# Inversion of well logs into lithology classes accounting for spatial dependencies by using Hidden Markov Models and Recurrent Neural Networks

Miao Tian[a,b], Henning Omre[b], Huaimin Xu[a,*]

[a] *College of Geosciences, China University of Petroleum Beijing, Beijing 102249, China*
[b] *Department of Mathematical Sciences, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

**Abstract**

Lithology is a crucial factor in reservoir characterization. Due to the limited availability of cores, the classes of the subsurface lithologies in boreholes need to be predicted from indirect measurements like well logs. However, the spatial interdependence between sediments and the spatial coupling between the well logs data pose challenges in this lithology classification. Numerous proposed classifiers are based on spatial element-wise independence and these classifiers usually fail to provide accurate predictions. In this study, we focus on two classification models from the Bayesian and the deep learning framework, which both take spatial context into account. We discuss a kernel-based hidden Markov (HM) model and a kind of recurrent neural network model named gated recurrent unit (GRU). Cross-validation results from these two models of three partially cored real wells are compared to result from a simple non-spatial deep neural network (DNN) model. The cross-validation results indicate that the lithology classifiers from models taking vertical spatial dependency into account are much more reliable in terms of classification accuracy and geological interpretation. The probabilistically defined HM model performs better than the neural network GRU model.

*Keywords:* Hidden Markov Model, Recurrent Neural Network, Lithology Classification, Well Logs Inversion

## 1. Introduction

Lithology is one of the most crucial factors to control the reservoir characteristics. To predict this categorical attribute, qualitative analyses of the well logging curves labeled by core-plugs are usually performed. Also, geological experience and geophysics forward modeling is involved in the classification procedure (Fresia et al., 2017). However, the number of cores are limited by the costly coring operation and the fragile

[*] Corresponding author
*Email addresses:* tiamo117@hotmail.com (Miao Tian), omre@math.ntnu.no (Henning Omre), xhmcup@126.com (Huaimin Xu)

borehole condition. Thus, the quantitative information hidden in the well logging data is needed in the classification models for the subsurface lithology distribution (Saggaf and Nebrija, 2000; Ghosh et al., 2016).

Proposed classifiers usually belong to one of two types, either multivariate statistics or artificial intelligence. We focus on Bayesian classifiers from the former type and the deep learning neural networks from the latter. In the Bayesian setting, the geological knowledge derived from the reservoir exploration experiences are captured in the prior model while the observation response can be represented in the likelihood model (Loures and Moraes, 2006; Lindberg et al., 2015; Corina and Hovda, 2018). The posterior model is fully defined by the prior and the likelihood models. The Bayesian framework is frequently used in geological inversion of well log data into lithology classes (Schumann, 2002; Eidsvik et al., 2004; Grana et al., 2012; Moja et al., 2018). The deep learning framework provides many data-driven models. Deep learning methods, for example the deep neural network (DNN), take advantage of the hidden information of the labeled data in order to predict attributes automatically based on few distributional assumptions (LeCun et al., 2015; Forgione et al., 2015). In recent years, with the enormous development in the processing power of computers, deep learning methods are applied in many research fields such as computer vision, medical diagnostics, natural language process and robotics. Also, in geoscience research deep learning is used in classification for lithologies and oil-water layers (Liu et al., 2009; Horrocks et al., 2015; Ghosh et al., 2016; Maxwell et al., 2019; Saporetti et al., 2019; Tian et al., 2019).

The lithology class and the corresponding physical properties of the local sediments are mainly controlled by the depositional environment. Since the depositional environment stays relatively stable during a period, the sediments during one period are more similar than those sediments deposited in another period. Hence there is a spatial dependency between the adjacent sediments. Note that we assume that there are no faults or other post-sedimentary formation reconstructions. Furthermore, limited by the resolution of well logging tools, the observed well log measurements may record a spatial convolution of the true physical properties of the subsurface sediments. Each recorded value of logging curves is a weighted summation of the adjacent sediments' logging responses (Lindberg et al., 2015). Hence there may also be spatial coupling between the observations. Traditional lithology classifiers in both the Bayesian and the deep learning framework treat each sample as a spatial element-wise independent event. Fortunately, advanced models can take the spatial dependency into account which is expected to improve lithology classification.

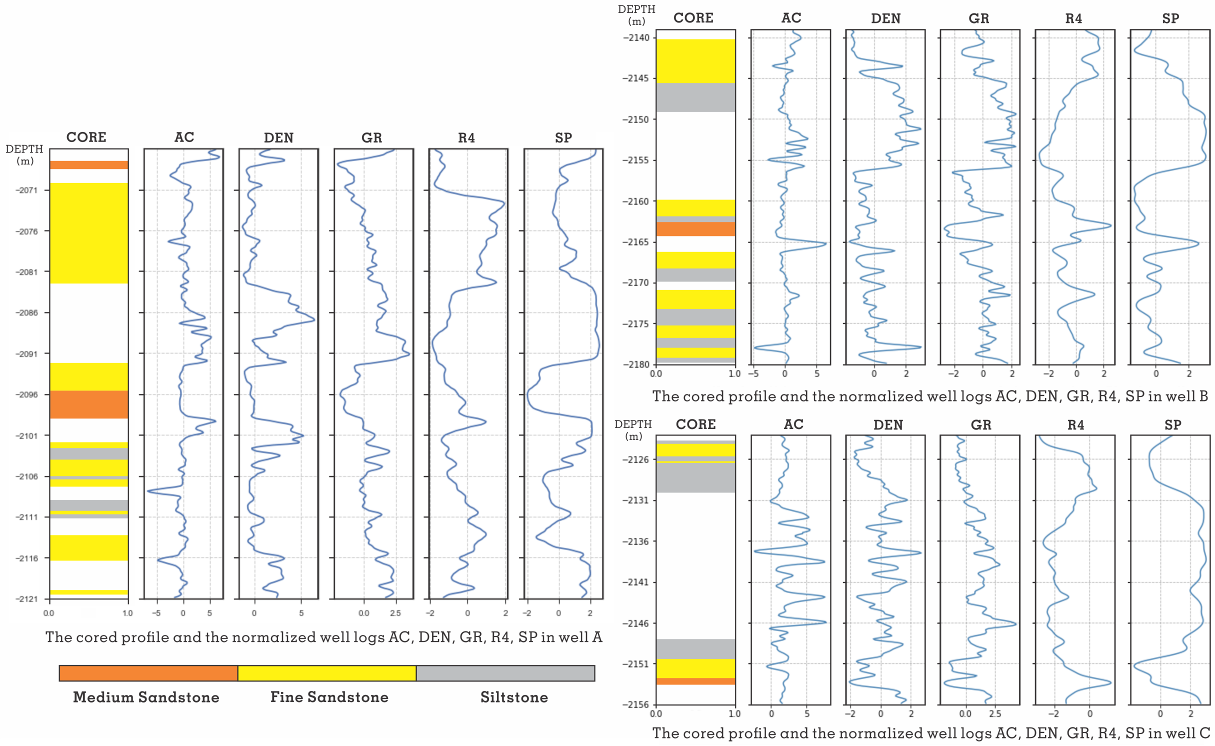
A first order hidden Markov (HM) model from the Bayesian setting and a recurrent neural network (RNN) classifier named bidirectional gated recurrent unit (GRU) model from the deep learning framework are discussed in this study. The HM model is frequently used in statistical analysis. For the lithology classification, the prior model with the Markov chain assumption is used to capture the vertical spatial dependency between classes (Eidsvik et al., 2004; Lindberg and Omre, 2014). The RNN model is specially designed for sequential data processing (Cleeremans et al., 1989; Schuster and Paliwal, 1997). An advanced version of RNN, the bidirectional GRU model, is implemented in this study as a classifier. The bidirectional

GRU model captures the hidden information of the observations in a neighborhood of the current sample (Schuster and Paliwal, 1997; Cho et al., 2014; Chung et al., 2014). Then the model is used for lithology classification.

In Section 2, we describe the lithology classification challenge and present the data in our case study. We also introduce suitble notation for the study. In Section 3, we introduce and discuss three classifiers including the DNN model, the bidirectional GRU model and the HM model. In Section 4, we compare the performance of the models based on the case study. In Section 5, we display the cross-validation results for three wells and discuss the results. Finally in Section 6, the conclusion and recommendation for further work are presented.

## 2. Problem definition, case study and notation

The clastic sediments of our study area are created in a beach depositional environment. Hence the predominant deposition is widespread fine-grained sediments. The lateral continuity of the fine-grained sedimentary layers ensure high correlation between wells. However, for sediments generated in fluvial system, lithology identification of beach and bar sedimentary bodies are a laborious task. The rarely changed grain size limits the discrimination of well log responses for various matrix. Low porosity and permeability restrain the effects of formation fluids on logging measurements. Therefore the categorical information is hidden in the multidimensional data space. Furthermore, the facies along a vertical 1D profile through the subsurface layers should be considered as a spatial inversion problem because of interdependence between sediments and convolution properties of well log data. In this case study we consider the lithology sequence along vertical wells in which well-log observations are available. The sedimentary lithologies are classified as medium sandstone (MS), fine sandstone (FS) and siltstone (SS). In the wells, the lithologies are partly observed in core samples, see Fig.1. Note that a large proportion of the well profile lacks core samples. Five normalized logging curves from three wells are available for us. These curves are acoustic log (AC), density log (DEN), gamma ray (GR), log-deep resistivity (R4) and spontaneous potential (SP). Pairwise scatterplots of the well logs sorted by lithologies are displayed in Fig.2. Although the MS layers may be identified by using primarily the GR log, there appear large overlaps between most of the log responses for varying lithologies. The 1D profile along the well path is discretized to $\mathcal{T} = \{1, \ldots, T\}$. At each $t \in \mathcal{T}$, a observation vector $\mathbf{d}_t = (d_{t,1}, \ldots, d_{t,5})$ is provided by five well logs with $\mathbf{d} = \{\mathbf{d}_t; t = 1, \ldots, T\}$ being the complete set of observations. At each depth t, we assign one of the three lithologies $\kappa_t \in \Omega_\kappa : \{MS, FS, SS\}$. The objective of our study is to assess the full lithology profile represented by the vector $\boldsymbol{\kappa} : \{\kappa_t; t = 1, \ldots, T\}$ given the observations $\mathbf{d}$, i.e. $[\boldsymbol{\kappa}|\mathbf{d}]$. The lithology variables of interest with associated well log observations are displayed in Fig.5a, and we need to connect these variables by a directed graph pointing from $\mathbf{d}$ to $\boldsymbol{\kappa}$ in order to make the lithology classification.



The cored profile and the normalized well logs AC, DEN, GR, R4, SP in well A

The cored profile and the normalized well logs AC, DEN, GR, R4, SP in well B

The cored profile and the normalized well logs AC, DEN, GR, R4, SP in well C

Figure 1: Normalized well logs AC, DEN, GR, R4, SP and the cored profiles in wells

## 3. Methodology

We discuss three classification models: the traditional DNN model, the bidirectional GRU model and the HM model. The former two are neural network models which are typical artificial intelligence classifiers while the latter one is cast in a statistical Bayesian framework. Both the HM and the GRU models capture the spatial vertical dependency in the problem, while the traditional DNN model ignores the spatial dependency.

### *3.1. The Traditional Deep Neural Network*

As mentioned above, the traditional DNN classifier, which is found to be superior to many other simple machine learning methods such as decision tree and k-means, provides us with the benchmark results. DNN model assumes the observations are spatial elementwise independent. Hence it does not take any vertical spatial dependency into account. The observations are directly processed through the DNN-layers $l_d = 1, \dots, L_d$ as

$$\mathbf{z}_t^{l_d} = f_{Relu}(\mathbf{\Gamma}_z^{l_d}\mathbf{z}_t^{l_d-1} + \boldsymbol{\beta}_z^{l_d})$$

with initial condition $\mathbf{z}_t^0 = \mathbf{d}_t$. The $\mathbf{z}_z^{l_d}$ is a $n_{l_d}$-vector with $n_0 = n_d$. The unknown model parameters are $\mathbf{\Gamma}_z^{l_d}$ being a $(n_{l_d} \times n_{l_d-1})$-matrix and $\boldsymbol{\beta}_z^{l_d}$ being a $n_{l_d}$-vector. The dimension at the last layer $n_{L_d}$ must be

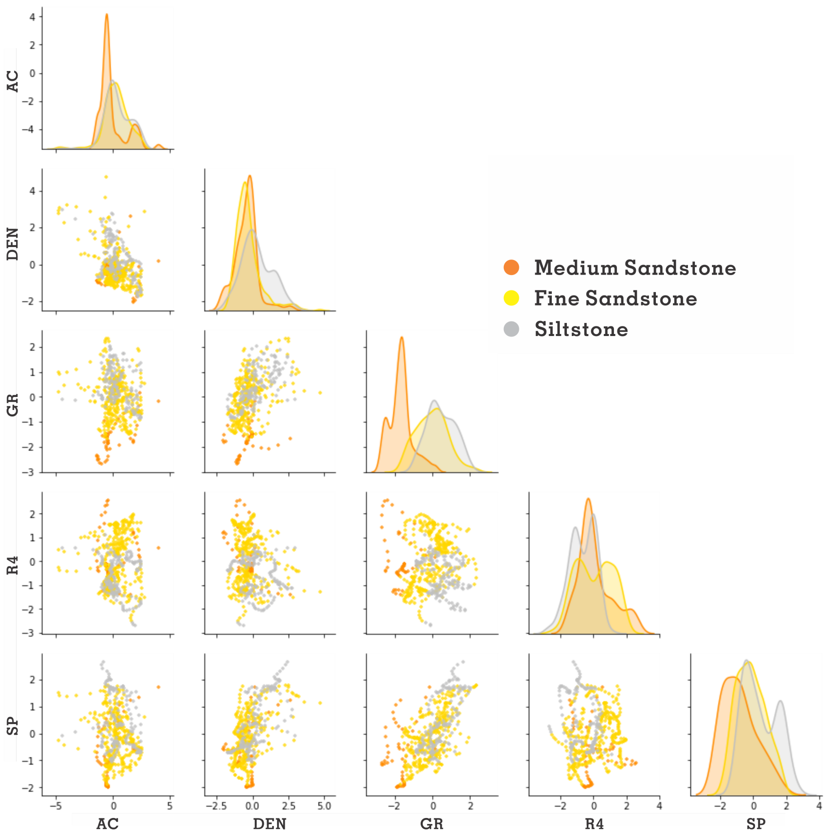


Figure 2: Pairwise scatterplots and histograms of all the well logs sorted by lithologies

three corresponding to $\Omega_\kappa : \{MS, FS, SS\}$. Hence the result $\mathbf{z}_t = \mathbf{z}_t^{L_d} = [z_{t,\kappa_t}]_{\kappa_t \in \Omega_\kappa}$ is just based on the corresponding measurements $\mathbf{d}_t$, see Fig.3.

The marginal probabilities for $\kappa_t \in \Omega_\kappa$ are then defined as

$$\begin{aligned} p(\kappa_t|\mathbf{d}) &= p(\kappa_t|\mathbf{d}_t) = p(\kappa_t|\mathbf{z}_t) \\ &= \frac{\exp z_{t,\kappa_t}}{\sum_{\kappa_t \in \Omega_\kappa} \exp z_{t,\kappa_t}} \end{aligned}$$

and a marginal maximum posterior (MMAP) criterion is used in the lithology profile prediction

$$\check{\boldsymbol{\kappa}}_{MMAP} = \{\check{\kappa}_t = \underset{\kappa_t}{\operatorname{argmax}}\{p(\kappa_t|\mathbf{d}_t)\}; t \in \mathcal{T}\}$$

and we use $p(\kappa_t|\mathbf{d}_t); \kappa_t \in \Omega_\kappa; t \in \mathcal{T}$ for uncertainty quantification. Note that the neural network model only provide marginal probabilities, hence no general statements about $\boldsymbol{\kappa}$ given $\mathbf{d}$ can be quantified probabilistically. Moreover, realizations from $[\boldsymbol{\kappa}|\mathbf{d}]$ cannot be generated. The unknown model parameters denoted by matrices $\boldsymbol{\Gamma}_:^:$ and vectors $\boldsymbol{\beta}_:^:$ are estimated from a training set of wells $[\boldsymbol{\kappa}^o, \mathbf{d}^o]_i; i = 1, \ldots, n_T$. We use the matching criterion for each well,

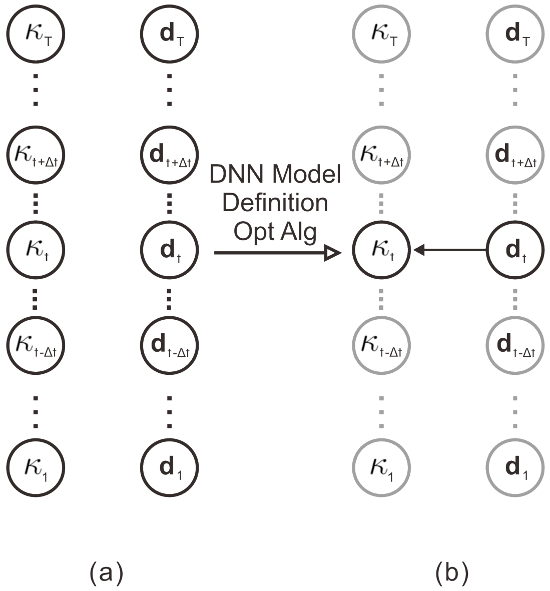


Figure 3: Dependence structure of the DNN; (a) the lithologies profile with the corresponding observations; (b) the posterior marginal distribution from DNN.

$$\mathcal{L}(\breve{\boldsymbol{\kappa}}_{MMAP}, \boldsymbol{\kappa}^o; \boldsymbol{\Gamma}^{\cdot}_{\cdot}, \boldsymbol{\beta}^{\cdot}_{\cdot}) = \sum_{t\in\mathcal{T}^o} I(\breve{\kappa}_{MMAP,t} = \kappa^o_t)$$

and define the estimator as

$$[\hat{\boldsymbol{\Gamma}}^{\cdot}_{\cdot}, \hat{\boldsymbol{\beta}}^{\cdot}_{\cdot}] = \underset{\Gamma^{\cdot}_{\cdot},\beta^{\cdot}_{\cdot}}{\operatorname{argmax}}\{\mathcal{L}(\breve{\boldsymbol{\kappa}}_{MMAP}, \boldsymbol{\kappa}^o; \boldsymbol{\Gamma}^{\cdot}_{\cdot}, \boldsymbol{\beta}^{\cdot}_{\cdot})\}$$

The actual optimization is made by the advanced gradient-based optimization method back-propagation through time (BPTT) with a fixed loss-function, and it is demanding computationally. Note that the model parameters have no intuitive interpretation and hence we have to rely fully on the algorithm to identify a optimum solution.

### *3.2. Bidirectional Gated Recurrent Unit Neural Network*

In recent decade, a lot of attention has been paid to data-driven methods represented by deep learning. Deep neural network methods are believed to automatically extract the hidden information in data concerning the variable of interest. Bidirectional GRU provides a memory unit with several gates to integrate the input data at current and nearby depths. The GRU system provides an artificial vector at each depth t called the recurrent hidden $n_a$-vector state, $\mathbf{a}_t = (a_{t,1}, \ldots, a_{t,n_a})$ by summarizing the observations $\mathbf{d}_t^{\Delta_t} = \{\mathbf{d}_t; t = t-\Delta_t, \ldots, t, \ldots, t+\Delta_t\}$ to capture the vertical spatial dependency in a $2\Delta_t+1$ neighborhood. Thereafter a general DNN classifier is activated on these artificial vectors hence the classifier is $\widetilde{\kappa}_t = g(\mathbf{a}_t)$, see Fig.4. The possible interaction in the lithology vector $\boldsymbol{\kappa}$ is not captured by the model, hence available geological and exploration experiences are ignored. The normalized results from the GRU model may be interpreted as marginal posterior pdfs $\mathbf{p}_{t|\mathbf{d}_t^{\Delta_t}} = [p(\kappa_t|\mathbf{d}_t^{\Delta_t})]_{\kappa_t\in\Omega_\kappa}; t \in \mathcal{T}$.

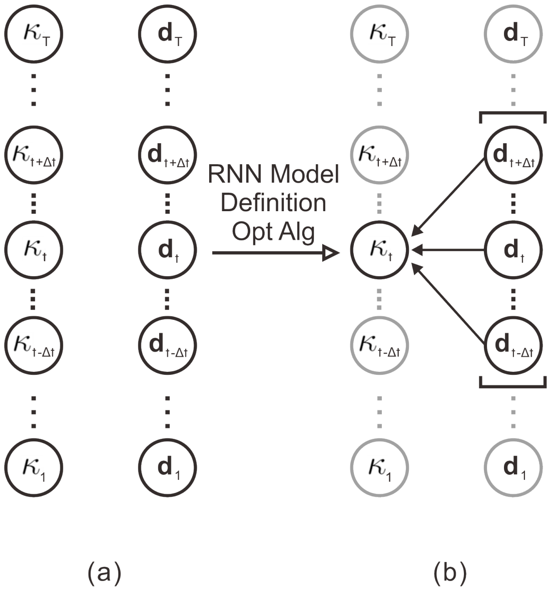


Figure 4: Dependence structure of the RNN; (a) the lithologies profile with the corresponding observations; (b) the posterior marginal distribution from RNN.

The bidirectional GRU model capture the information in the well logs in a neighborhood around t, represented by $\mathbf{d}_t^{\Delta_t}$. The model iterates over $t \in \mathcal{T}$ and for each GRU-layer $l = \{1, \ldots, L_g\}$ of dimension $n_l$. It sweeps $\Delta_t$ both from above to t and below to t, and generates the recurrent hidden state for each depth $t$ at the current hidden layer $l$, i.e. $\mathbf{a}_t^l$. The upward sweep of $\mathbf{a}_t^l$ is made recursively for $s = t - \Delta_t, \ldots, t$ as follows

$$\mathbf{a}_c = \begin{bmatrix} \dot{\mathbf{a}}_{s-1}^l \\ \mathbf{a}_s^{l-1} \end{bmatrix}$$

$$\boldsymbol{\alpha}_f^l = f_{sigmoid}(\boldsymbol{\Gamma}_f^l \mathbf{a}_c + \boldsymbol{\beta}_f^l)$$

$$\mathbf{c}_s^l = f_{tanh}(\boldsymbol{\Gamma}_c^l \left[\begin{bmatrix} \boldsymbol{\alpha}_f^l \\ \mathbf{i}_{2n_{l-1}} \end{bmatrix} \otimes \mathbf{a}_c\right] + \boldsymbol{\beta}_c^l)$$

$$\boldsymbol{\alpha}_u^l = f_{sigmoid}(\boldsymbol{\Gamma}_u^l \mathbf{a}_c + \boldsymbol{\beta}_u^l)$$

$$\dot{\mathbf{a}}_s^l = \boldsymbol{\alpha}_u^l \otimes \mathbf{c}_s^l + [\mathbf{i}_{n_l} - \boldsymbol{\alpha}_u^l] \otimes \mathbf{a}_{s-1}^l$$

The downward sweep of $\mathbf{a}_t^l$ is made in a similar way in the opposite direction for $s' = t + \Delta_t, \ldots, t$. The initial conditions are $\mathbf{a}_s^0 = \mathbf{d}_s$, $\mathbf{a}_{s'}^0 = \mathbf{d}_{s'}$ and $\dot{\mathbf{a}}_{t-\Delta_t-1}^l = \dot{\mathbf{a}}_{t+\Delta_t+1}^l = 0\mathbf{i}_{n_l}$. Hence $\mathbf{a}_c$ is a $n_l + 2n_{l-1}$-vector for $l = 2, \ldots, L_g$ and a $n_l + n_d$-vector for $l = 1$. $\boldsymbol{\alpha}_f^l$, $\boldsymbol{\alpha}_u^l$, $\mathbf{c}_s^l$ and $\dot{\mathbf{a}}_s^l$ are $n_l$-vectors. The unknown model parameters $\mathbf{\Gamma}^l_f$, $\mathbf{\Gamma}^l_c$ and $\mathbf{\Gamma}^l_u$ are $(n_l \times (n_l + 2n_{l-1}))$-matrices for $l = 2, \dots, L_g$ and $(n_l \times (n_l + n_d))$-matrices for $l = 1$, while $\boldsymbol{\beta}^l_f$, $\boldsymbol{\beta}^l_c$ and $\boldsymbol{\beta}^l_u$ are all $n_l$-vectors. The transfer functions $f_{sigmoid}(\cdot)$ and $f_{tanh}(\cdot)$ are activation functions which improve the nonlinear regression performance. More details of the upward and downward sweep are presented in Appendix A.

We set the final results of the sweeps as $\vec{\mathbf{a}}^l_t = \dot{\mathbf{a}}^l_{s=t}$, $\overleftarrow{\mathbf{a}}^l_t = \dot{\mathbf{a}}^l_{s'=t}$ and then stack them to get the hidden state for $t$ at $l^{th}$ layer

$$\mathbf{a}^l_t = \begin{bmatrix} \vec{\mathbf{a}}^l_t \\ \overleftarrow{\mathbf{a}}^l_t \end{bmatrix}$$

Hence the $\mathbf{a}^l_t$ is a $2n_l$-vector. The upward recursion at $t$ though $s$ after the last GRU-layer is performed to obtain the $n_{L_g}$-vector $\vec{\mathbf{a}}^{L_g}_t$. Similarly the downward recursion at $t$ though $s'$ after the last GRU-layer provides the $n_{L_g}$-vector $\overleftarrow{\mathbf{a}}^{L_g}_t$. The stacked $2n_{L_g}$-vector $\mathbf{a}^{L_g}_t = [\vec{\mathbf{a}}^{L_g}_t, \overleftarrow{\mathbf{a}}^{L_g}_t]^T$, which we believe has captured the spatial coupling in observations, is then processed by the following traditional DNN classifier through the DNN-layers $l_d = 1, \dots, L_d$ of dimension $n_{l_d}$ as

$$\mathbf{z}^{l_d}_t = f_{Relu}(\mathbf{\Gamma}^{l_d}_z \mathbf{z}^{l_d-1}_t + \boldsymbol{\beta}^{l_d}_z)$$

with initial condition $\mathbf{z}^0_t = \mathbf{a}^{L_g}_t$. The $\mathbf{z}^{l_d}_z$ is a $n_{l_d}$-vector with $n_0 = 2n_{L_g}$. The rest of the parameters are identical to the ones in the traditional DNN classifier defined in Section 3.1. The recursion of GRU model provides the $n_{L_d}$-vectors $\mathbf{z}_t = \mathbf{z}^{L_d}_t = [z_{t,\kappa_t}]_{\kappa_t \in \Omega_\kappa}$ which are dependent on the observation vector $\mathbf{d}^{\Delta_t}_t$ through the initiation at $\mathbf{a}^{L_g}_t$.

We define the corresponding marginal probabilities for $\kappa_t \in \Omega_\kappa$ as

$$\begin{aligned} p(\kappa_t|\mathbf{d}) &= p(\kappa_t|\mathbf{d}^{\Delta_t}_t) = p(\kappa_t|\mathbf{z}_t) \\ &= \frac{\exp z_{t,\kappa_t}}{\sum_{\kappa_t \in \Omega_\kappa} \exp z_{t,\kappa_t}} \end{aligned}$$

Lastly we use a MMAP criterion to predict the lithology class

$$\widetilde{\boldsymbol{\kappa}}_{MMAP} = \{\widetilde{\kappa}_t = \underset{\kappa_t}{\operatorname{argmax}}\{p(\kappa_t|\mathbf{d}^{\Delta_t}_t)\}; t \in \mathcal{T}\}$$

Here the uncertainty quantification is based on $p(\kappa_t|\mathbf{d}^{\Delta_t}_t); \kappa_t \in \Omega_\kappa; t \in \mathcal{T}$. The matching criterion and model parameter estimator used in Section 3.1 are also used here. Furthermore, the gradient-based optimizer and loss-function for parameters estimation are identical to the ones used for the traditional DNN.

### *3.3. The Hidden Markov Model*

According to Bayes' rule, the posterior model is provided by

$$p(\boldsymbol{\kappa}|\mathbf{d}) = \frac{1}{p(\mathbf{d})} \times p(\mathbf{d}|\boldsymbol{\kappa})p(\boldsymbol{\kappa})$$

where the likelihood model $p(\mathbf{d}|\boldsymbol{\kappa})$ defines the procedure of well logs data collection, the prior model $p(\boldsymbol{\kappa})$ represents the geological and exploration experience with the variable of interest $\boldsymbol{\kappa}$, and $p(\mathbf{d})$ in the equation is a normalizing constant. In general high-dimensional models the normalizing constant is computationally demanding to assess, but under certain model assumptions efficient recursive algorithms can be defined.

The likelihood model $p(\mathbf{d}|\boldsymbol{\kappa})$ links the observations to the lithology classes, which are in focus here. We assume that the likelihood model is conditional independent with single-site response, and can be expressed as

$$p(\mathbf{d}|\boldsymbol{\kappa}) = \prod_t p(\mathbf{d}_t|\boldsymbol{\kappa}) = \prod_t p(\mathbf{d}_t|\kappa_t)$$

Hence the likelihood function is defined by the location-wise likelihood functions

$$p(\mathbf{d}_t|\kappa_t); \quad \kappa_t \in \Omega_\kappa; t \in \mathcal{T}$$

which we assume to be stationary, hence independent of $t$.

The prior model, represents our knowledge about the geological setting and sediments. A Markov chain with the first-order Markov property is chosen to represent the spatial coupling of the lithologies. It can be defined as

$$p(\boldsymbol{\kappa}) = p(\kappa_1) \prod_{t \in \mathcal{T}_{-1}} p(\kappa_t|\kappa_{t-1}, \dots, \kappa_1) = p(\kappa_1) \prod_{t \in \mathcal{T}_{-1}} p(\kappa_t|\kappa_{t-1})$$

The initial pdf $\mathbf{p}_1$ and the set of transition matrices $\mathbf{P}_{t-1,t}; t \in \mathcal{T}_{-1}$ is defined as

$$\mathbf{p}_1 = [p(\kappa_1)]_{\kappa_1 \in \Omega_\kappa}$$

$$\mathbf{P}_{t-1,t} = [p(\kappa_t|\kappa_{t-1})]_{\kappa_{t-1},\kappa_t \in \Omega_\kappa}; \quad t \in \mathcal{T}_{-1}$$

and we assume the transition matrices to be stationary, hence independent of $t$. Moreover the initial pdf is defined as the stationary pdf of this transition matrix. The set of marginal prior pdfs $\mathbf{p}_t; t \in \mathcal{T}$ can be calculated by

$$\mathbf{p}_t = [p(\kappa_t)]_{\kappa_t \in \Omega_\kappa} = \mathbf{P}'_{t-1,t}\mathbf{p}_{t-1}; \quad t \in \mathcal{T}_{-1}$$

Due to the stationarity assumptions these marginals will be identical. The HM model is displayed in the directed graph Fig.5b. The prior and likelihood models given above will fully define the posterior model. The resulting posterior pdf is

$$
\begin{aligned}
p(\boldsymbol{\kappa}|\mathbf{d}) &= \frac{1}{p(\mathbf{d})} \times \prod_{t\in\mathcal{T}} p(\mathbf{d}_t|\kappa_t) \times p(\kappa_1) \prod_{t\in\mathcal{T}_{-1}} p(\kappa_t|\kappa_{t-1}) \\
&= \frac{1}{p(\mathbf{d})} \times p(\mathbf{d}_1|\kappa_1)p(\kappa_1) \times \prod_{t\in\mathcal{T}_{-1}} p(\mathbf{d}_t|\kappa_t)p(\kappa_t|\kappa_{t-1}) \\
&= p(\kappa_1|\mathbf{d}) \prod_{t\in\mathcal{T}_{-1}} p(\kappa_t|\kappa_{t-1}, \mathbf{d}_{t:T})
\end{aligned}
$$

The latter identity, which is varying through depth, entails that the posterior model is a non-stationary first-order Markov chain (Moja et al., 2018), see Fig.5c. From Fig.5b it is easy to interpret this posterior model. Focus on $\kappa_t$, and conditioning on $\kappa_{t-1}$ cuts off the lower part of the graph and hence only dependence on $\mathbf{d}_{t:T}$ remains. The recursive reverse algorithm, see Baum et al. (1970); Scott (2002); Moja et al. (2018), can efficiently provide the initial pdf and the transition matrices

$$
\begin{aligned}
\mathbf{p}_{1|\mathbf{d}} &= [p(\kappa_1|\mathbf{d})]_{\kappa_1\in\Omega_\kappa} \\
\mathbf{P}_{t-1,t|\mathbf{d}} &= [p(\kappa_t|\kappa_{t-1}, \mathbf{d}_{t:T})]_{\kappa_{t-1},\kappa_t\in\Omega_\kappa}; \quad t \in \mathcal{T}_{-1}
\end{aligned}
$$

The marginal posterior pdfs can be calculated as

$$
\mathbf{p}_{t|\mathbf{d}} = [p(\kappa_t|\mathbf{d})]_{\kappa_t\in\Omega_\kappa} = \mathbf{P}'_{t-1,t|\mathbf{d}}\mathbf{p}_{t-1|\mathbf{d}}; \quad t \in \mathcal{T}_{-1}
$$

and they will all depend on the full set of observations $\mathbf{d}$, see Fig.5d, since they are recursively obtained starting with $\mathbf{p}_{1|\mathbf{d}}$ which depends on $\mathbf{d}$. Hence the HM model has captured the spatial coupling in the variable $\boldsymbol{\kappa}$ and in the observations $\mathbf{d}$. From the posterior model we can generate posterior realizations rapidly according to the initial posterior pdf and the set of posterior transition matrices. Moreover, the recursive Viterbi algorithm provides the gobal maximum posterior (MAP) prediction (Viterbi, 1967),

$$
\hat{\boldsymbol{\kappa}}_{MAP} = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\{p(\boldsymbol{\kappa}|\mathbf{d})\}
$$

Lastly, the posterior probability profiles for a given $\kappa$ can be presented as

$$
\mathbf{p}_\kappa = [p(\kappa_t = \kappa|\mathbf{d})]_{t\in\mathcal{T}}; \quad \kappa \in \Omega_\kappa
$$

and we quantify the prediction uncertainty by these probability profiles. An alternative naive prediction can be defined as the MMAP predictor

$$
\hat{\boldsymbol{\kappa}}_{MMAP} : \{\hat{\kappa}_t = \underset{\kappa_t}{\operatorname{argmax}}\{\mathbf{p}_\kappa(\kappa_t|\mathbf{d})\}; \quad t = 1, \ldots, T\}
$$

where the maximizations are made marginally.

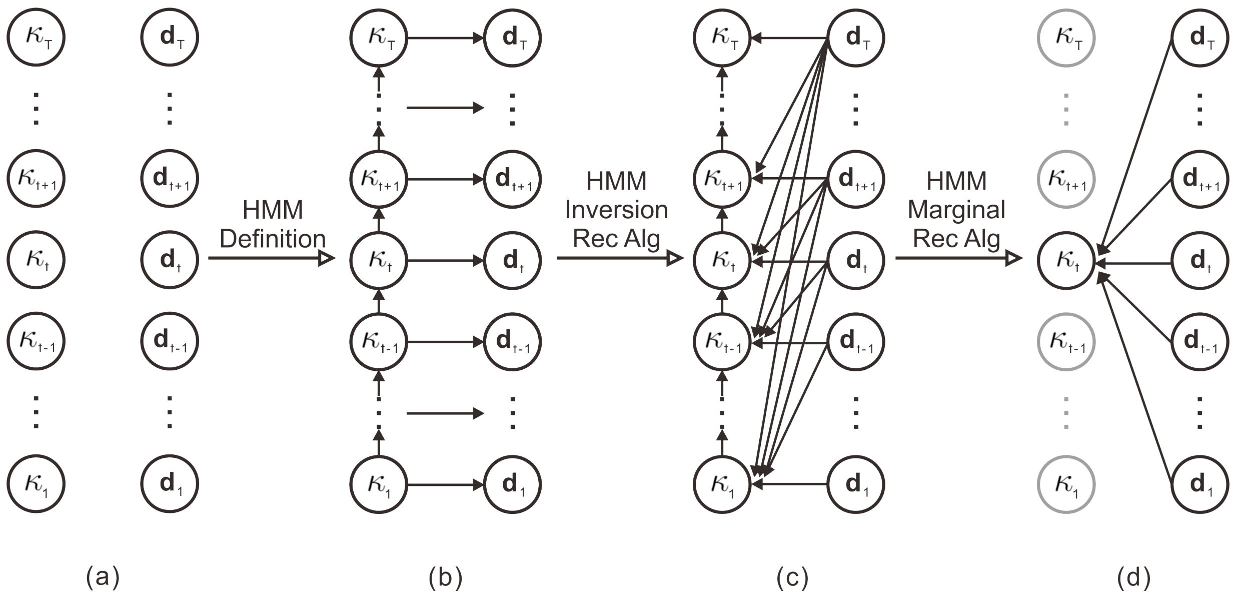


Figure 5: Dependence structure of the hidden Markov model; (a) the lithologies profile with the corresponding observations; (b) the definition of hidden Markov model; (c) the posterior Markov chain; (d) the posterior marginal distribution from HMM.

This HM model is completely probabilistically defined, and the posterior model is assessable by very fast recursive algorithms. The advantage of having the posterior model available is that the probability of any statement involving $\boldsymbol{\kappa}$ given $\mathbf{d}$ can be calculated. Example of relevant questions are:

- at depth t, what is the probability of having an unbroken string of three for a given lithology $\kappa'$ upwards?
- in the profile, what is the probability of having more than five transitions between two given lithologies $\kappa'$ and $\kappa''$?

Note also that the model can be extended to capture convolved observations, i.e. likelihood models being conditional independent with multi-site response. Also higher-order Markov chain prior models can be evaluated. These extensions, see Lindberg et al. (2015), define posterior models which also can be assessed recursively, although with considerable larger computational demands.

We use a kernel estimator to estimate $p(\mathbf{d}_t|\kappa_t)$, see Moja et al. (2018). Denote the observed well logs of cores of class $\kappa$ as $\mathbf{d}^\kappa = (\mathbf{d}_1^\kappa, \ldots, \mathbf{d}_{n_\kappa}^\kappa)$, see Fig.2, then the estimator is,

$$\hat{p}_k(\mathbf{d}_t|\kappa) = \frac{1}{n_\kappa h_\kappa} \sum_{i=1}^{n_\kappa} k(\frac{\mathbf{d}_t - \mathbf{d}_i^\kappa}{h_\kappa}); \quad \kappa \in \Omega_\kappa$$

where $k(\tau)$; $\tau \in \Re^n$ is the kernel function and $h_\kappa$ is the band width which defines the smoothness of the density distribution. The kernel function applied here is Gaussian kernel function,

$$k_G(\boldsymbol{\tau}) = \varphi_n(\boldsymbol{\tau}; \mathbf{0}_n, \hat{\boldsymbol{\Sigma}}_\kappa); \quad \boldsymbol{\tau} \in \mathcal{R}^n$$

with a zero $n$-vector $\mathbf{0}_n$ as the expectation of each class of observations $\mathbf{d}^\kappa$ and a $(n \times n)$-matrix $\hat{\boldsymbol{\Sigma}}_\kappa$ as the empirical covariance matrix. A cross validation psuedo-likelihood (CVL) method provides the optimum band width $h_\kappa$ as

$$CVL(h_\kappa) = \sum_{i=1}^{n_\kappa} \hat{p}_{k_{(-i)}}(\mathbf{d}_i^\kappa|\kappa)$$

where $\hat{p}_{k_{(-i)}}(\cdot)$ is the kernel estimator based on the observations $\mathbf{d}^\kappa = (\mathbf{d}_1^\kappa, \ldots, \mathbf{d}_{i-1}^\kappa, \mathbf{d}_{i+1}^\kappa, \ldots, \mathbf{d}_{n_\kappa}^\kappa)$. Then the estimator for the band width can be chosen as

$$\hat{h}_\kappa = \underset{h_\kappa}{\operatorname{argmax}}\{\log CVL(h_\kappa)\}$$

We use a naive counting estimator to provide the transition probabilities $p(\kappa_t|\kappa_{t-1})$, and denote the observed lithology classes $\boldsymbol{\kappa}^o = [\kappa_1^o, \ldots, \kappa_T^o]$, then

$$\hat{p}(\kappa_t|\kappa_{t-1}) = \frac{1}{T-1} \sum_{i \in \mathcal{T}_{-1}} I(\kappa_i^o = \kappa_t \cap \kappa_{i-1}^o = \kappa_{t-1}); \quad \kappa_{t-1}, \kappa_t \in \Omega_\kappa$$

The actual estimator is adjusted for missing observations. The associated initial pdf $\mathbf{p}_1$ is defined by the stationary pdf obtained by the expression

$$\mathbf{p}_1 = \hat{\mathbf{P}}'_{t-1,t}\mathbf{p}_1$$

The estimators for the model parameters $p(\mathbf{d}_t|\kappa_t); \kappa_t \in \Omega_\kappa$ and $p(\kappa_t|\kappa_{t-1}); \kappa_{t-1}, \kappa_t \in \Omega_\kappa$ can be justified intuitively from the interpretation of the parameters, and they are extremely fast to assess.

## 4. Model comparison

We use the three wells displayed in Fig.1 for the model comparison. We do cross-validation by removing one well at the time and estimate model parameters based on the two others. Then we make lithology classification $\boldsymbol{\kappa}^*$ with associated probability profiles $\mathbf{p}_{\boldsymbol{\kappa}} = [p(\kappa_t = \kappa|\mathbf{d})]_{t \in \mathcal{T}}$ for $\kappa \in \Omega_\kappa$ for the removed well. Hence we got three cross-validated wells, where the truth is denoted $\boldsymbol{\kappa}^o$. We propose three coefficients for the performance evaluation of the DNN, GRU and HM classifiers.

The results from the cross-validation of wells are presented as follows:

- visual inspection of truth $\boldsymbol{\kappa}^o$ with classification $\boldsymbol{\kappa}^*$ and probability profiles $\mathbf{p}_\kappa$ for $\kappa \in \Omega_\kappa$, for each of the models.

- classification match score $CS_\kappa$ of classification for each $\kappa \in \Omega_\kappa$

$$CS_\kappa = \frac{\sum_{t \in \mathcal{T}} \mathbf{I}(\kappa_t^* = \kappa_t^o = \kappa)}{\sum_{t \in \mathcal{T}} \mathbf{I}(\kappa_t^o = \kappa)}$$

  where $CS_\kappa \in [0,1]$, with $CS_\kappa = 0$ for complete mismatch and $MS_\kappa = 1$ for full match.

- probability score $PS_\kappa$ of probability profiles for each $\kappa \in \Omega_\kappa$

$$PS_\kappa = \frac{\sum_{t \in \mathcal{T}} \mathbf{I}(\kappa_t^o = \kappa) p(\kappa_t^* = \kappa | \mathbf{d})}{\sum_{t \in \mathcal{T}} \mathbf{I}(\kappa_t^o = \kappa)}$$

  where $PS_\kappa \in [0,1]$, with $PS_\kappa = 0$ for complete probability mismatch and $PS_\kappa = 1$ for full match.

- overfit loss $OL_\kappa$ for each $\kappa \in \Omega_\kappa$

$$OL_\kappa = \frac{CS_\kappa^{tr} - CS_\kappa^{cv}}{CS_\kappa^{tr}}$$

  where $CS_\kappa^{cv}$ is the CS for the cross validation well while $CS_\kappa^{tr}$ is the CS for training wells. Note $OL_\kappa \in [0,1]$ where $OL_\kappa = 0$ entails no overfit while $OL_\kappa = 1$ entails dramatic overfit.

## 5. Results and Discussion

The model parameters of each model are estimated by the two training wells in each cross-validation study. The DNN classifier is based on a 64-128-64-32-16-3 general hidden layer model, which does not take spatial dependency into account. The large number of parameters are estimated from the two training wells, and it requires about 2 hours on a regular Apple Mac Pro laptop. A GRU classifier with 64-128-64 memory unit layers followed by 32-16-3 general hidden layers is chosen. This model is based on a $\Delta_t = 2$ neighborhood hence including $\mathbf{d}_t^{\Delta_t = 2} : \{\mathbf{d}_t; t = t-2, \dots t+2\}$ in the classification at each depth $t$. The model parameters are also jointly estimated from the two training wells, and the computer demands are about 3.5 hours. Interpretation of the parameters from the DNN and GRU model is impossible. Hence the neural network based classifiers should be considered as a kind of black-box model. Lastly, the HM classifier is parameterized by a prior transition matrix between the lithologies and a likelihood Gaussian kernel. The transition matrices are assessed by counting estimators on the training wells, see Table.1, while the Gaussian kernel band width is estimated by cross-validation on the training wells, see Table.2. Note that in spite of a large proportion of missing core samples, the model parameter estimates in the HM classifier are fairly consistent. The estimation process is very efficient and the computing demand is less than 10 seconds. On the contrary with the DNN and GRU model, all the model parameters of HM classifier can be interpreted easily.

Table 1: The prior transition matrices and the corresponding stationary pdfs

| The training wells | The prior transition matrix | The stationary pdf |
|---|---|---|
| Well A & Well B | $\begin{array}{cc} & \begin{matrix} \text{MS} & \text{FS} & \text{SS} \end{matrix} \\ \begin{matrix} \text{MS} \\ \text{FS} \\ \text{SS} \end{matrix} & \begin{bmatrix} 0.9358 & 0.0600 & 0.0042 \\ 0.0068 & 0.9390 & 0.0542 \\ 0.0075 & 0.0625 & 0.9300 \end{bmatrix} \end{array}$ | $\begin{array}{cc} \begin{matrix} MS \\ FS \\ SS \end{matrix} & \begin{bmatrix} 0.0995 \\ 0.5041 \\ 0.3964 \end{bmatrix} \end{array}$ |
| Well A & Well C | $\begin{array}{cc} & \begin{matrix} \text{MS} & \text{FS} & \text{SS} \end{matrix} \\ \begin{matrix} \text{MS} \\ \text{FS} \\ \text{SS} \end{matrix} & \begin{bmatrix} 0.9489 & 0.0487 & 0.0024 \\ 0.0090 & 0.9640 & 0.0270 \\ 0.0012 & 0.0813 & 0.9175 \end{bmatrix} \end{array}$ | $\begin{array}{cc} \begin{matrix} MS \\ FS \\ SS \end{matrix} & \begin{bmatrix} 0.1210 \\ 0.6595 \\ 0.2195 \end{bmatrix} \end{array}$ |
| Well B & Well C | $\begin{array}{cc} & \begin{matrix} \text{MS} & \text{FS} & \text{SS} \end{matrix} \\ \begin{matrix} \text{MS} \\ \text{FS} \\ \text{SS} \end{matrix} & \begin{bmatrix} 0.9453 & 0.0498 & 0.0050 \\ 0.0068 & 0.9392 & 0.0541 \\ 0.0075 & 0.0522 & 0.9403 \end{bmatrix} \end{array}$ | $\begin{array}{cc} \begin{matrix} MS \\ FS \\ SS \end{matrix} & \begin{bmatrix} 0.1157 \\ 0.4616 \\ 0.4277 \end{bmatrix} \end{array}$ |

Table 2: The optimized band widths

| Lithology class<br>$\hat{h}_\kappa$ | MS | FS | SS |
|---|---|---|---|
| By using well A and B | 2.337 | 2.913 | 1.660 |
| By using well A and C | 1.360 | 2.930 | 1.500 |
| By using well B and C | 1.363 | 2.926 | 1.730 |

We separately trained these classifiers by using the labeled observations from two of the wells each time. The remaining well is taken as a blind test well to evaluate the performance of the classifiers. Hence three blind test results for each model are provided by our study.

Fig.6-8 display the blind test results for each of the three cross-validated wells. Each display contain from the left to right, the true profile $\boldsymbol{\kappa}^o$, DNN classification (MMAP) and probability profiles, GRU classification (MMAP) and probability profiles and two HM classifiers (MAP, MMAP), the probability profiles with four realizations.

Fig.6 displays the cross-validation results of well A. The DNN model provides random, frequent and thick SS layers with extremely thin MS layers next to them along the well profile. The GRU model provides more FS layers at the upper and middle part. However, SS layers with unreasonable thickness are also predicted. Both these neural network method miss the true MS layers which can be identified by core-plugs at the upper part. The HM model provides homogeneous prediction with thick FS layers. Wrong MS and SS layers are predicted in the bottom of the well. But the HM model is able to give us the correct prediction of the MS layer at depth 2068m and always provides the predicted MS and SS layers with reasonable thickness. Note that well A is the one with the most core-plugs, hence more samples are removed from the training set. All models are considered to provide unsatisfactory predictions in this cross-validation study.

Fig.7 displays the cross-validation results for well B. The DNN model predicts the bottom part as a thick SS layer with thin MS interlayers. Moreover, it arranges many layers with wrong depth for the remaining well profile. The GRU model supply a reliable reproduction of the lithology profile. It arranges scattered but wrongly located SS layers in the bottom of the profile. We also get a reliable reproduction by the HM model. The MMAP is closer to the truth than the corresponding MAP because it has larger prediction heterogeneity. The four realizations are of course even more heterogeneous prediction. According to the MMAP, the SS layer at the bottom of the well and the thinner MS layer in the middle are correctly predicted.

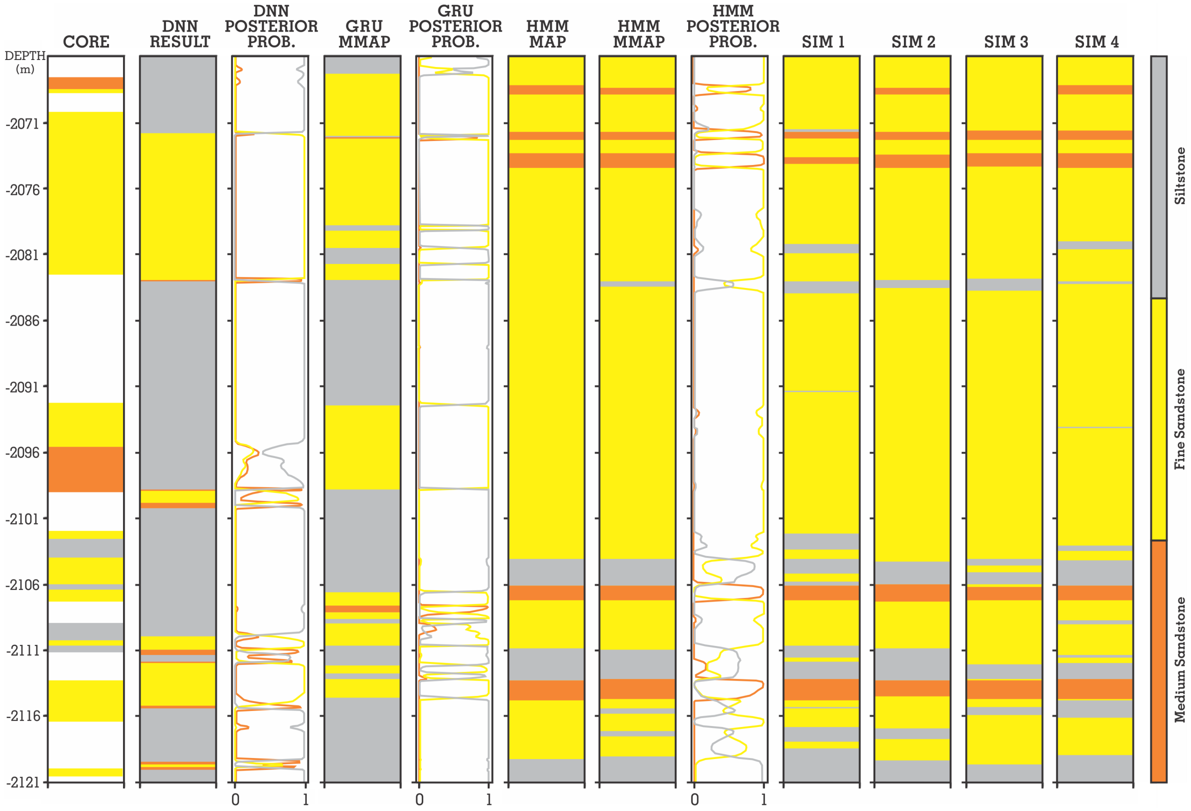


Figure 6: The cross-validation result for well A

Fig.8 displays the cross-validation results for well C. All models fail to predict the thick SS layers at the top of the well. The DNN model predicts more MS layers and frequent MS-SS transitions in the middle unknown part. This effect is not consistent with geological interpretations and observations in core plugs.

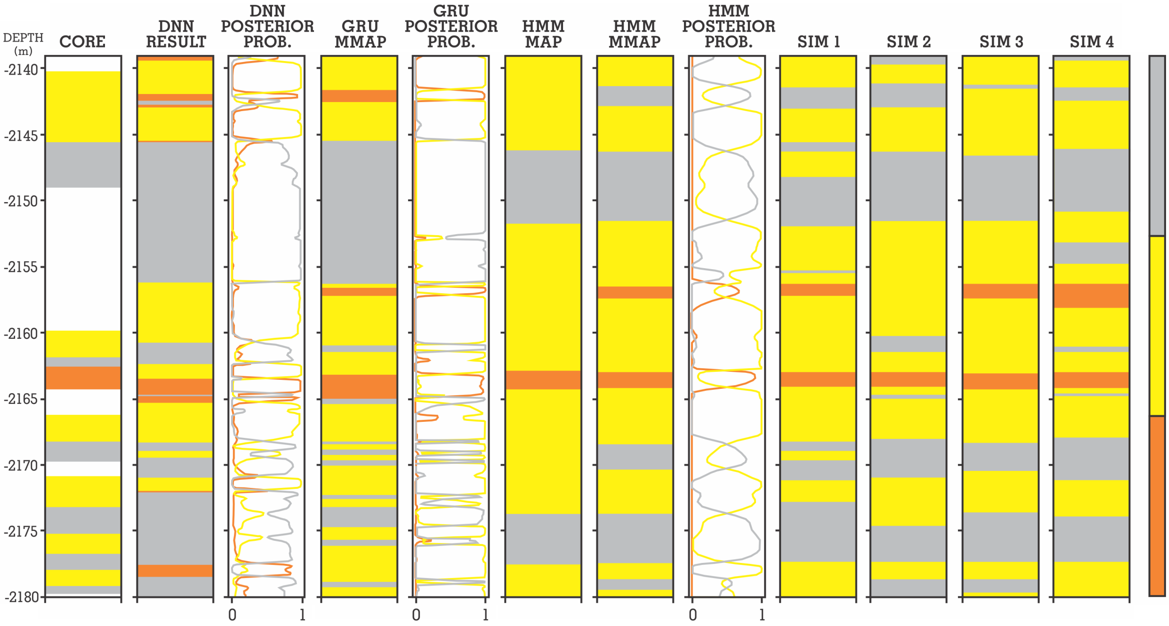


Figure 7: The cross-validation result for well B

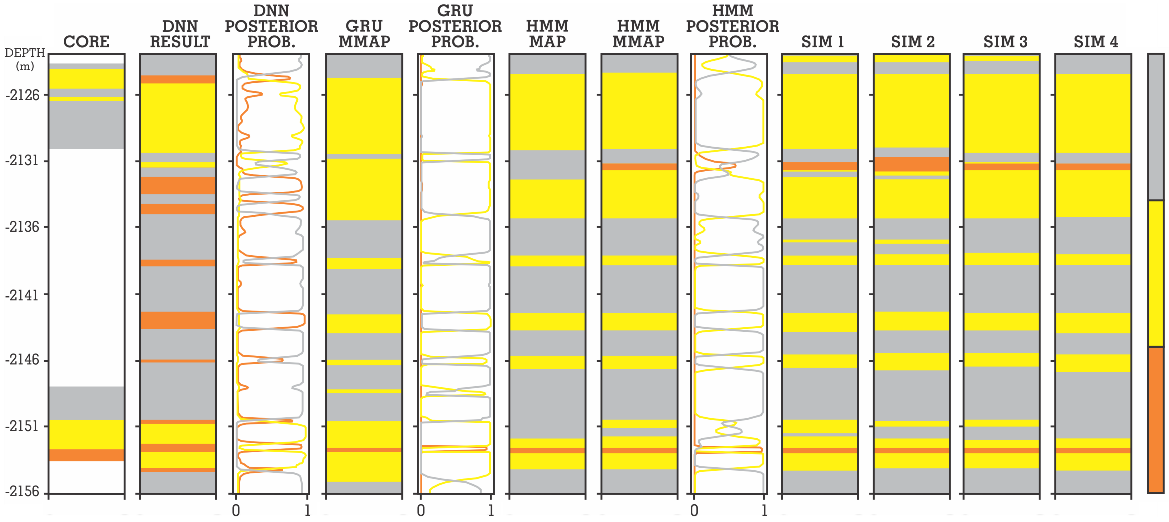


Figure 8: The cross-validation result for well C

The predictions from GRU and HM model are similar to each other. The proportion of SS increases with depth. At bottom of well both the GRU and HM models correctly arrange the MS interlayer around depth 2153m while the HM model predicts the thickness better.

The proposed evaluation coefficients provide us with quantitative information to compare the model performances. We calculate CS, PS and OL for each model, lithology and cross-validation. Then we take the average values as their finally scores, see Table 3. The models which take the vertical spatial dependency into account are superior in almost all scores except for lithology SS. The DNN model is superior for SS. The severe over-prediction of the thick SS layers by the DNN model causes this effect. For the models with spatial coupling, the HM model preforms better than the GRU model in all cases, also for lithology SS. Hence we consider the HM model to be the most general and robust lithology classifier in this study.

Table 3: Model Comparison by evaluation coefficients, with best values colored

| Coefficient | Lithology<br>Classifier | MS | FS | SS |
|---|---|---|---|---|
| CS | DNN model | 0.176 | 0.558 | 0.690 |
| | GRU model | 0.310 | 0.773 | 0.449 |
| | HM model | 0.448 | 0.786 | 0.452 |
| PS | DNN model | 0.242 | 0.570 | 0.620 |
| | GRU model | 0.309 | 0.748 | 0.488 |
| | HM model | 0.369 | 0.782 | 0.506 |
| OL | DNN model | 0.816 | 0.411 | 0.226 |
| | GRU model | 0.666 | 0.209 | 0.518 |
| | HM model | 0.552 | 0.024 | 0.456 |

## 6. Conclusion

According to the predictions from the three models and the corresponding comparisons, the spatial interdependence between the sediments and the spatial coupling of the observations are quite important for the lithology classification. The DNN classifier which takes no vertical spatial dependency into account provides the worst lithology classifications for all cross-validation. The GRU model which just captures the spatial coupling in observations has medium performance in each case. And the HM model, which represents the spatial interdependence between sediments and the spatial coupling in observations by using the Markov chain assumption, is the most reliable one of the classifications. Moreover, the neural network training is a time consuming process. The training time for the DNN and GRU models are dramatically longer than for the recursive reverse algorithm of the HM model. Last but not least, the neural network methods only provide the locationwise classifiers. Meanwhile the HM model provides a fully specified probabilistic model which is able to provide answers to any statement about $\boldsymbol{\kappa}$. In this case study, a small data set collected from just three wells is used for the experiments. Hence more comparisons and discussions based on big real data sets from variable geological settings are needed to obtain a more reliable conclusion.

## Acknowledgements

The PhD project of the first author is funded by China University of Petroleum Beijing. The study is made during a research visit of the first author at Department of Mathematical Sciences, Norwegian University of Science and Technology (NTNU), Trondheim, Norway during September 2018 to September 2019. The work is partly funded by the URE research initiative at IMF/NTNU. Miao Tian is grateful for the Norway & USA visiting study scholarship provided by Zhaojian Tian and Liping Yu. Finally, Weiqiao Huang, thank you for the warmest accompanying with me. I love you. And would you marry me?

## Appendix A. The Upward and Downward Sweeps for Bidirectional GRU Neural Network

**Algorithm 1:** The upward and downward sweep algorithm

```
for $l = 1, \dots, L_g$ do
    for $t = 1, \dots, T$ do
        for $s = t - \Delta_t, \dots, t$ do
            set $\dot{\mathbf{a}}^l_{t-\Delta_t-1} = 0\mathbf{i}_{n_l}$ and $\mathbf{a}^0_s = \mathbf{d}_s$;
            $\mathbf{a}_c = \begin{bmatrix} \dot{\mathbf{a}}^l_{s-1} \\ \mathbf{a}^{l-1}_s \end{bmatrix}$;
            $\vec{\boldsymbol{\alpha}}^l_f = f_{sigmoid}(\vec{\boldsymbol{\Gamma}}^l_f \mathbf{a}_c + \vec{\boldsymbol{\beta}}^l_f)$;
            $\vec{\mathbf{c}}^l_s = f_{tanh}(\vec{\boldsymbol{\Gamma}}^l_c \left[ \begin{bmatrix} \vec{\boldsymbol{\alpha}}^l_f \\ \mathbf{i}_{2n_{l-1}} \end{bmatrix} \otimes \mathbf{a}_c \right] + \vec{\boldsymbol{\beta}}^l_c)$;
            $\vec{\boldsymbol{\alpha}}^l_u = f_{sigmoid}(\vec{\boldsymbol{\Gamma}}^l_u \mathbf{a}_c + \vec{\boldsymbol{\beta}}^l_u)$;
            $\dot{\mathbf{a}}^l_s = \vec{\boldsymbol{\alpha}}^l_u \otimes \vec{\mathbf{c}}^l_s + [\mathbf{i}_{n_l} - \vec{\boldsymbol{\alpha}}^l_u] \otimes \dot{\mathbf{a}}^l_{s-1}$
        end
        $\vec{\mathbf{a}}^l_t = \dot{\mathbf{a}}^l_{s=t}$;
        for $s' = t + \Delta_t, \dots, t$ do
            set $\dot{\mathbf{a}}^l_{t+\Delta_t+1} = 0\mathbf{i}_{n_l}$ and $\mathbf{a}^0_{s'} = \mathbf{d}_{s'}$;
            $\mathbf{a}_c = \begin{bmatrix} \dot{\mathbf{a}}^l_{s'+1} \\ \mathbf{a}^{l-1}_{s'} \end{bmatrix}$;
            $\overleftarrow{\boldsymbol{\alpha}}^l_f = f_{sigmoid}(\overleftarrow{\boldsymbol{\Gamma}}^l_f \mathbf{a}_c + \overleftarrow{\boldsymbol{\beta}}^l_f)$;
            $\overleftarrow{\mathbf{c}}^l_{s'} = f_{tanh}(\overleftarrow{\boldsymbol{\Gamma}}^l_c \left[ \begin{bmatrix} \overleftarrow{\boldsymbol{\alpha}}^l_f \\ \mathbf{i}_{2n_{l-1}} \end{bmatrix} \otimes \mathbf{a}_c \right] + \overleftarrow{\boldsymbol{\beta}}^l_c)$;
            $\overleftarrow{\boldsymbol{\alpha}}^l_u = f_{sigmoid}(\overleftarrow{\boldsymbol{\Gamma}}^l_u \mathbf{a}_c + \overleftarrow{\boldsymbol{\beta}}^l_u)$;
            $\dot{\mathbf{a}}^l_{s'} = \overleftarrow{\boldsymbol{\alpha}}^l_u \otimes \overleftarrow{\mathbf{c}}^l_{s'} + [\mathbf{i}_{n_l} - \overleftarrow{\boldsymbol{\alpha}}^l_u] \otimes \dot{\mathbf{a}}^l_{s'+1}$
        end
        $\overleftarrow{\mathbf{a}}^l_t = \dot{\mathbf{a}}^l_{s'=t}$;
        $\mathbf{a}^l_t = \begin{bmatrix} \vec{\mathbf{a}}^l_t \\ \overleftarrow{\mathbf{a}}^l_t \end{bmatrix}$;
    end
    $\mathbf{a}^l = [\mathbf{a}^l_t, \dots, \mathbf{a}^l_T]$;
end
$\mathbf{a} = \mathbf{a}^{L_g} = [\mathbf{a}^{L_g}_1, \dots, \mathbf{a}^{L_g}_T]$
```

## References

Baum, L.E., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics 41, 164–171.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 .

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .

Cleeremans, A., Servan-Schreiber, D., McClelland, J.L., 1989. Finite state automata and simple recurrent networks. Neural computation 1, 372–381.

Corina, A.N., Hovda, S., 2018. Automatic lithology prediction from well logging using kernel density estimation. Journal of Petroleum Science and Engineering 170, 664–674.

Eidsvik, J., Avseth, P., Omre, H., Mukerji, T., Mavko, G., 2004. Stochastic reservoir characterization using prestack seismic data. Geophysics 69, 978–993.

Forgione, M., Bombois, X., Van den Hof, P.M., 2015. Data-driven model improvement for model-based control. Automatica 52, 118–124.

Fresia, B., Ross, P.S., Gloaguen, E., Bourke, A., 2017. Lithological discrimination based on statistical analysis of multi-sensor drill core logging data in the matagami vms district, quebec, canada. Ore Geology Reviews 80, 552–563.

Ghosh, S., Chatterjee, R., Shanker, P., 2016. Estimation of ash, moisture content and detection of coal lithofacies from well logs using regression and artificial neural network modelling. Fuel 177, 279–287.

Grana, D., Pirrone, M., Mukerji, T., 2012. Quantitative log interpretation and uncertainty propagation of petrophysical properties and facies classification from rock-physics modeling and formation evaluation analysis. Geophysics 77, WA45–WA63.

Horrocks, T., Holden, E.J., Wedge, D., 2015. Evaluation of automated lithology classification architectures using highly-sampled wireline logs for coal exploration. Computers & geosciences 83, 209–218.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436.

Lindberg, D.V., Omre, H., 2014. Blind categorical deconvolution in two-level hidden markov models. IEEE Transactions on Geoscience and Remote Sensing 52, 7435–7447.

Lindberg, D.V., Rimstad, E., Omre, H., 2015. Inversion of well logs into facies accounting for spatial dependencies and convolution effects. Journal of Petroleum Science and Engineering 134, 237–246.

Liu, H., Wen, S.m., Li, W.h., Xu, C.b., Hu, C.q., 2009. Study on identification of oil/gas and water zones in geological logging base on support-vector machine, in: Fuzzy Information and Engineering Volume 2. Springer, pp. 849–857.

Loures, L.G., Moraes, F.S., 2006. Porosity inference and classification of siliciclastic rocks from multiple data sets. Geophysics 71, O65–O76.

Maxwell, K., Rajabi, M., Esterle, J., 2019. Automated classification of metamorphosed coal from geophysical log data using supervised machine learning techniques. International Journal of Coal Geology 214, 103284.

Moja, S.S., Asfaw, Z.G., Omre, H., 2018. Bayesian inversion in hidden markov models with varying marginal proportions. Mathematical Geosciences , 1–22.

Saggaf, M.M., Nebrija, E.L., 2000. Estimation of lithologies and depositional facies from wire-line logs. AAPG bulletin 84, 1633–1646.

Saporetti, C.M., da Fonseca, L.G., Pereira, E., 2019. A lithology identification approach based on machine learning with evolutionary parameter tuning. IEEE Geoscience and Remote Sensing Letters .

Schumann, A., 2002. Hidden markov models for lithological well log classification. Terra Nostra 4, 373–378.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681.

Scott, S.L., 2002. Bayesian methods for hidden markov models: Recursive computing in the 21st century. Journal of the American Statistical Association 97, 337–351.

Tian, M., Xu, H., Cai, J., Wang, J., Wang, Z., 2019. Artificial neural network assisted prediction of dissolution spatial distribution in the volcanic weathered crust: A case study from chepaizi bulge of junggar basin, northwestern china. Marine and Petroleum Geology 110, 928–940.

Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE transactions on Information Theory 13, 260–269.