

Machine Learning Algorithm for Preterm Birth Classification from EHG Signals

Design Project | Spring 2019

Dian Guo | 3

Michael Isaf | 3

Alexandra Melehan | 3

Brandi Nevius | 3

Executive Summary

Preterm births contribute to many negative health outcomes including fatality for newborns. Currently, there are few legitimate methods for treating or preventing preterm births, largely because they are so difficult to predict [1]. Recent research suggests that uterine electromyogram (EMG), also known as electrohysterogram (EHG), signals show promise in predicting preterm labor [2].

Our unique method for predicting preterm labor consists of:

- Surveying previous studies to determine whether digital signal processing or artificial intelligence techniques were more effective predictors of preterm labor, and
- Using patient medical history in conjunction with EHG features to provide a more comprehensive feature set for a neural network.

Both of the above points were pursued in an attempt to improve preterm labor prediction specificity and sensitivity. Moreover, while manipulating EHG features is becoming common practice for preterm prediction, few, if any, previous studies researched the possibilities of patient medical history as an effective tool in preterm prediction. This project first explores the anticipated usefulness of combined EHG data and patient medical history in predicting preterm labor in existing studies, and then presents a novel prediction algorithm adopting said methods and its results.

Introduction and Significance

According to the World Health Organization, preterm birth is any delivery that occurs before the full gestation period of 37 weeks [1]. Preterm birth impacts 1 in 10 babies worldwide and reigns as the number one cause of infant death, and is on the rise [1]. Interestingly, 75% of preterm births are preventable by medical interventions; however, these interventions inherently rely on detection, which as of yet is not an exact science for preterm birth [1].

Some biological factors have been associated with preterm birth, like hypertension or diabetes [1]. Genetics appears to play a role as well, as do behavioral influences like smoking or the number of

previous abortions [1]. However, an expectant mother with some or one of these conditions is

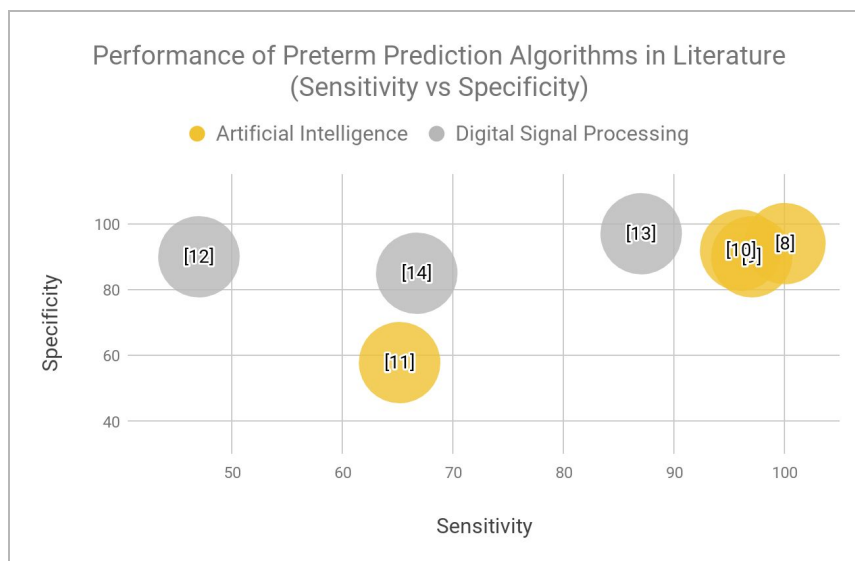


Figure 1: A plot of the reported sensitivities versus specificities of seven preterm classification algorithms. Three groups employed digital signal processing while the other four used artificial intelligence algorithms. Besides Far et. al [7], the AI algorithms form a visible cluster in the upper right portion of the graph with both high sensitivity and specificity [6, 9, 10]. We have therefore chosen to proceed with an AI approach.

not guaranteed to prematurely deliver, and perhaps more alarmingly, many preterm births occur spontaneously and idiopathically. In other words, many preterm births occur without any apparent warning at all.

As such, in the past decade many research groups have looked to underlying physiological signals to predict preterm labor before it happens. Specifically, electrohysterography (EHG), also known as uterine electromyogram (EMG), has been endorsed as a window into uterine electrical activity which appears to be predictive of the onset of preterm labor.

A 2008 study gathered EHG signals from 300 pregnant women including both term and preterm births which was published on the Physionet website entitled the Term-Preterm EHG Database (TPEHG DB) [2, 3]. Developments in the past few years have also included the use of four by four electrode arrays, which yield more detailed data sets than previous studies, and were also published to Physionet [4, 5, 6, 7].

Since the data publications, research groups have developed many prediction algorithms using either digital signal processing (DSP) or more sophisticated artificial intelligence (AI) algorithms. **Figure 1** depicts the resulting sensitivities and specificities of seven such algorithms, some using DSP and the remainder AI. Considering the superior success of the AI algorithms, the team chose to proceed with an AI algorithm as opposed to plain DSP. However, there are countless AI approaches and just as many signal features to consider. Previous groups like Fele-Žorž et. al explicitly recommended pursuing certain features like median frequency and sample entropy [2]. **Figure 2** presents the methods of just eight of the many AI preterm prediction algorithms and their associated features employed thus far. Considering the potential lethal cost of failing to predict a preterm pregnancy, the team hopes to further improve the accuracy of a preterm prediction algorithm by also considering the medical history of the expectant mother in addition to the many features suggested in literature.

Reference	Classification Method	Features	Validation Method
[14]	Artificial neural network	Mean power spectrum, Standard deviations power spectrum (PS), Peak frequency, Burst duration, Number of bursts per unit time, Total burst activity	Not listed
[13]	Artificial neural network Multilayer Perceptron (MLP)	Not listed (abstract only)	Not listed (abstract only)
[6]	Artificial neural network simple perceptron	10 parameters of power spectrum density (doesn't specify which)	Simulated testing data
[5]	Multilayer perceptron	Sample entropy Cepstral coefficients	Holdout CV
[15]	Support Vector Machines (SVM) with RBF kernel	Power Median frequency	Not listed
[9]	linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), uncorrelated normal density based classifier (UDC), polynomial classifier (POLYC), logistic classifier (LOGLC), 3-NN (KNNC), decision tree (TREC), parzen classifier (PARZENC), support vector classifier (SVC)	Root mean square Median frequency Peak frequency Sample entropy	Holdout CV, K-fold CV, Receiver Operator Curve (ROC), Area Under the Curve (AUC)
[10]	backpropagation trained feed-forward neural network classifier (BPXNC), levenberg-marquardt trained feed-forward neural network classifier (LMNC), automatic neural network classifier (NEURC), radial basis function neural network classifier (RBNC), random neural network classifier (RNNC), perceptron linear classifier (PERLC)	Root mean square Median frequency Peak frequency Sample entropy	Holdout CV, K-fold CV, Receiver Operator Curve (ROC), Area Under the Curve (AUC)
[7]	Support Vector Machines (SVM) with RBF kernel	Mean, Median, Standard deviation, Zero-crossing, Entropy, Cepstrum	Not listed

Figure 2: A compilation of the classification methods for preterm birth prediction using machine learning from related literature. While a variety of artificial intelligence algorithms were employed, many of the features consistently appeared across groups, especially root mean square, median frequency, peak frequency, and sample entropy.

Project Narrative

Data Retrieval

Data was retrieved from the Term-Preterm EHG Database (TPEHG DB) from Physionet published by Fele-Žorž et. al in 2008 [2, 3]. The data consisted of 300 EHG records from 300 pregnancies. 292 of those records were obtained from term pregnancies while 38 were obtained from preterm. The records consist of three channels which were recorded using four electrodes. The header file for each patient contained the record number, pregnancy duration, the gestation duration at the time of recording, the maternal age, the number of previous deliveries, the number of previous abortions, the weight at the time of recording, whether hypertension occurred, whether the patient was diabetic, the placental position, whether there was bleeding during the first trimester, whether there was bleeding the second trimester, whether funneling occurred, and whether the patient was a smoker. Many of these values were numerical but a few were boolean [3]. The published data already had a bandwidth filter applied from 0.08-4.0 Hz as uterine EMG ranges from 0 to <5 Hz [2]. From there, data from any subjects without a full medical history was eliminated, considering that our classifier includes features from patient history. This elimination process yielded 169 total subjects worth of data out of the original 300, 19 of which are preterm and 150 term.

Feature Selection and Extraction

Classifier Features	
Signal Features	Median frequency
	Peak frequency
	Sample Entropy
	RMS
Medical History Features	Age
	Weight
	Placental Position
	Hypertension
	Diabetes
	Smoking
	Number of Previous Abortions
	Number of Previous Deliveries
	Funneling
	Bleeding First Trimester
	Bleeding Second Trimester

Figure 3: The 15 features considered in our classifier, 4 from the EHG signals, and the remaining 11 from the patient medical history.

Many papers that have explored this same space (building classifiers for preterm pregnancies from EHG signals) utilized median frequency, peak frequency, sample entropy and RMS in their detection algorithms, and some even explicitly recommended these features for future studies among other features due to their relative success [9, 10, 13, 15]. Fortunately, these four signal features were already extracted and explicitly labeled in downloadable data files from the TPEHG DB, along with patient medical history including but not limited to age, weight, height, diabetes, hypertension, and smoking. In total, 15 features were selected to be considered in the classifier, as shown in **Figure 3**.

Balancing and Splitting the Dataset

Considering the disparity between the number of term and preterm records, the data set was balanced by using the widely accepted Synthetic Minority Oversampling Technique (SMOTE). This algorithm was also employed by similar studies that used this exact data set [9, 10, 13]. Essentially, this algorithm generates synthetic data points by randomly

interpolating values between the real data points. After completing SMOTE, the preterm data grew from 19 subjects to 150, now evenly matching the 150 term signals. Using the Python toolkit “scikit-learn”, the data was split 80-20 into training and testing groups respectively.

Neural Network Design Process

Although research shows that a feed-forward neural network performs the best, the network architecture was not specified. The challenge then is to design a neural network that would be able to adequately train off the relatively small dataset. The neural network has a 15-neuron input layer and 2-neuron output layer, to account for all of the features and the two output probabilities. For simplicity, the Adam optimizer, sparse categorical cross-entropy loss function, 80-20 holdout, 20% dropout rate, and the accuracy metric was chosen. Following the guideline for designing such architecture [16], two hidden layers were selected to map the uncertain relationship between the 15 input features and the two outputs. To narrow down how many neurons would be inside of each hidden layer, a trial was conducted. The trial includes all 144 configurations, two layers of neurons each ranging from 3 to 15, of the neural network, each model trained on randomly split data and evaluated, and only storing the first 10 models with above 80% accuracy. Because the models were trained on randomly split data, their accuracy results may not be reliable. After the preliminary screening, the models were evaluated again in 30 random data split trials, storing their accuracy, specificity, and sensitivity per trial, as well as averages.

Classifier Results

The models showed an interesting pattern. The lower number models and the higher number models showed worse results than the middle ones (**Figure 4**). This pattern can be attributed to the lower number models not having enough neurons, thus under-fitting, and the higher number models having too many neurons, thus over-fitting. The algorithm was thus refined to reflect the mid-level neural network, which was shown to not overfit or underfit. The average sensitivity of the classifier under such conditions was 92.6%. That being said, the algorithm performed worse than most of the literature review algorithms, with both a lower sensitivity and specificity (**Figure 5**).

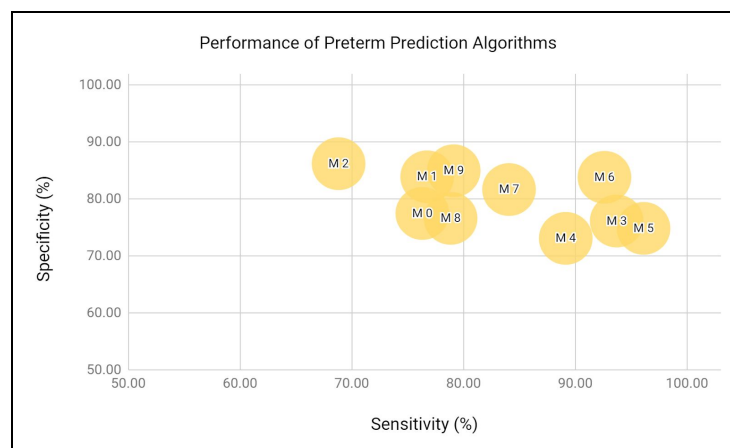


Figure 4. The performance summary of the 10 models under 30 random trials.

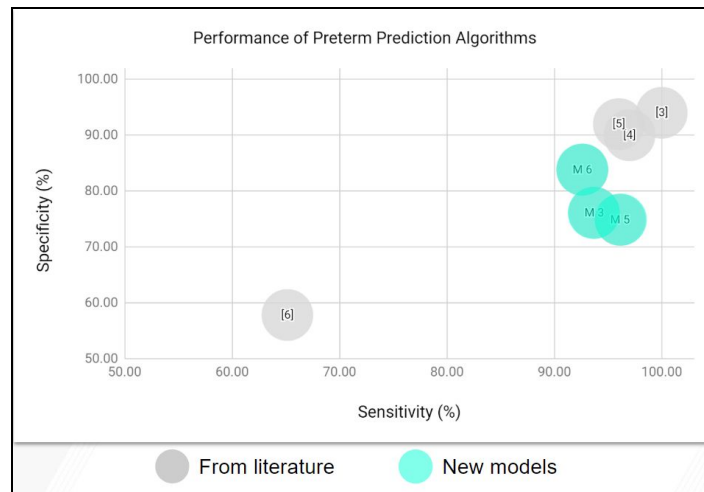


Figure 5. The performance summary of the this algorithm compared to algorithms in literature review.

As Model 5 had the highest sensitivity, it was further evaluated for performance. This model performed better with time, as the accuracy improved across the run time (**Figure 6**). Additionally, this model did not overfit the data, as the loss for an overfitted model will tend to increase after a certain interval of time, overtraining the model, and this model did not do so (**Figure 7**).

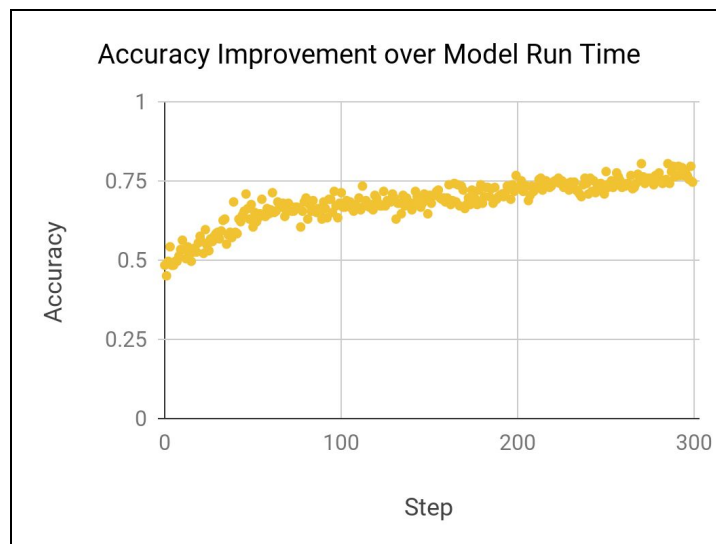


Figure 6. The relationship between Model 5 accuracy and run time.

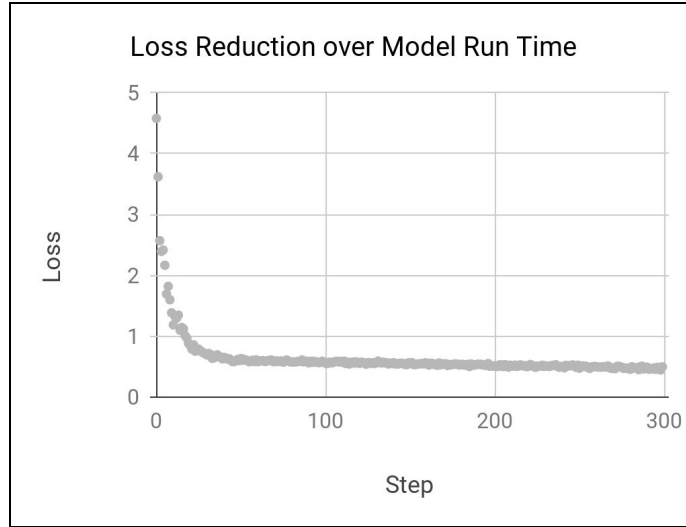


Figure 7. The loss reduction of Model 5 with respect to run time.

Conclusions and Future Directions

Conclusions

As the sensitivity is indicative of the algorithm's ability to correctly identify preterm patients, this measure is the most important result for infant survival. The average sensitivity of the classifier was 92.6%, which was similar but slightly worse than the sensitivity of other algorithms from the literature review. A unique aspect of this algorithm was the usage of patient medical history in the prediction algorithm, but this was not found to improve the diagnostic accuracy.

Future Directions

Moving forward, several changes could potentially improve the performance of this algorithm. In the short term, three routes for improvement have been identified. First, the accuracy could potentially benefit by normalizing the features and batches when training the classifier. For instance, some features like body weight have large magnitudes which could inadvertently eclipse the neural network weight of other smaller-magnitude features like age. Normalizing these features initially could help equalize these relative importances and thus improve the AI performance. Additionally, resources suggest that testing on synthetic data may not be ideal due to data "bleeding" [17]. Data bleeding describes the phenomenon that occurs when a data set is balanced before splitting. Because balancing data involves interpolating existing data, synthetic data is directly related to the real data, and may skew test results. To prevent this in the future, we would split the data into testing and training groups first, and then balance using SMOTE. The last short term adjustment would be to split data into training and testing sets based on *subjects*, not just signals. In other words, 80% and 20% of subjects would be assigned to training and testing categories respectively as opposed to just 80% and 20% of the total data. In the case that multiple channels exist from the same subject, this method of splitting data further prevents data bleeding (the algorithm would not be testing on the same subject on which it was trained).

If given much more time, the team would run more trials and gather more data. This would yield a stronger and more generalizable algorithm than current data sets allow for. Finally, a wearable device could be developed to continuously monitor electrical signals and perhaps reduce the number preterm births.

References

- [1] “Preterm birth,” World Health Organization. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/preterm-birth>.
- [2] G. Fele-Žorž, G. Kavšek, Ž. Novak-Antolič, and F. Jager, “A comparison of various linear and non-linear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups,” *Medical & Biological Engineering & Computing*, vol. 46, no. 9, pp. 911–922, 2008.
- [3] The Term-Preterm EHG Database (TPEHG DB). [Online]. Available: <https://physionet.org/physiobank/database/tpehgdb/>.
- [4] S. Sim, H. Ryou, H. Kim, J. Han, and K. Park, “Evaluation of Electrohysterogram Feature Extraction to Classify the Preterm and Term Delivery Groups,” *IFMBE Proceedings The 15th International Conference on Biomedical Engineering*, pp. 675–678, 2014.
- [5] A. Alexandersson, T. Steingrimsdottir, J. Terrien, C. Marque, and B. Karlsson, “The Icelandic 16-electrode electrohysterogram database,” *Scientific Data*, vol. 2, p. 150017, 2015.
- [6] The Term-Preterm EHG DataSet with Tocogram (TPEHGT DS). [Online]. Available: <https://physionet.org/physiobank/database/tpehgt/>.
- [7] Icelandic 16-electrode Electrohysterogram (EHG) Database. [Online]. Available: <https://physionet.org/physiobank/database/ehgdb/>.
- [8] M. O. Diab, A. El-Merhie, N. El-Halabi, and L. Khoder, “Classification of uterine EMG signals using supervised classification method,” *Journal of Biomedical Science and Engineering*, vol. 03, no. 09, pp. 837–842, 2010.
- [9] P. Fergus, P. Cheung, A. Hussain, D. Al-Jumeily, C. Dobbins, and S. Iram, “Prediction of Preterm Deliveries from EHG Signals Using Machine Learning,” *PLoS ONE*, vol. 8, no. 10, 2013.
- [10] I. O. Idowu, P. Fergus, A. Hussain, C. Dobbins, and H. A. Askar, “Advance Artificial Neural Network Classification Techniques Using EHG for Detecting Preterm Births,” 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems, 2014.
- [11] D. T. Far, M. Beiranvand, and M. Shahbakhti, “Prediction of preterm labor from EHG signals using statistical and non-linear features,” 2015 8th Biomedical Engineering International Conference (BMEiCON), 2015.

- [12] I. Verdenik, M. Pajntar, and B. Leskošek, "Uterine electrical activity as predictor of preterm birth in women with preterm contractions," *European Journal of Obstetrics & Gynecology and Reproductive Biology*, vol. 95, no. 2, pp. 149–153, 2001.
- [13] F. Jager, S. Libenšek, and K. Geršak, "Characterization and automatic classification of preterm and term uterine records," *Plos One*, vol. 13, no. 8, 2018.
- [14] K. A. Lee, M. H. Chang, M.-H. Park, H. Park, E. H. Ha, E. A. Park, and Y. J. Kim, "A Model for Prediction of Spontaneous Preterm Birth in Asymptomatic Women," *Journal of Women's Health*, vol. 20, no. 12, pp. 1825–1831, 2011.
- [15] W. L. Maner and R. E. Garfield, "Identification of Human Term and Preterm Labor using Artificial Neural Networks on Uterine Electromyography Data," *Annals of Biomedical Engineering*, vol. 35, no. 3, pp. 465–473, 2007.
- [16] J. Heaton. "Feedforward Backpropagation Neural Networks," *Introduction to Neural Networks for Java*, 2nd ed. Available: <https://web.archive.org/web/20140721050413/http://www.heatonresearch.com/node/707/>.
- [17] N. Becker, "The Right Way to Oversample in Predictive Modeling," Nick becker, 23-Dec-2016. [Online]. Available: <https://beckernick.github.io/oversampling-modeling/>. [Accessed: 16-Apr-2019].