Classification of Imbalanced Cardiotocography Data using SMOTE, ADASYN and CNN Techniques

William Schneble[1], Dong Si[2]
[1]University of Washington, Bothell, USA
schnwil@uw.edu, dongsi@uw.edu

Abstract:

Fetal cardiotocography (CTG) is commonly used to monitor the fetal heart rate (FHR) and uterine contractions (UC) during pregnancy and determine the overall well-being of the fetus. Medical intervention, such as a caesarean section, can be performed if the CTG data shows an abnormal FHR pattern that could be indicative of hypoxia or other complications that require immediate action. This project aims at reducing the medical expertise and time required to perform evaluation of the CTG data and to deliver accurate and informative results of the data. However, real world data tends to be imbalanced and noisy and the CTG dataset is no exception. To achieve these goals we have performed several, multiclass, supervised learning models along with SMOTE and ADASYN sampling and have collected their corresponding classification accuracy, precision, recall, and f-scores for comparison, discussion and selection. Re-sampling was found to increase accuracy among under-represented samples at an expense of majority samples but we were also able to reduce false negatives which are of critical importance in preventing damage or death of the fetus and patient. Also combining re-sampling with, grid search and ensemble voting we were able to build an accurate, high recall and F-score classifier.

Keywords:

Machine Learning, Cardiotocography, CTG, medical intervention, Fetal Heart Rate, FHR, Grid Search, SMOTE, ADASYN, Condensed Nearest Neighbor, CNN, class imbalance, neural network

1. Introduction

The use of machines in assistance or consultation on diagnosis is not new, in the 1970's INTERNIST-1, MYCIN, and CADUCEUS were being developed and used experimentally. While MYCIN had an 'acceptable recommendation' about 75% of the time, it and other computer consultants were not used beyond the experimental stages [1]. This was generally due to a few major faults:

- Early models were generic and complexity of interacting diseases, the evolution of a disease or infection over time and other factors proved too intractable for an all purpose model [2]

- Interpretation of results, such as from INTERNIST-1 which assigned and accumulated scores that corresponding to building evidence, are difficult to trace and explain to physicians [2]
- Difference in standards for data collection which can result in missing data or data with different dimensions, is not supported well or at all in classification models

For these reasons our project focuses only on evaluating the fetus' status by only using the CTG data which includes standard, required fields such as uterine contractions, baseline FHR, variability in FHR and presence of accelerations and decelerations. The fetus's FHR is classified into a FHR pattern: normal where no action needs to be taken at time of observation, suspect where continued surveillance and reevaluation is needed, and pathological that requires immediate evaluation and management. Bringing machine learning to the problem can reduce the time medical staff needs to interpret the CTG results and take immediate action if necessary. Also, an accurate model would give confidence to the medical staff as well as the patient that certain standards are being met and followed that do not rely solely on availability to medical expertise which can vary from region to region, person to person.

Sunder and other researchers have implemented supervised artificial neural networks (ANN) and achieved an F-score of 0.9784 for normal, 0.4514 for suspect, and 0.9725 for pathological on the same data this project uses [3]. This project considers several sampling techniques and machine learning algorithms and applies grid search techniques to find the most optimal, fined tuned and accurate model. Recall, which includes false negatives, will be of particular importance among the suspect and abnormal classes because this represents poor fetus well being misclassified and may lead to a missed medical intervention opportunity which could have severe repercussions such as brain damage or death for the fetus.

1.1 The CTG Dataset

The dataset used for this project is from the University of California Irvine (UCI) Machine Learning Repository [4]. There are 23 features and this project uses the 23rd feature, NSP, for classification and the first 21 features as training attributes (see Table 1). The 22nd feature is an alternative class label that will not be used in this project. The set includes 2126 samples with 1655 being normal labels, 295 suspect, and 176 pathological.

Table 1: Data features and descriptions. NSP is used as class label.

Feature	Description
LB - FHR	FHR baseline (bpm)
AC	# accelerations per second
FM	# fetal movements per second
UC	#uterine contractions per second
DL	# light decelerations per second
DS	# severe decelerations per second
DP	# prolonged decelerations per second
ASTV	% time with abnormal short term variability
MSTV	mean value of short term variability
ALTV	% time with abnormal long term variability
MLTV	mean value of long term variability
Width	width of FHR histogram
Min	minimum of FHR histogram
Max	maximum of FHR histogram
Nmax	# histogram peaks
Nzeros	# histogram zeros
Mode	histogram mode
Mean	histogram mean
Median	histogram median
Variance	histogram variance
Tendency	histogram tendency
CLASS	FHR pattern class (not used)
NSP	fetal state code (normal, suspect, pathologic)

2. Strategy and Methods

2.2 Preprocessing

The dataset contains a large class imbalance with the suspect label being under-represented by a factor of 5.6 (1655 / 295) and the pathological label by a factor of 9.4 (1655 / 176). To address this imbalance we can either over sample the minority classes or under sample the majority classes. Because the dataset is already fairly small at 2126 samples and under sampling will throw away some data, we will focus on over sample the minority classes with two oversampling techniques, ADASYN and Synthetic Minority Over-sampling TEchnique (SMOTE) and one undersampling technique, Condensed Nearest Neighbor (CNN).

SMOTE uses a KNN algorithm and creates synthetic data points along line segments to k-nearest neighbors. For extensive oversampling this technique can yield better results than standard replication because it “effectively forces the decision region of the minority class to become more general” [5]. ADAYSN also creates synthetic data to reduce bias from class imbalance and tries to shift the decision boundary toward hard-to-classify labels [7]. The CNN method for under sampling the majority class is also a KNN algorithm that seeks to reduce the samples to points close to the decision boundary [6].

Additional preprocessing includes standardization of the data for distance based algorithms such as k-nearest neighbor (KNN) and support vector machines (SVM). For the tree based models, such as random forest classifier (RFC), the data does not need to be standardized but for simplicity and consistency we will also be using the standardized data for these models.

2.3 Strategy – Grid Search

Knowing the most optimal hyper-parameters or the most effective model ahead of time is generally not possible, not all data is the same in dimensional complexity, noise, or distribution. Thus, our approach is to perform a simple loop for several different models, each iteration trying a different combination of hyper-parameters values and sampling in order to find the most optimal parameters for each model and the most accurate model overall. Scikit-learn has a built in implementation of this strategy called GridSearchCV that also supports cross validation and threading.

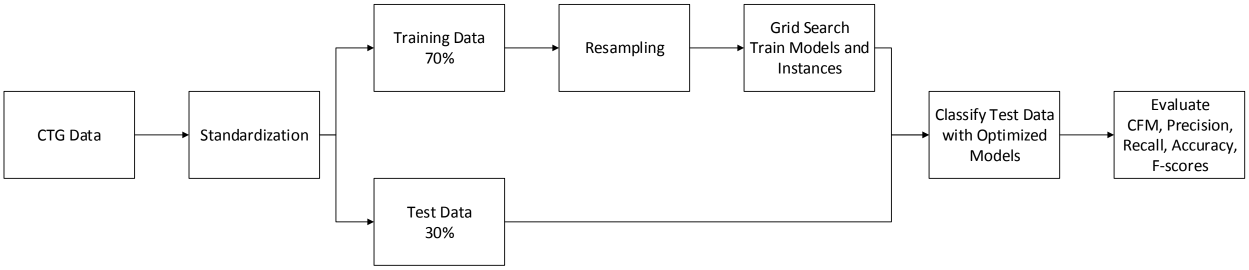
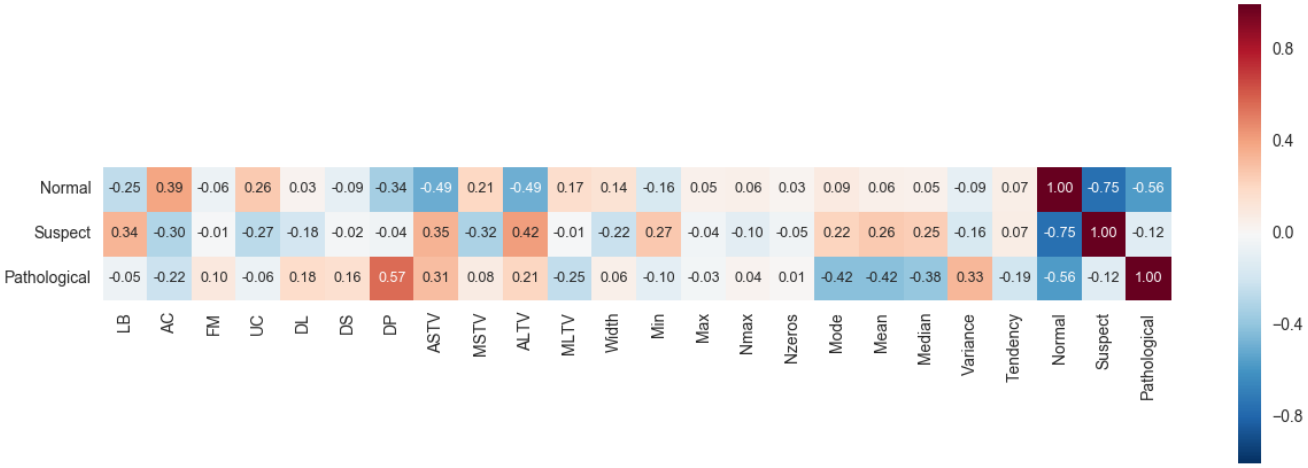


Fig. 1. Flow diagram of overall strategy



	LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	ALTV	MLTV	Width	Min	Max	Nmax	Nzeros	Mode	Mean	Median	Variance	Tendency	Normal	Suspect	Pathological
Normal	-0.25	0.39	-0.06	0.26	0.03	-0.09	-0.34	-0.49	0.21	-0.49	0.17	0.14	-0.16	0.05	0.06	0.03	0.09	0.06	0.05	-0.09	0.07	1.00	-0.75	-0.56
Suspect	0.34	-0.30	-0.01	-0.27	-0.18	-0.02	-0.04	0.35	-0.32	0.42	-0.01	-0.22	0.27	-0.04	-0.10	-0.05	0.22	0.26	0.25	-0.16	0.07	-0.75	1.00	-0.12
Pathological	-0.05	-0.22	0.10	-0.06	0.18	0.16	0.57	0.31	0.08	0.21	-0.25	0.06	-0.10	-0.03	0.04	0.01	-0.42	-0.42	-0.38	0.33	-0.19	-0.56	-0.12	1.00

Fig. 2. Correlation coefficient (R) heat map: a visualization of the covariance matrix (C) that has the equation below (1). Simply, as dimension one increases does dimension two also increase? A value of 1 indicates perfect, positive correlation, -1 indicates a perfect, negative correlation and 0 represents no correlation.

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} + C_{jj}}} \tag{1}$$

If we were to use all the models available in our grid search it would probably be too expensive to run within a realistic timeframe. Using the correlation coefficients, see figure 2, we can see that the data is probably not linearly separable in low dimensional space because no single feature shows strong correlation. The DP feature is moderately correlated with the pathological label, with an R value of 0.57, but in general we can rule out linear models like logistic regression, linear activation functions in NN (see figure 3), and linear kernels in our SVM.

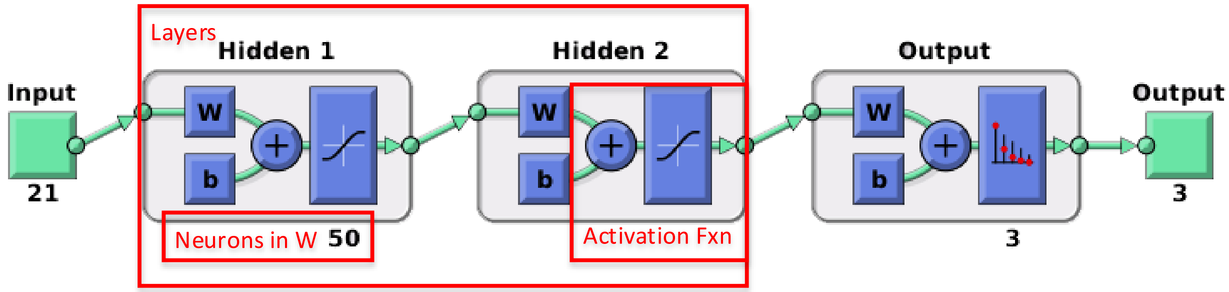


Fig. 3. Example of feed forward neural network in MATLAB with parameters of interest to optimize highlighted (not shown but also considered is the training function).

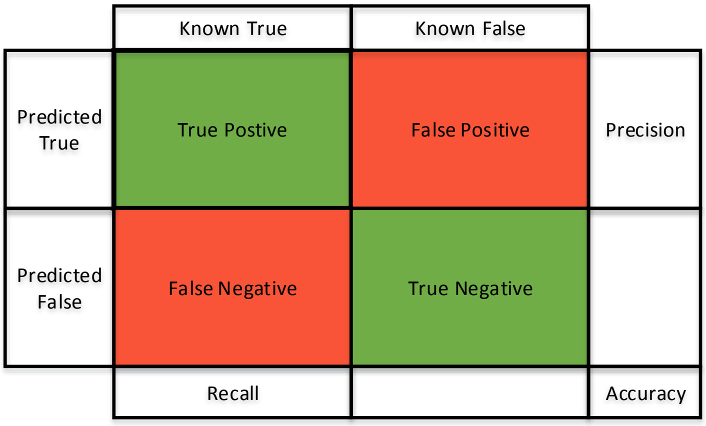
2.4 Metrics for Evaluation

The accuracy is calculated for every classifier. The precision, recall and F-Score are calculated for each class label in each classifier. In addition for data visualization purposes, confusion matrixes are created for each classifier and for MATLAB's PNN an additional performance plot that details the cross-entropy over training epochs.

Accuracy is simply the sum of the correct classifications over the total sample size.

$$Acc = \frac{\sum TP + \sum TN}{Population} \tag{2}$$

Confusion Matrix is a visualization of the true and predicted labels in a grid as shown in figure 4.



	Known True	Known False	
Predicted True	True Postive	False Positive	Precision
Predicted False	False Negative	True Negative	
	Recall		Accuracy

Fig. 4. Confusion Matrix

Precision is a measurement of relevancy of results. It is defined as the sum of true positives over the sum of true positives and false positives.

$$Prec = \frac{\sum TP}{\sum TP + \sum FP} \tag{3}$$

Recall is a measurement of completeness of the relevant results. It is defined as the sum of true positives over the sum of true positives and false negatives.

$$Rec = \frac{\sum TP}{\sum TP + \sum FN} \tag{4}$$

F-Score is a measurement of test accuracy and uses both precision and recall.

$$F1 = 2 * \frac{Prec * Rec}{Prec + Rec} \tag{5}$$

3. Results and Discussion

The accuracy score of the classifiers is a poor indicator of classifier evaluation because of the class imbalance. The accuracy score thus strongly reflects the underlying distribution of the class labels which is evident in confusion matrix using no re-sampling, having a high accuracy and recall in normal classes but some of the lowest recalls for suspect and pathological classes (see Table 2). The confusion matrixes not only let us visualize the extent of the class imbalance but also the classifier's difficulty in classifying the underrepresented samples (see figure 5). The under sampling technique, CNN, was better at classifying the pathological class with a recall of 0.951 but was worse in general accuracy with an F-score of 0.827. ADASYN performed better in general over the other sampling techniques gaining the majority of the top spots among any category other than those specific to the pathological label.

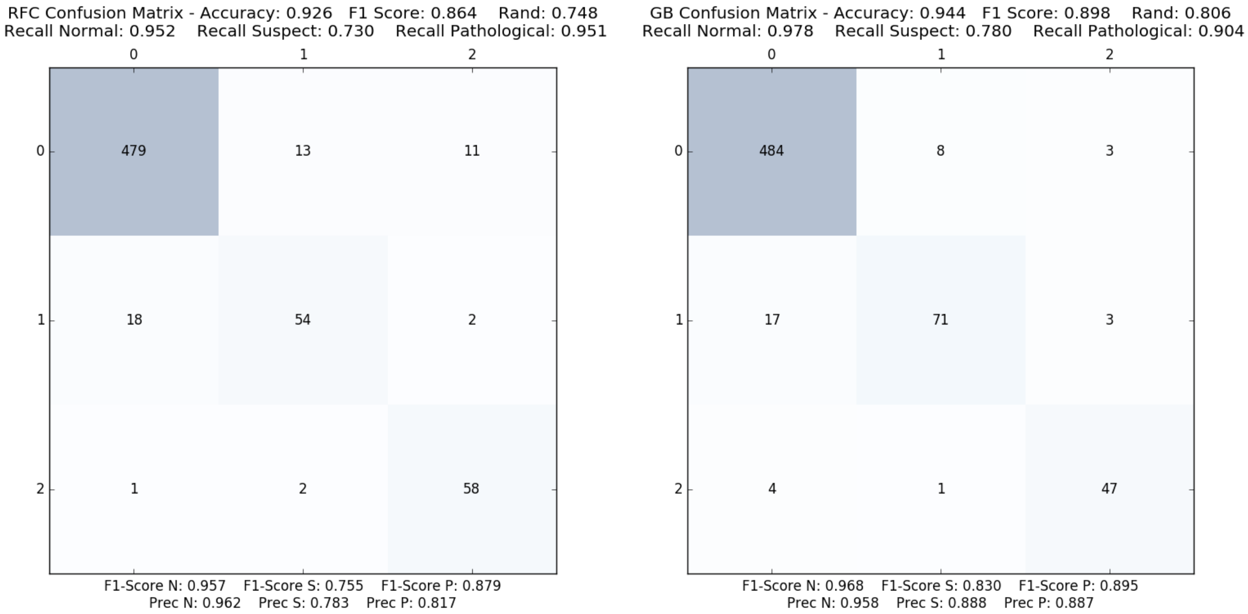


Fig. 5. Confusion matrixes from the under sampling CNN technique and random forest (left) and oversampling with ADASYN and gradient boosting (right). X-axis is target labels while Y-axis is classifier predicted labels.

Table 2: Results of non-network models, highest scores per category highlighted. N = normal, S = suspect, P = pathological labels. F-Score is the mean of each label's F-Score.

Model	Sampling	Avg. Acc Scores		Recall			Precision			F-Score		
		Accuracy	F-Score	N	S	P	N	S	P	N	S	P
Bagging	ADASYN	0.926	0.864	0.972	0.714	0.865	0.951	0.802	0.882	0.961	0.755	0.873
	SMOTE	0.926	0.871	0.956	0.744	0.938	0.962	0.744	0.882	0.959	0.744	0.909
	CNN	0.918	0.853	0.940	0.730	0.967	0.961	0.794	0.756	0.950	0.761	0.849
	None	0.934	0.877	0.994	0.667	0.846	0.935	0.912	0.957	0.964	0.770	0.898
SVM	ADASYN	0.915	0.848	0.956	0.725	0.865	0.954	0.742	0.849	0.955	0.733	0.857
	SMOTE	0.918	0.837	0.966	0.686	0.833	0.951	0.756	0.833	0.958	0.719	0.833
	CNN	0.895	0.793	0.946	0.581	0.852	0.946	0.581	0.852	0.946	0.581	0.852
	None	0.928	0.874	0.982	0.677	0.865	0.936	0.851	0.957	0.958	0.754	0.909
RFC	ADASYN	0.928	0.861	0.978	0.714	0.827	0.949	0.812	0.896	0.963	0.760	0.860
	SMOTE	0.920	0.865	0.946	0.756	0.938	0.964	0.707	0.882	0.955	0.731	0.909
	CNN	0.926	0.864	0.952	0.730	0.951	0.962	0.783	0.817	0.957	0.756	0.879
	None	0.937	0.881	0.992	0.710	0.827	0.942	0.892	0.956	0.967	0.791	0.887
KNN	ADASYN	0.897	0.815	0.945	0.670	0.827	0.944	0.678	0.827	0.944	0.674	0.827
	SMOTE	0.881	0.770	0.942	0.570	0.792	0.937	0.590	0.792	0.939	0.580	0.792
	CNN	0.831	0.714	0.869	0.568	0.836	0.948	0.400	0.708	0.907	0.469	0.767
	None	0.897	0.811	0.970	0.570	0.788	0.916	0.746	0.911	0.942	0.646	0.845
Gradient Boost	ADASYN	0.944	0.898	0.978	0.780	0.904	0.958	0.888	0.887	0.968	0.831	0.895
	SMOTE	0.929	0.878	0.948	0.826	0.917	0.974	0.732	0.880	0.961	0.776	0.898
	CNN	0.886	0.809	0.897	0.757	0.951	0.964	0.675	0.667	0.929	0.714	0.784
	None	0.931	0.872	0.976	0.753	0.827	0.949	0.843	0.896	0.962	0.795	0.860

The results show that the accuracy and F-score with re-sampling were lower compared to using no re-sampling technique such as with the bagging classifier. However this can be misleading, the accuracy scores without re-sampling is only reprehensive of the underlying distribution which can be seen by the individual recall and precision scores, a 0.994 for normal but 0.667 and 0.846 for suspect and pathological respectively. Re-sampling helped address these problems by sacrificing some score in the normal label for raising the minority class' scores.

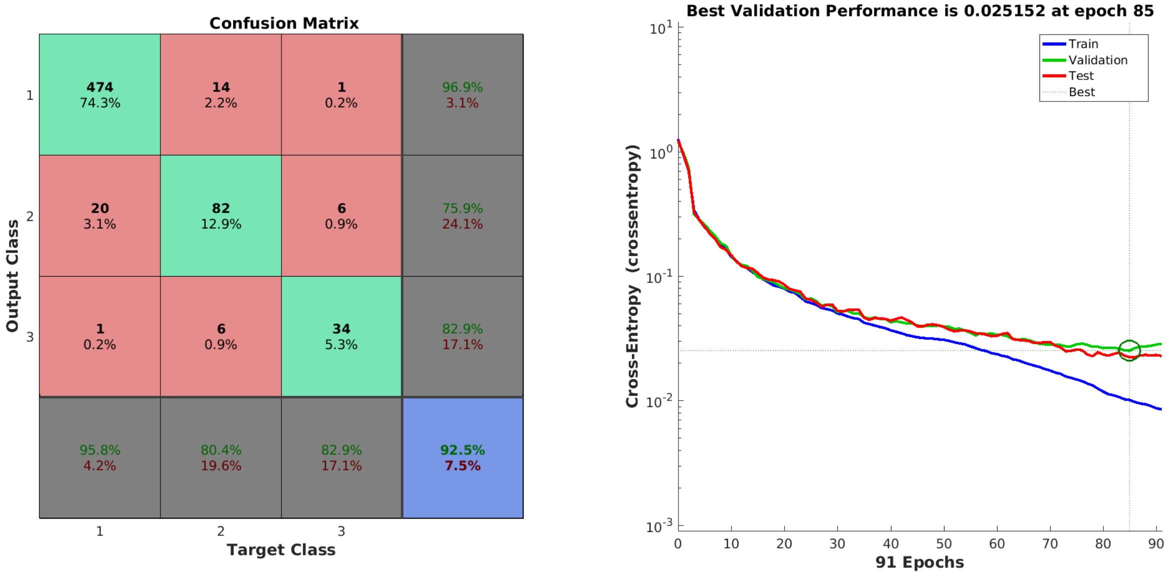
Table 3: Evaluation of the neural networks, sorted by accuracy. Lookup network details with ID number in Table 4.

ID	Sampling	Avg. Acc Scores		Recall			Precision			F-Score		
		Accuracy	F-Score	N	S	P	N	S	P	N	S	P
1	ADASYN	0.925	0.846	0.976	0.654	0.868	0.921	0.853	0.805	0.948	0.740	0.835
2	ADASYN	0.925	0.856	0.971	0.751	0.829	0.954	0.824	0.829	0.962	0.786	0.829
3	SMOTE	0.923	0.878	0.975	0.718	0.885	0.939	0.849	0.902	0.957	0.778	0.893
4	SMOTE	0.923	0.884	0.979	0.712	0.875	0.933	0.849	0.961	0.956	0.774	0.916
5	SMOTE	0.923	0.884	0.983	0.702	0.875	0.931	0.860	0.961	0.955	0.773	0.916
6	ADASYN	0.922	0.856	0.977	0.727	0.842	0.945	0.863	0.780	0.956	0.789	0.810
7	None	0.906	0.830	0.945	0.674	0.914	0.957	0.690	0.803	0.960	0.681	0.851
8	None	0.895	0.805	0.931	0.651	0.940	0.963	0.643	0.712	0.951	0.647	0.810
9	None	0.898	0.819	0.931	0.667	0.979	0.961	0.690	0.697	0.946	0.678	0.814
10	CNN	0.882	0.788	0.947	0.639	0.716	0.933	0.589	0.906	0.940	0.613	0.800
11	CNN	0.875	0.781	0.942	0.587	0.767	0.925	0.600	0.868	0.933	0.593	0.814
12	CNN	0.868	0.773	0.959	0.591	0.677	0.903	0.722	0.792	0.930	0.650	0.730

Table 4: Top network parameters. Also training time in the number of epochs before validation check limit (6) reached (1000 max epochs) and error measured via cross-entropy.

ID	Neurons per Layer	Number of Layers	Training Function	L1 Transfer Function	L2 Transfer Function	Number of Epochs	Lowest Cross-Entropy
1	50	2	trainscg	radbas	radbas	78	0.033
2	40	1	trainscg	radbas	NA	106	0.026
3	40	1	trainscg	radbas	NA	118	0.018
4	60	2	trainscg	tansig	radbas	98	0.035
5	40	2	trainscg	tansig	radbas	111	0.034
6	60	2	trainscg	tansig	radbas	107	0.025
7	60	2	trainscg	radbas	logsig	28	0.068
8	60	2	trainscg	tribas	radbas	25	0.071
9	40	2	trainscg	radbas	logsig	25	0.057
10	60	1	traingdx	tribas	NA	131	0.094
11	40	2	trainscg	tansig	logsig	23	0.176
12	40	2	trainscg	logsig	radbas	14	0.193

The neural networks show a closer battle between SMOTE and ANASYN. SMOTE was able to better classify the normal and pathological labels but ANASYN did better at the suspect cases which is also reflected in the average F-scores (see Table 3). CNN brought no benefits to the neural network's ability to classify the data, doing worse in raw accuracy and F-scores than no re-sampling. Common trends (Table 4) among the top networks included the pervasive scaled conjugate gradient backpropagation for training and the radial basis transfer function.



Confusion Matrix

Output Class \ Target Class	1	2	3	
1	474 74.3%	14 2.2%	1 0.2%	96.9% 3.1%
2	20 3.1%	82 12.9%	6 0.9%	75.9% 24.1%
3	1 0.2%	6 0.9%	34 5.3%	82.9% 17.1%
	95.8% 4.2%	80.4% 19.6%	82.9% 17.1%	92.5% 7.5%

Fig. 6. Confusion matrix and performance plot for the top neural network (ID 1): using ADASYN, two layers 50 neurons each, with radial basis transfer functions.

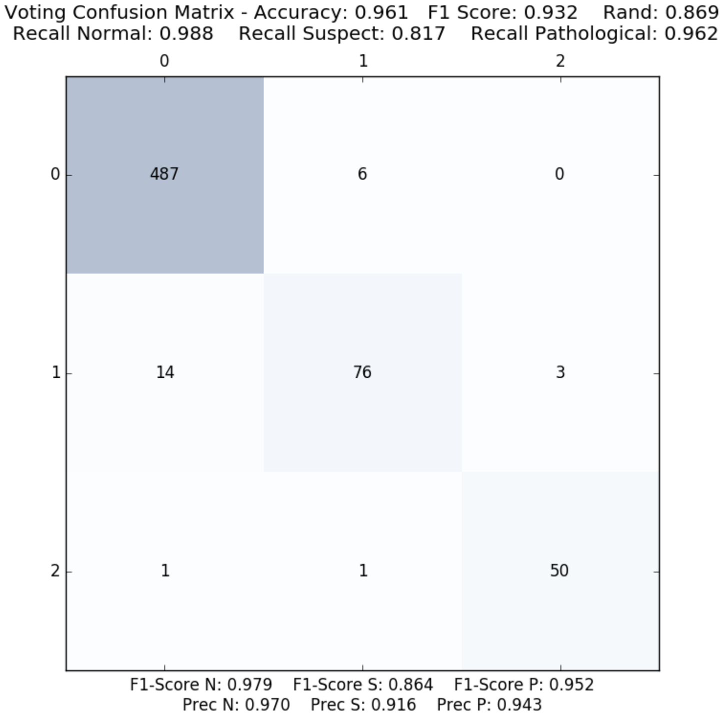
4. Conclusion

The sampling techniques were able to increase the classification accuracy of the imbalanced, minority classes at cost to the majority classes. There was no clear winner among the models, gradient boost with ADASYN performed well in general but performed poorly in the pathological class. This class is very important to classify because these are fetuses that are doing poorly and may need intervention. A low recall score could mean fetuses that are in distress are going unnoticed and a chance for medical intervention may be missed. On the other hand, low precision scores are not immediately alarming because testing can be done again to confirm a false positive. Similarly, we are willing to sacrifice some accuracy in the classification of the normal labels because these can be retested for validity, thus we can have a relatively low normal recall. This additional testing and validation should be at most an annoyance for the patient, and so still worth reducing, but suspect and pathological recalls are our primary concern due to the severe repercussions of failing to detect fetus distress.

The suspect class in particular is difficult to classify as seen by the general low F-scores for this label. Unfortunately the neural networks did not show an improved ability to classify this label but they did show better performance with their F-scores over the non-network models for normal and pathological labels. Future improvements could include a second round of grid searches. Parameters and functions that performed poorly in the previous iteration could be dropped for new or further refinement of semi-optimized parameters. Additional work is required since the suspect class can be falsely classified as normal which, as discussed above, could have serious consequences.

A solution to this problem could be to use ensemble voting techniques to create a classifier that is made up of several classifiers. Each individual classifier may be better at classifying a certain label, such as suspect, and have a more weighted vote when it goes to vote for the suspect class. Such a combination of models would also reduce the variation between runs. With a single instance of a model, four runs may have a slightly, +/-1%, accuracy deviation. However with a large number of instances, this variation between runs is reduced and we have a more predictable, stable model [8]. A downside of such a setup is the increased training time to train all of the individual instances and each instance's voting weight would be a parameter that would have to be optimized and known before hand.

The results to the right (figure 7) show an example a voting ensemble output. The voting was not able to achieve an F-score for the suspect class better than the gradient boost with ADASYN which scored 0.831 but the voting setup was able to achieve better overall accuracy, F-score and Rand index over that instance however. With more fine tuning and finding more optimized classifiers to add to the voting ensemble, this may be able to provide better classification of all classes at the expense of increased training and development time.


Voting Confusion Matrix - Accuracy: 0.961 F1 Score: 0.932 Rand: 0.869
Recall Normal: 0.988 Recall Suspect: 0.817 Recall Pathological: 0.962

	0	1	2
0	487	6	0
1	14	76	3
2	1	1	50

F1-Score N: 0.979 F1-Score S: 0.864 F1-Score P: 0.952
Prec N: 0.970 Prec S: 0.916 Prec P: 0.943

Fig. 7. Confusion matrix for multi-classifier, weighted voting.

5. References

1. Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C., & Cohen, S. N. (1974). Computer-Based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System. *Computers and Biomedical Research*, *8*(4), 303–320. Retrieved from http://www.sciencedirect.com/science/article/pii/001048097590009
2. Lisboa, P. J. G. (2002). A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Networks*, *15*(1), 11–39. http://doi.org/10.1016/S0893-6080(01)00111-3
3. Sundar, C. (2012). Classification of Cardiotocogram Data using Neural Network based Machine Learning Technique. *International Journal of Computer Applications*, *47*(14), 19–25. Retrieved from http://research.ijcaonline.org/volume47/number14/pxc3880279.pdf
4. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. http://doi.org/10.1613/jair.953
6. Gowda, K. C., & Krishna, G. (1979). The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood. *IEEE Transactions on Information Theory*, *25*(4), 488–490. https://doi.org/10.1109/TIT.1979.1056066
7. He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks*, (3), 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969
8. Bryll, R., Gutierrez-Osuna, R., & Quek, F. (2003). Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, *36*(6), 1291–1302. https://doi.org/10.1016/S0031-3203(02)00121-8