Cardiotocographic Diagnosis of Fetal Health based on Multiclass Morphologic Pattern Predictions using Deep Learning Classification

Julia H. Miao[1], Kathleen H. Miao[1, 2]
[1]Cornell University, Ithaca, NY 14850, USA
[2]New York University School of Medicine, New York, NY 10016, USA

***Abstract*—Medical complications of pregnancy and pregnancy-related deaths continue to remain a major global challenge today. Internationally, about 830 maternal deaths occur every day due to pregnancy-related or childbirth-related complications. In fact, almost 99% of all maternal deaths occur in developing countries. In this research, an alternative and enhanced artificial intelligence approach is proposed for cardiotocographic diagnosis of fetal assessment based on multiclass morphologic pattern predictions, including 10 target classes with imbalanced samples, using deep learning classification models. The developed model is used to distinguish and classify the presence or absence of multiclass morphologic patterns for outcome predictions of complications during pregnancy. The testing results showed that the developed deep neural network model achieved an accuracy of 88.02%, a recall of 84.30%, a precision of 85.01%, and an *F*-score of 0.8508 in average. Thus, the developed model can provide highly accurate and consistent diagnoses for fetal assessment regarding complications during pregnancy, thereby preventing and/or reducing fetal mortality rate as well as maternal mortality rate during and following pregnancy and childbirth, especially in low-resource settings and developing countries.**

***Keywords*—Activation function; deep learning; deep neural network; dropout; ensemble learning; multiclass; regularization; cardiotocography; complications during pregnancy; fetal heart rate**

I. INTRODUCTION

In 2012, approximately 213 million pregnancies occurred worldwide [1]. Of those pregnancies, 190 million (89%) occurred in developing countries and 23 million (11%) were in developed countries. In 2013, complications of pregnancy resulted in 293,336 deaths due to maternal bleeding, complications of abortion, high blood pressure, maternal sepsis, and obstructed labor [2]. According to the World Health Organization [3], roughly 303,000 women died during and following pregnancy and childbirth in 2015, and in 2016, about 830 women died every day from pregnancy-related or childbirth-related complications around the world.

Medical complications of pregnancy and pregnancy-related deaths, impacting mothers and/or their babies, continue to remain a major global challenge today. Maternal death rate is especially concentrated in several specific areas of the world. In fact, almost 99% of all maternal deaths occur in developing countries [3]. This high and uneven mortality distribution reflects global inequities of access to medical services and medical treatment. There are large mortality differences not only between countries but also within countries. These disparities in mortality rates persist even between high-income and low-income women, as well as between women living in rural areas and urban areas. Complications during pregnancy and childbirth are thus among the leading causes of death in developing countries [2], [3]. Most of these complications develop during pregnancy, while other complications may happen before pregnancy but are further worsened over the course of pregnancy. However, almost all of these maternal deaths during pregnancy occurred in low-resource settings, and most of them could have been prevented or treated.

Complications of pregnancy may include disorders of high blood pressure, gestational diabetes, infections, preeclampsia, pregnancy loss and miscarriage, preterm labor, and stillbirth. Other complications include severe nausea, vomiting, and iron-deficiency anemia [4], [5]. Thus, pregnancy may be affected due to these conditions, which require additional ways of assessing and evaluating fetal well-being. These conditions may involve medical problems in the mother that could impact on the fetus, pregnancy-specific problems, and diseases of the fetus [6]. In association with increased risk to the fetus, medical problems in the mother include essential hypertension, pre-eclampsia, renal and autoimmune disease, maternal diabetes, and thyroid disease [7]-[10]. Other medical problems in pregnancy, which pose increased risk to fetal health, are prolonged pregnancy, vaginal bleeding, reduced fetal movements, and prolonged ruptured membranes [11]. Additionally, intrauterine growth restriction, fetal infection, and multiple pregnancies increase the risks to the fetuses [11], [12]. As a result, these conditions could lead to neurodevelopmental problems in infancy, such as non-ambulant cerebral palsy, developmental delay, auditory and visual impairment, and fetal compromise, which might lead to morbidity or mortality in the newborn.

In order to assess fetal well-being and monitor for increased risk of complications of pregnancy, cardiotocography (CTG) is widely used as a technical method of continuously measuring and recording the fetal heart rate (FHR) and uterine contractions during pregnancy. This provides the possibility of monitoring the development of fetal hypoxia and intervening appropriately before severe asphyxia or death occurs [13]. In association with uterine contractions, the FHR along with its variability, reactivity, and possible decelerations are important measurements for assessment of fetal well-being [14]. The

FHR can be obtained via an ultrasound transducer placed on the mother's abdomen. The CTG, which depends on FHR, uterine contractions, and fetal movement activity, is utilized to detect and identify dangerous situations for the fetus. During the antepartum and intrapartum periods in pregnancy and childbirth, the CTG is often used for assessment and evaluation of fetal conditions by obstetricians.

Recently, advanced technologies in modern medical practices have successfully allowed robust and reliable machine learning and artificial intelligence techniques to be utilized in providing automated prognosis based on early detection outcomes in many medical applications [15]-[18]. The implementation and feasibility of machine learning tools can significantly aid medical practitioners in making informed medical decisions and diagnoses, potentially reducing maternal and fetal mortality and complications during pregnancy and childbirth and significantly aiding populations in both developing and developed countries. Diagnosing the FHR multiclass morphologic pattern is a challenging task, but computer-aided detection (CAD) based on machine learning technologies have been developed to provide automated classifications for fetal state during pregnancy [19]. Previous research reports used CAD approaches to diagnose the fetal state in pregnancy based on a method of support vector machines (SVM) with a Gaussian kernel function [20], [21]. Other research reports included classification of cardiotocograms using Neural Network and Random Forest classifiers [22], [23]. However, these above mentioned machine learning methods and approaches were designed and developed to classify and predict only binary outcomes of normal versus abnormal cases during medical diagnosis in patients or as normal versus pathological cases in pregnancy using clinical diagnostic datasets of patients and CTG data in pregnancy, respectively.

In this research, an alternative and enhanced artificial intelligence approach is proposed for CTG diagnosis of fetal assessment based on multiclass morphologic pattern predictions, including 10 target classes, using deep learning classification models. The designed and developed deep learning classification and prediction models include two systems: a deep learning-based training classification model and a deep learning-based prediction model (also known as a diagnosis model). The training classification model is based on a multilayer perceptron with a multiclass softmax classification using deep learning technologies. The diagnosis model is used to distinguish and classify the presence or absence of multiclass morphologic patterns for outcome predictions of complications during pregnancy. By uniquely integrating multiclass morphologic patterns and predictions instead of binary predictions of normal versus pathological cases, the models provide a more reliable and specific diagnosis based on fetal health assessment with CTG. The performances of the deep learning-based classification and prediction model for diagnosing multiclass morphologic patterns in pregnancy were evaluated using multiclass measures based on averages of recalls (also known as sensitivities), precisions, *F*-scores, misclassification errors, and diagnostic accuracies.

II. Cardiotocography Data Description

In this section, descriptions and characteristics of the CTG data sets regarding complications in pregnancy are introduced. The CTG data sets, which have been used in this research paper, were obtained from the CTG databases available in the UCI Machine Learning Repository [24]. These databases consist of data information on measurements of FHR and uterine contraction features during pregnancy based on Cardiotocograms, which were contributed by the Biomedical Engineering Institute, Porto, Portugal, and the Faculty of Medicine, University of Porto, Portugal in September 2010. These data sets were collected based on clinical instances in pregnancy in 1980 and periodically from 1995 to 1998, resulting in a constantly increasing dataset size.

There are a total of 2,126 clinical instances, representing different complications of pregnancy on fetal cardiotocograms in the CTG dataset. The clinical instances on the fetal cardiotocograms were automatically processed, and their respective diagnostic features were measured. These clinical instances were also classified with respect to a morphologic pattern by three expert obstetricians and had consensus classification labels assigned to each of them. Each clinical instance in the CTG dataset contains 21 input attributes and one multiclass attribute as well as one fetal state. The multiclass attribute represents the multiclass morphologic patterns, which includes the 10 target classes. Additionally, this multiclass attribute is represented by an integer valued from "1" to "10", where each of integers represents one of the morphologic patterns in pregnancy. The fetal state is assigned one of the 3 classes, including normal, suspect, or pathologic cases. Thus, the CTG dataset can be used for building classification and predictive models based on the 10-class, 3-class, or even the 2-class classification experiments by eliminating the suspect class category in fetal state. In previous reports [20]-[23], several machine learning-based classification models eliminated all suspect cases in fetal state and were established based on only a binary classification of the fetal state in terms of normal and pathologic cases.

In this research paper, the multiclass morphologic patterns, including all 10 of the target classes, have been utilized for developing the deep neural network classification and prediction models. Pattern recognition and prediction of multiple target outcomes is a challenging task in the field of machine learning and artificial intelligence; since multiclass morphologic patterns with imbalanced sample sizes of the 10 target classes in the CTG data will be used, this task thus proves advanced. Ultimately, however, the integration of all 10 target classes and multiclass morphologic patterns, compared to previous research model's use of binary classification, allows a more reliable and realistic diagnosis and prediction of multiclass outcomes, thus aiding patients with an accurate and more specific fetal health assessment.

III. Deep Neural Network Architectures and Classifiers

In this section, we present a CTG classification and diagnosis model for the multiclass morphologic pattern prediction, representing the 10 target classes for fetal outcome forecasting in pregnancy using the deep neural network

classification and prediction models along with corresponding algorithms, methods, approaches, and architecture. Furthermore, we discuss some of the special techniques that can be used to prevent overfitting and enhance the deep neural network classification and prediction model performances for multiclass morphologic pattern prediction and CTG diagnosis.

Deep learning consists of neural networks that teach themselves and make decisions autonomously. Deep learning methods and architectures have gained significant attention in the area of artificial intelligence in recent years. It has recently expanded exponentially in both academic communities and global high-tech industries since 2011 [25]. One of the most important deep learning architectures is a Deep Belief Network, which is built by stacking a set of restricted Boltzmann machines (RBM) [26]-[28]. The RBM is a generative stochastic artificial neural network that can learn from a probability distribution over its input data. Depending on an objective task, the RBM can also be trained either for supervised learning or for unsupervised learning. Another important deep learning architecture is called deep auto-encoder [29]. The deep auto-encoder is also an artificial neural network, usually used for unsupervised learning [29]-[31]. The deep auto-encoder is capable of learning an encoding representation based on a set of input data and has become more widely used for learning generative models of data.

A traditional multilayer perceptron neural network model can be considered as a processor that acquires and stores experiential knowledge through a machine learning process during a training process [15], [17]. In order to retain the knowledge, synaptic weights that resemble interneuron connections are used. The training process of a learning algorithm involves the modification of the synaptic weights of the model in order to obtain a desired objective. The multilayer perceptron neural network model uses a back-propagation approach for training the neural network classification unit during the training process. The back-propagation approach based on the Widrow-Hoff learning rule [15], [17], [32] can be used to minimize the objective function for the neural network model. The input data and the corresponding output data are used to train the neural network classification model until the model appropriately approximated a function within a prior defined error value. During the training process, a learning algorithm is used to adjust weights and biases by utilizing the derivative vectors of errors back-propagated through the neural network classification unit.

In theory, deep neural networks and the multilayer perceptron neural networks, which have the exact same network structure and computations, perform similarly if the deep neural networks and multilayer perceptron neural networks have been given the same conditions. Both deep neural networks and the multilayer perceptron neural networks consist of perceptrons in terms of linear and nonlinear transformation functions. The nonlinear transformation functions between layers of perceptrons enable neural networks to be used for modeling nonlinear behaviors.

Deep neural networks differ from multilayer perceptron neural networks in terms of the network depth, which is determined according to the number of hidden layers in the network. In general, a neural network with three or more hidden layers is considered as a deep neural network. In that case, the higher layers are building new abstractions on top of previous layers, usually leading to learning a better solution with the deep neural network. On the other hand, the number of hidden layers in the network also entails difficulties to train the network in practice. This is because increasing the number of hidden layers in the networks leads to two major issues:

1) Vanishing gradients: The back-propagation approach [15], [17] becomes less helpful in passing information from the higher layers to the lower layers. The gradients become small relative to the weights of the networks and begin to almost vanish when information is passed back.

2) Overfitting: The deep neural network classification model performs very well with a training dataset, but the model shows poorer performances on a real testing dataset. Overfitting is the central problem in the field of machine learning and artificial intelligence.

Fig. 1 shows the deep neural network architectures based on the multilayer perceptron neural networks in detail. Architecturally, the simplest form of the deep neural networks is a feedforward and non-recurrent neural network very similar to the multilayer perceptron neural network, which has an input layer (green color), an output layer (green color), and one or more hidden layers connecting them, but the number of hidden layers consist of a set of active nodes (blue color) and in-active nodes (red color). In this research paper, we have used this type of deep neural network architecture as a fundamental network system to develop the deep learning-based neural network classification and prediction models. The designed deep neural network architecture allows us to classify the multiclass morphologic patterns with imbalanced sample sizes of the 10 target classes in the CTG data for fetal assessment during pregnancy.

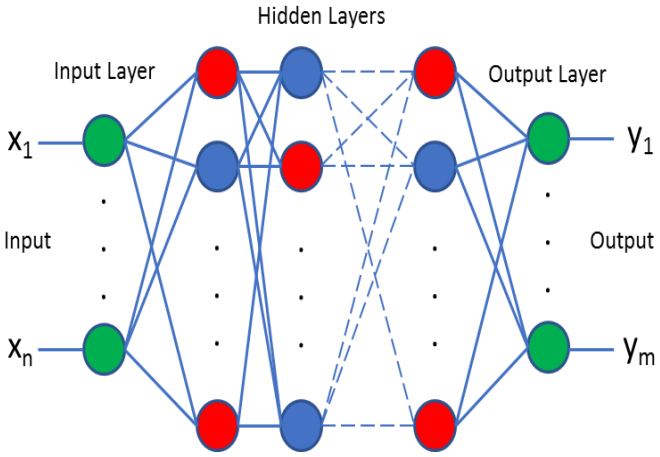

Fig. 1. A deep neural network architecture with an input layer, the number of hidden layers, and an output layer, where the blue cycles are active nodes, the red cycles are in-active nodes in the number of the hidden layers, and the green cycles present an input layer and an output layer.

The deep neural network architecture is composed of the multiple perceptrons, which are stacked one after the other in a layer-wise fashion. The input matrix data $\boldsymbol{X}$ is fed into the input layer, which is a multidimensional perceptron with a weight matrix $\boldsymbol{W}_1$, bias vector $\boldsymbol{B}_1$, and a transfer function Φ_1. The output of the input layer is then fed into the first hidden layer, which is a perceptron with another weight matrix $\boldsymbol{W}_2$, bias vector $\boldsymbol{B}_2$, and a transfer function Φ_2. This process continues for every one of the L hidden layers, which is again a perceptron with another weight matrix $\boldsymbol{W}_L$, bias vector $\boldsymbol{B}_L$, and a transfer function Φ_L until we reach the output layer. As can be seen, according to Fig. 1, we refer to the first layer as the input layer, the last layer as the output layer, and every other layer as a hidden layer in the network architecture.

The deep neural network architecture with one hidden layer has a mathematical representation:

$$\boldsymbol{Y} = \Phi_2(\Phi_1(\boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{B}_1)\boldsymbol{W}_2 + \boldsymbol{B}_2), \quad (1)$$

where $\boldsymbol{Y}$ is an output matrix data. The deep neural network architecture with two hidden layers computes a function in the following:

$$\boldsymbol{Y} = \Phi_3(\Phi_2(\Phi_1(\boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{B}_1)\boldsymbol{W}_2 + \boldsymbol{B}_2)\boldsymbol{W}_3 + \boldsymbol{B}_3), \quad (2)$$

and, in general, the deep neural network architecture with the number of (*L-1*) hidden layers calculates an output function given by:

$$\boldsymbol{Y} = \Phi_L((\cdots \Phi_3(\Phi_2(\Phi_1(\boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{B}_1)\boldsymbol{W}_2 + \boldsymbol{B}_2)\boldsymbol{W}_3 + \boldsymbol{B}_3)\cdots)\boldsymbol{W}_L + \boldsymbol{B}_L), \quad (3)$$

where the transfer function Φ_n, $n = 1, 2, \ldots, L$, can be either a linear or a nonlinear transfer function.

A. Activation Function

An activation function of a neural node in the neural network defines an output of that neural node given a set of inputs. In an artificial neural network or deep neural network architecture, this activation function is also called a transfer function, which can be a linear or non-linear transfer function. The most common transfer functions that are used in deep learning or deep neural network architectures are as follows:

1) Scaled exponential linear unit

$$F(x) = \lambda \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \quad (4)$$

where α and λ are hyper-parameters to be adjusted, $\alpha > 0$ and $\lambda > 0$. When $\alpha = 0$ and $\lambda = 1$, (4) is called the *Rectified linear unit* (ReLU); where $\lambda = 1$, (4) is known as the *exponential linear unit* (ELU).

2) Sigmoid function unit

$$F(x) = \frac{1}{1+e^{-x}} \quad (5)$$

Equation (5) is a *sigmoid function*, which is real-valued and differentiable, having a non-negative or non-positive first derivative, one local minimum, and one local maximum. In general, the sigmoid function exists as a range from 0 to 1 and is used for binary classification. It is especially used for classification models where we want to predict a probability as an output.

3) Softmax function

$$F(x_j) = \frac{e^{x_j}}{\sum_k^K e^{x_k}} \quad (6)$$

where x is a vector of the inputs to the output layer, and $j = 1, 2, \ldots, K$, indexes the output units. The *softmax function* is often used for any number of multiclass classifications.

In this research paper, the ReLU function has been used in the input and hidden layers, and the softmax function is used in the output layer for the deep neural network architecture.

B. Dropout

One of the primary pitfalls of machine learning and artificial intelligence is overfitting when the model catastrophically sacrifices generalization performances for the purposes of minimizing training loss. In other words, a deep neural network model performs really well based on a training dataset. However, in practice, the deep neural network model performs much more poorly on testing data or real unseen testing data. Indeed, overfitting is one of the key critical problems in the field of machine learning and artificial intelligence.

Deep neural networks, which consist of multiple linear and non-linear hidden layers, have a self-learning capability of capturing very complicated relationships between their inputs and outputs. However, with a limited size of a training dataset, many of these complicated relationships could be the result of sampling noise, that is, they may exist in the training set but not in the real testing data. This leads to overfitting. In addition, large deep neural networks are slow to train for use in applications, thereby making it difficult to handle overfitting by combining the predictions of many different large neural nets at testing time.

Dropout is one of several effective and powerful regularization techniques to prevent deep neural network architectures from overfitting [33], [34]. The key idea of the dropout, based on a theory of the role of sex in evolution [35], is to randomly eliminate units along with their network connections from the deep neural networks during a training process. In other words, the central idea of the dropout is to take a deep neural network classification model, which overfits easily, and to train only smaller subsets of the classification models from the deep neural network architectures. As a result, the dropout technique can prevent the deep neural network units from co-adapting too much, thereby avoiding a single node specializing to a task.

Additionally, a dropout technique for the deep neural networks can be viewed as an alternative form of ensemble learning, in which each member of the ensemble learning is trained based on a different subsample of the input data, thereby resulting in learning only a subset of the entire input feature space. At each training step within a batch size, the dropout technique creates a different deep neural network by randomly removing some of the neural units from the hidden layer and/or even input and output layers. Conceptually, the dropout technique actually achieves a similar outcome such that an ensemble learning system uses many different deep neural networks at each of the steps (or batch size) with a subset of input data during the training process. During the

testing process, the deep neural network is only used with the scaled down weights (or partial weights in the network) instead of using entire neural units. Thus, from a point of mathematical view, the dropout technique approximates ensemble averaging using the geometric mean as average [36].

The dropout technique for the deep neural networks has been especially successful in applications because of its simplicity and remarkable effectiveness as a regularization function as well as its interpretation as an exponentially large ensemble learning for the deep neural networks. As a result, this dropout technique implemented in the deep neural network model significantly reduces overfitting issues and provides major improvements over other regularization methods.

C. Regularization

Regularization is one of the key elements of machine learning and artificial intelligence, especially in deep learning. The regularization allows deep neural networks to generalize well to testing data even when the networks are trained based on a finite training set or an imperfect optimization procedure [37], [38]. In other words, regularization can be considered as any modification to a learning algorithm in which is intended to reduce its network test error but not its network training error. In essence, regularization is a supplementary technique that can be used to make the model performance better in general and to produce better results on the testing data [38]. In conjunction with the dropout technique, regularization is another mathematical method for combating overfitting for the deep neural networks.

One of the most popular regularizations is L_2 regularization also known as weight decay, which takes a more direct approach than the dropout technique for regularizing. Generally, a common underlying cause for overfitting is that the deep neural network classification model is too complex in terms of large parameters for the problem based on a training data set. In other words, the regularization can be used to decrease complexity of the deep neural network classification model while maintaining the same number of large parameters. Thus, in order to minimize a L_2 norm, the regularization does so by penalizing weights with large magnitudes using a hyper-parameter λ to specify the relative importance of the L_2 norm for minimizing the loss on the training data set.

Formally, training a deep neural network f_θ is to find a weight function $\theta(\boldsymbol{w}, \boldsymbol{b})$, where $\boldsymbol{w}$ and $\boldsymbol{b}$ denote weights and bias, respectively, such that the expected regularized loss can be minimized:

$$E(\theta, D) = arg\ \min_\theta \left\{\frac{1}{D}\sum_{(x_i,t_i)\in D} E(f_\theta(x_i), t_i)\right\} + \lambda\|\theta\|_p, \quad (7)$$

where D is a training data and (x_i, t_i) are samples in the training data D; the x_i are inputs and t_i are targets. The hyper-parameter λ can be used to control the relative importance of the regularization function. The first item and second item in (7) are referred to as an error function and a regularization error, where

$$\|\theta\|_p = \left(\sum_{j=0}^{N} |\theta_j|^p\right)^{\frac{1}{p}}, \quad (8)$$

which is the L_p norm of θ. If p = 1, (8) is L_1 regularization. If p = 2, (8) is L_2 regularization. Note that the error function in (7), which is dependent on the targets, assigns a penalty to model predictions according to whether or not the model predictions are consistent with the targets. The regularization error assigns a penalty to the model depending on anything except the targets.

D. Initialization

Training deep neural networks is difficult because of vanishing or exploding activations and gradients. The central challenge in training deep neural networks is about how to deal with the strong dependencies that exist during training between the parameters across a large number of hidden layers. This is because a solution to a non-convex optimization algorithm, such as the method of stochastic gradient descent, heavily depends on initialization weights in the deep neural networks. In other words, if the initialization weights in the deep neural networks start too small, then the initialization weights shrink as they pass through each of the hidden layers until they are too tiny to be useful. On the other hand, if the initialization weights in the deep neural networks begin too large, then the initialization weights quickly rise as they pass through each hidden layer until they are too large to be useful. These behaviors are referred to as saturation in training deep neural networks because of nonlinear activation functions embedded in the hidden layers.

Note that deep neural networks with linear and/or nonlinear activation functions initialized from unsupervised pre-training methods, such as deep RBM and deep auto-encoder [39]-[41], do not suffer from these saturation behaviors. Consequently, another important note is that even in the presence of very large amounts of training data in a supervised learning, stochastic gradient descent (SGD) is still subject to a degree of overfitting to the training data. In that sense, unsupervised pre-training method based on the deep RBM and deep auto-encoder interacts intimately with the optimization process. The positive effect of the unsupervised pre-training method is seen not only in generalization error but also in training error when the amount of training data becomes large.

Training deep RBM and deep auto-encoder as an unsupervised pre-training method for the deep neural networks can be considered a breakthrough in effective training strategies [26], [42]-[44]. The unsupervised pre-training method is generally based on greedy layer-wise unsupervised pre-training followed by supervised fine-tuning [41]. Each layer is pre-trained with an unsupervised learning algorithm by learning nonlinear activation functions of their inputs from the previous layers, which capture the major variations in their inputs. Lastly, the unsupervised pre-training method establishes the stage for a final training phase in which the deep neural networks is fine-tuned with respect to a supervised learning criterion of the gradient-based optimization.

Another initialization method for the deep neural networks is known as *Xavier* initialization [39], which is used to make sure that the weights are in a reasonable range of values throughout many hidden layers. Assume that there is an input X with N components and a linear neuron (or combination) with random weights W:

$$Y = \sum_{i=1}^{N} w_i X_i. \quad (9)$$

The variance of this liner combination Y is given by [45]:

$$Var(\sum_{i=1}^{N} w_i X_i) = \sum_{i=1}^{N} w_i^2 Var(X_i) + \sum_{i \neq j}^{N} w_i w_j Cov(X_i, X_j). \quad (10)$$

If the random variables $X_1, \ldots, X_N$ are independent and identically distributed, this always leads to uncorrelated random variables such that

$$Cov(X_i, X_j) = 0, \ for\ i \neq j. \quad (11)$$

Thus, (11) is rewritten to

$$Var(\sum_{i=1}^{N} w_i X_i) = \sum_{i=1}^{N} w_i^2 Var(X_i). \quad (12)$$

In addition, we further assume that the deep neural network weights w_i and inputs X_i are uncorrelated and both have zero-mean:

$$\sum_{i=1}^{N} w_i^2 Var(X_i) = \sum_{i=1}^{N} Var(W) Var(X) = NVar(W) Var(X). \quad (13)$$

Comparing (12) to (13), we obtain a result as follows:

$$Var(\sum_{i=1}^{N} w_i X_i) = NVar(W) Var(X). \quad (14)$$

Equation (14) implies that the variance of the output is the variance of the input with a scaled function by *NVar(W)*. If we further want to make the variance of the input to be the same as the variance of the output, it must hold $Var(W) = \frac{1}{N}$ for the inputs so that we are able to preserve variance of the inputs after passing through a number of the hidden layers. For the backpropagation update, we also need to ensure that $Var(W) = \frac{1}{M}$ for the outputs. Thus, in general, for implementation of the initialization on the deep neural networks, the variance of the weights for the deep neural networks can be set to their average based on the inputs and outputs, that is,

$$Var(W) = \frac{1}{N+M}. \quad (15)$$

IV. Multiclass Evaluation Methods on Deep Neural Network

In this section, multiclass evaluation methods for the performances of the deep neural network classification model in multiclass morphologic pattern prediction based on the CTG data are discussed in detail.

The evaluation of the model performances for the deep neural networks is typically based on testing data sets, rather than analytically in the field of machine learning and artificial intelligence. The classification effectiveness of machine learning models, deep neural networks, and/or any other type of models can usually be measured in terms of model sensitivity (also known as recall), specificity, precision, *F*-score, accuracy, and misclassification error [45], [16]. In this section, we extend the evaluation methods for the effectiveness measurements of the deep neural network classification model from a binary classification to a multiclass classification problem.

Let $C_1, \ldots, C_K$ be multiclass labels, in which we want to predict K labels using deep neural network classification models. For correct decisions, let *TP* be a decision to assign similar multiclass to the same cluster, and let *TN* be a decision to assign dissimilar multiclass to different clusters. On the other hand, for incorrect decisions, let *FP* be a decision to assign dissimilar multiclass to the same cluster, and let *FN* be a decision to assign similar multiclass to different clusters.

For the effectiveness measurement of the deep neural network classification models, a global calculation of the *TP*, *TN*, *FP*, and *FN* can be computed in the following:

$$TP = \sum_{i=1}^{N} TP_i, \quad (16)$$

$$TN = \sum_{i=1}^{N} TN_i, \quad (17)$$

$$FP = \sum_{i=1}^{N} FP_i, \quad (18)$$

and

$$FN = \sum_{i=1}^{N} FN_i, \quad (19)$$

where TP_i, FP_i, TP_i, and TP_i denote the local measures, representing the number of *true positive*, *false positive*, *true negative*, and *false negative* test examples with respective to the *i*-th class label. This evaluation method is referred to as *micro average* [46], [47].

Then, *sensitivity* for the multiclass classification is defined as the probability of correctly identifying those with the true positive rate [16]:

$$Sensitivity = \frac{TP}{TP+FN}, \quad (20)$$

where sensitivity is known as *recall* in the field of machine learning and deep learning. The *specificity* is defined as the probability of correctly identifying true negative rate:

$$Specificity = \frac{TN}{FP+TN}, \quad (21)$$

The *precision* is defined as

$$Precision = \frac{TP}{TP+FP}, \quad (22)$$

In performance measures of the deep neural networks, the recall in (20) is a measure of quantity while the precision in (22) is a measure of quality. Both the recall and precision are in a mutual relationship based on the understanding and measure of relevance.

Another method, which is similar to the one-vs-all classification technique, is to calculate all the local recalls R_i and precisions P_i for each C_i of the multiclass. This method is referred to as *macro average*. Then, the average of recalls is as follows:

$$\bar{R} = \frac{1}{K} \sum_{i=1}^{K} R_i, \quad (23)$$

and average of precisions is given by,

$$\bar{P} = \frac{1}{K} \sum_{i=1}^{K} P_i. \quad (24)$$

Note that micro and macro average methods represent different calculation behaviors, thereby leading to different results in the multiclass evaluation of the classification model effectiveness for the deep neural networks.

V. Results

In this research paper, the deep neural network classification model is proposed for accurate diagnosis of fetal state based on the CTG data and the multiclass morphologic pattern outcome predictions. The proposed deep neural network classification model has a deep neural network

architecture, including 21 input units, first and second hidden layers, and 11 binary output units. The 11 binary output units, which allow us to form 10 unique sequences, can be used to represent the 10 target classes in the morphologic pattern outcomes on fetal assessment for multiclass classifications and predictions. The first hidden layer contains 105 units with each of the ReLU activation functions and 25% dropout rate of the network. The second hidden layer has 42 units, also connected with the ReLU activation functions, and 20% dropout rate of the network. Each of the 11 output units in the last stage of the deep neural network architecture is connected to a softmax activation function. For each batch process during the training of the deep neural network, the dropout rates in the first and second hidden layer are randomly applied to the deep neural network, thereby resulting in random connections within the deep neural network architecture. Doing so allows us to generate an alternative form of ensemble learning as well as reduce and/or prevent overfitting issues for the deep neural network classification model.

In the CTG data, each clinical instance consists of 40 raw attributes. Among all of the raw attributes, only 23 of them can be used for developing the deep neural network classification and prediction models. The other 13 attributes are not recommended to be used according to the attribute restriction in the CTG data. Table I lists the detailed 23 raw attributes, which had been used for the development of the deep neural network classification model. The variable of the "Class" in Table I is referred to as a target variable, which includes 10 integers from 1 to 10, representing different morphologic pattern behaviors of complications in pregnancy.

The proposed deep neural network classification and prediction models were applied to all clinical instances, which represent complications of pregnancy based on fetal assessments in the CTG data, and were used to predict the multiclass morphologic patterns with the 10 target class outcomes. Table II shows the details of the clinical instances in terms of the number of cases and percentages of the presence or absence of complications during pregnancy based on fetal assessments in each of the 10 morphologic pattern outcome data sets.

As can be seen in Table II, there are large differences in terms of percentages of the number of cases within each of the multiclass morphologic pattern outcomes in the CTG data. Class C3 has the lowest percentage of number of cases at 2.49%, while class C2 has the highest percentage of number of cases at 27.23%. As can be seen, the number of the sample distributions in the multiclass morphologic pattern outcomes would lead to a challenge in multiclass classification with imbalanced sample sizes for the CTG data in the field of machine learning and deep learning.

In order to evaluate the effectiveness of the performances of the developed deep neural network classification and prediction models, the model accuracy and misclassification error as well as the recall, precision, and *F*-score were estimated using a nonparametric approach based on a holdout method [45]. The holdout method applied by partitioning the CTG data into two mutually exclusive data sets, training data and testing data, respectively. The deep neural network classification model was first trained using the training data, and then it was tested using the testing data.

In this research, the entire CTG data was randomly separated into 70% training and 30% testing data sets using the holdout method. The deep neural network classification and prediction models were trained and tested by using the 70% training and 30% testing data sets, respectively. The training and testing processes for the deep neural network classification and prediction models were repeated 24 times based on the different 70% training and 30% testing data sets. Doing so would determine an average of the testing performance results for the deep neural network classification and prediction models.

TABLE I. THE RAW ATTRIBUTES OF THE VARIABLE NAMES AND DESCRIPTIONS IN THE CTG DATA SET

Variable Name	Descriptions	Variable Name	Descriptions
LB	FHR baseline (beats per minute)	Min	Minimum of FHR histogram
AC (Second)	Number of accelerations	Max	Maximum of FHR histogram
FM (Second)	Number of fetal movements	Nmax	Number of histogram peaks
UC (Second)	Number of uterine contractions	Nzeros	Number of histogram zeros
DL (Second)	Number of light decelerations	Mode	Histogram mode
DS (Second)	Number of severe decelerations	Mean	Histogram mean
DP (Second)	Number of prolongued decelerations	Median	Histogram median
ASTV	Percentage of time with abnormal short term variability	Variance	Histogram variance
MSTV	Mean value of short term variability	Tendency	Histogram tendency
ALTV	Percentage of time with abnormal long term variability	Class	FHR pattern class code (1 to 10): 1-calm sleep 2-REM sleep 3-calm vigilance 4-activevigilance 5-shift pattern 6-accelerative or decelerative pattern (stress situation) 7-decelerative pattern (vagal stimulation) 8-largely decelerative pattern; 9-flat-sinusoidal pattern (pathological state) 10-suspect pattern
MLTV	Mean value of long term variability	NSP	fetal state class code (N=normal; S=suspect; P=pathologic)
Width	Width of FHR histogram		

TABLE II. CLINICAL INSTANCES OF COMPLICATIONS DURING PREGNANCY BASED ON FETAL ASSESSMENTS OF THE MULTICLASS MORPHOLOGIC PATTERNS IN THE 10 TARGET CLASS OUTCOMES IN THE CTG DATA

MP	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
NC	384	579	53	81	72	332	252	107	69	197
%	18.06	27.23	2.49	3.81	3.39	15.62	11.85	5.03	3.25	9.27

Note: MP means Morphologic patterns; NC means Number of cases.

Fig. 2 shows a graph plot of the designed and developed deep neural network classification model performances using the training dataset at each of the epochs for 80,000 iterations during the training process. The accuracy of the deep neural network classification model is 97.32% based on the training dataset. Furthermore, Fig. 3 illustrates a graph plot of the deep neural network classification model loss function error throughout the 80,000 iterations during the training process. The loss function error of the deep neural network classification model is 0.0941 in an optimal sense of minimum mean square error (MMSE). In general, the higher the accuracy that the deep neural network classification model can achieve the lower the loss function error is obtained during the training process.

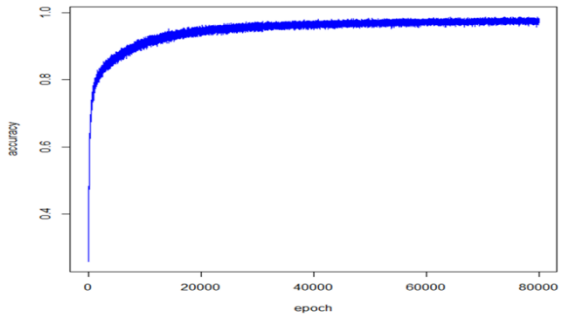

Fig. 2. A graph plot of the deep neural network classification model performance in terms of accuracy at each of the epochs for 80,000 iterations during the training process, where x-axis denotes each of the epochs and y-axis represents the model accuracy (the value of 1.0 at the y-axis represents a trained model with 100% accuracy).

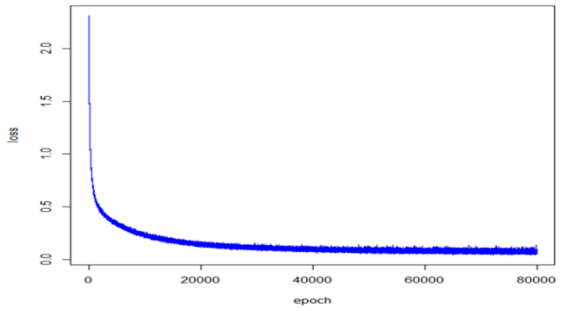

Fig. 3. A graph plot of the deep neural network classification model loss function error at each of the epochs for 80,000 iterations during the training process, where x-axis denotes each of the epochs and y-axis represents the model loss function error in MMSE.

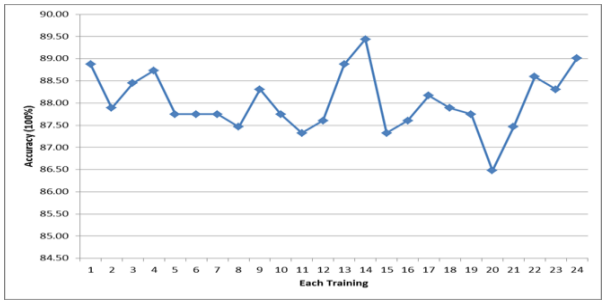

Fig. 4. A graph plot of the deep neural network classification model performances for 24 accuracy measures based on the 24 different testing data sets, where x-axis represents each of the 24 tests and y-axis is the tested model accuracy in percentage.

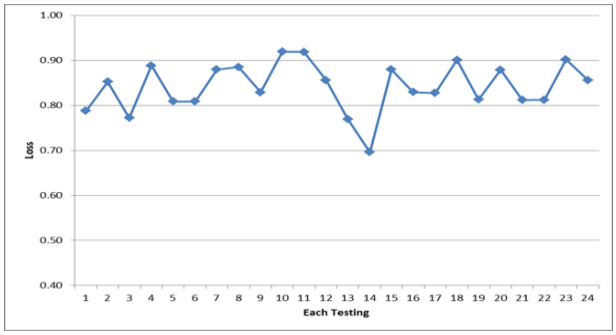

Fig. 5. A graph plot of the deep neural network classification model performances with 24 loss function error measures based on the 24 different testing data sets, where x-axis represents each of the 24 tests and y-axis is the tested model loss function error.

In this research paper, we repeated the same training and testing processes 24 times for the deep neural network classification model. For each of the 24 times, the entire CTG data was randomly divided into 70% training and 30% testing data sets. Then the deep neural network classification and prediction models were trained using the training data sets and tested using the testing data sets, respectively. The deep neural network classification model performances were recorded based on the 24 testing data sets. Displaying the results, Fig. 4 shows a graph plot of the deep neural network classification model performances in terms of 24 accuracy measures based on the 24 different testing data sets. As can be seen, the highest testing accuracy of the deep neural network classification model is 89.44%; the lowest testing accuracy is 86.48%. This leads to an average model accuracy of 88.02% with a standard deviation of 0.67%. The average misclassification error of the model is 11.98%. Correspondingly, Fig. 5 is a graph plot of the deep neural network classification model performances regarding the 24 loss function error measures based on the 24 different testing data sets as well. The best MMSE of the loss function error is 0.70 while the worst MMSE is 0.92. The average MMSE of the loss function errors is 0.84 with a standard deviation of 0.05.

In general, according to the training and testing results, whether the deep neural network classification model falls into a global or local minimum in an optimal sense is inconsequential. If the deep neural network overfitting in a minimum sense can be controlled, this deep neural network classification model would be determined to have realistically accurate diagnoses for fetal assessment during pregnancy based on the multiclass morphologic patterns of the 10 target class predictions.

Thus, in order to optimize the deep neural network classification model, the “rmsprop” method was used during the training process in this research. The “rmsprop” method is one of the mini-batch learning methods, which divides the learning rate for a weight by a running average of the magnitudes of recent gradients for the weight [48] and keeps a moving average of the squared gradient for each weight. In this research, a mini-batch size of 80 was used along with a learning rate of 0.00005 during the training processes.

Table III shows a combined confusion matrix (also known as an error matrix), which is a special table, using a summation

of the 24 individual confusion matrices based on 24 independently individual testing results. This combined confusion matrix allows us to visualize the deep neural network classification model performances in detail. Each row of the combined confusion matrix represents the number of clinical instances in a predicted class, while each of the columns represents the number of clinical instances in an actual multiclass. In statistics, this combined confusion matrix can also be called a contingency table along with two dimensions in terms of "actual" and "predicted" as well as identical sets of "classes" in both dimensions.

Based on results of the combined confusion matrix in Table III, corresponding values of recall, precision, and *F*-score for each of the multiclass morphologic patterns based on the 10 target classes using the deep neural network classification model were estimated as shown in Table IV. As can be seen, the averages of recall and precision are 84.30% and 84.91% along with standard deviations 8.39% and 6.89%, respectively. This leads to an average of the *F*-scores that is equal to 0.8453 with a standard deviation of 0.0737.

TABLE III. A COMBINED CONFUSION MATRIX BASED ON THE 24 INDEPENDENTLY INDIVIDUAL CONFUSION MATRICES USING THE 24 DIFFERENT TESTING DATA SETS

		Actual Multiclass Cases									
		C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
PC		2584	97	127	0	123	22	27	0	3	154
		101	4736	0	60	14	140	0	0	10	0
		126	3	371	0	0	2	4	0	0	0
		0	110	0	540	1	0	0	0	0	0
		31	27	0	0	457	3	9	0	9	28
		24	67	0	0	0	2278	55	1	4	0
		15	40	6	0	0	28	1798	75	4	0
		0	0	0	0	0	47	0	617	32	0
		0	0	0	0	0	0	0	1	556	100
		143	8	0	0	29	0	3	2	126	1062

Note: PC means predicted cases.

TABLE IV. RECALL, PRECISION, AND *F*-SCORE FOR EACH OF THE MULTICLASS MORPHOLOGIC PATTERNS

Multiclass	**Recall**	**Precision**	***F*-Score**
C1	0.8545	0.8237	0.8388
C2	0.9308	0.9358	0.9333
C3	0.7361	0.7332	0.7347
C4	0.9000	0.8295	0.8633
C5	0.7324	0.8103	0.7694
C6	0.9040	0.9378	0.9206
C7	0.9483	0.9145	0.9311
C8	0.8865	0.8865	0.8865
C9	0.7473	0.8463	0.7937
C10	0.7902	0.7735	0.7817
Average	**0.8430**	**0.8491**	**0.8453**
Standard Deviations	**0.0839**	**0.0689**	**0.0727**

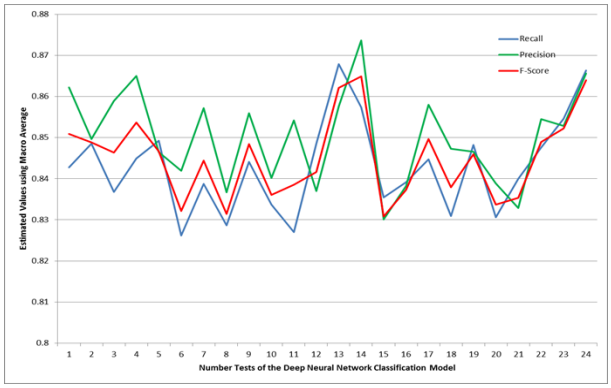

Fig. 6. A graphic plot of the macro average of recall, precision, and *F*-score based on the 24 individual confusion matrices for the deep neural network classification model using the 24 different testing data sets.

In this research, another method utilizing the macro average, which is the one-vs-all classification technique, was also used to estimate recall, precision, and *F*-score. Fig. 6 shows a graphic plot regarding the estimated values of recall, precision, and *F*-score based on the 24 independently individual confusion matrices. The macro average was used to compute all of the local recall and precision values for each of the multiclass morphologic patterns based on each of the 24 confusion matrices. The averages of recall and precision were then calculated according to (23) and (24) based on each of the confusion matrices, thereby leading to the 24 estimated values of recall, precision, and *F*-score as shown in Fig. 6. Based on the macro average, the averages of the recall, precision, and *F*-score are 84.30%, 85.01%, and 0.8508, respectively. Correspondingly, the standard deviations for the averaged recall, precision, and *F*-score are 1.13%, 1.14%, and 0.0100, respectively. As can be seen, in general, the *F*-score curve is displayed between the recall and precision curves.

Furthermore, it is noted that there are some differences in terms of averages and standard deviations of the recall, precision, and *F*-score for the deep neural network classification model using the combined confusion matrix as shown in Tables III and IV as well as using the macro average as shown in Fig. 6. However, generally speaking, there are no significant differences for the averages and standard deviations of the recall, precision, and *F*-score for the deep neural network classification model using the methods of micro and macro averages. Utilizing the 24 independently individual confusion matrices based on the 24 different testing data sets for estimating the deep neural network classification model performances allows us to establish and demonstrate a new and alternative way of representing the model recall, precision, and *F*-score in a dynamic representation.

VI. CONCLUSION AND FUTURE WORK

In this research paper, the deep neural network classification and prediction models were designed and developed for CTG diagnosis and prediction based on fetal assessment in pregnancy with the multiclass morphologic patterns of the 10 target classes with imbalanced samples. In conjunction with the dropout technique, regularization was applied to combat overfitting for the deep neural networks during the training process. As a result, the developed deep

neural network architecture allowed us to not only show a strong, alternative form of largely exponential ensemble learning but also reduce overfitting issues for the deep neural network classification and prediction models. Therefore, the developed deep neural network classification and prediction models can provide highly accurate and consistent diagnoses for fetal assessment regarding complications during pregnancy based on the multiclass morphologic patterns, thereby preventing and/or reducing fetal morbidity or mortality rate as well as maternal mortality rate during and following pregnancy and childbirth, especially in developing countries or in low-resource settings.

The dropout technique was used to enable us to randomly drop neural units with their connections in the deep neural network architecture. It can be treated as a large exponential ensemble learning for the deep neural networks. This significantly reduces overfitting and provides major improvements over traditional regularization methods. However, one of the problems with the dropout technique is that the training period is typically longer than that of a standard deep neural network architecture. This is because the parameter updates in the networks are very noisy. Moreover, the dropout technique can be considered as an alternative way of adding noise to the hidden units in the networks. This becomes a trade-off requirement between overfitting and training time. In other words, by increasing training time, one can already use a high dropout rate and encounter fewer overfitting problems for the deep neural network architectures. Thus, for future work, an interesting direction to take is to speed up the dropout technique during the training processes despite the large deep neural network architecture. Furthermore, another future direction is to use the dropout technique as an adaptive regularization for adaptive ensemble learning to further prevent overfitting, thereby enhancing the model performances of the deep neural network architectures and diagnoses of fetal health assessment with cardiotocography in clinical cases.

ACKNOWLEDGMENT

The authors would like to thank Dr. Joaquim P. Marques de Sa from the Biomedical Engineering Institute, Porto, Portugal; and Dr. Joao Bernardes and Dr. Diogo Ayres-de-Campos from the Faculty of Medicine, University of Porto, Portugal, whose Cardiotocography datasets of clinical instances were contributed to and made available in the Cardiotocography Databases of the UCI Machine Learning Repository.

REFERENCES

[1] G. Sedgh, S. Singh, and R. Hussain, "Intended and unintended pregnancies worldwide in 2012 and recent trends," Studies in Family Planning, Vol. 45, Issue 3, pp. 301-314, September 2014.

[2] C. J. Murray, "Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the global burden of disease study 2013." Lancet, Vol 385, pp. 117-171, January 2015.

[3] World Health Organization, Maternal mortality: fact sheet, Updated November 2016: Available http://www.who.int/mediacentre/factsheets/fs348/en/.

[4] National Institutes of Health, What are some common complications of pregnancy? US Department of Health and Human Services: Available https://www.nichd.nih.gov/health/topics/pregnancy/conditioninfo/Pages/complications.aspx.

[5] Office on Women's Health, Pregnancy Complications, US Department of Health and Human Services: Available https://www.womenshealth.gov/pregnancy/youre-pregnant-now-what/pregnancy-complications.

[6] R. M. Grivell, Z. A. Gillian, M. L. Gyte, and D. Devane, "Antenatal cardiotocography for fetal assessment," Cochrane Database of Systematic Reviews, Issue 9. No. CD007863, pp. 1-57, 2015, John Wiley & Sons, Ltd.

[7] C. Nelson-Piercy and C. Williamson, Medical disorders in pregnancy, In: Chamberlain G, Steer P editor(s), Turnbull's Obstetrics, 3rd Edition, Edinburgh: Churchill Liingstone, pp. 275–97, 2001.

[8] C. Lloyd, Hypertensive disorders of pregnancy, In: Fraser DM, Cooper MA editor(s), Myles Textbook for Midwives. 14th Edition, Edinburgh: Churchill Livingstone, pp. 357–71, 2003.

[9] C. Lloyd, Common medical disorders associated with pregnancy, In: Fraser DM, Cooper MA editor(s), Myles Textbook for Midwives, 14th Edition, Edinburgh: Churchill Livingstone, pp. 321–55, 2003.

[10] National Institute for Health and Clinical Excellence, Diabetes in pregnancy: management of diabetes and its complications from pre-conception to the postnatal period, London, pp. 1-39, March 2008.

[11] C. Gribbin and J. Thornton, Critical evaluation of fetal assessment methods, In: James DK, Steer PJ, Weiner CP editor(s), High Risk Pregnancy Management Options, Elsevier, 2006.

[12] N. M. Fisk and R. P. Smit, Fetal growth restriction; small for gestational age, In: Chamberlain G, Steer P editor(s), Turnbull's Obstetrics, 3rd Edition, Edinburgh: Churchill Livingstone, pp. 197–209, 2001.

[13] I. Ingemarsson, "Fetal monitoring during labor," Neonatology, Vol. 95, No. 4, June 2009.

[14] FIGO News, "Report of the FIGO study group on the assessment of new technology: evaluation and standardization of fetal monitoring," Organized by G. Rooth, A. Huch, and R. Huch, International Journal of Gynecology & Obstetrics, Vol. 25, pp. 159-167, 1987.

[15] G. J. Miao, K. H. Miao, and J. H. Miao, "Neural pattern recognition model for breast cancer diagnosis," Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Bioinformatics, August Edition, pp. 1-8, September 2012.

[16] K. H. Miao, J. H. Miao, and G. J. Miao, "Diagnosing coronary heart disease using ensemble machine learning," International Journal of Advanced Computer Science and Applications, Vol. 7, No. 10, pp. 30-39, 2016.

[17] K. H. Miao and G. J. Miao, "Mammographic diagnosis for breast cancer biopsy predictions using neural network classification model and receiver operating characteristic (ROC) curve evaluation," Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Bioinformatics, September Edition, Vol. 3, Issue 9, pp. 1–10, October 2013.

[18] J. H. Miao, K. H. Miao, and G. J. Miao, "Breast cancer biopsy predictions based on mammographic diagnosis using Support Vector Machine learning," Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Bioinformatics, Vol. 5, No. 4, pp. 1–9, 2015.

[19] D. Ayres-de-Campos, J. Bernardes, A. Garrido, J. Marques-de-Sá, and L. Pereira-Leite, "Sisporto 2.0: A program for automated analysis of cardiotocograms," The Journal of Maternal-Fetal Medicine, Vol. 9, Issue 5, pp. 311-318, October 2000.

[20] P. A. Warrick, E. F. Hamilton, R. E. Kearney, and D. Precup, "A machine learning approach to the detection of fetal hypoxia during labor and delivery," Proceedings of the Twenty-Second Innovative Applications of Artificial Intelligence Conference, pp. 1865-1870, 2010.

[21] Z. Comert and A. F. Kocamaz, "Comparison of machine learning techniques for fetal heart rate classification," Special issue of the 3rd International Conference on Computational and Experimental Science and Engineering, Vol. 132, pp. 451-454, 2017.

[22] C. Sundar, M. Chitradevi, and G. Geetharamani, "Classification of cardiotocogram data using neural network based machine learning technique," International Journal of Computer Applications, Vol. 47, No. 14, pp. 19-25, June 2012.

[23] M. Arif, "Classification of cardiotocograms using Random Forest classifier and selection of important features from cardiotocogram

signal," Biomaterials and Biomedical Engineering, Vol. 2, No. 3, pp. 173-183, 2015.

[24] The UCI Machine Learning Repository, Cardiotocography Data Set: Available http://archive.ics.uci.edu/ml/datasets/Cardiotocography.

[25] J. Wang, "Deep learning: an artificial intelligence revolution," ARK Invest, pp. 1-41, New York, June 2017.

[26] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets." Neural Computation, Vol. 18, pp. 1527–1554, 2006.

[27] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Computation, Vol. 14, No. 8, pp. 1711–1800, 2002.

[28] G. E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, Department of Computer Science, University of Toronto, Canada, UTML TR 2010-003, Version 1, August 2010.

[29] Y. Bengio, "Learning deep architectures for AI," Journal Foundations and Trends in Machine Learning, Vol. 2, pp. 1-127, January 2009.

[30] C. Y. Liou, J. C. Huang, and W. C. Yang, "Modeling word perception using the Elman network," Neurocomputing, pp. 3150–3157, 2008.

[31] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," JMLR: Workshop and Conference Proceedings, pp. 37-50, 2002.

[32] S. Haykin, Neural Network: A Comprehensive Foundation, Macmillan College Publishing Company, 1994.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Neural and Evolutionary Computing, pp. 1-18, July 2012.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, 2014.

[35] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," Proceedings of the National Academy of Sciences, Vol. 107, No. 4, pp. 1452–1457, 2010.

[36] D. Warde-Farley, I. J. Goodfellow, A. Courville, and Y. Bengio, "An empirical analysis of dropout in piecewise linear networks," Machine Learning, pp. 1-10, January 2014.

[37] I. J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[38] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: a taxonomy," Artificial Intelligence, pp 1-27, October 2017.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Vol. 9, Chia Laguna Resort, Sardinia, Italy, 2010.

[40] C. Ryali, G. Nallamala, W. Fedus, and Y. Prabhuzantye, "Efficient encoding using deep neural networks," pp. 1-8, 2015.

[41] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" Journal of Machine Learning Research, Vol. 11, pp. 625-660, 2010.

[42] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," In Advances in Neural Information Processing Systems, Vol. 12, pp. 153–160, 2007.

[43] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," Department of computer science and operations research, University of Montreal, Canadian Institute for Advanced Research, pp. 1-30, April 2014.

[44] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," Proceedings of the 19th International Conference on Neural Information Processing Systems, pp. 1137-1144, 2006.

[45] G. J. Miao and M. A. Clements, Digital Signal Processing and Statistical Classification, Artech House, Inc., 2002.

[46] M. L. Zhang and Z. H. Zhou, "A review on multi-lable learning algorithms," IEEE Transactions on Knowledge and Data Engineering, Vol. 26, Issue 8, pp. 1819-1837, March 2013.

[47] E. Gibaja and S. Ventura, "A tutorial on multi-label learning," ACM Computing Surveys, Vol. 47, Issue 3, April 2015.

[48] G. Hinton, with N. Srivastava, and K. Swersky, "Neural network for machine learning: Lecture 6a overview of mini-batch gradient descent," Computer Science Department, University of Toronto, Winter 2014.