary (the support vectors) are considered in determining the location of the hyperplane. Finally, the margin is the smallest perpendicular distance from the hyperplane to the support vectors (where margin is typically maximised).

MLP – Key Pros

- Highly flexible with ability to adapt many features, including: network design (network depth, number of neurons, activation functions, learning rates, epochs.
- Computationally cheap, so can produce fast results.
- Well understood, with broad software & academic support which develops network understanding, e.g., development tools like GridSearch support network development.

MLP – Key Cons

- Typically require a lot of labelled data to train an individual model.
- The high number of parameters can make development of a research 'pipeline' complex.
- Non trivial MLP concepts (e.g., Gradient descent, Backpropagation) can make challenging the use of the technique by non-specialists.

SVM – Key Pros

- Can classify extremely complex dataset patterns.
- Wide array of available kernels offers very high functionality of the technique.
- Very powerful on small to medium sized datasets.

SVM – Key Cons

- Computationally very expensive on large datasets.
- Requires a 'human in the loop' with an advanced understanding of SVM techniques to get the most from them.
- Sensitive to data errors impacting classification results.
- The high level of complexity involved in some SVM concepts (e.g., 'gamma', 'kernel trick').

Hypothesis Statement

SVMs are an extremely powerful mathematical technique in classification. Although MLPs can be combined with tools like GridSearch to optimise performance, the power of SVMs as a classification technique can be difficult to beat if the technique is well designed and the dataset has a low number of errors. Our hypothesis was that we expected our best SVMs to be a more powerful classifier vs our best performing MLP. Our preferred evaluation metric is Recall, given the imbalanced nature of the dataset and the need to avoid Type II (False Negative) errors in the CTG technique. Our SVM model Test produced an overall Recall score of 0.824 versus the MLP Test Recall score of 0.801, a +2.9% superior performance. Although this only represents a single event, such that it is not statistically significant, this result tentatively met our initial expectation that our best performing SVM would outperform our best performing MLP.

Experiments & Evaluation Methodology

Before reviewing the methodology used in our experimental process, it's worth repeating the important issue of the class imbalance present in the dataset. Here, class 0 is classified as a normal reading; class 1 suspect (requiring further investigation) and class 2 is pathologic (where abnormality is present). This class imbalance was managed using a borderline SMOTE technique. The observation mix, pre the application of borderline SMOTE had 1488 total observations, of which 78% were class 0 (normal), 13% class 1 (suspect) and 9% class 2 (pathologic). The Borderline

SMOTE process created 1165 observations in each of the three classes (with all matching the number of the original normal ‘0’ class.

We then progressed to constructing the multilayer perceptron. We have used multiple approaches using 1) MLP classifier from Sklearn, creating a multilayer perceptron, and 2) using PyTorch with skorch for our multilayer perceptron. Our first MLP classifier approach used the GridSearch functionality from Sklearn. We have tried to experiment with different parameters to know our best hidden layer sizes, activation functions, and solver for weight optimization. Since False Negatives are critical for this domain problem, the scoring parameter used is recall_macro.

The GridSearch process indicated that the best activation function to use was the ReLU (Rectified Linear Unit), with 2 hidden layers containing 128 and 64 neurons, using an ‘Adam’ optimiser with 200 iterations being used in the training process.

These insights were used to experiment with our intial PyTorch/Skorch MLP model. The process outlined above took us to our Final MLP model, implemented by using a combination of PyTorch & Skorch. The architecture & setup of this model involved: 2 hidden layers, of 128 and 64 neurons respectively; ReLU activation function; a loss function of ‘Cross Entropy Loss’; a dropout rate of 0.1 to help reduce overfitting; 200 maximum epochs & applying a batch size of 100; a learning rate of 0.1 along with a rate decay of 0.001 and an Adam optimiser. Again, as mentioned before, because of the importance of minimising False Negative (Type II) errors in this domain problem, the Epoch scoring parameter is optimised on the basis recall_macro. This is also a more appropriate evaluation measure for an imbalanced data set. We have not used the softmax activation function on the output layer, because the use of the ‘Cross Entropy Loss’ criterion function removed the need to use it.

Moving to the Support Vector Machine (SVM) experiment & evaluation methodology. We conducted a GridSearch process, which stated that the optimum parameters to use were: a radial basis function (rbf) kernel; a ‘C’ value of 1000; degree = 2 & gamma = 1). Although the parameters were established by GridSearch, we were concerned (on the basis of the classification report below) that the model might be overfitting.

Output Class Value	Precision	Recall	F1 Score	Support
0	0.94	0.95	0.94	490
1	0.70	0.72	0.71	98
2	0.86	0.72	0.78	50
accuracy			0.89	638
macro average	0.83	0.80	0.81	638
weighted average	0.90	0.89	0.89	638

Fig.2. SVM Classification Report (Variables: kernel = rbf, C = 1000, degree = 2, gamma =1)

Due to the overfitting concern, we manually adjusted the parameters for our SVM classifier, maintaining the rbf kernel and degree = 2, but changing C = 100 (from 1000) & gamma = 0.1 (from 1). Our experimental results are shown in the next section.

Experimental Results and Final Models

Our Final MLP model described earlier produced the following Recall Score, Accuracy Score & Confusion Matrix for the Train & Test data in turn.

Final MLP Model Training Recall Score, Accuracy Score & Confusion Matrix				**Final MLP Model Test Recall Score, Accuracy Score & Confusion Matrix**			
Final MLP Training Scores				**Final MLP Test Scores**			
Recall Score	0.878			**Recall Score**	0.801		
Accuracy Score	0.878			**Accuracy Score**	0.824		
Confusion Matrix	942	211	12	**Confusion Matrix**	411	75	4
	4	1053	108		8	75	15
	12	80	1073		6	4	40

Fig.3. Results of our final MLP model

The related loss curves during this training & validation process are shown below, where there was no evidence of overfitting in the train & test cycle (as no crossing of the validation set loss curve above the training set seen below):

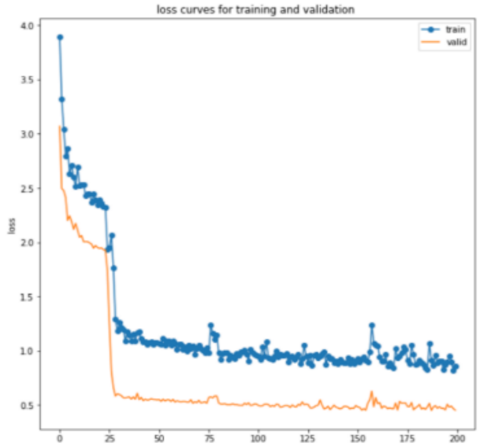


Fig.4. Loss Curves for Final MLP Model (loss y axis vs Epochs x axis)

The final ROC Curve and Confusion Matrix for the Test stage of the Final MLP model are shown below. The confusion matrix is a repeat of the earlier test plot. The ROC curves have values across each of the three classes which are very close to 1.0, indicating that the model is a highly skilled classifier. This is particularly notable for class 2 (pathologic class), which at 0.98 is the closest of all the classes to a perfect skill classifier score of 1.0.

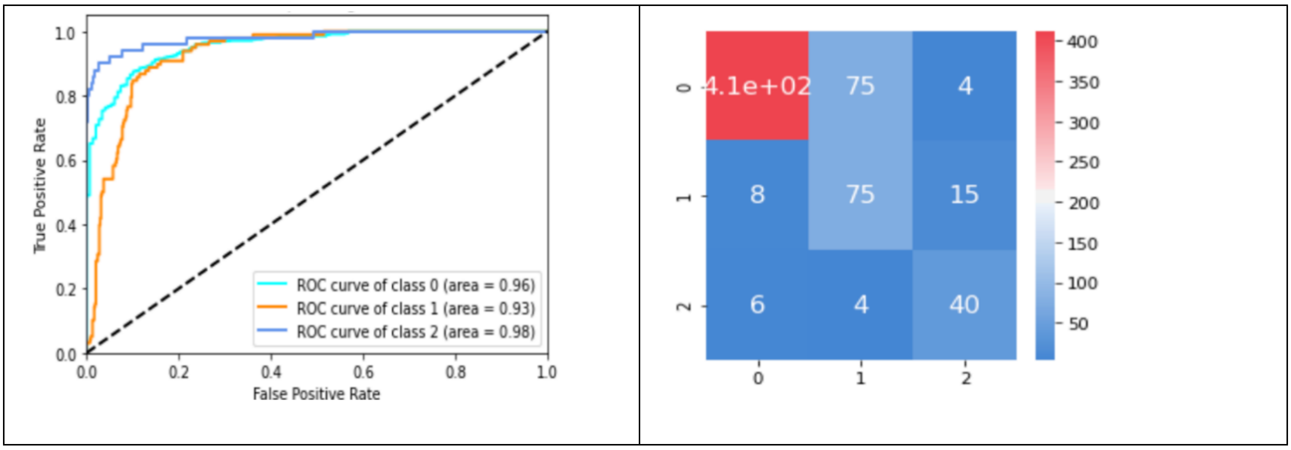


Fig.5. Receiver Operator Characteristic (ROC) Curves (below left) and Confusion Matrix for Final MLP Model

Our Final SVM model produced the following Recall Score, Accuracy Score & Confusion Matrix for the Train & Test data.

Final SVM Model Training Recall Score, Accuracy Score & Confusion Matrix				Final SVM Model Test Recall Score, Accuracy Score & Confusion Matrix			
Final SVM Training Scores				Final SVM Test Scores			
Recall Score	0.906			Recall Score	0.824		
Accuracy Score	0.906			Accuracy Score	0.861		
Confusion Matrix	1005	156	4	Confusion Matrix	434	54	2
	24	1046	95		11	73	14
	17	33	1115		6	2	42

Fig.6. Final SVM model results

The final SVM Model ROC Curve is shown below. The approach produced a ROC curve which has identical values for the Final MLP ROC curve model, indicating that the classifier is a skilled one. As before, this is especially important for class 2 pathologic events, which has the highest score of all three classes.

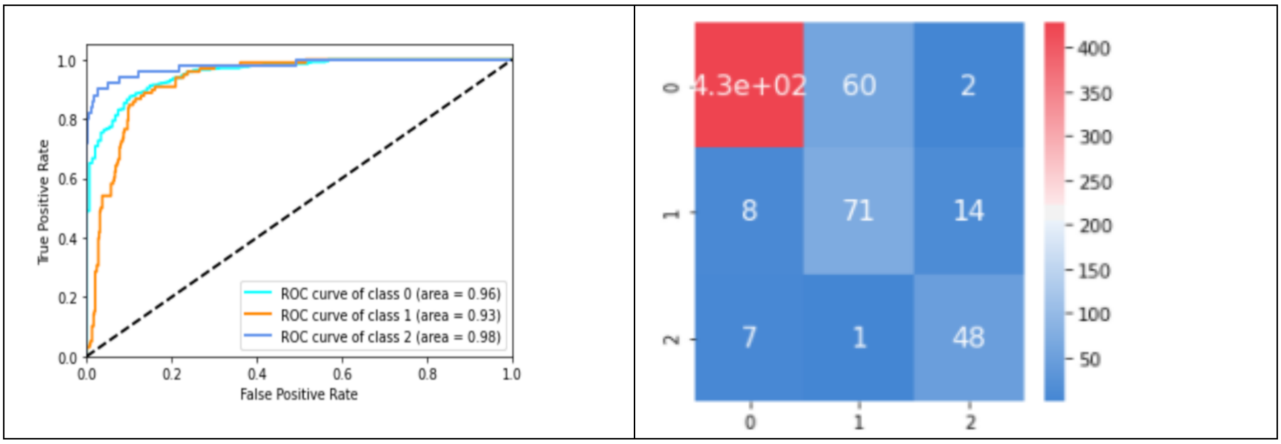


Fig.7. Receiver Operator Characteristic (ROC) Curves (below left) and Confusion Matrix for Final SVM Model

<u>Analysis & Critical Evaluation of the Results</u> - The essence of our analysis is the maximisation of the Recall (or Sensitivity) evaluation technique. The formula for Recall = TP/(TP+FN). Given the importance of avoiding false negatives in the CTG process, we wish to maximise recall, so that false negatives are minimised. This may come at a price of an increase in False positives as we scale up the project but that would be a domain requirement. Compared to our MLP approach, SVM is the superior technique as it produces a higher Recall score at both the overall indicator level, but also at the critical individual Class 2 level. We can demonstrate this in the following analysis, based on the experimental results shown in the last section.

In terms of overall Recall power, our MLP approach generated an overall Recall Score of 0.801. This is a good score. However, our SVM approach generated an overall Recall Score of 0.824, a superior score by 2.9%.

The results are also interesting when we review the recall scores of each individual class within the MLP and SVM techniques. These figures are computed from the confusion matrices of the Final MLP Test Confusion Matrix and the Final SVM Test Confusion Matrix. The two tables below illustrate the TP score of each class under each technique.

Recall Scores of Classes 0,1 & 2 from MLP Test Confusion Matrix		**Recall Scores of Classes 0,1 & 2 from SVM Test Confusion Matrix**	
Recall = TP/(TP+FN)		Recall = TP/(TP+FN)	
MLP		SVM	
Recall 0 Class	0.839	Recall 0 Class	0.886
Recall 1 Class	0.765	Recall 1 Class	0.745
Recall 2 Class	0.800	Recall 2 Class	0.840

Fig.7. Recall Scores for both the models

The table above shows the Recall Score for each of the Classes 0,1 & 2. Under SVM Test, the Recall score for 0, the normal Class, is superior at 0.886 vs 0.839; an improvement in performance of 5.6%. However, although a superior Recall score is important for normal fetal tests, a higher Recall score is key for Class 1 (suspect) and Class 2 (pathologic) fetal tests.

Although SVM has a slightly inferior Class 1 score vs MLP, at 0.745 vs 0.765 (an underperformance of 2.6%), it generates a superior performance in the critical Class 2, which looks to detect pathological issues. Here the Recall score for SVM in Class 2 is 0.840 vs MLP Recall of 0.800; a superior performance of 5%. Although such a difference in performance can be marginal, in practice this would mean that, when Class 2 situations face the healthcare team, SVM will detect 5 more out of every 100 than the MLP approach would. This and the overall superior performance of the SVM approach make it the superior detection technique to employ. Although this is only a single study, and therefore not statistically significant, it is a positive inference for the performance of the SVM approach versus the MLP technique.

This evaluation offers a means to discriminate between the two approaches that goes beyond the mere identical ROC curve performance of both the MLP and SVM approaches, especially since ROC curves can at times lead to errors in imbalanced datasets, as the area of the curve can be sensitive to small changes in values.

As such, the performance of SVM versus MLP matches our hypothesis that it would deliver a superior performance.

Conclusion & Future Work

Based on the work presented in this paper, we believe that a SVM classification approach is superior to its equivalent MLP approach, on the basis of SVM's higher Recall score, particularly in the key Class 2. In terms of future work, it would be interesting to test these results on a broader number of datasets, to better establish statistical significance. It would also be worthwhile to look at retuning the SVM approach, testing different parameters, perhaps closer to those first presented in the GridSearch technique.

References

1] Diogo Ayres-de-campos, João Bernardes, Antonio Garrido, Joaquim Marques-de-sá & Luis Pereira-leite (2000) SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms, Journal of Maternal-Fetal Medicine, 9:5, 311-318, DOI: 10.3109/14767050009053454

2] https://archive.ics.uci.edu/ml/datasets/cardiotocography

3] Hofmann, M., 2006. Support vector machines-kernels and the kernel trick. Notes, 26.

4] Han, S., Qubo, C. and Meng, H., 2012, June. Parameter selection in SVM with RBF kernel function. In World Automation Congress 2012 (pp. 1-4). IEEE.

5] Ruck, Dennis W., et al. "The multilayer perceptron as an approximation to a Bayes optimal discriminant function." IEEE Transactions on Neural Networks 1.4 (1990): 296-298.

Apendix

Glossary

1] Multicollinearity: **Multicollinearity** is the occurrence of high intercorrelations among two or more independent variables

2] borderline SMOTE: An Over-Sampling Method

3] gradient descent: **Gradient descent** is an optimization algorithm used to minimize the error between

4] activation function: weighted sum of the **input** is transformed into an **output** from a node in layers of neural network

5] GridSearch: **Grid search** is a tuning technique to get the optimum values of hyperparameters

6] rbf kernel: **radial basis function kernel in SVM**

7] ROC curve: **receiver operating characteristic curve**,

8] Backpropagation: backward propagation of errors

9] epochs: **Epoch** is when an entire dataset is passed forward and backward through the **neural network** only once.

Implementation details

Since the default optimiser for skorch was SGD, we tried using Stochastic gradient descent (SGD optimiser) in our final optimal MLP model. The model took higher number of epochs to train on our training dataset and varying the batch size resulted in a better accuracy and recall score. However, SGD wasn't as optimal as the Adam optimiser.

In Adam optimiser what we observed was varying the learning rate and weight decay really improved our model performance. We also experimented with sigmoid activation but the confusion matrix results we received for the same showed us that one of the classes was not being predicted.

With the use of only 'recall' (not recall_macro) as our metric in mlp models our accuracy results were 12% hence we chose to ignore it.

The grid search use on SVM originally gave us a overfitted model with an accuracy of 98% for training data and 78 %accuracy for test data. We had to manually try different values of C and gamma to get better models.

Additionally, we have used of 5-fold cross-validation for both SVM and MLP.