

Comparison of Machine Learning Techniques for Fetal Heart Rate Classification

Z. CÖMERT^{a,*} AND A.F. KOCAMAZ^b

^aBitlis Eren University, Department of Computer Engineering, Bitlis, Turkey

^binönü University, Department of Computer Engineering, Malatya, Turkey

Cardiotocography is a monitoring technique providing important and vital information on fetal status during antepartum and intrapartum periods. The advances in modern obstetric practice allowed many robust and reliable machine learning techniques to be utilized in classifying fetal heart rate signals. The role of machine learning approaches in diagnosing diseases is becoming increasingly essential and intertwined. The main aim of the present study is to determine the most efficient machine learning technique to classify fetal heart rate signals. Therefore, the research has been focused on the widely used and practical machine learning techniques, such as artificial neural network, support vector machine, extreme learning machine, radial basis function network, and random forest. In a comparative way, fetal heart rate signals were classified as normal or hypoxic using the aforementioned machine learning techniques. The performance metrics derived from confusion matrix were used to measure classifiers' success. According to experimental results, although all machine learning techniques produced satisfactory results, artificial neural network yielded the rather well results with the sensitivity of 99.73% and specificity of 97.94%. The study results show that the artificial neural network was superior to other algorithms.

DOI: [10.12693/APhysPolA.132.451](https://doi.org/10.12693/APhysPolA.132.451)

PACS/topics: cardiotocography, machine learning techniques, classification

1. Introduction

Cardiotocography (CTG), which consists of fetal heart rate (FHR) and uterine contraction (UC) signals, is used widely by obstetricians during antepartum and intrapartum periods for the biophysical assessment of fetal condition [1]. The signals are recorded simultaneously using electronic fetal monitoring (EFM) devices. CTG relies upon FHR, UC, and fetal movement activity and it is employed to detect dangerous situations for the fetus [2]. The aim of intrapartum fetal monitoring is to detect hypoxia or asphyxia and prevent fetal injury [3]. Oxygen deficiency of the fetus during labor is evaluated in three stages, namely, hypoxemia, hypoxia, and asphyxia [4]. The fetal defense mechanism manages this process with the help of the sympathetic and parasympathetic nervous systems. Although CTG has a high false-positive rate, it is a useful tool to observe situations that lead to fetal distress, such as prolonged spontaneous rupture of membranes, prematurity, and intrauterine growth restriction. The advantages of this surveillance technique have not been proven clearly, however it has also become a standard of care in many developed countries [5].

The advances in modern obstetric practice allowed many robust and reliable machine learning techniques to be utilized in classifying FHR patterns [6]. Huang and Hsu [7] have offered discriminant analysis (DA), decision tree (DT), and artificial neural network (ANN) to evaluate fetal distress. Yilmaz and Kılıkçier [8] have suggested using least square (LS) support vector machine (SVM) with particle swarm optimization and binary DT.

Ocak [9] has developed a medical decision support system based on SVM and genetic algorithm (GA). Sahin and Subasi [10] have compared the performances of eight different machine learning techniques by using WEKA software. As seen in mentioned works, some works focus on the classifying performance, whereas some of them try to find the most relevant features in order to reduce the dimension of feature space. The role of machine learning approaches in diagnosing diseases is becoming increasingly essential and intertwined [11]. In this context, the present study offers a comparison that is focused on the performances of five different machine learning techniques on FHR classification problem in terms of sensitivity (Se), specificity (Sp), geometric mean of Se and Sp (GM), and F-measure (F1). Also, unlike other works, extreme learning machine (ELM) algorithm with five different activation functions was evaluated extensively. The performance results of classifiers were compared to each other. Although all machine learning techniques produced rather well performances, ANN was superior to others.

2. Materials and methods

Machine learning techniques have been used in wide application range. Gradient descent, standard optimization, and least-square based methods are the main approaches to the training of networks [12]. This work has concentrated on most used and efficient machine learning methods, such as ANN, SVM, ELM, radial basis function network (RBFN), and random forest (RF). In this section, the data set and chosen machine learning techniques are described briefly.

2.1. Data collection

A publicly accessible data set was used in this study. The data set includes 2126 instances with 21 features

*corresponding author; e-mail: comertzafer@gmail.com

comprising of 8 continuous and 13 discrete values. The data set was created automatically by the software called SisPorto 2.0 [13]. As seen in Table I, the suspected instances were excluded from the original data set because these cases do not have any contribution to establishing of the diagnosis [10]. The general features of the data set are shown in Table I. Each algorithm performance was tested several times using 10-fold cross-validation during the training phase.

TABLE I

The summary of SisPorto data set.

| Data set | # Samples | # Features | # Class | # Samples per class |
|------------|-----------|------------|---------|---------------------|
| original | 2126 | 21 | 3 | 1655–295–176 |
| restricted | 1831 | 21 | 2 | 1655–176 |

2.2. Artificial neural network

ANN is an artificial computational technique for function approximation or categorization of multivariate data. It is composed of a series of neurons that is organized in layers. The structure of ANN comprises input layer, one or more hidden layer(s), and output layer [14]. Equation (1) presents the value as an output of a layer. σ demonstrates the activation function, N indicates the number of input neurons, w_{ij} represents the weights, x_j symbolizes the inputs, and b denotes the threshold value for hidden neurons.

$$a^i = \sigma \left(\sum_{j=1}^N w_{ij} x_j + b^i \right). \quad (1)$$

2.3. Support vector machine

SVM is a kernel mapping technique that can be applied to separable and non-separable data for regression, classification, and other learning tasks. Kernel-based algorithms have good generalization performance, and they work well in practice [15]. The similarity or dissimilarity of data objects is measured by kernels used to store data. The training data, which is separable by a hyperplane, is modeled in the following formula:

$$\begin{cases} x_i w + b \geq +1 & \text{for } y_i = +1, \\ x_i w + b \leq -1 & \text{for } y_i = -1, \\ y_i(x_i w + b) - 1 \geq 0 & \forall i \end{cases} \quad (2)$$

2.4. Extreme learning machine

ELM is a special training algorithm for single-hidden layer feedforward neural network (SLFN), which randomly chooses hidden nodes and provides emergent learning [16]. The essence of ELM is only one hidden layer, which does not need to be tuned. Its advantages include extremely fast learning speed, less human intervention, and excellent computational scalability [17]. Standard SLFNs with N arbitrary ($x_i t_i$) samples are modeled as follows:

$$f_L(x) = \sum_{i=1}^L \beta_i h_i(x_j) = \sum_{i=1}^L \beta_i h(w_i x_j + b_j) = o_i,$$

$$j = 1, \dots, N, \quad (3)$$

where h_i denotes the activation function, L is the number of hidden layer nodes, w_i is the weight vector between the i -th hidden node and the input nodes, β_i is the weight vector between the i -th hidden node and output nodes. The generalized inverse matrix of H is calculated according to Moore-Penrose. ELM can overcome slow training speed and being stuck in a local minimum of traditional training algorithm and have better generalization and is extensively used in regression and classification problems [18].

2.5. Radial basis function network

RBFN is a useful feedforward neural network architecture using radial basis functions as activation functions. It is typically configured with a single-hidden layer that uses Gaussian or some other kernel functions [19]. It includes an input vector, a layer of radial basis function (RBF) neurons, and an output layer. Unsupervised learning technique is used in the hidden layer. The linear combination of weights is employed in the output of the network.

2.6. Random forest

RF is a classifier that grows on many classification trees, which increases classification accuracy and accomplishes better generalization, for large databases in ensemble learning. RF uses multiple prediction models by aggregating a group of classifiers that is known as base classifiers, which are independent of each other [20]. Multiple trees are built randomly with a subspace of features. The forest selects the classification having the most votes. Majority voting is frequently used to aggregate the base classifier. The majority voting error rate (ε_{mv}) is given as follows:

$$\varepsilon_{mv} = \sum_{i=|\frac{M}{2}|+1}^M \binom{M}{i} \varepsilon^i (1 - \varepsilon)^{M-i}. \quad (4)$$

M represents the number of base classifier and ε represents the identical error rate for all base classifiers.

3. Results

Several performance metrics are required to evaluate a classifier success after the training phase. Several performance criteria have been obtained by using the confusion matrix and are given in Table II with their mathematical form and short description.

The cross-validation is a suitable technique, which consists of dividing the data into k subsamples, for moderate-sized data set. The average error rate is obtained by using each subsample for $k - 1$ times. In this context, the data set was partitioned as training and testing set. After this, 10-fold-cross validation was employed to estimate the average accuracy of the model.

During the configuration of ANN, twelve training algorithms were used so as to determine the most efficient one. The network employed the Levenberg–Marquardt backpropagation training algorithm (LM) and only one

TABLE II

The performance metrics obtained from confusion matrix for comparison of the algorithms.

| Metrics | Formula | Description |
|----------------|--------------------------------------|--|
| sensitivity | $\left(\frac{TP}{TP+FN}\right)$ | the measure of correctly classified positive cases |
| specificity | $\left(\frac{TN}{TN+FP}\right)$ | the measure of the false positive errors |
| geometric mean | $\sqrt{Se \times Sp}$ | the geometric mean of Se and Sp |
| F-score | $\left(\frac{2TP}{2TP+FP+FN}\right)$ | the measure of the harmonic mean of the precision and recall |

Herein, true positive (TP) and true negative (TN) represent true classification for positive and negative samples respectively, whereas false positive (FP) and false negative (FN) represent false classification for positive and negative samples respectively.

hidden layer with fifteen neurons in the study. LM occurs in a combination of gradient descent and the Gauss–Newton method. For this reason, LM can be guaranteed as the solution in many cases [21]. In the scope of choosing training algorithm, LM algorithm produced the best results regarding the classification accuracy. On the other hand, ANN was customized by means of training parameters. Momentum decrease factor, momentum increase factor, and maximum validation failures were set as 0.15, 12, and 10, respectively. The training with these parameters took place in less than thirty epochs, and it occurred at an average of 2.236 s. The training time of ANN takes longer according to other machine learning algorithms, and this case constitutes a disadvantage for the network. In summary, ANN with Se of 99.73% and Sp of 97.94% values was superior to other machine learning algorithms.

Kernel-based algorithms have great generalization performance for SVM. In the experiment, the linear, quadratic, cubic, and Gaussian kernel functions have been tested. Kernel was chosen as the Gaussian RBF. Through trial and error, the sigma value was set as 3, and the penalty factor was set at 0.5. According to experimental results, SVM performed quite well results as Se of 99.21% and Sp of 97.02%.

As mentioned earlier, ELM can overcome limitations of the traditional learning algorithms with better generalization performance, low computational process, and especially extremely fast learning ability [17, 18]. However, the ELM structure has some drawbacks. For example, the performance of ELM directly depends on the number of hidden neurons. Also, ELM may encounter an overfitting problem in the case that the number of neurons in the hidden layer is greater than the training set [12]. Therefore, the classification success and training time of ELM were investigated depending on the number of hidden nodes. The number of hidden nodes was increased gradually from 20 to 1000. Figure 1 shows that the classification success was grown until the number of hidden nodes approached 200. Afterward, no significant change

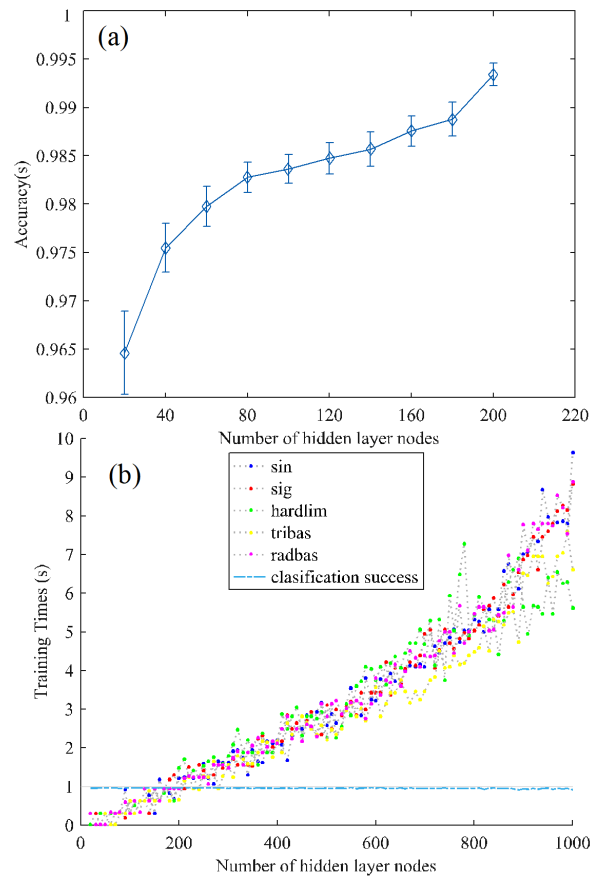


Fig. 1. Performance values of ELM. (a) Obtained accuracy versus the number of hidden nodes. (b) Training times versus the activation functions and the number of hidden nodes.

in the classification accuracy of ELM was observed. Consequently, the number of hidden layer node was set at 200. In addition, the training time showed a leap at this point. ELM was employed over fifty different times for each hidden layer node change. Se and Sp values of ELM were obtained as 96.34% and 95.30%, respectively.

RBFN consisting of one input layer, one hidden layer, and one output layer is a special version of SLFN. The experimental results pointed that this algorithm was the best in terms of network training time. Se and Sp values achieved 96.83% and 88.49%, respectively, by using RBFN.

RF also presented satisfactory results. The obtained Se and Sp values were 99.07% and 93.31%, respectively. In addition, the training time of the network was only 0.332 s. The performance results of all machine learning algorithms are presented in Table III.

Se , Sp , GM, and F1 metrics have been calculated using confusion matrix. Bold values in Table III show the highest value in each row.

4. Conclusions

Consequently, the widely used and efficient five machine learning techniques were examined on FHR classification task, and the network performances evaluated

TABLE III

Performance results [%] of machine learning algorithms.

| Metrics | ANN | SVM | ELM | RBFN | RF |
|-------------------------|--------------|-------|-------|--------------|-------|
| sensitivity (Se) | 99.73 | 99.21 | 99.34 | 96.83 | 99.07 |
| specificity (Sp) | 97.94 | 97.02 | 95.30 | 88.49 | 93.31 |
| geometric mean (GM) | 98.83 | 98.10 | 97.29 | 95.56 | 96.14 |
| F-measure ($F1$) | 0.997 | 0.992 | 0.993 | 0.969 | 0.990 |
| training time [s] | 2.236 | 0.720 | 0.889 | 0.073 | 0.332 |

regarding several metrics (Se , Sp , GM , and $F1$) obtained from confusion matrix. According to experimental results, ANN produced more efficient results (Se of 99.73% and Sp of 97.94%) than other machine learning techniques. ELM was examined comprehensively. More specifically, the tuning parameters, such as activation functions and the number of node in hidden layers were considered in a large range. The obtained results of each classifier were compared to each other considering the performance metrics. Although aforementioned machine learning techniques produced rather good results, ANN was revealed as the most effective algorithm.

References

- [1] D. Ayres-de-Campos, C.Y. Spong, E. Chandraharan, *Int. J. Gynecol. Obstet.* **131**, 13 (2015).
- [2] I. Ingemarsson, *Neonatology* **95**, 342 (2009).
- [3] A. Pinas, E. Chandraharan, *Best Pract. Res. Clin. Obstet. Gynaecol.* **30**, 33 (2016).
- [4] A.-K. Sundström, D. Rosén, K.G. Rosén, , Neovanta Medical AB, Göteborg 2005.
- [5] M.J. Stout, A.G. Cahill, *Clin. Perinatol.* **38**, 127 (2011).
- [6] Z. Cömert, A.F. Kocamaz, *Int. J. Comput. Appl.* **156**, 26 (2016).
- [7] M.-L. Huang, Y.-Y. Hsu, *J. Biomed. Sci. Eng.* **5**, 526 (2012).
- [8] E. Yılmaz, Ç. Kılıkçier, *Comput. Math. Methods Med.* **2013**, 1 (2013).
- [9] H. Ocak, *J. Med. Syst.* **37**, 9913 (2013).
- [10] H. Sahin, A. Subasi, *Appl. Soft Comput.* **33**, 231 (2015).
- [11] E.D. Reddy, V. Gondlekar, V. Gauns, *Acta Phys. Pol. A* **130**, 78 (2016).
- [12] O.F. Alcin, A. Sengur, S. Ghofrani, M.C. Ince, *Measurement* **55**, 126 (2014).
- [13] D. Ayres-de-Campos, J. Bernardes, A. Garrido, J. Marques-de-Sá, L. Pereira-Leite, *J. Matern. Fetal. Med.* **9**, 311 (2000).
- [14] Z. Cömert, A.F. Kocamaz, in: *Int. Artificial Intelligence and Data Processing Symp. (IDAP)*, Ed: A. Karci, IDAP, Malatya (Turkey) 2016, p. 569.
- [15] J. Spilka, J. Frecon, R. Leonarduzzi, N. Pustelnik, P. Abry, M. Doret, *IEEE J. Biomed. Health Inform.* **21**, 664 (2016).
- [16] Z. Cömert, A.F. Kocamaz, in: *25th Signal Processing and Communications Applications Conf. (SIU)*, Antalya (Turkey) 2017, p. 1.
- [17] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, *Neurocomputing* **70**, 489 (2006).
- [18] Z. Cömert, A.F. Kocamaz, S. Gungor, in: *24th Signal Processing and Communication Application Conf. (SIU)*, 2016.
- [19] K. Warwick, R. Craddock, *IEEE Conf. Decis. Control* **1**, 464 (1996).
- [20] P. Tomáš, J. Krohová, P. Dohnálek, P. Gajdoš, in: *36th Int. Conf. on Telecommunications and Signal Processing (TSP)*, IEEE, Rome 2013, p. 620.
- [21] D. Marquardt, *J. Soc. Ind. Appl. Math.* **11**, 431 (1963).