Neural network modeling and an uncertainty analysis in Bayesian framework: A case study from the KTB borehole site

Saumen Maiti[1] and Ram Krishna Tiwari[2]

Received 11 January 2010; revised 5 June 2010; accepted 25 June 2010; published 26 October 2010.

[1] A new probabilistic approach based on the concept of Bayesian neural network (BNN) learning theory is proposed for decoding litho-facies boundaries from well-log data. We show that how a multi-layer-perceptron neural network model can be employed in Bayesian framework to classify changes in litho-log successions. The method is then applied to the German Continental Deep Drilling Program (KTB) well-log data for classification and uncertainty estimation in the litho-facies boundaries. In this framework, a posteriori distribution of network parameter is estimated via the principle of Bayesian probabilistic theory, and an objective function is minimized following the scaled conjugate gradient optimization scheme. For the model development, we inflict a suitable criterion, which provides probabilistic information by emulating different combinations of synthetic data. Uncertainty in the relationship between the data and the model space is appropriately taken care by assuming a Gaussian a priori distribution of networks parameters (e.g., synaptic weights and biases). Prior to applying the new method to the real KTB data, we tested the proposed method on synthetic examples to examine the sensitivity of neural network hyperparameters in prediction. Within this framework, we examine stability and efficiency of this new probabilistic approach using different kinds of synthetic data assorted with different level of correlated noise. Our data analysis suggests that the designed network topology based on the Bayesian paradigm is steady up to nearly 40% correlated noise; however, adding more noise (~50% or more) degrades the results. We perform uncertainty analyses on training, validation, and test data sets with and devoid of intrinsic noise by making the Gaussian approximation of the a posteriori distribution about the peak model. We present a standard deviation error-map at the network output corresponding to the three types of the litho-facies present over the entire litho-section of the KTB. The comparisons of maximum a posteriori geological sections constructed here, based on the maximum a posteriori probability distribution, with the available geological information and the existing geophysical findings suggest that the BNN results reveal some additional finer details in the KTB borehole data at certain depths, which appears to be of some geological significance. We also demonstrate that the proposed BNN approach is superior to the conventional artificial neural network in terms of both avoiding "over-fitting" and aiding uncertainty estimation, which are vital for meaningful interpretation of geophysical records. Our analyses demonstrate that the BNN-based approach renders a robust means for the classification of complex changes in the litho-facies successions and thus could provide a useful guide for understanding the crustal inhomogeneity and the structural discontinuity in many other tectonically complex regions.

Citation: Maiti, S., and R. K. Tiwari (2010), Neural network modeling and an uncertainty analysis in Bayesian framework: A case study from the KTB borehole site, *J. Geophys. Res.*, *115*, B10208, doi:10.1029/2010JB000864.

1. Introduction

[2] Litho-facies classification based on experimental data is one of the important problems in geophysical well-log studies. Since permeability and fluid saturation for a given porosity varies considerably among the litho-facies at a

[1]Indian Institute of Geomagnetism, Navi-Mumbai, India.
[2]National Geophysical Research Institute, Hyderabad, India.

0148-0227/10/2010JB000864

certain depths, classification of litho-facies and their adequate representation in a 3-D cellular geophysical/geological model is vital for understanding the crustal inhomogeneity, the permeability, and the fluid saturation for exploration of oil and gas. The best sources of litho-facies information are core samples of reservoir rocks collected from wells. However, cores are not commonly taken due to high costs. The availability of core samples is also limited in comparison to the number of drilled wells in the geological/geophysical field. Hence, in a situation where core information is not available, the down-hole geophysical logs can be used as an alternative to infer the nature of surrounding rocks/lithology [*Benaouda et al.*, 1999].

[3] During the past decades, several researchers have attempted to solve the problems of litho-facies classification using conventional methods like, graphical cross-plotting and multivariate statistical analyses [*Rogers et al.*, 1992]. In the graphical cross-plotting technique [*Pickett* 1963; *Gassaway et al.*, 1989], two or more well-logs data are cross-plotted to yield lithologies. Multivariate statistical methods such as "principle component" and "cluster analysis" [*Busch et al.*, 1987; *Wolff and Pelissier-Combescure*, 1982] and "discriminant function analysis" [*Delfiner et al.*, 1987] have also invariably been used for the interpretation of borehole data. These techniques are, however, semi-automatic and require a large amount of data, which is costly and not easily available. Further the existing methods are also very tedious and time-consuming, particularly when it deals with the large number of noisy and complex borehole data.

[4] Appropriate mathematical modeling and statistical techniques can be applied to extract the meaningful information about the subsurface properties of the real earth (e.g., lithology, porosity, density, hydraulic conductivity, resistivity, salinity, water/oil saturation, etc.) using surface and/or borehole measurements [*Aristodemou et al.*, 2005]. In order to convalesce the model parameters correctly, an error function, which is a measure of discrepancy between the observables and the predictions from a forward-modeling calculation, is minimized [*Mosegaard and Tarantola*, 1995; *Tarantola*, 1987, 2006; *Devilee et al.*, 1999; *Bosch*, 1999; *Bosch et al.*, 2001; *Aristodemou et al.*, 2005; *Meier et al.*, 2007]. However, well-log signals, which are a proxy of lithology/litho-facies, are essentially the result of complex nonlinear geophysical processes arising primarily due to the variability and interactions of several factors, such as pore fluid, effective pressure, fluid saturation, pore shape, and grain size. Further, the well-log records are often found to be contaminated with inescapable correlated red noise primarily due to the deplorable borehole conditions. Therefore, estimation of lithology/litho-facies from well-log signals essentially constitutes a nonlinear geophysical inverse problem.

[5] In the recent past, artificial neural network (ANN)-based techniques have been extensively applied to solve nonlinear problems in almost all branches of geophysics [*Van der Baan and Jutten*, 2000; *Poulton*, 2001]. Examples include (1) seismic event classification [*Dystart and Pulli*, 1990; *Dai and MacBeth*, 1995], (2) well-log analysis [*Baldwin et al.*, 1990; *Rogers et al.*, 1992; *Helle et al.*, 2001; *Aristodemou et al.*, 2005; *Maiti et al.*, 2007; *Maiti and Tiwari*, 2009], (3) first arrival picking [*Murat and Rudman*, 1992; *McCormack et al.*, 1993], (4) earthquake time series modeling [*Feng et al.*, 1997], (5) inversion [*Raiche*, 1991; *Roth and Tarantola*, 1994; *Devilee et al.*, 1999; *Meier et al.*, 2007], (6) parameter estimation in geophysics [*Calderon-Macias et al.*, 2000], (7) prediction of aquifer water level [*Coppola et al.*, 2005], (8) magneto-telluric data inversion [*Spichak and Popova*, 2000], (9) magnetic interpretations [*Bescoby et al.*, 2006], and (10) signal discrimination [*Maiti and Tiwari*, 2010]. This type of network, however, does yield mean solutions to the inverse problem whose solution is essentially probabilistic in nature [*Devilee et al.*, 1999]. In addition to this, there are several other drawbacks in conventional neural network approaches [*Bishop*, 1995; *Coulibaly et al.*, 2001; *Aires*, 2004; *Maiti and Tiwari*, 2009, 2010]. One of the major limitations in the conventional ANN approach is frequent appearance of local and global minima in the modeled results. To surmount this problem, the network is trained by maximizing a likelihood function of the parameters or equivalently minimizing an error function to obtain the best set of parameters starting with an initial random set of parameters. Sometimes a regularization term with an error function is also included in the process of analysis for preventing over-fitting. However, in the conventional ANN approach, a complex model can fit training data well, but it does not necessarily guarantee smaller errors in the unseen data [*Bishop* 1995; *Coulibaly et al.*, 2001]. This is because the conventional ANN does not take account of uncertainty in the estimation of parameters [*Bishop* 1995; *Nabney*, 2004].

[6] *Roth and Tarantola* [1994] have assessed the stability and the effectiveness of an ANN inversion scheme in the presence of noise while inverting seismogram records to recover the crustal velocity structure. After several experiments, they have concluded that the ANN-based methods are not stable for analyzing strongly correlated noisy geophysical signals; however, the methods could be used for solving the nontrivial inverse problem. More recently, *Devilee et al.* [1999] proposed an efficient probabilistic ANN-based approach to determine the Eurasian crustal thickness from surface wave velocities data. Following *Devilee et al.* [1999], some researchers [*Meier et al.*, 2007] have provided a similar ANN-based approach by a mixture density network (MDN), which actually combines the concept of both histogram and median type network, to estimate the global crustal thickness from the surface wave data. Solving the inverse problems requires precise estimation of uncertainties to know what it means to fit the data. *MacKay* [1992] introduced first a fully Bayesian approach where the scalar hyperparameters of a network are estimated via the so-called evidence program. While inverting the remote-sensing data, *Aires* [2004] has given a theoretical treatment for ANN uncertainty estimation using the Bayesian statistics. This made use of the fully Bayesian concept via the evidence program. However, this approach is not tested on the data contaminated with different levels of correlated/colored noise.

[7] In the present work, we employ a newly developed BNN probabilistic approach [*Bishop*, 1995; *Nabney*, 2004], for classification of changes in litho-facies units from the German Continental Deep Drilling Project (KTB) well-log data. We use multiple output node histogram networks, which return probabilistic information equidistantly on geophysical inverse problems by emulating the solution

Table 1. Showing Physical Properties, Recording Tool, and Vertical Resolution of Well Log Data Used in the Study

Physical Properties	Tool	Vertical Resolution	Unit	Principle
Bulk density	Litho density tool (LDT)	1 m	Grams per cubic centimeter	Absorption/scattering of gamma rays
Neutron porosity	Compensated neutron porosity (CNT)	1 m	% (limestone porosity unit)	Absorption of neutrons
Gamma ray intensity	Natural gamma ray spectrometer (NGS)	0.5 m	American Petroleum Institute (API)	Natural gamma ray emissions

from samples. The stability and effectiveness of the Bayesian neural network (BNN) approach on noisy as well as on noise-free data are also examined. The algorithm essentially allows us to estimate uncertainty in the data mixed with or devoid of correlated/colored noises. We compared our results of KTB well-log data with the existing results from other methods. Our results suggest that the BNN technique is superior to the other conventional ANN techniques in a sense that it takes care of the problems of uncertainty analysis, over-fitting and under-fitting in a natural way even if the data are contaminated with some percentage of noise. The comparison of regression results between the BNN and the super self adaptive back-propagation (SSABP) [*Maiti et al.*, 2007] and the result of uncertainty analysis along the entire length of the litho-section are quite encouraging. Thus, besides introducing a new ANN approach based on the Bayesian paradigm for modeling the international quality of well-log data, the present analysis has also brought out some new results thus, exploring the generality of the method from the point of view of the actual application in other domains.

2. About the KTB

[8] The German Continental Deep Drilling Project (KTB) explores a metamorphic complex in northeastern Bavaria, southern Germany [*Maiti et al.*, 2007, Figure 1]. Lithologically, the continental crust at the drill site consists of three main facies units: paragneisses, metabasites, and heterogeneous series having alternations of gneiss-amphibolites, with minor occurrence of marbles, calcsilicates, orthogeneisses, lamprophyres, and diorites [*Franke*, 1989; *Berckhemer et al.*, 1997; *Emmermann and Lauterjung*, 1997; *Pechnig et al.*, 1997; *Leonardi and Kumpel*, 1998, 1999]. The detailed information concerning the KTB data and its geophysical significance can be found in several earlier papers [*Franke*, 1989; *Berckhemer et al.*, 1997; *Pechnig et al.*, 1997; *Emmermann and Lauterjung*, 1997; *Leonardi and Kumpel*, 1998, 1999]. We used here three types of well-log data, density, neutron porosity, and gamma ray intensity for constraining the litho-facies boundaries for the KTB modeling study. Total depths of the main hole and pilot hole are 9101 m and 4000 m, respectively. The borehole data are sampled at 0.15 m (6 inch) intervals [*Maiti et al.*, 2007].

[9] The rocks were metamorphosed at a pressure of 6–8 kbar and a temperature of 650°C–700°C. This medium grade metamorphism took place in the Lower to middle Devonian (410–380 Ma ago) [*Leonardi and Kumpel*, 1998]. The crustal heterogeneities at the KTB borehole site are well documented [*Leonardi and Kumpel*, 1998]. These records are a complete, continuous, and uninterrupted series and hence could be appropriately utilized for the classification of the litho-facies units in a new perspective of Bayesian framework. A brief summary of the data and its geophysical/geological significance pertinent to this study is, however, presented here to preserve self-sufficiency of the paper.

2.1. Data

[10] Density data were measured using gamma rays emanating from a ^{137}Cs source that enters the wall rocks and are backscattered to a gamma detector (Litho Density Tool, LDT). Its vertical resolution is 1 m (Table 1). Neutron porosity values too are determined by using devices with radioactive sources. For the porosity logging, neutrons are emanated from an Am-Be radioactive source, and the rocks response, in the form of either gamma rays or fast neutrons or slow neutrons, is determined by an appropriate sensor (Neutron Compensated Log, NCL). For a constant borehole geometry. the response is a measure of the concentration of hydrogen atoms, which in the case of fluids-saturated rocks is related to the porosity of the geological formations. The NCL has a vertical resolution of 1 m (Table 1).

[11] The gamma ray radiation was measured using a Neutral Gamma Ray Spectrometer (NGS). This tool quantifies the natural gamma ray spectra of the isotopes ^{40}K, ^{232}Th, and ^{238}U. The bulk gamma ray intensity, deduced from the NGS data, is directly proportional to the concentration of the corresponding isotopes in a geological formation. The variations within a recorded log thus indicate changes in the lithology. High concentrations of K, Th, and U in crystalline successions reveal the presence of acidic rocks such as paragneisses, whereas the basic compositions are reflected by a scarcity of radio nuclides [*Leonardi and Kumpel*, 1999]. The gamma intensity is measured in the conventional American Petroleum Institute (API) unit. The vertical resolution is estimated to be 0.5 m (Table 1) [*Leonardi and Kumpel*, 1998].

2.2. Relationship Between the Log Response and Regional Geology

[12] Gamma ray activity exhibits a general increase from the most mafic rocks (ultramafites) to the most acidic rocks (potassium-feldspar gneisses) because of the chemical composition. In the KTB crystalline metamorphic basement, the total gamma ray is the most crucial physical parameter for differentiating the succession. In general, amphibolites and metagabbros, which are the main rock types of the massive metabasite units, are physically characterized by lower gamma ray activity and higher density than the rocks of the paragneisses sections. This is related to the mineral content, which within metabasites consists of more mafic and dense minerals like hornblende and garnet biotite. Paragneisses sections are composed mainly of quartz, plagioclase, and micas [*Berckhemer et al.*, 1997; *Emmermann and Lauterjung*, 1997; *Pechnig et al.*, 1997]. Since pore space in the crystalline basement is very low, neutron

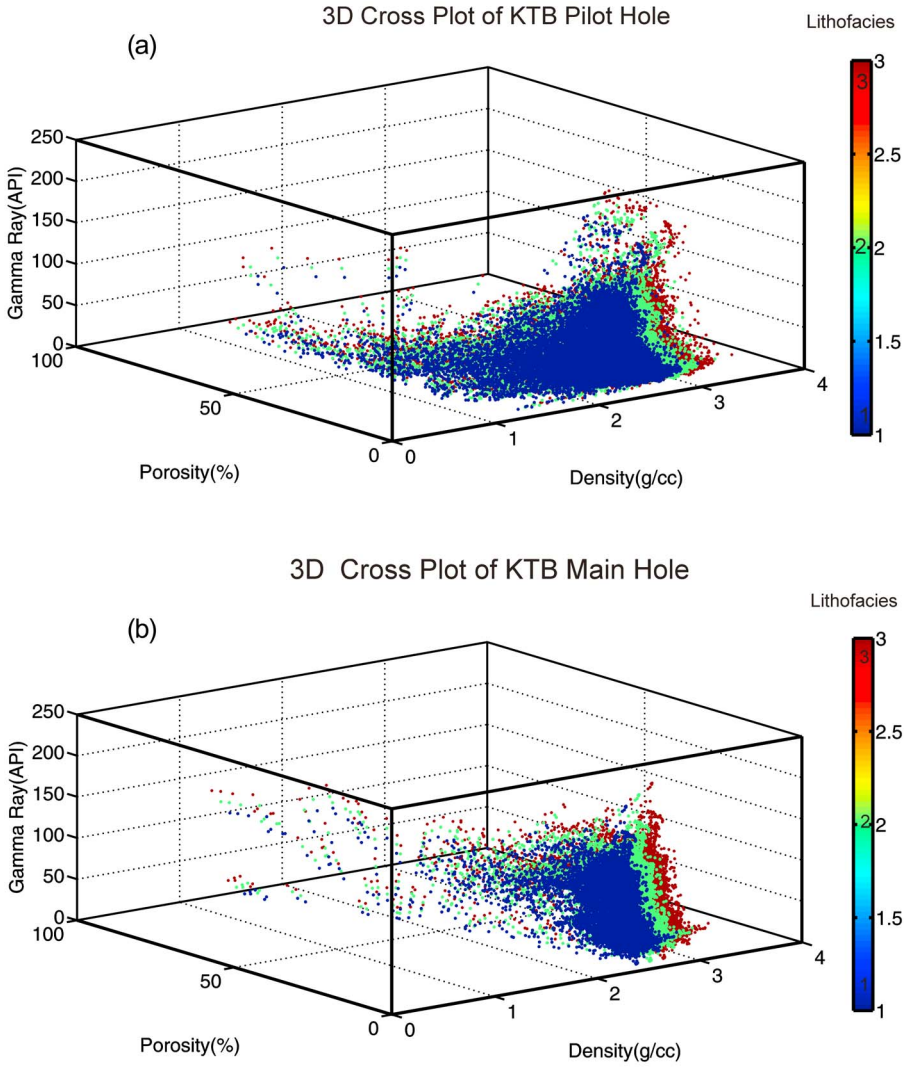


Figure 1. (a) Cross plot of density (g/cc), porosity (%), and gamma ray intensity (API) of KTB pilot hole showing strong nonlinearity and difficulty to establish parametric boundaries (b) same for KTB main hole.

porosity response is found to be dependent upon the mineralogical composition. Enhanced porosity is, in general, restricted to discrete zones of faulting and fracturing; however, neutron porosity in undisturbed depth sections is predominantly reacting to the water bound minerals like phyllosilicates or amphiboles [*Pechnig et al.*, 1997]. Hence, rock types poor in these minerals, such as quartz and feldspar-rich gneisses show very low neutron porosities. In contrast, rocks with high phyllosilicate and amphibole contents produce high values in the neutron porosity. We note that general log response knowledge is used for constructing network samples.

[13] The 3-D cross plot of density (g/cc), neutron porosity (%) and gamma ray (API) of the KTB main hole and pilot hole shows the strong overlapping/superposing well-log signal characteristics in 3-D parameter space (Figure 1). This overlapping signals could be characterized partly due to the physics (amalgamated rock structure limited by resolving power of data characteristics) and partly due to the noise (that may be of any kind, white Gaussian, red, pink, blue, etc.) in the well-log data. The complex nonlinear overlapping pattern of observed well-log data apparently evident in Figures 1a and 1b suggests that it may not be appropriate to apply linear inversion method to classify litho-facies units. It is therefore prudent to employ a nonlinear inversion scheme to obtain a general probability density function (pdf) in Bayesian framework to explore precisely the successions of litho-facies units.

3. Probabilistic Solution of Inverse Problem

[14] The solution to a general inverse problem can be given in the form of a *pdf*, P(x, d) [*Tarantola*, 1987; *Sen and Stoffa*, 1996; *Bosch*, 1999; *Sambridge and Mosegaard*, 2002], where d represents a set of distinct measurements and x is a set of model parameters. In the Bayesian framework, the solution can be given by

$$\mathrm{P}(x \mid d) = \frac{\mathrm{P}(d \mid x)\mathrm{P}(x)}{\mathrm{P}(d)} \tag{1}$$

Where, P() stands for probability, d represents a set of distinct measurements, and x is a set of model parameters. Thus, P($d|x$) represents a *pdf* of observed data given the model, (likelihood); P(d) is the *pdf* of the data d (scale factor in version represents limitations on the data space imposed by the physics and prior constraints on the model space), and P(x) is the *pdf* of the model parameter x, independent of the data (prior information on model). The solution of the inverse problem for a specific experiment may be approximated by the sampling based method according to P(x, d). But forming the solution requires too many forward calculations [*Devilee et al.*, 1999]. Instead, a neural network properly trained on a finite data set $\{d, x\}$ can emulate the conditional pdf P($x|d$).

4. Artificial Neural Networks

[15] An artificial neural network is an abstract model of the brain, consisting of simple processing units—similar to "neurons" in the human brain—connected layer by layer to form a network [*Rosenblatt*, 1958; *Meier et al.*, 2007]. A multi-layer perceptron (MLP) is a special configuration of an ANN [*Bishop*, 1995] that ranges among the most powerful methods for solving the nonlinear classification and boundaries detection problems. The MLP model consists of one input layer, one output layer, and at least one intermediate hidden layer between the input and output layer. In a fully connected MLP, a neuron (node) of each layer is connected to a neuron of the next layer through a synaptic weight. Output of the input layer is then fed to the input of the hidden layer and the output of the hidden layer is then fed to the input of the output layer (Figure 2a). The information propagates in one direction from the input layer to the output layer.

[16] The ANN works by adapting weights and biases in order to minimize error functions between the "network output" and the "target values" via a suitable algorithm. The popularly known back-propagation (BP) algorithm uses the scaled conjugate gradients (SCG) and quasi-Newton methods for optimization of synaptic weights and biases [*Rumelhart et al.*, 1986] (Figure 2a). In our work we use a nonlinear hyperbolic tan sigmoid transfer function of the form

$$f_j\left(net_j^{(l)}\right) = \frac{e^{\beta\left(net_j^{(l)}\right)} - e^{-\beta\left(net_j^{(l)}\right)}}{e^{\beta\left(net_j^{(l)}\right)} + e^{-\beta\left(net_j^{(l)}\right)}}. \tag{2}$$

Here, e denotes the basis of the natural logarithm (Figure 2a), and β is a constant that determines the stiffness of the function near

$$net_j^{(l)} = \sum_{i=1}^{n} w_{ji}^{(l)} d_i^{(l-1)} - \Theta_j^{(l)} = 0 \tag{3}$$

in layer (l) [*Roth and Tarantola*, 1994]. $net_j^{(l)}$ is a value received by the jth node in layer (l), $w_{ji}^{(l)}$ is a connection weight between the ith node in layer ($l-1$) and the jth node in layer (l), and d_i be an input is a variable for the ith node in layer ($l-1$). $\Theta_j^{(l)}$ is a bias unit for the jth node in layer (l). The value of β is adopted as 1.0 to keep the transfer function in sigmoidal shape [*Roth and Tarantola*, 1994]. Henceforth, it might be more convenient to put the first and second layer synaptic weight and bias term into a single network parameter w.

[17] The principle goal of the neural network approach is to learn the relationship between an input and an output in space/domain from a finite data set $s = \{d_k, x_k; k = 1, ..., N\}$ by adjusting network parameters w (weight and biases). This is done by maximizing the likelihood of the data set S (or equivalently by minimizing its negative logarithm), which forms a conventional least squares error measure in the form

$$E = \frac{1}{2}\sum_{k}^{N}\left\{x_k - o_k(d_k; w_k)^2\right\}, \tag{4}$$

where x_k and o_k are, respectively, the target/desired and the actual output at each node in the output layer. We note that input d consists of three types of well-log data (viz. density, neutron porosity, and gamma ray intensity) and output x consists of three types of litho-facies present over the KTB super deep borehole. We construct the histogram network introduced by *Devilee et al.* [1999] which emulates the conditional pdf P($x|d$) directly to the new input d. We will return to that in section 4.1.

4.1. Histogram Network

[18] *Devilee et al.* [1999] introduced a histogram-type network which provides a finite discritization of P(x, d). The k-output of a histogram network emulates equidistantly sampled approximation of the solution, i.e., *pdf* P($x|d$). Suppose, we consider the case of a scalar x and apply the following operator to discritize its value using k-segments with length Δx:

$$x_k(x) = \begin{cases} 1 & k\Delta x < x < (k+1)\Delta x \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

The k-output o_k (d) of the optimally trained network gives

$$o_k(d) = \int_{x0+k\Delta x}^{xo+(k+1)\Delta x} \mathrm{P}(x \mid d)dx \approx \mathrm{P}_k(d). \tag{6}$$

For each set of inputs d, the trained network has k-outputs o_k (d) which return the probabilities that x takes a value in the kth window of width Δx. With $k = 3$, we obtain our application of classification of input into one of the three states. We note that the solution approximately satisfies

$$\sum \mathrm{P}_k = 1. \tag{7}$$

4.2. Bayesian Neural Networks

[19] In a conventional neural network approach, often a regularization term is incorporated to solve equation (4) and optimize the objective function:

$$E(w) = \mu E_S + \lambda E_R. \tag{8}$$

Here λ and μ, which control other parameters (synaptic weight and biases), are known as hyperparameters. For $E_R = \frac{1}{2}\sum_{i=1}^{R} w_i^2$, R is the total number of weights and biases in

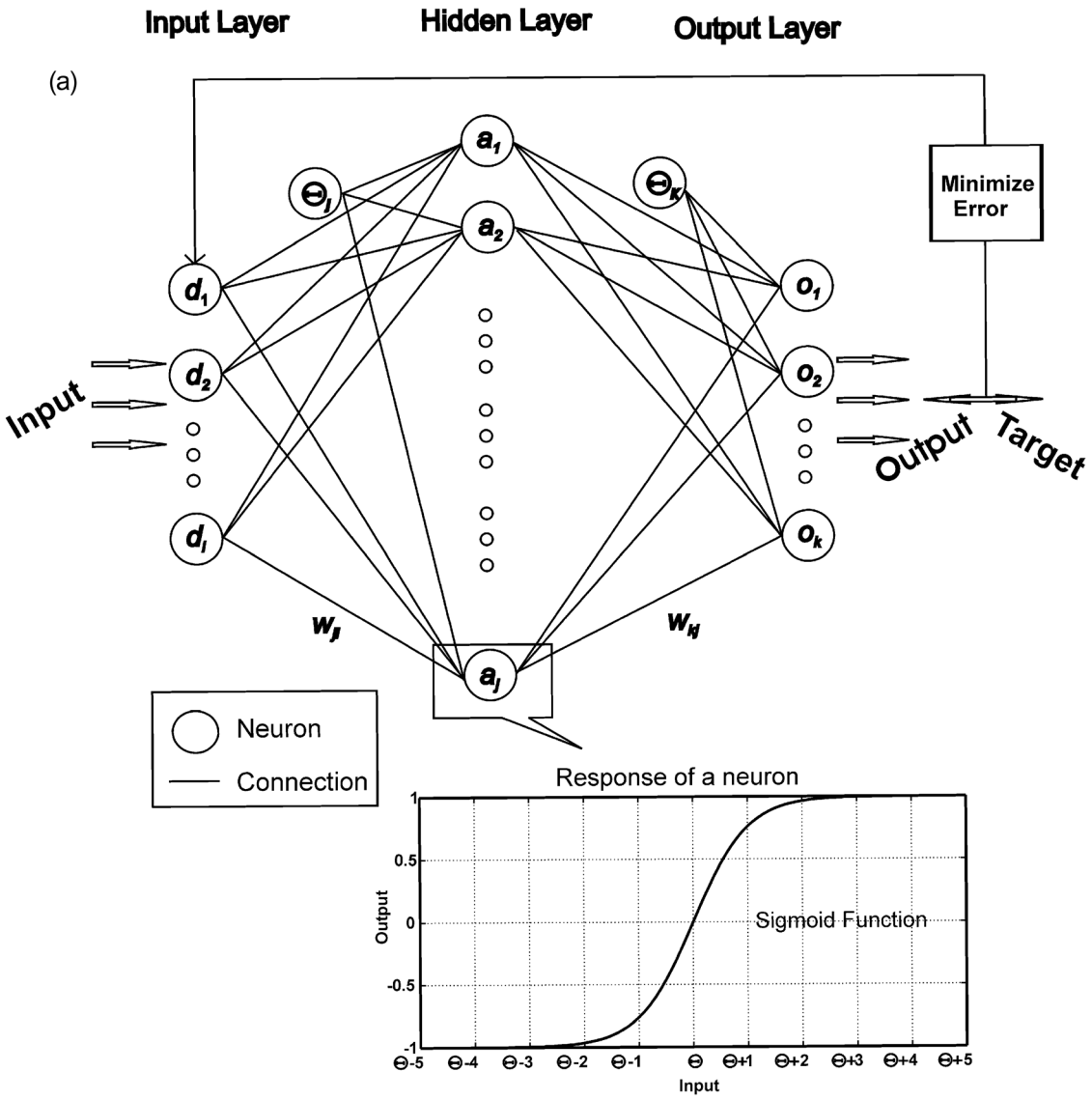


Figure 2. (a) Layout of the MLP with a three layer neural networks with d representing input and subscript i representing the number of "nodes" in the input layer. At each node of the hidden layer, the net arguments are squashing through a nonlinear activation function of type hyperbolic tangent where w_{ji} represents the connection weight between the ith node in the input layer and the jth node in the hidden layer. w_{jk}, represents the connection weight between the jth node in the hidden layer and the kth node in the output layer; Θ_j and Θ_k are bias vectors for the hidden and the output layer. Now $a_j = f_j(net_j)$ is the output of the weighted sum through the activation function following equation (2). When the sum of the argument of a neuron is comparable to the threshold value Θ_j, the sigmoid function squashes linearly; otherwise, it saturates with value +1; –1 gives nonlinearity for nonlinear mapping between an input and an output space. (b) A simple example of the application of BNN to a "regression" problem. Here 50 data points have been generated by sampling the function defined by equation (25); the network consists of a MLP with five hidden units having tanh activation functions and one linear output unit/node. The "network" shows the BNN results with the weight vector set to w_{MLP} corresponding to the maximum of the posterior distribution, and the "error bar" representing $\pm 1\delta$ from equation (45). Notice how the error bars are larger in regions of low data density. (c) Same when the standard deviation of noise distribution is 0.5. (d) Example when no regularization is used and the standard deviation of noise distribution is 0.1. (e) Example when the data are sampled at point 0.25 and 0.90 and the standard deviation of noise distribution is 0.1.

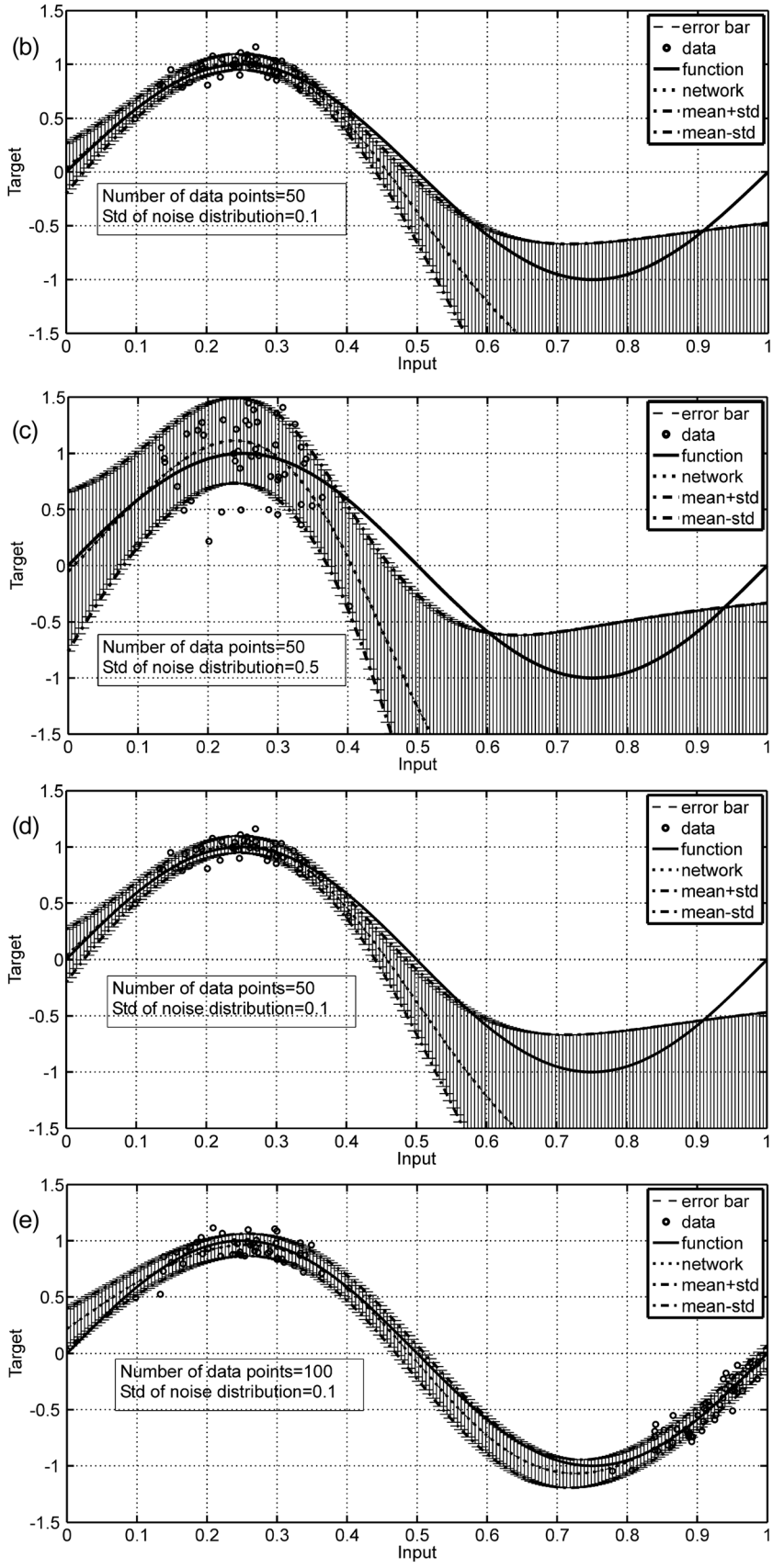


Figure 2. (continued)

the network. This corresponds to the use of simple weight-decay regularizer, which gives a prior distribution of the form [*Bishop*, 1995] of

$$\mathrm{P}(w) = \frac{1}{Q_R(\lambda)} \exp\left(-\frac{\lambda}{2} \| w \|^2\right). \tag{9}$$

Thus, when $\|w\|$ is large, E_R is large, and P(w) is small, and so this choice of prior distribution says that we expect the weight values to be small rather than large. Thus, the regularization term favors small values for network weight and biases and decreases the tendency of a model to "over-fit" noise in the training data. In the traditional approach, the training of a network begins by initializing a set of weights and biases and ends up with the single best set of weights and biases given the objective function is optimized.

[20] In the Bayesian approach, a suitable prior distribution, say P(w) of weights is considered before observing the data instead of a single set of weights. Using Bayes' rule, one can write *a posteriori* probability distribution for the weights, as [*Bishop*, 1995; *Aires*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(w \mid s) = \frac{\mathrm{P}(s \mid w)\mathrm{P}(w)}{\mathrm{P}(s)}, \tag{10}$$

where P($s \mid w$) is a data set likelihood function and the denominator P(s) is a normalization factor. Integrating out over the weight space, we obtain [*Bishop*, 1995; *Aires* 2004; *Nabney*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(s) = \int \mathrm{P}(s \mid w)\mathrm{P}(w) dw. \tag{11}$$

Equation (11) ensures that the left-hand side of equation (10) gives unity when integrated over all the weight space. Once the posterior has been calculated, all types of inference are obtained by integrating out over that distribution. Therefore, implementing the Bayesian method, expressions for the prior distributions P(w) and likelihood function P($s|w$) are needed. The prior distribution P(w) can be expressed in terms of a weight decay regularizer, E_R of the conventional learning method. For example, if a Gaussian prior is considered, we can write the distribution as an exponential of the form [*Bishop*, 1995; *Nabney*, 2004]

$$\mathrm{P}(w) = \frac{1}{Q_R(\lambda)} \exp(-\lambda E_R), \tag{12}$$

where Q_R (λ) is a normalization factor and can be given by [*Bishop*, 1995; *Nabney*, 2004]

$$Q_R(\lambda) = \int \exp(-\lambda E_R) dw. \tag{13}$$

Equation (12) ensures that $\int p(w) dw = 1$. The hyperparameter λ can be fixed or could be optimized as part of the training process [*Nabney*, 2004]. Note that in practice it is difficult to know the prior state of information about the network weight in advance. It is the Bayesian machinery employed to provide a neat, tractable, and sound mathematical framework to update the network parameter distribution. Here a Gaussian prior is chosen because it simplifies the calculation of the normalization coefficient Q_R (λ) using equation (13) giving [*Bishop*, 1995; *Nabney*, 2004]

$$Q_R(\lambda) = \left(\frac{2\pi}{\lambda}\right)^{R/2}. \tag{14}$$

[21] Alternative choices for the prior P(w) have been discussed in great length by *Buntine and Weigend* [1991], *Neal* [1993], and *Williams* [1995]. The data-dependent likelihood function in Bayes' theorem can be formed in terms of the error function, E_S of the conventional method. For instance, if the noise (error) model is Gaussian, an equation we write for likelihood function is [*Bishop*, 1995; *Aires*, 2004; *Nabney*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(s \mid w) = \frac{1}{Q_S(\mu)} \exp(-\mu E_S). \tag{15}$$

The function Q_S (μ) is a normalization factor given by

$$Q_S(\mu) = \int \exp(-\mu E_S) ds, \tag{16}$$

where $\int ds = \int dx_1 \ldots dx_N$ represent integration out over the target variables. If it is assumed that the target data are generated from a smooth function with additive zero mean Gaussian noise, the probability of observing the data value x for a given input vector d would be

$$\mathrm{P}(x \mid d, w) \propto \exp\left(-\frac{\mu}{2}\{x - o(d; w)\}^2\right), \tag{17}$$

where $o(d; w)$ represents a network function governing the mean of the distributions, w denotes the corresponding weight vector, and the parameter μ controls the variance of the noise. Provided the data points are drawn independently from this distribution, we have the expression for the likelihood as [*Bishop*, 1995; *Aires*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(s \mid w) = \prod_{k=1}^{N} \mathrm{P}(x_k \mid d_k, w) = \frac{1}{Q_S(\mu)} \exp\left(-\frac{\mu}{2} \sum_{k=1}^{N} \{x_k - o_k(d_k; w)\}^2\right). \tag{18}$$

The expression in equation (16) for the normalization factor Q_S (μ) is then the product of N independent Gaussian integrals, which have been evaluated by *Bishop* [1995]. Accordingly, we can write

$$Q_S(\mu) = \left(\frac{2\pi}{\mu}\right)^{N/2}. \tag{19}$$

After deriving the expressions for prior and likelihood functions and the posterior distribution of weights and using those expressions in equation (10), we obtain [*Bishop*, 1995; *Aires*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(w \mid s) = \frac{1}{Q_E} \exp(-\mu E_S - \lambda E_R) = \frac{1}{Q_E} \exp(-E(w)), \tag{20}$$

where

$$E(w) = \mu E_S + \lambda E_R = \frac{\mu}{2}\sum_{k=1}^{N}\{x_k - o_k(d_k; w)\}^2 + \frac{\lambda}{2}\sum_{i=1}^{R} w_i^2 \quad (21)$$

and

$$Q_E(\lambda, \mu) = \int \exp(-\mu E_S - \lambda E_R) dw. \quad (22)$$

In equation (20), the objective function in the Bayesian method corresponds to the inference from the posterior distributions of the network parameters w. After defining the posterior distributions, the network is trained with a suitable optimization algorithm to minimize the error function $E(w)$ or equivalently to maximize the posterior distribution P(w|s). Using the rules of conditional probability, the distribution of outputs for a given input vector, d, we can write in the form of [*Bishop*, 1995; *Aires*, 2004; *Khan and Coulibaly*, 2006]

$$\mathrm{P}(x \mid d, s) = \int \mathrm{P}(x \mid d, w)\mathrm{P}(w \mid s) dw. \quad (23)$$

We note that P(x | d, w) is simply the model for distribution of noise on the target data for a fixed value of weight vector w_{MLP} and can be expressed by equation (17), and P(w | s) is the posterior probability distribution of weight. If the data set is large, the posterior distribution P(w | s) may be approximated to Gaussian distribution [*Walker*, 1969]. After some simplification, we can write the integral of equation (23) as [*Bishop*, 1995]

$$\mathrm{P}(x \mid d, s) = \frac{1}{(2\pi\delta_i^2)^{1/2}} \exp\left(-\frac{\{(x - o(d; w_{MLP})\}^2}{2\delta_i^2}\right), \quad (24)$$

whose mean is $o(d; w_{\mathrm{MLP}})$, and variance is given by [*Bishop*, 1995]

$$\delta_i^2 = \frac{1}{\mu} + g^T H^{-1} g. \quad (25)$$

Here, μ is a hyperparameter and is actually the inverse variance of the noise model, and g denotes the gradient of $o(d; w)$ with respect to the weights w evaluated at w_{MLP} and H is the Hessian matrix of the total(regularized) error function with elements, which we can write [*Bishop*, 1995; *Aires*, 2004; *Khan and Coulibaly*, 2006]

$$H = \nabla\nabla E(w_{\mathrm{MLP}}) = \mu\nabla\nabla E_S(w_{\mathrm{MLP}}) + \lambda I, \quad (26)$$

where I is an identity matrix. The standard deviation δ_i of the predictive distribution for the target model x can be interpreted as error bar on the mean value $o(d; w_{\mathrm{MLP}})$.

4.3. Evidence Approximation

[22] The evidence approximation is the main important concept when the Gaussian approximation of Bayesian neural network is used. It is an iterative algorithm for determining optimal weights and hyperparameters. The evidence method has been discussed in detail by *MacKay* [1992] and is similar to the type II maximum likelihood method (MLM).

[23] In this approach, the posterior distribution of network weights can be written as [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\mathrm{P}(w \mid s) = \int\int \mathrm{P}(w, \lambda, \mu \mid s) d\lambda d\mu = \int\int \mathrm{P}(w \mid \lambda, \mu, \mid s)\mathrm{P}(\lambda, \mu \mid s) d\lambda d\mu. \quad (27)$$

The evidence procedure approximates the posterior density of the hyperparameters P(λ, μ|s) which is sharply peaked around λ_{MLP} and μ_{MLP}, the most probable values of the hyperparameters. This is known as Laplace approximation [*Nabney*, 2004]. Then we can write [*Bishop*, 1995]

$$\mathrm{P}(w \mid s) \approx \mathrm{P}(w \mid \lambda_{\mathrm{MLP}}, \mu_{MLP}, s) \int\int \mathrm{P}(\lambda, \mu \mid s) d\lambda d\mu \approx \mathrm{P}(w \mid \lambda_{\mathrm{MLP}}, \mu_{\mathrm{MLP}}, s). \quad (28)$$

This suggests that we need to evaluate the values of hyperparameters which maximize the posterior probability of weight and then perform the remaining calculations with the hyperparameter values set to these evaluated values [*Bishop*, 1995]. For the case of a nonlinear model like MLP, it is more complex to perform an integral of equation (28). This approximation should be viewed as a purely local one around a particular mode w_{MLP} based on a second-order Taylor series expansion of $E(w)$ [*Mackay*, 1992; *Nabney*, 2004]:

$$E(w) \approx E(w_{\mathrm{MLP}}) + \frac{1}{2}(w - w_{\mathrm{MLP}})^T H (w - w_{\mathrm{MLP}}). \quad (29)$$

Since the error function is a negative log probability of the weight posterior probability, it is clear that the weight posterior probability would be Gaussian. Thus, we can write [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\mathrm{P}(w|\lambda, \mu, s) = \frac{1}{Q_S^*} \exp\left(-E(w_{\mathrm{MLP}}) - \frac{1}{2}\Delta w^T H \Delta w\right), \quad (30)$$

where $\Delta w = w - w_{\mathrm{MLP}}$ and Q_s^* are the normalization constants for approximating Gaussian and can be therefore written as [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$Q_s^*(\lambda, \mu) = \exp\left(-E(w_{\mathrm{MLP}})(2\pi)^{R/2}(\det H)^{-1/2}\right). \quad (31)$$

In order to evaluate λ_{MLP} and μ_{MLP}, we can consider the modes of their posterior distribution: [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004],

$$P(\lambda, \mu \mid s) = \frac{P(s \mid \lambda, \mu)P(\lambda, \mu)}{P(s)} \quad (32)$$

The term P(λ, μ) denotes a prior over the hyperparameters, it is called a hyperprior. The denominator in equation (32) is independent of λ and μ; hence, the maximum posterior values for these hyperparameters could be obtained by maximizing the likelihood term P(s|λ, μ). This term is called

the evidence for λ and μ [*Bishop*, 1995]. We evaluate this term by integrating the data likelihood over all possible weights w [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]:

$$\mathrm{P}(s \mid \lambda, \mu) = \int \mathrm{P}(s \mid w, \lambda, \mu)\mathrm{P}(w \mid \lambda, \mu)dw \tag{33}$$

$$= \int \mathrm{P}(s \mid w, \mu)\mathrm{P}(w \mid \lambda)dw. \tag{34}$$

Here, we have used the fact that the prior is independent of μ and the likelihood function is independent of λ. Using equations (12) and (15), we obtain [*MacKay*, 1992; *Nabney*, 2004; *Bishop*, 1995]

$$\mathrm{P}(s \mid \lambda, \mu) = \frac{1}{Q_S(\mu)}\frac{1}{Q_R(\lambda)}\int \exp(-E(w)dw = \frac{Q_E(\lambda, \mu)}{Q_S(\mu)Q_R(\lambda)}. \tag{35}$$

We can write the log of evidence as [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\ln \mathrm{P}(s \mid \lambda, \mu) = -\lambda E_R^{\mathrm{MLP}} - \mu E_S^{\mathrm{MLP}} - \frac{1}{2}\ln|H| + \frac{R}{2}\ln\lambda + \frac{N}{2}\ln\mu - \frac{N}{2}\ln(2\pi). \tag{36}$$

Here we must consider the problem of finding the maximum with respect to λ. In order to differentiate $\ln|H|$ with respect to λ, we write $H = H_{\mathrm{UR}} + \lambda I$, where $H_{\mathrm{UR}} = \nabla\nabla E_S$ is the Hessian of the unregularized error function. Let $\psi_1, \ldots, \psi_R$ be the eigenvalues of the data Hessian H. Then H has eigenvalues $\psi_i + \lambda$, and we have [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\frac{d}{d\lambda}\ln|H| = \frac{d}{d\lambda}\ln\left(\prod_i^R(\psi_i + \lambda)\right) = \frac{d}{d\lambda}\sum_i^R \ln(\psi_i + \lambda) = \sum_{i=1}^R \frac{1}{\psi_i + \lambda} = \mathrm{tr}\left(H^{-1}\right) \tag{37}$$

Note that in this derivation we have implicitly assumed that the eigenvalues ψ_i do not themselves depend on λ [*Bishop*, 1995]. For nonlinear network models, the Hessian H is a function of w. Since the Hessian is evaluated at w_{MLP}, and since w_{MLP} depends on λ, we can find the result of equation (37) which actually neglects the term involving $\frac{d\psi_i}{d\lambda}$ [*MacKay*, 1992].

[24] With this approximation, the maximization of equation (36) with respect to λ is then straightforward with the result that, at maximum, we can now write [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$2\lambda E_R^{\mathrm{MLP}} = R - \sum_{i=1}^R \frac{\lambda}{\psi_i + \lambda} = \gamma, \tag{38}$$

where the quantity γ is defined by [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\gamma = \sum_{i=1}^R \frac{\psi_i}{\psi_i + \lambda}. \tag{39}$$

Since ψ_i is the eigenvalue of $H_{\mathrm{UR}} = \mu\nabla\nabla E_S$, it follows that ψ_i is directly proportional to μ and hence that [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004],

$$\frac{d\psi_i}{d\mu} = \frac{\psi_i}{\mu}. \tag{40}$$

Thus, we have [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\frac{d}{d\mu}\ln|H| = \frac{d}{d\mu}\sum_i^R \ln(\psi_i + \lambda) = \frac{1}{\mu}\sum_i^R \frac{\psi_i}{\psi_i + \lambda}. \tag{41}$$

This leads to the following condition satisfied at the maximum of equation (36) with respect to μ [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]:

$$2\mu E_S^{\mathrm{MLP}} = N - \sum_{i=1}^R \frac{\psi_i}{\psi i + \lambda} = N - \gamma. \tag{42}$$

In a practical implementation of this approach, one should begin by finding the optimum value of w_{MLP} using a standard iterative optimization algorithm, while periodically the values of λ and μ are restricted as [*MacKay*, 1992; *Bishop*, 1995; *Nabney*, 2004]

$$\lambda_{\mathrm{new}} = \frac{\gamma}{2E_R}, \tag{43}$$

$$\mu_{\mathrm{new}} = \frac{N - \gamma}{2E_S}. \tag{44}$$

4.4. Synthetic Example

[25] We use the recently developed scaled conjugate gradient algorithm [*Moller*, 1993] for the optimization process, which avoids the expensive line-search procedure of conventional conjugate gradients. Experiments have shown this is an extremely efficient algorithm outperforming both the conjugate gradients and quasi-Newton algorithms when the cost function and gradient evaluation is relatively small [*Nabney*, 2004].

[26] We conduct a practical numerical experiment following *Nabney* [2004]. For example, to illustrate the application of the Bayesian techniques to a "regression" problem, we consider a one-input one–output example involving data generated from the smooth function

$$f(x) = 0.25 + 0.07\sin(2\pi x), \tag{45}$$

with additive Gaussian noise having a standard deviation of $\delta = 0.1$. The values for x were generated by sampling an isotropic Gaussian mixture distribution. Figures 2b–2e show the trained BNN approximation corresponding to the mean of the predictive distributions. The error bar represents one standard deviation ($\pm 1\delta$) of the predictive distribution. This predictive distribution allows us to provide error bars for the network output instead of just a single answer. It may be noted that the size of the error bar varies approximately with the inverse data density [*Williams*, 1995]. A prior of the form of $\mathrm{P}(w)$ (equation (9)) and weight-decay regularization were used, and the values of μ and λ were chosen by an online evidence re-estimation schedule. In this first set of experiments we have used that (1) the number of nodes used

Table 2. A Priori Information on Model Parameter to Generate Forward Model for Neural Network Training Indicating that Gamma Ray Intensity Value is the Most Crucial Factor to Categorize Litho-Facies Unit in Metamorphic Area

Litho-Facies Unit	Density (g/cc)	Neutron Porosity (%)	Gamma Ray Intensity (API)	Desired Output/Binary Code
Paragneisses	2.65–2.85	5–15	70–130	100
Metabasites	2.75–3.1	5–20	0–50	010
Heterogeneous series	2.60–2.9	1–15	40–90 & 120–190	001

in a single hidden layer is 5, (2) the number of training samples used in the experiments is 50, (3) the initial prior hyperparameters values are $\lambda = 0.01$ and $\mu = 50.0$, (4) the tolerance for the weight optimization is set to a very low value (10^{-7}). This is because the Gaussian approximation to the weight posterior depends on being at a minimum of the error function [*Nabney*, 2004]. The converged values for the hyperparameters are $\mu = 80.512$ and $\lambda = 0.176$. In this case, the true value of μ is 100, so that the procedure is slightly overestimating the noise variance. Notice that the true function still lies within the error bar limit in the region where no training sampled data is available (>0.5) (Figures 2b–2d). We see that the error bar in this region is entirely due to regularization, whereas no regularization is used to smooth the function in practice because we obtain the identical results without the use of real prior information ($\lambda \approx 0.0$) on the smoothness of the function (Figure 2d). After several experiments with varying initial sets of hyperparameters, it is concluded that with a sufficient number of training samples $N > 5$, the true function lies within the error bound while we keep the ratio $\frac{\lambda}{\mu} < 0.015$. When λ and μ are kept fixed, as N increases, the first term of equation (21) becomes more and more dominant, until the second term becomes insignificant. The maximum likelihood solution is then a very good approximation to the most probable solution w_{MLP}. But, for very small data sets, the prior term plays an important role in determining the location of the most probable solution. An effective value of the regularization parameter (the coefficient of the regularizing term) depends only on the ratio $\frac{\lambda}{\mu}$, since an overall multiplicative factor is found to be unimportant [*Bishop*, 1995]. The second experiments demonstrate that if the underlying function is sampled more densely over the entire range (>0.5), the chances of the true function to lie beyond the error bound is less (Figure 2e). Note that for all experiments of the second phase we set the initial hyperparameters as $\lambda = 0.01$, $\mu = 50.0$, and the standard deviation of additive Gaussian noise as 0.1 ($\delta = 0.1$). Our experiment clearly shows that the BNN estimates the true function well where the data samples are more condensed and less noisy and the uncertainty is more where the true function is poorly sampled. Given that there are no training samples, the error bars in that interval are entirely due to the posterior distribution of network weights. In the next section we will discuss the modeling strategy and initialization of all model parameters.

5. Model Initiation and Implementation

5.1. Hidden Layers, Connection Weights, and Output

[27] The network has three nodes in the input layer. Each node takes the well-log data of density (g/cc), neutron porosity (%), and gamma ray intensity (API). In Bayesian neural network modeling, we do not necessarily need to estimate the optimal number of the weights (hidden layer) to have a good generalization [*Bishop*, 1995]. However, in the present classification problem, we find by trial and error that a single hidden layer with twenty individual nodes is appropriate. The output layer of network consists of three nodes which return a *posterior pdf* corresponding to the three types of litho-facies units viz. paragneisses, metabasites, and heterogeneous series.

5.2. Number of Training Samples

[28] The published results of core sample analysis from the KTB site [*Pechnig et al.*, 1997] are shown in Table 2. We consider a total of 702 representative input/output pairs within the bounds defined in Table 2 for the BNN training. The purpose for considering a limited number of training set was to maintain comparative status with the published histogram model. In order to get desired accuracy of $(1 - \varepsilon)$ of a MLP network on unseen data, at least (N/ε) examples must be provided [*Van der Bann and Jutten*, 2000], where ε is an error and N represents the network internal variables (weights and biases) which can be found as

$$N \approx [(N_i + 1)N_h + (N_h + 1)]N_o. \tag{46}$$

Here, N_i (Input) = N_o (Output) = 3; N_h (hidden layer) = 20 (Figure 2a). According to this relation, the present network has $N \approx [(3 + 1) \times 20 + (20 + 1) \times 3)] \approx 143$ internal variables which is less than the number of training samples (351) available. This ensures that the present network would provide at least 70% accuracy in prediction [*Van der Bann and Jutten*, 2000].

5.3. Input Data Scaling and Model Parameterization

[29] We scaled all the input/output pair values between 0 and 1 (−1 and +1) by using a simple linear transformation algorithm, [*Poulton*, 2001]; normalized input = 2 × (input-minimum input)/(maximum input-minimum input) − 1. We initialize the model by computing the probability distribution functions of the model parameters. In view of computational simplicity [*Bishop*, 1995] and to avoid large curvature [*Nabney*, 2004], the initial values of model parameters (synaptic weight and biases) are formed by following a Gaussian prior distribution function of a zero mean and an inverse variance, λ (also known as regularization coefficient or prior hyperparameter). The implementation of zero mean Gaussian prior can be done in two ways: one way is to consider a single hyperparameter λ for all the weights and alternatively to consider a separate hyperparameter for different groups of weights [*Nabney*, 2004]. We used a single hyperparameter λ for all the weights in a network following the equation (9). This is useful because a broad prior weight distribution is usually appropriate for this case as we use a

Table 3. Showing Estimated Network Training Hyperparameters μ and λ Via "Evidence Program" to Enable Efficient Learning of the Present Problem[a]

MLP Structure with Different Hidden Node	Epoch (0–250)	Epoch (250–500)	Epoch (500–750)	Epoch (750–1000)
3-5-3	$\mu = 10.20$	$\mu = 10.58$	$\mu = 10.69$	$\mu = 10.80$
	$\lambda = 0.06$	$\lambda = 0.04$	$\lambda = 0.03$	$\lambda = 0.03$
3-10-3	$\mu = 10.05$	$\mu = 10.45$	$\mu = 10.65$	$\mu = 10.71$
	$\lambda = 0.11$	$\lambda = 0.10$	$\lambda = 0.09$	$\lambda = 0.09$
3-20-3	$\mu = 8.98$	$\mu = 9.57$	$\mu = 9.84$	$\mu = 10.14$
	$\lambda = 0.25$	$\lambda = 0.22$	$\lambda = 0.21$	$\lambda = 0.20$
3-30-3	$\mu = 8.99$	$\mu = 9.66$	$\mu = 9.86$	$\mu = 9.9$
	$\lambda = 0.33$	$\lambda = 0.29$	$\lambda = 0.26$	$\lambda = 0.26$
3-40-3	$\mu = 8.49$	$\mu = 8.99$	$\mu = 9.24$	$\mu = 9.40$
	$\lambda = 0.45$	$\lambda = 0.45$	$\lambda = 0.40$	$\lambda = 0.36$

[a]The parameters μ and λ which control other parameters (weight and biases) of MLP network are known as hyperparameters. The re-estimation of the hyperparameters is carried out four times.

local nonlinear optimization algorithm for online re-estimating hyperparameter values. In order to define an objective function in the Bayesian framework, an error model for the data likelihood is required. Having assumed that the target data is formed from a smooth function with additive zero mean Gaussian noise, we estimated the hyperparameters μ is for both hidden and output layer weights.

[30] After defining the prior and the likelihood functions, the objective function has been estimated by the posterior distribution of weights. The maximum a posteriori (MAP) model is derived from an iterative optimization process that maximizes a posteriori probability distribution or equivalently minimizes the objective/cost functions. In Bayesian neural networks, the objective function/cost function is optimized by the SCG with evidence re-estimation of hyperparameters scheme. The evidence approximations could be performed in two steps: (1) computing the w_{MLP} in maximizing penalized likelihood and (2) periodically re-estimating the hyperparametersλ_{MLP}. In our experiments, the first step has been attained by optimizing the penalized likelihood with the SCG algorithm for 250 iterations. The re-estimation of the hyperparameters λ_{MLP} was carried out four times, and the Hessian matrix was calculated with the re-estimated hyperparameters (Table 3).

5.4. Data Division for Model Validation and Testing

[31] Over-fitting is one of the serious drawbacks of ANN modeling. In that, the selected training set is memorized in such a way that performance of the network is excellent only on this set but not on other data. In order to circumvent this problem, several researchers have recommended cross validations and early stopping skills [*Van der Bann and Jutten*, 2000; *Maiti et al.*, 2007]. Attaining good "generalization" of a model is, however, somewhat difficult where the data are complex, nonlinear and beset with deceptive noise. The Bayesian learning approach control "effective complexity" by considering many adjustable parameters and the parameter uncertainty is considered in form of probability distribution [*Bishop*, 1995; *Nabney*, 2004]. For the BNN learning, we keep the validation and the test sets separately: one is for noise analysis and another is for improving the training set in case the training set is not appropriately representative of all types of "model" information. But the validation error is not explicitly monitored during training [*Bishop*, 1995]. For noise analysis, the total database is shuffled appropriately and partitioned into three random subsets [*Maiti et al.*, 2007]: the training, the validation and the test set. The first 50% of the total data set is used for training. The remaining 50% is used for examining the "generalization" capability of the trained network. Here, again 24.93% (i.e., 175) of the data is kept for validation and the remaining 25.07% (i.e. 176) is for testing (Figures 3a–3h).

5.5. Generalization Capacity of the Network

[32] Generalization capacity of the trained network can be evaluated by error analysis on the validation and the test data sets. Error deviation is simply calculated by taking the difference between the target (binary output target, 100 for paragneisses, 010 for metabasites, and 001 for hetero-series) and the predicted network values at the output layer. We can denote error deviation as

$$e_i = d_i - o_i, \tag{47}$$

where d_i is the target and o_i is the network output. Figures 3a–3h show error-deviation plots of the validation and the test data sets for the paragneisses, metabasites and heterogeneous series. Table 4 shows the overall accuracy of the network prediction corresponding to the three types of lithofacies units. Evidently the accuracy of network prediction on the validation data set is comparatively better than the test data sets (Figure 3). Particularly the error prediction for the metabasites units is comparatively less than the paragneisses and the heterogeneous units (Figure 3). This could be explained by the fact that the heterogeneous series unit is composed of some components as a result of the alteration between the paragneisses and the metabasites unit.

6. Network Sensitivity to Correlated Noise

[33] In many geological/geophysical situations, we invariably observe some kind of deceptive/correlated noise which dominates the field observations and corrupts the signal. In the present case, we do not have any precise idea about the "percentage" of noise present in the actual well-

Figure 3. Error deviation and error bar map of validation and test data pertaining to paragneisses, metabasites, and heterogeneous series. (a)–(d) When the input generalization set is corrupted with 30% red noise. (e)–(h) When the input generalization set is corrupted with 50% red noise. Error bar defines 90% confidence limit.

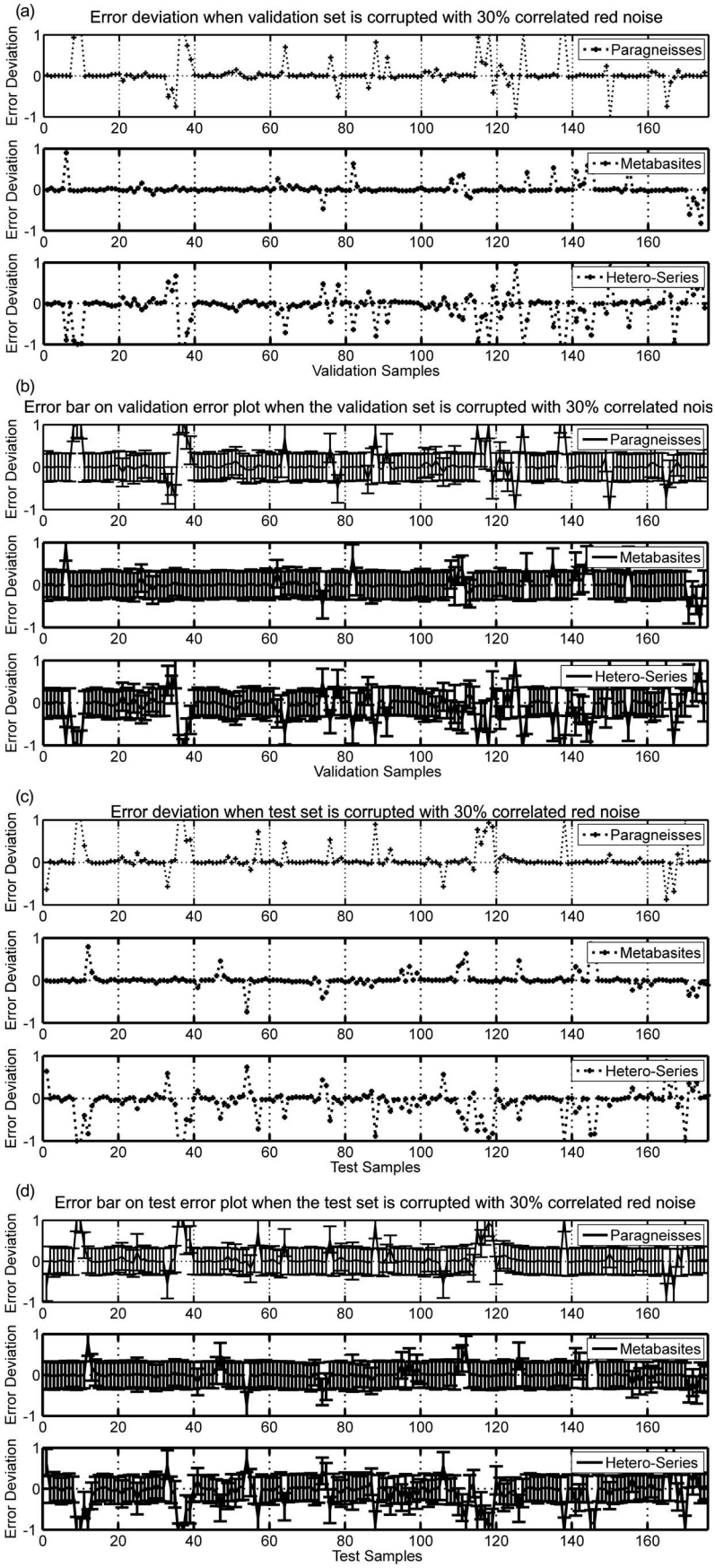

(a)
Error deviation when validation set is corrupted with 30% correlated red noise
Error Deviation
Paragneisses
Metabasites
Hetero-Series
Validation Samples
(b)
Error bar on validation error plot when the validation set is corrupted with 30% correlated nois
Error Deviation
Paragneisses
Metabasites
Hetero-Series
Validation Samples
(c)
Error deviation when test set is corrupted with 30% correlated red noise
Error Deviation
Paragneisses
Metabasites
Hetero-Series
Test Samples
(d)
Error bar on test error plot when the test set is corrupted with 30% correlated red noise
Error Deviation
Paragneisses
Metabasites
Hetero-Series
Test Samples

Figure 3

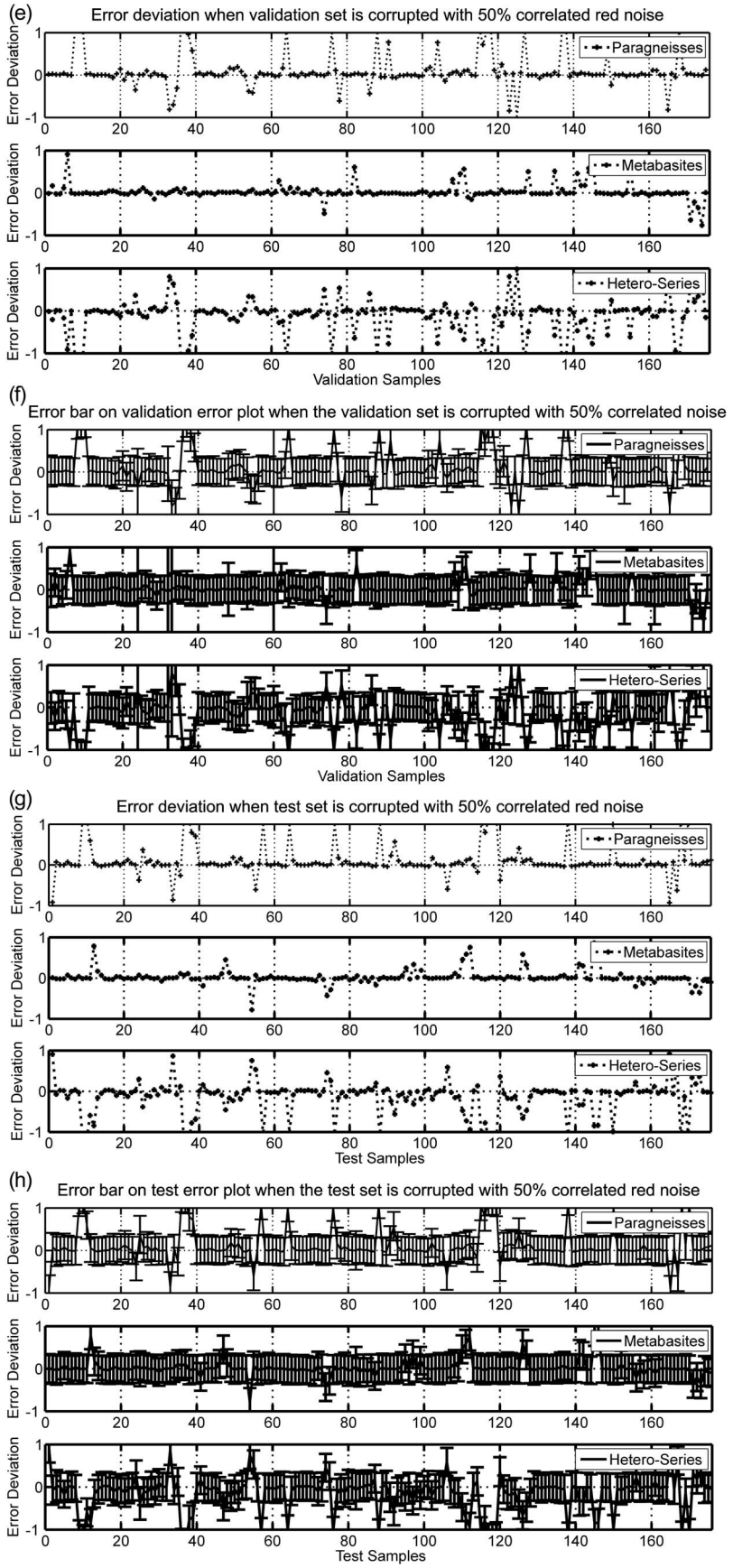


Figure 3. (continued)

Table 4. Showing Percentage of Accuracy While Validation and Test Data Sets are Corrupted with Different Level of Red Noise

	Percentage of Accuracy in Generalization Data Set						
	Validation Data Set			Test Data Set			
Red Noise Level	Paragneisses	Metabasites	Heterogeneous Series	Paragneisses	Metabasites	Heterogeneous Series	Average Stability ± 5% Error Limits
0%	89.14%	85.14%	74.29%	89.20%	82.95%	72.16%	82.14%
10%	86.16%	86.29%	73.14%	87.50%	82.95%	70.45%	81.08%
20%	84.00%	85.71%	68.57%	84.66%	82.92%	67.31%	78.86%
30%	81.14%	85.71%	66.29%	82.39%	81.82%	64.77%	77.02%
40%	80.57%	85.71%	65.71%	80.68%	80.11%	61.36%	75.69%
50%	78.86%	84.57%	62.86%	78.98%	79.55%	59.09%	73.98%

log data. Assuming that there is some possibility of inescapable noise in the data, it would be prudent to test the robustness and the stability of the results. For this, we generated correlated noise using the first-order autoregressive model [*Fuller*, 1976],

$$d(n) = Ad(n-1) + \varepsilon_n. \tag{48}$$

Here, $d(n)$ is a stationary space series at space n and ε_n is a Gaussian white noise with zero mean and unit variance. The constant A represents the maximum likelihood estimator (MLE) and can be computed from the data as [*Fuller*, 1976]

$$A(n) = \frac{\sum[(d_{n-1} - d_m)(d_n - d_m)]}{\sqrt{\left[\sum (d_{n-1} - d_m)^2 \sum (d_n - d_m)^2\right]}}, \tag{49}$$

where $d(n)$ is a data value at the point n, and d_m is the mean value. Using equation (49), the MLE constant $A(n)$ is estimated as 0.52 for the density data series, 0.21 for the porosity data series, and 0.53 for the gamma ray data series from the validation data sets and 0.47 for the density series, 0.23 for the porosity series, and 0.55 for the gamma ray series from the test data sets. We generated the Gaussian noise ε_n using a MATLAB library function. The correlated noisy time series (equation (48)) for each well-log data set is, then, normalized between [−1 and +1] using

$$\varepsilon_c^{\text{nor}}(n) = \frac{2 \times \{\varepsilon_c(n) - \varepsilon_{c\min}\}}{(\varepsilon_{c\max} - \varepsilon_{c\min})} - 1, \tag{50}$$

where $\varepsilon_c^{\text{nor}}(n)$ is the normalized correlated noise and $\varepsilon_c(n)$ is the unnormalized correlated noise; $\varepsilon_{c\max}$ and $\varepsilon_{c\min}$ are represented as the maximum and the minimum value of $\varepsilon_c(n)$, respectively. The normalized correlated noise for a data series is, then, taken to the level of the well-log data using

$$d\varepsilon_c(n) = d(n) \times \varepsilon_c^{\text{nor}}, \tag{51}$$

where $d(n)$ is the synthetic data series (we assume synthetic training data is noise free). Now, the correlated noise $d\varepsilon_c(n)$ from equation (51) is added to the $d(n)$ according to the equation

$$d_u(n) = d(n) + \left(\frac{u}{100}\right) \times d\varepsilon_c(n). \tag{52}$$

Here $d_u(n)$ is the data corrupted with a certain "percentage" of correlated noise in the above equation and $u = 1, 2, 3, \ldots, 100$. We prepared the individual data sets corrupted with different level of correlated red noise. The results of stability in presence of correlated noise are presented in Table 4. The accuracy of network prediction is examined using the validation and the test data sets when both the data sets are contaminated with different levels (10%–50%) of deceptive red noise. The error-deviation plot for well-log data corrupted by 30% and 50% correlated noise is presented in Figures 3a–3h. Our analyses suggest that the predictive efficiency of the BNN is considerably robust, even if the input well-log data is contaminated with "red" noise up to 40% or so; however, the predictive capability is degenerated beyond 50% noise contamination.

7. Uncertainty Analysis

[34] The uncertainty at the network output, (covariance matrix $\text{Cov}_o = \text{Cov}_\mu + g^T H^{-1} g$) is due to the intrinsic noise in the data embodied in μ and the theoretical error described by the posterior distribution of the weight vector w embodied in $g^T H^{-1} g$ [*Aires*, 2004]. For the forward linear operator, the posterior *pdf* will be Gaussian, and in that case, one can assume that all uncertainties are also Gaussian. However, for the nonlinear neural network, even if the *pdf* of the neural network weight is Gaussian, the *pdf* of the output can be non-Gaussian [*Aires*, 2004]. The derivative of forward function is evaluated at w_{MLP}. Here, the mean standard deviation (STD) (= $\sqrt{\text{diag}(\text{Cov}_o)}$) is estimated by taking the square root of the diagonal terms in Cov_o. The elements along the main diagonal of output covariance matrix shows the "variances" of the fluctuations about the mean of the Gaussian probability densities that characterizes the uncertainties, and the off-diagonal elements show the extent to which these fluctuations are correlated [*Tarantola*, 1987]. Figure 3 shows error bars plotted on error deviation curve for the input data which are corrupted with different level of correlated deceptive noise. All error bars are ± unit standard-deviations estimated from the a posteriori covariance matrix. Figures 4 and 5 present the mean standard deviation or three outputs: paragneisses, metabasites, and heterogeneous series. The ± unit standard deviations/error bar shows the 90% confidence interval (CI) [*Nabney*, 2004]. The minimum, maximum, and average values of the standard deviation of output error over the entire length of litho-section are documented in sections 8–10.

8. Examples

[35] We used three sets of borehole data viz. density, neutron porosity, and gamma ray obtained from the German

Continental Deep Drilling Program site. These data are applied to the trained BNN network to classify the litho-facies succession, and the BNN-based results are compared with the results of the published geological section [*Emmermann and Lauterjung*, 1997], and our earlier results based on super self adaptive back-propagation neural networks [*Maiti et al.*, 2007]. *Maiti et al.* [2007] have developed a classification scheme based on the very fast SSABP neural network theory where the learning rate is variable and adaptive to the complexity of the error surface. In that approach the solution obtained is based on maximum likelihood method, where a single "best" set of weight values is evaluated by minimization of a suitable error function. The BNN approach considers a probability distribution function over weight space instead of a single "best" set of weights. Thus, it is quite sensible to compare the BNN results with the results based on SSABP. The comparative results obtained by both neural network techniques for the pilot and the main borehole data are displayed in Figures 4 and 5, respectively. The outputs of the network represent the posterior probability distribution. Further, the standard deviation error maps, corresponding to the three types of litho-facies are presented to quantify the prediction uncertainties of the network output over the entire KTB litho-section. All the comparative results shown in Figures 4 and 5 are presented in a three-column gray-shaded matrix with black representing 1 and white representing 0. The interpretation of the maximum a posteriori geological section (MAPGS) is as follows: if the litho-facies of a particular class exists, the output value of the node in the last layer is 1 or very close to 1, and if not, it is 0 or very close to 0.

8.1. Comparisons of the BNN Results With the Published Results

[36] The published results of litho-facies successions by *Emmermann and Lauterjung* [1997] were re-drawn for the sake of clarity [*Maiti et al.*, 2007, Figure 2]. The MAPGS derived from the BNN modeling via the SCG optimization for both the KTB boreholes are displayed at 500 m data windows for critical and thorough examination and compared with published subsections ("Subsection of KTB-VB/HB" in Figures 4 and 5). A careful visual inspection of these figures suggests that the BNN results correlate fairly well with the published results over the entire litho-section.

8.2. Pilot Borehole (KTB-VB) (up to 4000 m Depth)

[37] The results based on the BNN modeling confirm, in general, the presence of paragneisses, metabasites, and heterogeneous series within the first 500 m depths. However, a close examination and comparison of the BNN results with the published results reveal some dissimilarity too (Figure 4a). For instance, in the depth range of 30–100 m, the BNN model indicates the presence of heterogeneous series, whilst the published subsection shows the paragneisses unit. Likewise at the depth range of 240–250 m, the BNN results indicate the presence of paragneisses, while the published subsection shows the heterogeneous series. If we go further down, at the depth range of 305–340 m, the BNN results show the presence of paragneisses, heterogeneous series and metabasites instead of the heterogeneous series alone. There is also some mismatch at 400–430 m depth range where the BNN shows the metabasites, instead of the heterogeneous series, and at the 430–490 m depth range, the BNN indicates the paragneisses instead of the heterogeneous series.

[38] There is a positive correlation between the BNN model results and the published results (Figures 4b and 4c) at the 500–1500 m depth range which shows good conformity with the three litho-facies successions (e.g., paragneisses, metabasites and heterogeneous series). However there are some divergences too. The BNN results suggest the presence of the paragneisses instead of the heterogeneous series at the depth intervals 520–556 m, the heterogeneous series instead of the paragneisses at the depth interval of 579–582 m, the heterogeneous series instead of the paragneisses at the depth interval of 598–601 m, the heterogeneous series instead of the paragneisses at the depth interval of 707–712 m, and the metabasites instead of the paragneisses at the depth interval of 1145–1168 m. The present results also show good match within the depth range of 1500–2500 m (Figures 4d and 4e) with a few exceptions, for example, the presence of heterogeneous series at the depth range of 1597–1618 m and the paragneisses at the depth range of 1744–1817 m and 2442–2467 m. The BNN results show the dominance of the paragneisses at the depth intervals of 2500–3000 m, 2562–2623 m, and 2852–2953 m in addition to an inter-bedded thin metabasites structures at the depth range of 2640–2646 m (Figure 4f). Figure 4g exhibits the presence of the paragneisses and the heterogeneous series that are consistent with the published results except for the depth range of 3408–3421 m. In addition to this, the BNN results also suggest the presence of the paragneisses and the heterogeneous series at the depths intervals of 3204–3248 m and 3409–3418 m, respectively. Comparison of the present results at the depth range of 3500–4000 m shows a depositional sequence of the paragneisses and the metabasites that exactly match with the published results (Figure 4h). The minor deviations and some differences observed between the published and the present result is explained in the discussion section. The average standard deviation estimated at the network output corresponding to the paragneisses, metabasites and heterogeneous series is ±0.30 for the entire KTB pilot hole (down to 4000 m), except for depths 58.52 m, 381.60 m, 1186.75 m, 1238.09 m, and 1286.86 m, where the STD is ±0.58, ±0.78, ±0.99 ±0.54 and ±0.79, respectively.

Figure 4. (a) Comparison of the maximum a posteriori geological section (MAPGS) obtained by BNN with the SCG approach with a maximum likelihood geological section (MLGS) obtained by the SSABP neural network, and the published litho-facies subsection of pilot hole (KTB-VB) (published litho-subsection is redrawn after *Emmermann and Lauterjung* [1997]) and the standard deviation (std) error map estimated at the network output by BNN approach at the depth interval of 0–500 m. In this interval 0–28 m, data are not available. (b)–(h) Same for the depth range of 500–1000 m, …, 3500–4000 m in KTB pilot hole(KTB-VB).

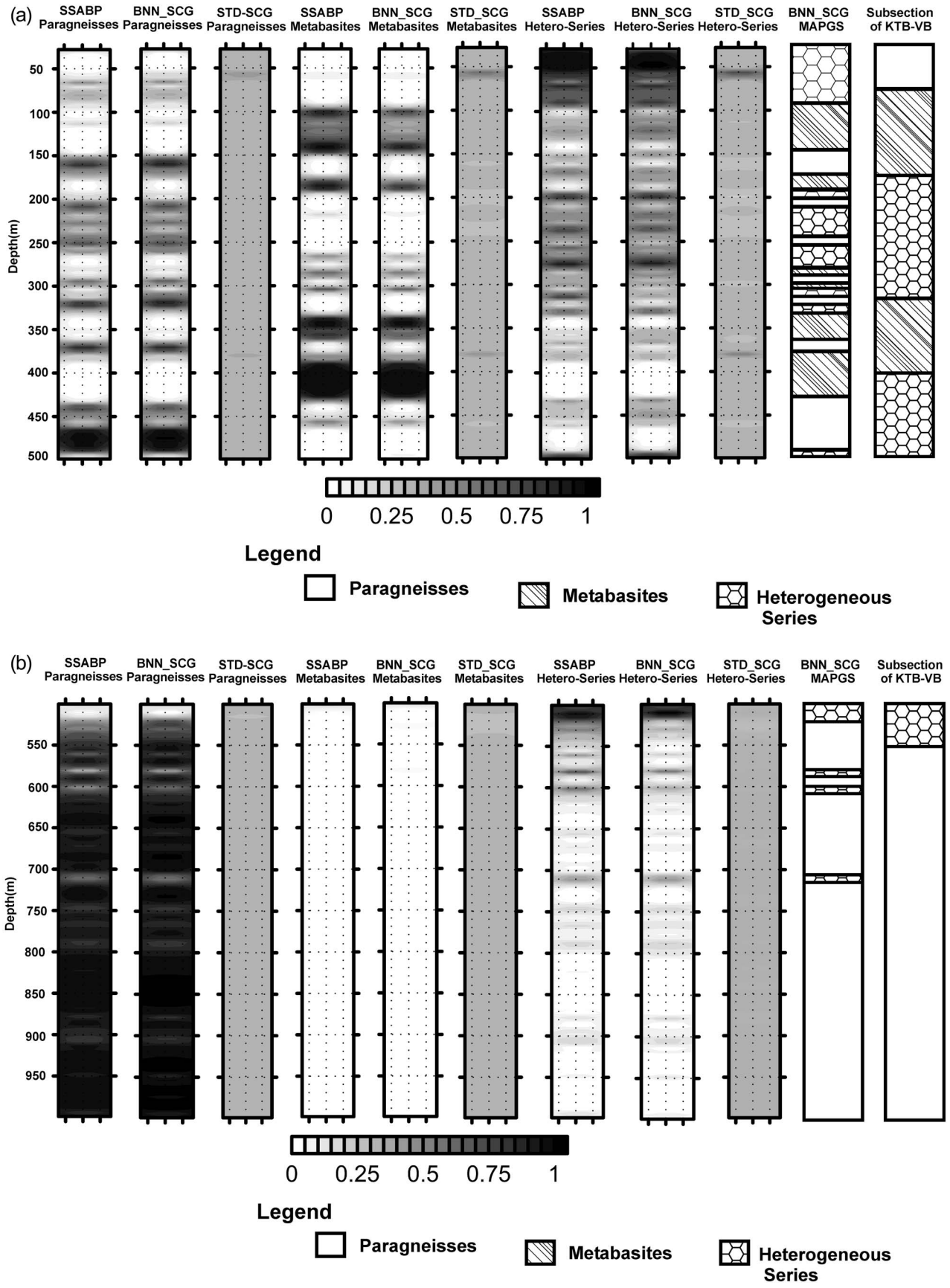
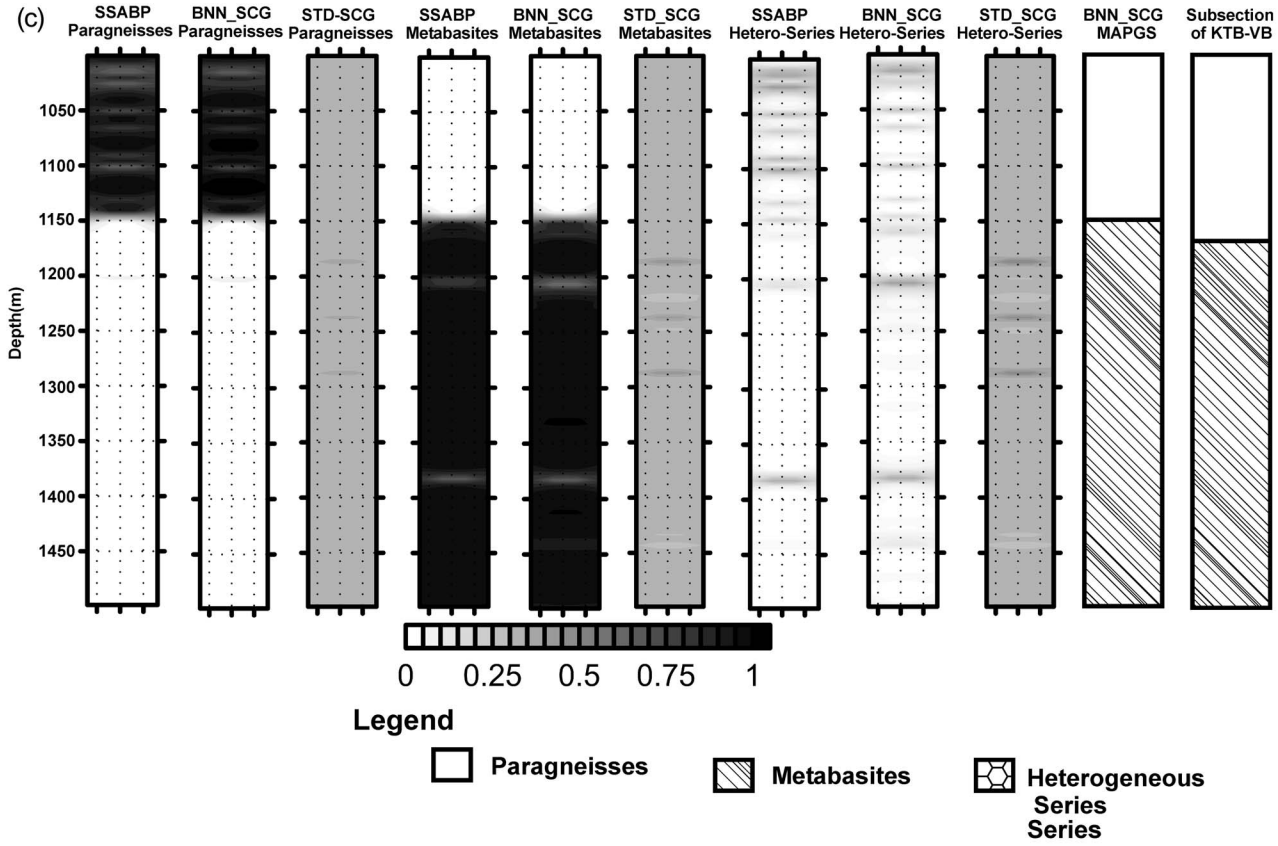

(a)
SSABP Paragneisses
BNN_SCG Paragneisses
STD-SCG Paragneisses
SSABP Metabasites
BNN_SCG Metabasites
STD_SCG Metabasites
SSABP Hetero-Series
BNN_SCG Hetero-Series
STD_SCG Hetero-Series
BNN_SCG MAPGS
Subsection of KTB-VB
Depth(m)
50
100
150
200
250
300
350
400
450
500
0
0.25
0.5
0.75
1
Legend
Paragneisses
Metabasites
Heterogeneous Series
(b)
SSABP Paragneisses
BNN_SCG Paragneisses
STD-SCG Paragneisses
SSABP Metabasites
BNN_SCG Metabasites
STD_SCG Metabasites
SSABP Hetero-Series
BNN_SCG Hetero-Series
STD_SCG Hetero-Series
BNN_SCG MAPGS
Subsection of KTB-VB
Depth(m)
550
600
650
700
750
800
850
900
950
0
0.25
0.5
0.75
1
Legend
Paragneisses
Metabasites
Heterogeneous Series

Figure 4



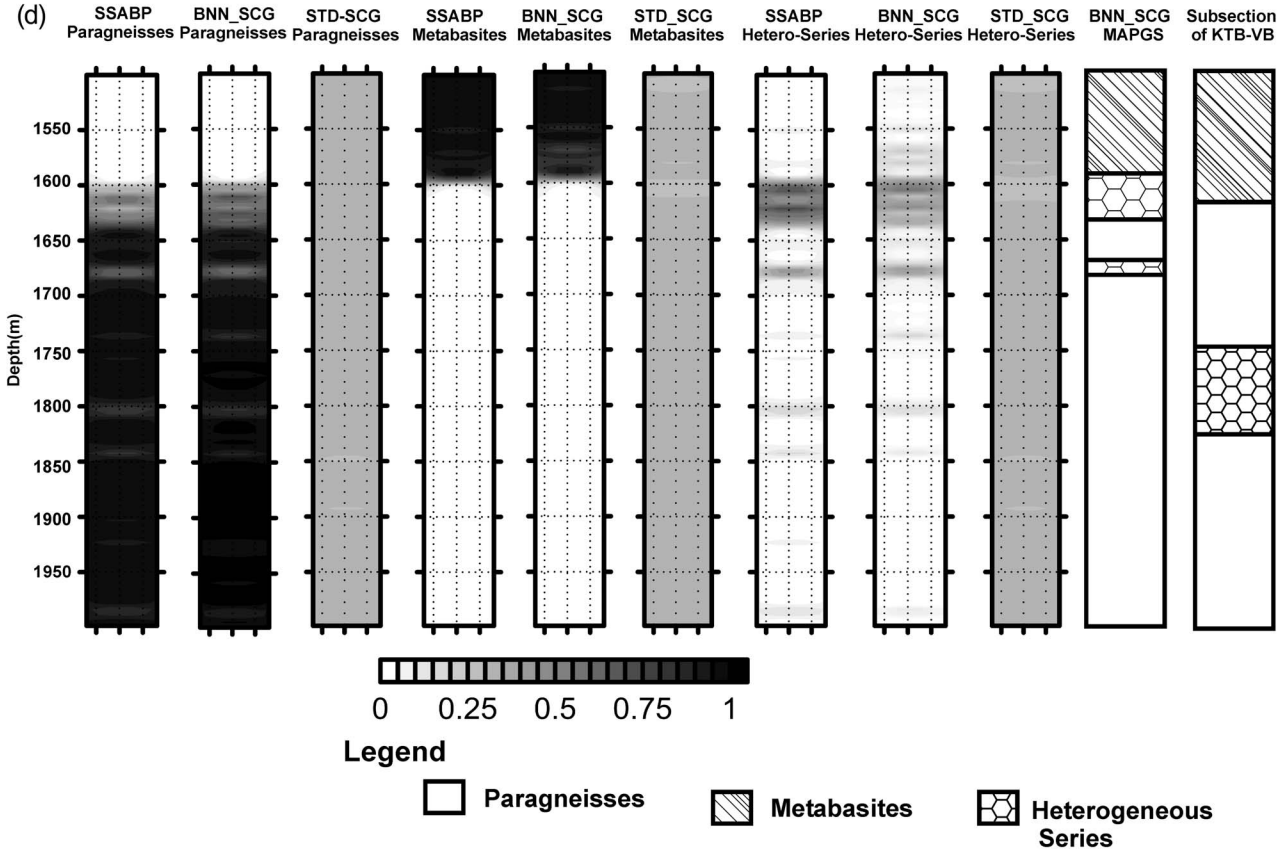


Figure 4. (continued)

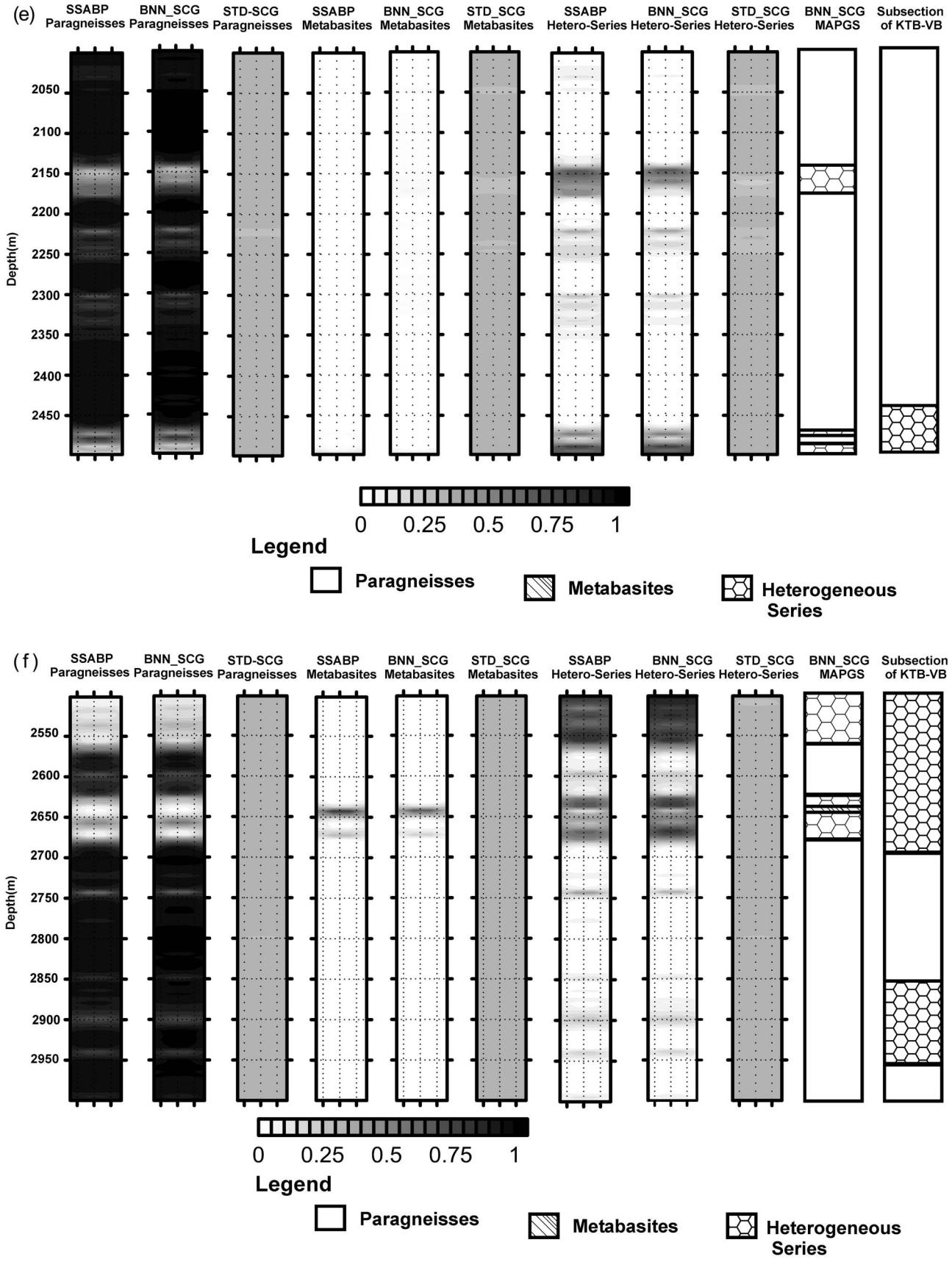
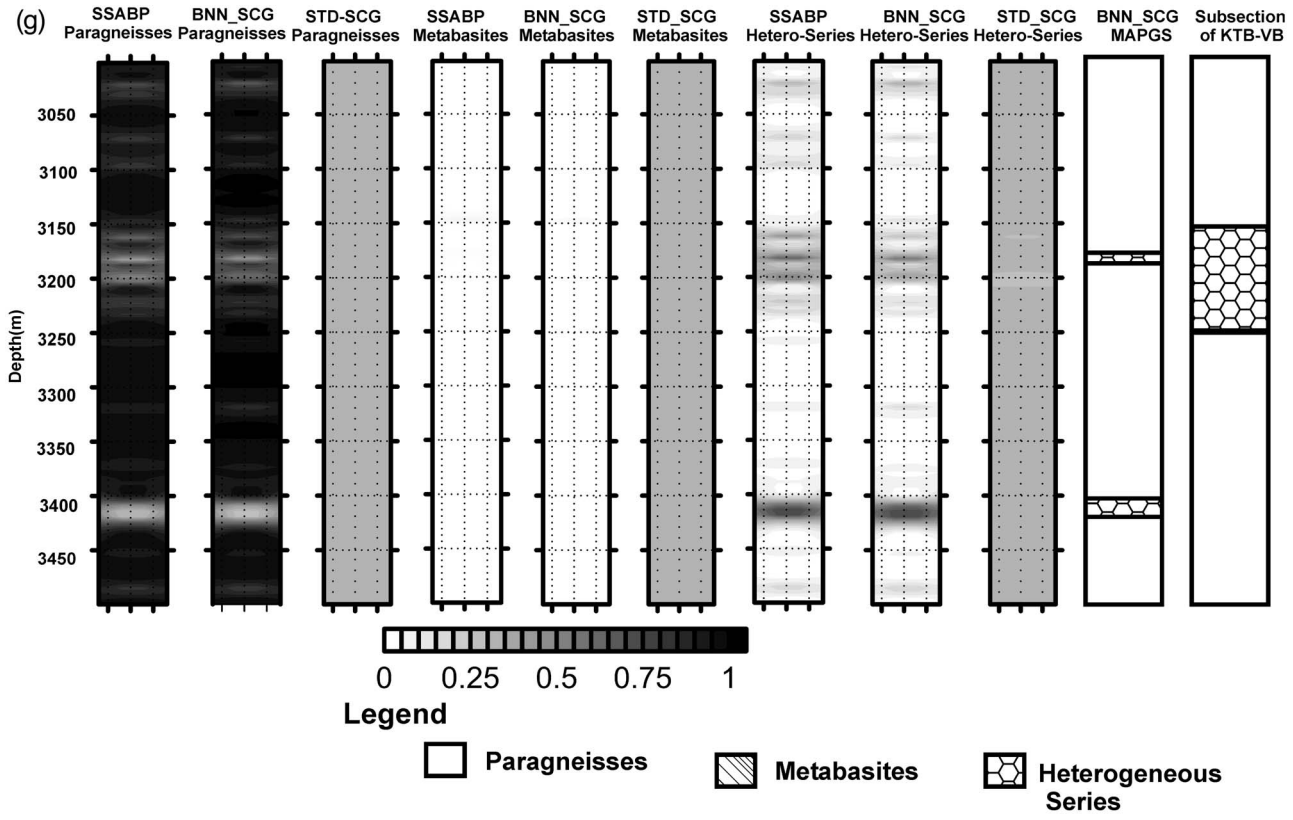


Figure 4. (continued)



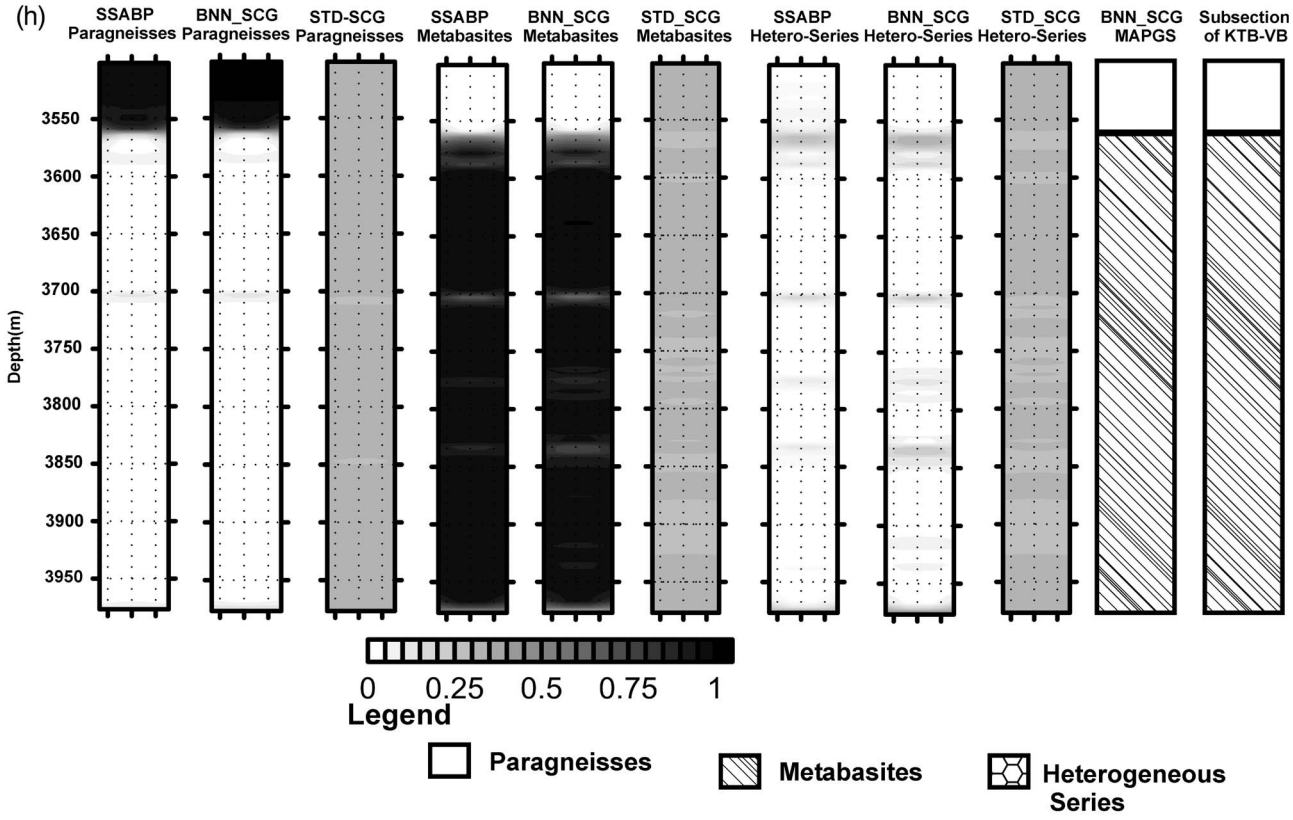


Figure 4. (continued)

8.3. Main Borehole (KTB-HB) (up to 7000 m Depth)

[39] Comparison of the BNN modeling result of the main KTB borehole data also exhibits, in general, good correlation with the published results of *Emmermann and Lauterjung* [1997] (Figures 5a–5d). However, there are some variations too. For instance, there is an evidence of bimodal combination of the hetero-series and the metabasites sequence at the depth interval of 4400–4500 m instead of a single depositional sequence of the metabasites as reported in previous investigation. Further there is also evidence for the thinner heterogeneous series at the depths range of 4580–4583 m, 4809–4812 m, and 4580–4583 m instead of a single metabasites unit. Again at the depth interval of 5440–5500 m, there is evidence of a bimodal sequence of the hetero-series and the metabasites instead of the metabasites only. Further, a closer look at Figures 5a–5d reveal the presence of heterogeneous series, metabasites, and paragneissess at the depth range of 5500–5650 m instead of only metabasites and results also suggest the presence of heterogeneous series at the depth range of 5820–5840 m instead of the metabasites unit. It is interesting to note here that the BNN modeling results also reveal successions of an additional structure at the depth ranges of 6017–6026 m, 6322–6334 m, and 6400–6418 m in the heterogeneous series (Figure 5e), in addition to the main classes at the depth interval of 5543–5560 m (Figure 5d). Figure 5f shows the dominance of heterogeneous series. But the BNN results show the presence of heterogeneous series at the depth range of 6550–6665 m instead of the metabasites and the heterogeneous series and results also indicate the presence of heterogeneous series and the metabasites at the depth interval of 6710–7000 m instead of a single metabasites unit. The average standard deviation estimated at the network output corresponding to the paragneisses, metabasites, and heterogeneous series is ±0.30 for the entire KTB main hole (here, 4000–7000 m), except for depths 4624.73 m, 5647.33–5650.38 m, and 5682.38 m, where the standard deviation is ±0.44, ±0.68, and ±0.87, respectively.

9. Computation Time

[40] Table 5 compares performances in terms of the execution(CPU) time, error bars, number of iterations, number of parameters, and processor/memory used between the current SCG-based BNN method and the hybrid Monte Carlo (HMC)/Markov Chain Monte Carlo (MCMC)-based BNN method [*Maiti and Tiwari*, 2009, 2010]. We selected a fixed number (100) of iterations instead of an arbitrary stopping criterion. For the comparison, the simulations were done with MLP network with 20 hidden nodes. The initial prior hyperparameters and the number of training examples (351) were also kept unchanged for both the simulation. Table 5 shows that the SCG-based BNN requires less execution time at the cost of error bars compared to the HMC/MCMC-based BNN. Experimental results show that using large amounts of training data the SCG-based BNN and HMC/MCMC-based BNN performance are similar, but the MCMC method seems superior on smaller-sized training sets. We note that the disadvantage of the present SCG-based BNN method for small data sets, namely that a large part of the computational effort is taken up with the inversion of the Hessian. Thus, the Bayesian approach is undeniably expensive in computational load, but in complex real-world problems there are few cheap alternatives.

10. Discussions

[41] Comparison of the MAPGS with the published litho-species section of *Emmermann and Lauterjung* [1997] exhibits more or less matching patterns (Figures 4 and 5). In addition to this, the BNN model reveals some finer structural details, which might be geologically significant. We note, however, that in such complex geological situations, it is somewhat intricate to assert an exact geological interpretation for these thin successions as to whether these apparently visible finer details inferred from our study are truly meaningful geological structures or simply an artifact of our analysis. To examine the authenticity of these structural details, we manually checked a few samples produced by the trained network with the limited core knowledge defined in Table 2 (comparison is given in Table 6). It is interesting to see that the trained network produces more or less identical results that are consistent with the training data. We note, however, that the present probabilistic histogram model of litho-facies classification cannot be uniquely constrained and/or compared with the existing litho-section [*Maiti et al.*, 2007, Figure 2]. The reason being that the published results are mostly gross-average depth sections estimated from the litho-sections. The second reason could be that the observed data may be biased with some deceptive "red noise" signals with nonzero mean. Hence, it is likely that there could be some possibility of error due to lack of resolution in the KTB data. It is noteworthy, however, that most of the deviations and differences between the published and the BNN model were observed in the pilot borehole and moreover in the upper part of the crust.

[42] Further we note that there are some mismatches between the present model and the published model. The standard deviation error map, which shows uncertainty in the classification of the litho-facies boundaries, is prepared based on the clear distinction between the winner and the non-winner node values. A standard deviations with a larger than average value shows more uncertainty in prediction and vice versa. In some cases, however, we found that the standard deviation values are more even when the maximum a posteriori probability is comparable to the litho-facies units. This seems to have occurred in the estimation of uncertainty in terms of probability distribution functions of

Figure 5. (a) Comparison of a maximum a posteriori geological section (MAPGS) obtained by BNN with the SCG approach with the maximum likelihood geological section (MLGS) obtained by the SSABP neural network approach with the published litho-facies subsection of main hole (KTB-HB) (published litho-subsection is redrawn after *Emmermann and Lauterjung* [1997]) and the standard deviation (std) error map estimated at the network output by the BNN approach at the depth interval of 4000–4500 m. (b)–(f) Same for the depth range of 4500–5000 m, …, 6500–7000 m in KTB main hole (KTB-HB).

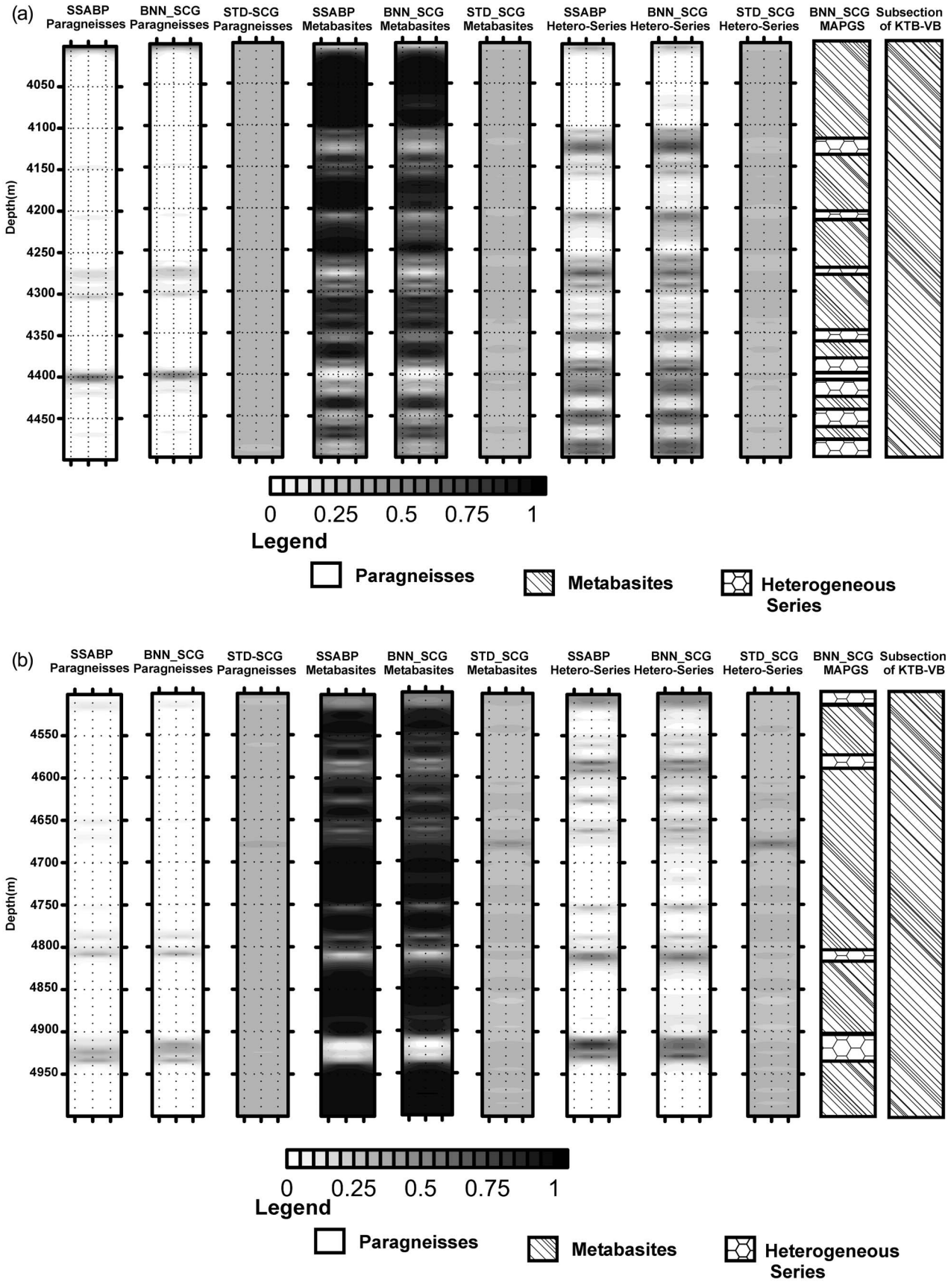

(a)
SSABP Paragneisses
BNN_SCG Paragneisses
STD-SCG Paragneisses
SSABP Metabasites
BNN_SCG Metabasites
STD_SCG Metabasites
SSABP Hetero-Series
BNN_SCG Hetero-Series
STD_SCG Hetero-Series
BNN_SCG MAPGS
Subsection of KTB-VB
Depth(m)
4050
4100
4150
4200
4250
4300
4350
4400
4450
0 0.25 0.5 0.75 1
Legend
Paragneisses
Metabasites
Heterogeneous Series
(b)
SSABP Paragneisses
BNN_SCG Paragneisses
STD-SCG Paragneisses
SSABP Metabasites
BNN_SCG Metabasites
STD_SCG Metabasites
SSABP Hetero-Series
BNN_SCG Hetero-Series
STD_SCG Hetero-Series
BNN_SCG MAPGS
Subsection of KTB-VB
Depth(m)
4550
4600
4650
4700
4750
4800
4850
4900
4950
0 0.25 0.5 0.75 1
Legend
Paragneisses
Metabasites
Heterogeneous Series

Figure 5

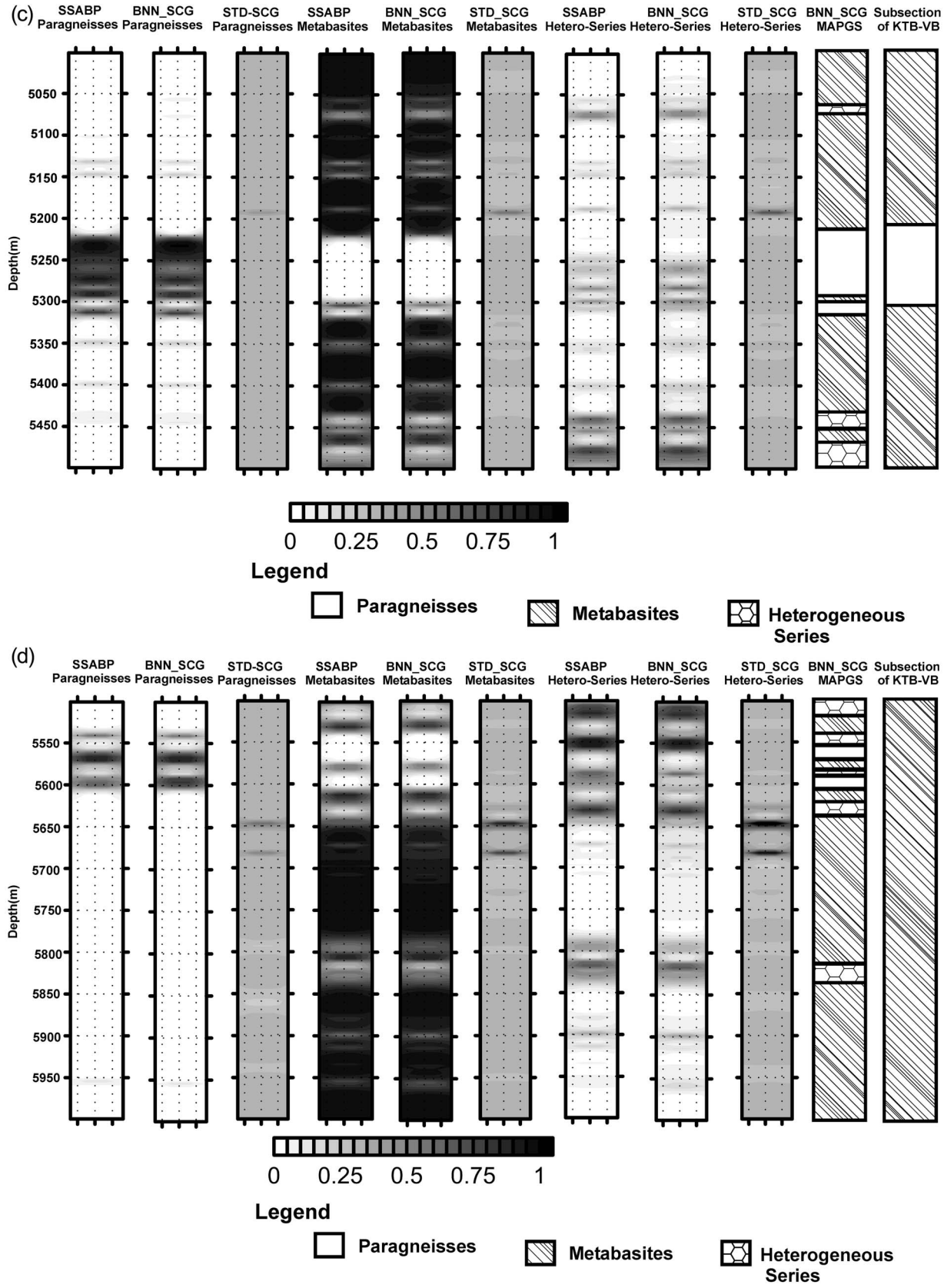
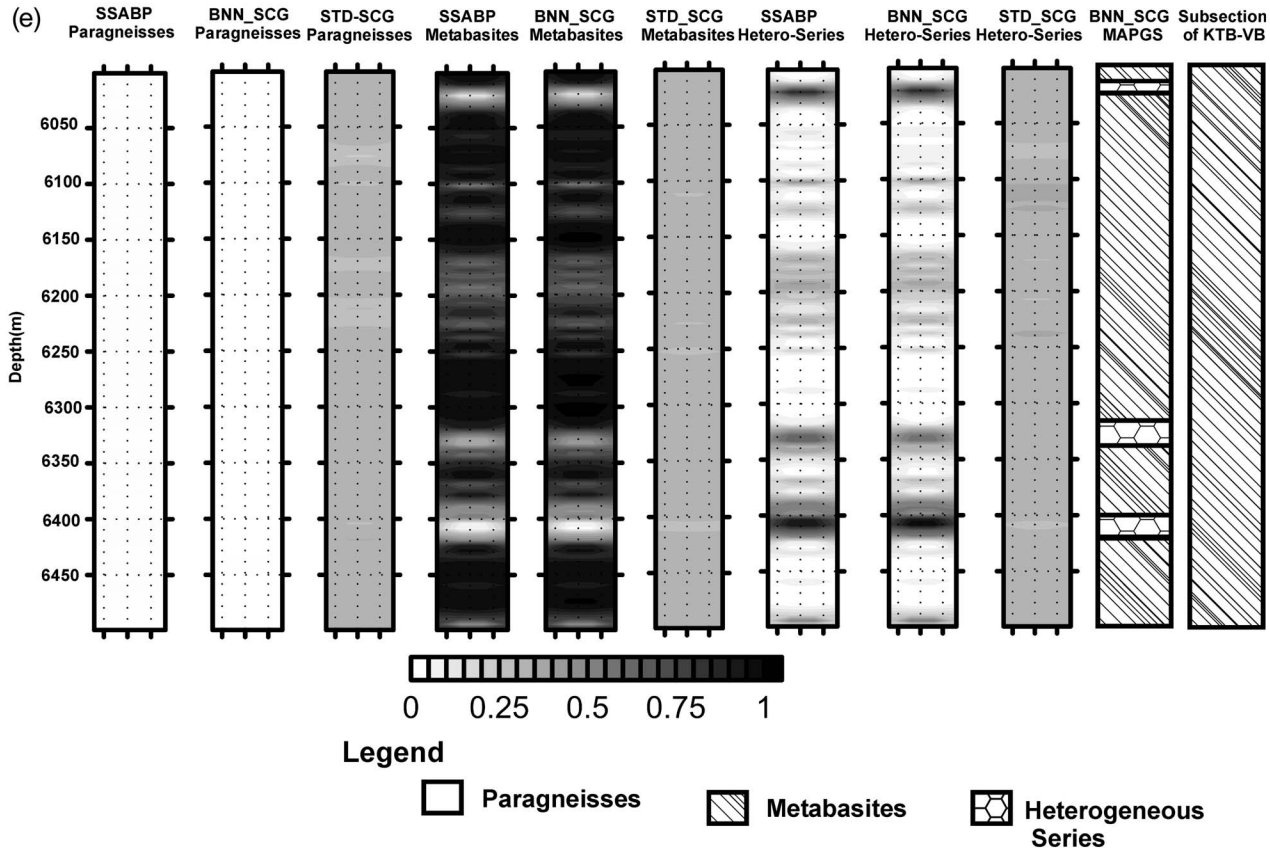


Figure 5. (continued)



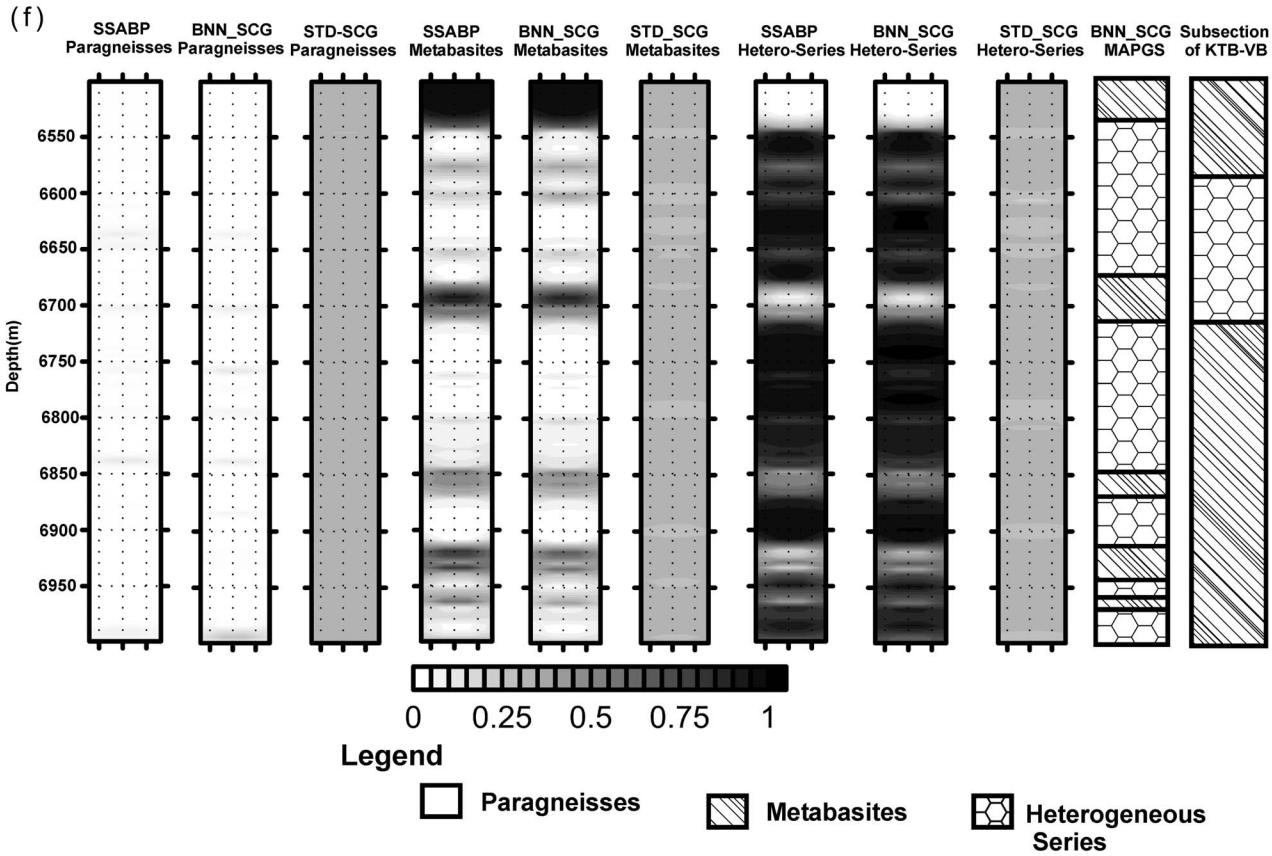


Figure 5. (continued)

Table 5. Showing the Comparison of the Performances Between the Present SCG-Based BNN and the HMC/MCMC-Based BNN

Methods	Number of Iterations	Time Taken (s)	Error Bars	Number of Parameters	Computer Processor Used With Memory
SCG-BNN	100	3	0.30	143	Intel(R) Core(TM) 2 Due CPU E7500@ 2.93GHz RAM 4GB
HMC/MCMC-BNN	100	15	0.15	143	Intel(R) Core(TM) 2 Due CPU E7500@ 2.93GHz RAM 4GB

noisy data as mentioned above. We mention, however, that while interpreting prediction of the network's output node with the maximum a posteriori value, in most of the cases, we found an excellent separation between the winning and non-winning output node values rendering overall the actual patterns with high correlation (correlation coefficients ~0.943).

[43] We mention that prior to applying the method to the real KTB data analysis, we thoroughly examined the sensitivity of neural network hyperparameters with adequate

Table 6. Analysis of Real Data Taken From Both KTB Pilot Hole (KTB-VB) and KTB Main Hole (KTB-HB) From Different Depth With ± Prediction Error

Borehole (samples of data taken)	Depth (m)	Density (g/cc)	Neutron Porosity (%)	Gamma Ray Intensity (API)	Desired Output/ Binary Code			Neural Networks Output ± STD		
KTB-VB	3119.17	2.71	10.80	107.1	1	0	0	1.00 ± 0.30	0.00 ± 0.30	0.03 ± 0.30
KTB-VB	1574.29	2.94	12.34	45.00	0	1	0	0.00 ± 0.31	0.98 ± 0.31	0.01 ± 0.31
KTB-VB	89.00	2.73	10.46	82.05	0	0	1	0.05 ± .0.31	0.02 ± 0.31	0.83 ± 0.31
KTB-VB	305.86	2.94	14.06	32.61	0	1	0	0.00 ± 0.31	0.81 ± 0.29	0.18 ± 0.28
KTB-VB	893.82	2.82	12.94	128.41	0	0	1	0.67 ± 0.31	0.00 ± 0.31	0.31 ± 0.31
KTB-VB	1393.54	3.01	11.24	23.26	0	1	0	0.00 ± 0.30	0.99 ± 0.30	0.01 ± 0.31
KTB-VB	2252.47	2.80	14.94	119.17	1	0	0	0.75 ± 0.30	0.00 ± 0.30	0.22 ± 0.30
KTB-VB	3864.25	2.94	13.59	19.10	0	1	0	0.00 ± 0.30	0.98 ± 0.30	0.01 ± 0.31
KTB-VB	3559.45	2.74	12.11	104.35	1	0	0	1.00 ± 0.30	0.01 ± 0.30	0.03 ± 0.30
KTB-VB	771.60	2.68	5.33	112.73	1	0	0	0.96 ± 0.31	0.00 ± 0.31	0.01 ± 0.31
KTB-VB	1072.43	2.776	9.72	115.58	1	0	0	1.00 ± 0.30	0.0 ± 0.30	0.02 ± 0.30
KTB-VB	1145.74	2.745	12.03	106.69	1	0	0	0.99 ± 0.31	0.01 ± 0.30	0.00 ± 0.31
KTB- VB	1374.03	2.97	8.73	15.83	0	1	0	0.00 ± 0.30	0.99 ± 0.31	0.01 ± 0.31
KTB- VB	1715.26	2.70	10.08	105.55	1	0	0	1.02 ± 0.30	0.01 ± 0.31	0.00 ± 0.30
KTB- VB	1878.48	2.72	12.03	111.65	1	0	0	1.00 ± 0.30	0.01 ± 0.30	0.00 ± 0.30
KTB- VB	2084.83	2.75	16.51	120.17	1	0	0	1.03 ± 0.30	0.00 ± 0.31	0.01 ± 0.30
KTB- VB	2290.42	2.75	14.70	119.58	1	0	0	0.96 ± 0.30	0.01 ± 0.30	0.02 ± 0.30
KTB- VB	2697.17	2.65	10.00	103.20	1	0	0	1.02 ± 0.31	0.00 ± 0.30	0.01 ± 0.31
KTB- VB	2891.33	2.71	8.76	103.30	1	0	0	1.00 ± 0.31	0.01 ± 0.31	0.02 ± 0.31
KTB- VB	3177.84	2.77	9.69	96.31	1	0	0	0.78 ± 0.31	0.02 ± 0.31	0.20 ± 0.30
KTB- VB	3801.16	3.04	14.10	30.46	0	1	0	0.00 ± 0.31	0.99 ± 0.30	0.01 ± 0.31
KTB- VB	3889.55	3.00	8.677	29.21	1	0	0	0.00 ± 0.30	1.00 ± 0.31	0.03 ± 0.30
KTB-HB	6515.86	3.00	14.82	20.32	0	1	0	0.00 ± 0.30	0.99 ± 0.31	0.00 ± 0.31
KTB-HB	6807.09	2.74	11.17	55.44	0	0	1	0.00 ± 0.30	0.05 ± 0.29	0.95 ± 0.30
KTB-HB	6470.14	2.72	25.11	16.72	0	1	0	0.00 ± 0.31	1.00 ± 0.31	0.01 ± 0.31
KTB-HB	6999.12	2.85	12.47	49.45	0	0	1	0.00 ± 0.31	0.38 ± 0.31	0.61 ± 0.31
KTB-HB	5677.35	2.95	1.66	26.75	0	1	0	0.00 ± 0.31	1.00 ± 0.31	0.03 ± 0.31
KTB-HB	5372.25	2.97	15.23	16.83	0	1	0	0.00 ± 0.31	1.00 ± 0.31	0.02 ± 0.31
KTB-HB	5217.87	2.93	4.08	29.53	0	1	0	0.00 ± 0.31	1.00 ± 0.31	0.12 ± 0.31
KTB-HB	4547.00	2.82	8.67	35.92	0	1	0	0.00 ± 0.30	0.79 ± 0.29	0.22 ± 0.29
KTB-HB	4427.37	2.81	4.52	38.63	0	1	0	0.00 ± 0.31	0.88 ± 0.31	0.17 ± 0.31
KTB-HB	4433.16	2.75	4.89	42.33	0	1	0	0.00 ± 0.31	1.02 ± 0.31	0.01 ± 0.31
KTB-HB	4442.00	2.92	10.21	37.06	0	1	0	0.00 ± 0.31	0.99 ± 0.31	0.01 ± 0.31
KTB-HB	4950.56	2.90	12.09	33.75	0	1	0	0.01 ± 0.31	0.99 ± 0.31	0.03 ± 0.31
KTB-HB	6325.36	2.75	15.72	13.35	0	1	0	0.01 ± 0.31	0.87 ± 0.31	0.01 ± 0.31
KTB-HB	4006.59	2.95	15.73	36.147	0	1	0	0.00 ± 0.31	0.84 ± 0.31	0.15 ± 0.31
KTB-HB	4002.93	2.84	16.83	109.94	1	0	0	1.00 ± 0.31	0.00 ± 0.30	0.01 ± 0.31
KTB-HB	4206.24	2.90	14.51	44.58	0	1	0	0.00 ± 0.30	1.00 ± 0.27	0.05 ± 0.27
KTB-HB	4311.39	2.92	17.16	22.42	0	1	0	0.00 ± 0.31	0.98 ± 0.31	0.02 ± 0.31
KTB-HB	4548.22	2.87	7.96	34.12	0	1	0	0.00 ± 0.31	0.64 ± 0.29	0.41 ± 0.29
KTB-HB	4556.76	2.97	5.31	25.51	0	1	0	0.00 ± 0.31	1.01 ± 0.26	0.03 ± 0.25
KTB-HB	4630.67	2.89	2.73	24.59	0	1	0	0.01 ± 0.31	0.85 ± 0.31	0.14 ± 0.31
KTB-HB	4873.75	3.08	3.00	28.48	0	1	0	0.00 ± 0.31	1.02 ± 0.31	0.01 ± 0.31
KTB-HB	5054.34	2.87	10.66	32.95	0	1	0	0.00 ± 0.31	0.99 ± 0.31	0.00 ± 0.31
KTB-HB	5058.30	2.69	14.47	135.41	0	0	1	0.16 ± 0.31	0.22 ± 0.31	0.62 ± 0.31
KTB-HB	5149.90	2.97	9.91	19.61	0	1	0	0.00 ± 0.31	1.01 ± 0.31	0.03 ± 0.31
KTB-HB	5321.19	2.86	6.56	40.54	0	1	0	0.03 ± 0.31	0.64 ± 0.28	0.380 ± 0.28
KTB-HB	5743.95	3.06	6.10	18.39	0	1	0	0.00 ± 0.31	1.01 ± 0.31	0.00 ± 0.31
KTB-HB	5840.27	2.65	14.53	16.09	0	1	0	0.00 ± 0.31	0.98 ± 0.31	0.00 ± 0.31
KTB-HB	6271.56	2.96	18.45	13.75	0	1	0	0.00 ± 0.31	1.00 ± 0.31	0.00 ± 0.31
KTB-HB	6486.90	2.78	15.27	42.39	0	1	0	0.00 ± 0.31	0.77 ± 0.31	0.22 ± 0.31

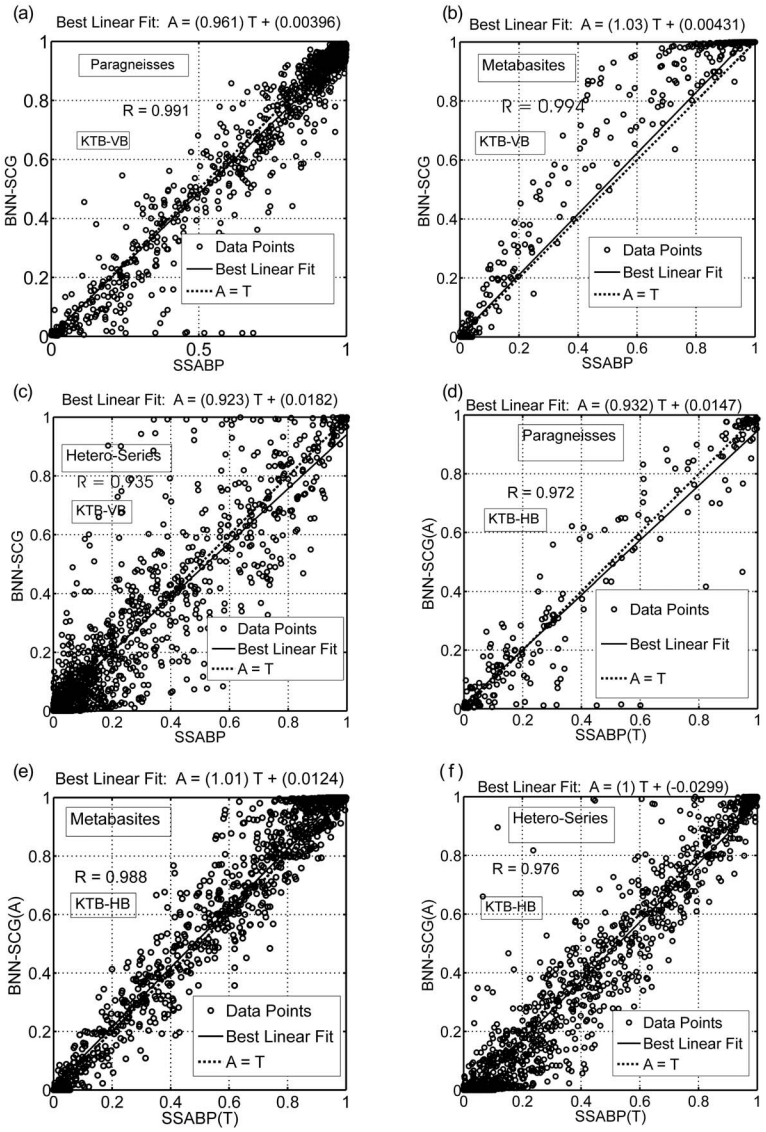


Figure 6. (a–f) Regression analysis of pilot hole (KTB-VB) and KTB main hole (KTB-HB) corresponds to the paragneisses, metabasites, and hetero-series shows very good agreement between the BNN and the SSABP results. A dashed line indicates the best linear fit (slope 1 and y-intercept 1). The solid line in the figure shows the perfect fit. The correlation coefficient (R), between the two approaches actually, measures how well the variation is in two results. If this number is equal to 1, there is a perfect correlation between the two approaches.

empirical examples for network prediction. The experiment guided us to choose appropriate hyperparameters for the actual data analysis. In this way, we could also fairly rule out the uncertainty due to choosing the appropriate hyperparameters in interpretation. The experimental data analysis suggests that the true prediction heavily depends on the ratio $\frac{\lambda}{\mu}$. Accordingly, the hyperparameter was set in for the real KTB data analysis. Further regression analysis of both borehole results and between two approaches (SSABP and BNN–SCG) also shows consistent and good agreements ($R \sim 0.97$) (Figures 6a–6f).

[44] It may thus be emphasized that the BNN algorithm employed here combined with its validation, test and regression analyses do provide credentials to the present results. As discussed above, despite various sources of errors, the apparently visible changes in litho-logs successions appears to be the inter-bedded geological structures that remained ambiguous/unrecognized in earlier qualitative investigations. The output of the histogram type networks for choosing a maximum a posterior probability value, as discussed in detail by *Bishop* [1995], is more appropriate and provides better guidance on data analysis for such a complex data analysis.

11. Conclusions

[45] A new BNN approach [*Nabney*, 2004] is employed to decode changes in layer successions from well-log data. The

stability and the efficiency of the BNN approach are examined on empirically generated noisy as well as noise-free data sets. The BNN has inherent ability to approximate the functional relationship between the input and the output space/domain by learning through examples, even if there is no deterministic relationship between the input and the output space/domain. The method provides a neat and tractable mathematical framework in which the network weights could be adjusted in a fully probabilistic way. The proposed method is robust for uncertainty analysis and takes care of over-fitting and under-fitting in a natural way. The BNN technique is also an efficient and cost-effective tool to interpret a large amount of borehole log data. This provides a good testimony to use the BNN-based techniques to solve the nonlinear inversion problem for borehole geophysics.

[46] The BNN approach is then successfully applied to classify changes in the litho-facies boundaries from the real well-log data obtained from the German Continental Deep Drilling site. The method essentially allows us to estimate uncertainty in network prediction along the entire length of litho-section of the KTB. Some mismatching observed in the BNN analysis might result due to the presence of red signals with nonzero mean and/or lack of resolution in the KTB data. Over all the results of present analyses suggest that, besides corroborating well with the existing results on the KTB site, the method also uncovers additional finer details of intervening layer successions in the bigger geological units, which seem to be of some geological significance and should form the basis for more detailed quantitative examination. Thus, besides introducing a new probabilistic inversion scheme to the problems of well-log data for litho-facies classification, the present analysis also reveals some new results and thus explores the generality of the new method for its actual application to other domains of earth sciences. Because of its computational efficiency, it is proposed that the BNN methods could be further exploited for analyzing a large amount of borehole data in some other geologically complex areas of interest.

[47] **Acknowledgments.** We would like to thank Ian T. Nabney for providing the Netlab tool box from which some of the routines have been used in the work. Saumen Maiti expresses sincere thanks to Department of Science and Technology, Government of India to carry out the research work. We are also grateful to the Director of the National Geophysical Research Institute (CSIR) and the Director of the Indian Institute of Geomagnetism, Navi-Mumbai, for their kind permission to publish this work. We are also thankful to Prof. Hans-Joachim Kumpel for providing the KTB data.

References

Aires, F. (2004), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 1. Network weights, *J. Geophys. Res.*, *109*, D10303, doi:10.1029/2003JD004173.

Aristodemou, E., C. Pain, C. Oliveira, and T. Goddard (2005), Inversion of nuclear well-logging data using neural networks, *Geophys. Prospect.*, *53*, 103–120.

Baldwin, J., A. R. M. Bateman, and C. L. Wheatley (1990), Application of neural network to the problem of mineral identification from well logs, *Log Anal.*, *31*, 279–293.

Benaouda, D., G. Wadge, R. B. Whitmarsh, R. G. Rothwell, and C. MacLeod (1999), Inferring the lithology of borehole rocks by applying neural network classifiers to downwhole logs: An example from the Ocean Drilling Program, *Geophys. J. Int.*, *136*, 477–491.

Berckhemer, H., A. Rauen, H. Winter, H. Kern, A. Kontny, M. Lienert, G. Nover, J. Pohl, T. Popp, A. Schult, J. Zinke, and H. C. Soffel (1997), Petrophysical properties of the 9-km deep crustal section at KTB, *J. Geophys. Res.*, *102*(B8), 18337–18361.

Bescoby, D. J., G. C. Cawley, and P. N. Chroston (2006), Enhanced interpretation of magnetic survey data from archaeological sites using artificial neural networks, *Geophysics*, *71*(5), H45–H53.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford.

Bosch, M. (1999), Lithologic tomography: From plural geophysical data to lithology estimation, *J. Geophys. Res.*, *104*(B1), 749–766.

Bosch, M., and J. McGaughey (2001), Joint inversion of gravity and magnetic data under lithologic constraints, *Leading Edge*, *20*, 877–881.

Bosch, M., A. Gullen, and P. Ledru (2001), Lithologic tomography: An application to geophysical data from the Cadomian belt of northern Brittany, France, *Tecton.*, *331*, 197–227.

Buntine, W. L., and A. S. Weigend (1991), Bayesian back propagation, *Complex Syst.*, *5*, 603–643.

Busch, J. M., W. G. Fortney, and L. N. Berry (1987), Determination of lithology from well logs by statistical analysis, *SPE Form. Eval.*, *2*, 412–418.

Calderon-Macias, C., M. K. Sen, and P. L. Stoffa (2000), Artificial neural networks for parameter estimation in geophysics, *Geophys. Prosp.*, *48*, 21–47.

Coppola, E. A., Jr., A. J. Rana, M. M. Poulton, F. Szidarovszky, and V. W. Uhl (2005), A neural network model for predicting aquifer water level elevations, *Ground Water*, *43*(2), 231–241.

Coulibaly, P., F. Anctil, and B. Bobee (2001), Multivariate reservoir inflow forecasting using temporal neural networks, *J. Hydrol. Eng.*, *6*, 367–376.

Dai, H., and C. Macbeth (1995), Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophys. J. Int.*, *120*, 758–774.

Delfiner, P., O. Peyret, and O. Serra (1987), Automatic determination of lithology from well logs, *SPE Form. Eval.*, *2*, 303–310.

Devilee, R. J. R., A. Curtis, and K. Roy-Chowdhury (1999), An efficient probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness, *J. Geophys. Res.*, *104*(12), 28,841–28,856.

Dystart, P. S., and J. J. Pulli (1990), Regional seismic event classification at the NORESS array: Seismological measurements and use of trained neural network, *Bull. Seis. Soc. Am.*, *80*, 1910–1933.

Emmermann, R., and J. Lauterjung (1997), The German Continental Deep Drilling Program KTB: Overview and major results, *J. Geophys. Res.*, *102*, 18179–18201.

Feng, X.-T., M. Seto, and K. Katsuyama (1997), Neural dynamic modeling on earthquake magnitude series, *Geophys. J. Int.*, *128*, 547–556.

Franke, W. (1989), The geological framework of the KTB drill site, in *The German Continental Deep Drilling Program (KTB)*, edited by R. Emmermann and J. Wohlenberg, pp. 38–54, Springer, Berlin.

Fuller, W. A. (1976), *Introduction to Statistical Time Series*, John Wiley, New York.

Gassaway, G. R., D. R. Miller, L. E. Bennett, R. A. Brown, M. Rapp, and V. Nelson (1989), Amplitude variations with offset: Fundamentals and case histories, SEG Continuing Education Course Notes.

Helle, H. B., A. Bhatt, and B. Ursin (2001), Porosity and permeability prediction from wireline logs using artificial neural networks: A North Sea case study, *Geophys. Prosp.*, *49*, 431–444.

Khan, M. S., and P. Coulibaly (2006), Bayesian neural network for rainfall-runoff modeling, *Water Resour. Res.*, *42*, W07409, doi:10.1029/2005WR003971.

Leonardi, S., and H. Kumpel (1998), Variability of geophysical log data and signature of crustal heterogeneities at the KTB, *Geophys. J. Int.*, *135*, 964–974.

Leonardi, S., and H. Kumpel (1999), Fractal variability in super deep borehole-implications for the signature of crustal heterogeneities, *Tectonophysics*, *301*, 173–181.

MacKay, D. J. C. (1992), A practical Bayesian framework for back-propagation networks, *Neural Comput.*, *4*(3), 448–472.

Maiti, S., and R. K. Tiwari (2009), A hybrid Monte Carlo method based artificial neural networks approach for rock boundaries identification: A case study from the KTB bore hole, *Pure Appl. Geophys.*, *166*, 2059–2090, doi:10.1007/s00024-009-0533-y.

Maiti, S., and R. K. Tiwari (2010), Automatic discriminations of geophysical signals using the Bayesian neural networks approach, *Geophysics*, *75*(1), E67–E78, doi:10.1190/1.3298501.

Maiti, S., R. K. Tiwari, and H. J. Kumpel (2007), Neural network modeling and classification of lithofacies using well log data: A case study from KTB borehole site, *Geophys. J. Int.*, *169*, 733–746.

McCormack, M. D., D. Zaucha, and D. Dushek (1993), First break refraction event picking and seismic data trace editing using neural networks, *Geophysics*, *58*, 67–78.

Meier, U., A. Curtis, and J. Trampert (2007), Global crustal thickness from neural network inversion of surface wave data, *Geophys. J. Int.*, *169*, 706–722.

Moller, M. (1993), A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks*, *6*, 525–533.

Mosegaard, K., and A. Tarantola (1995), Monte Carlo sampling of solutions to inverse problem, *J. Geophys. Res.*, *100*(B7), 12,431–12,447.

Murat, M. E., and A. J. Rudman (1992), Automated first arrival picking: A neural network approach, *Geophys. Prosp.*, *40*, 587–604.

Nabney, I. T. (2004), *Netlab Algorithms for Pattern Recognition*, Springer, New York.

Neal, R. M. (1993), Bayesian learning via stochastic dynamics, in *Advances in Neural Information Processing Systems*, vol. 5, edited by C. L. Giles et al., pp. 475–482, Morgan Kaufmann, San Francisco, Calif.

Pechnig, P., S. Haverkamp, J. Wohlenberg, G. Zimmermann, and H. Burkhardt (1997), Integrated interpretation in the German Continental Deep Drilling Program: Lithology, porosity, and fracture zones, *J. Geophys. Res.*, *102*, 18,363–18,390.

Pickett, G. R. (1963), Acoustic character logs and their application in formation evaluation, *J. Petr. Tech.*, *15*, 659–667.

Poulton, M., (Ed.) (2001), *Computational Neural Networks for Geophysical Data Processing*, Pergamon, Oxford, U.K.

Raiche, A. (1991), A pattern recognition approach to geophysical inversion using neural nets, *Geophys. J. Int.*, *105*, 629–648.

Rogers, S. J., J. H. Fang, C. L. Karr, and D. A. Stanley (1992), Determination of lithology from well logs using a neural network, *AAPG Bull.*, *76*(5), 731–739.

Rosenblatt, F. (1958), The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol. Rev.*, *65*, 386–408.

Roth, G., and A. Tarantola (1994), Neural networks and inversion of seismic data, *J. Geophys. Res.*, *99*, 6753–6768.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning representations by back-propagating errors, *Nature*, *323*, 533–536.

Sambridge, M., and K. Mosegaard (2002), Monte Carlo methods in geophysical inverse problems, *Rev. Geophys.*, *40*(3), 1009, doi:10.1029/2000RG000089.

Sen, M. K., and P. Stoffa (1996), Bayesian inference, Gibbs' sampler and uncertainty estimation in geophysical inversion, *Geophys. Prospect.*, *44*, 313–350.

Spichak, V., and I. Popova (2000), Artificial neural network inversion of Magnetotelluric data in terms of three-dimensional earth macroparameters, *Geophys. J. Int.*, *142*, 15–26.

Tarantola, A. (1987), *Inverse Problem Theory*, Elsevier, New York.

Tarantola, A. (2006), Popper, Bayes and the inverse problem, *Nat. Phys.*, 492–494.

Van der Baan, M., and C. Jutten (2000), Neural networks in geophysical applications, *Geophysics*, *65*, 1032–1047.

Walker, A. M. (1969), On the asymptotic behaviour of posterior distributions, *J. R. Stat.*, *31*, 80–88.

Williams, P. M. (1995), Bayesian regularization and pruning using a Laplace prior, *Neural Comput.*, *7*, 117–143.

Wolff, M., and J. Pelissier-Combescure (1982), FACIOLOG: Automatic electrofacies determination: SPWLA Annual Logging Symposium paper FF, 6–9.

S. Maiti, Indian Institute of Geomagnetism, Navi-Mumbai-410218, India. (saumen_maiti2002@yahoo.co.in)

R. K. Tiwari, National Geophysical Institute (CSIR), Hyderabad-500007, India.