IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 August 2023, accepted 18 August 2023, date of publication 22 August 2023, date of current version 25 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307355

RESEARCH ARTICLE

Epistemic Uncertainty and Model Transparency in Rock Facies Classification Using Monte Carlo Dropout Deep Learning

TOUHID MOHAMMAD HOSSAIN[1], MAMAN HERMANA[1], AND SAID JADID ABDULKADIR[2], (Senior Member, IEEE)

[1]Centre for Subsurface Imaging (CSI), Department of Geosciences, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia
[2]Department of Computer and Information Sciences (CIS), Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia

Corresponding author: Touhid Mohammad Hossain (touhid.hossain@utp.edu.my)

This work was supported by Yayasan Universiti Teknologi PETRONAS (YUTP) Grant, Cost Center 015LC0-398, Universiti Teknologi PETRONAS, Malaysia.

ABSTRACT Although Deep Learning (DL) architectures have been used as efficient prediction tools in a variety of domains, they frequently do not care about the uncertainty in the predictions. This may prevent them from being used in practical applications. In seismic reservoir characterisation, predicting facies from seismic data is typically viewed as an inverse uncertainty quantification issue. The goal of the current study is to analyse the dependability of rock facies classification model in order to quantify the uncertainty while maintaining the high accuracy by using and evaluating monte carlo dropout based deep learning (MCDL), a computationally efficient technique. The proposed method is unique since it can quantify the epistemic uncertainty of the classified facies in blind or unseen well conditioned on Seismic attributes in the bayesian approximation achieved by MCDL framework. The findings show that MC dropout is successful in terms of accuracy and reliability, with a blind test F1-scores of 98% and 82% in predicting facies from synthetic and seismic datasets respectively. Moreover, the applications in a 2D section indicate that the internal regions of the seismic sections are generally classified with less epistemic uncertainty than their boundaries, as calculated from the different realizations of the MCDL network. For comparison, a plain DL and support vector machine (SVM) are also implemented and the findings suggest that our method outperformns the other models in comparison which indicates the potential of the model to be implemented in a robust rock facies classification.

INDEX TERMS Uncertainty, facies classification, Monte Carlo dropout, deep learning.

I. INTRODUCTION

Machine learning (ML) employs computational techniques to find significant patterns in datasets. In order to make judgements based on the data at hand and the domain expertise, machine learning (ML) and deep learning (DL) mehtods have been applied in a variety of fields in recent years. The accuracy and dependability of these models both are taken into account while assessing them. For regression tasks accuracy is commonly evaluated by statistical error measures such as the Mean-Squared-Error (MSE), coefficient of determination (R2), and for classification tasks, precision, recall, F1 score are mainly considered.

However, Although ML has a lot of potential, the uncertainty might cause the outputs from such models to become unreliable. Situations containing unknown, incomplete or imperfect knowledge are referred as uncertainty, which can arise from the whole analytical process, including the gathering, organising, and analysis of huge data [1]. The majority of ML approaches have a significant hurdle when dealing with vague data [1]. Reference [2] (2020) concentrated on how uncertainty affects the effectiveness of learning from

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

large datasets; nonetheless, reducing the inherent uncertainty in a sizable dataset is of special significance [2]. Even little faults can aggravate any flaws in the whole analytic process when these problems are scaled up to the level of huge data through positive feedback loops [1]. As a result, even if ML models can provide an ideal solution depending on the training set, if the uncertainty in the model perameters and data are not taken into account, such optimal models run a high chance of failing in real-world deployment. Having trust in the expected solution's quality is crucial since decisions based on inaccurate forecasts might result in large losses. Datasets of the right size and accuracy are necessary for the anticipated answers from such models to be trustworthy.

The scientific community has been increasingly interested in recent years in quantifying the uncertainty around ML models. Epistemic and aleatoric uncertainties are the two basic forms of uncertainty that may be modelled in the context of ML [3], [4]. The uncertainty in the model parameters is known as epistemic uncertainty. Aleatoric uncertainty has something to do with the noise that the training dataset itself contains. When using ML models for a particular issue domain, the literature has examples of several modelling strategies to assess aleatoric or epistemic uncertainty or both. Although quantifying uncertainty can enable its mitigation, uncertainty cannot be eliminated. This stochasticity is captured by Uncertainty Quantification (UQ) as a probability distribution. As a result, UQ does not necessarily result in an improvement in accuracy; rather, because of its probabilistic character, UQ enables the inclusion of a confidence interval in the output of the present model, enhancing its reliability and transparency.

There are two basic forms of uncertainty that ML and DL models confront in terms of dependability. The intrinsic unpredictability or variability in the data itself causes aleatoric uncertainty, also known as irreducible uncertainty or data uncertainty. Even with more samples, it cannot be removed. The information used to construct the model may originate from a variety of sources, including simulations or programmed data as well as experimental observations. Data dispersion and inaccuracies brought about by measurement, gathering, or creation procedures are referred to as noise in the data. Uncompleted domain coverage is another source of uncertainty, as models are frequently built using a small set of data and may not generalise successfully.

The source of epistemic uncertainty, also known as knowledge uncertainty or subjective uncertainty, is the model. By giving the model sufficient training examples, it may be decreased. Epistemic uncertainty may be caused by a variety of things, such as selecting extremely straightforward or intricate model structures, stochastic optimisation techniques, or certain statistical error measures.

Uncertainty Quantification (UQ) approaches are useful for reducing how uncertainties affect decision-making processes. The three basic categories of UQ methods–Bayesian techniques (e.g. Monte Carlo dropout and variational inference), ensemble methods (e.g. deep ensemble and deep ensemble Bayesian), and alternative approaches e.g. deep Gaussian Process are discussed. Making more informed judgements is made possible by using these strategies for analysing and controlling model prediction uncertainty.

Facies are sedimentary rock formations that can be distinguished from one another by their physical characteristics (such as sedimentary structure and grain size), and which were created in a specific depositional setting due to the influence of a hydrodynamic regime that was relatively consistent. Facies classification goals to predict lateral variations and subsurface depositional sequences [5] in order to identify suitable reservoir zone for hydrocarbon and geothermal extraction [6].

Facies are often described by geologists using core analysis, stratigraphy, and sedimentology. Practically speaking, however, facies categorization is frequently carried out utilising data from core samples, well log response, and seismic response. The manual interpretation of facies from seismic data frequently requires a great deal of the interpreters' time, talent, and knowledge. In this context, data-driven techniques might help interpreters by easing several difficulties connected with facies analysis from seismic data. However, due to the complexity of seismic-signals and different degrees of resolution, machine learning based facies classification at the seismic scale is typically less accurate than the observed facies characterization. Although data is frequently available, the requisite volume and precision cannot always be attained, and the data itself may occasionally change as a result of future environmental unpredictability. Moreover, due to the ambient noise and/or instrumental noise that affect the seismic data, uncertainty arises during model training which makes it essential to build models that can address the uncertainty in the prediction.

In the past, researchers have classified facies from seismic and well log data using a variety of statistical and mathematical models. In the past, these techniques have relied on algorithms for pattern recognition and clustering [7], [8], which aim to group the dataset into the appropriate number of facies based on the similarity in characteristics (elastic or petrophysic parameters). In a research Monte Carlo classification method using well logs combined with rock physics models is employed to calculate posterior uncertainty [9], [10]. To classify seismic facies, Z. Liu et al. worked on convolutional neural network (CNN). In another research, J. S. Dramsc et al. worked on the performance Analysis of State-of-the-Art CNN Architectures in seismic Facies classification.

In their study, Grana et al. evaluated DL algorithms and Monte Carlo probablistic method for the classification of facies using seismic datasets. To meet the needs of a huge number of tagged objects, unsupervised approaches and semisupervised generative adversarial networks are also used in recent years [13], [14], [15].

Seismic datasets driven facies classification models, however, include certain difficulties on account of limited training examples and the presence of noise in the training data, and that makes models inefficient if an adequate uncertainty handling framework is not taken into consideration. To measure the epistemic uncertainty along with the predictions, numerous techniques have been employed in recent years. Conventionally, in this particular field uncertainty quantification is commonly being practiced using Bayesian inferences [16]. When using CNN on picture datasets, Das et al. [17] utilised an approximated Bayesian computing technique to estimate the posterior distribution of predictions. In order to estimate the aleatoric uncertainties in the lithology classification model, Rough Set Theory (RST) is employed [18]. Liu and Grana [19] used an ensembled approach to analyse the related uncertainty in the facies categorization in a geostatistical inversion problem.

However, the majority of these available methods operate on the foundation of mapping one input to one output and hence they do not output the uncertainty consequent to the model parameters (epistemic uncertainty) [11], [14], [20]. Retraining such a Neural Network (NN) model results in a different prediction due to differing starting values of the parameters, indicating that the model's uncertainty is not acknowledged or taken into consideration [21]. In his model, Ghahramani [21] presented a probabilistic strategy for examining epistemic uncertainty throughout the deep learning training phase. In a recent study Touhid et al. proposed Bayesian deep learning (BDL) approach to quantify the DL model uncertainty in the facies prediction model using seismic data [22]. Abdullah et al. in a reseach workrd on a comprehensive review of common BDL models used in the healthcare field [23]. Athough BDL can produce good results it is computationally expensive and takes a long time to conduct the inference [24]. Another difficulty in BDL network is the prior probability distribution, commonly referred to as the prior which needs the expert's opinion about the data and model parameters (weights) [23]. The computational cost of BDL and gaussian process is a disadvantage since it becomes unaffordable for very large high dimension networks.

Recently, a technique called MC-dropout (MCD) was created that is computationally more effective [25], [26], [27]. It is possible to think of a NN with any number of hidden layers as a bayesian approximation problem of the probabilistic deep gaussian process if dropout is employed before the weight layers [24], [26]. In essence, dropout works as a regularisation approach during training to reduce over-fitting in DL. In a MCDL model, a distinct set of neurons are being dropped out in each iteration of training, because in every training iteration it randomly samples the neurons to be dropped out in each layer (based on the dropout rate of that particular layer). As a result, the architecture of the model varies a bit in every iteration, and the final result may be thought of as an average of several neural nets that were each trained on a single batch of data, driving the network to acquire more reliable and comprehensive representations. During MCDL inference or testing entails making several forward runs through the network while dropout is enabled and considering all the predictions give the epistemic uncertainty of the model. MCDL is simple and significantly less complex than BDL because in MCDL the model doesn't need to approximate the true posterior distribution with the model parameters during inference (Variational Inference). variational inference in BDL performs to provide a probabilistic framework for estimating intractable posterior distributions [22]. Another advantage of MCDL over BDL is, Monte Carlo Dropout-based neural networks do not explicitly require a prior distribution to construct the posterior distribution.

Therefore, at test time, the prediction is no longer deterministic. By utilizing the stochastic outputs, epistemic uncertainty is calculated int his model. In this article, Monte Carlo dropout based deep learning (MCDL) algorithm is employed as an alternative of BDL [24] and the application is shown in two different cases; identify four facies classes: shale, brine sand, hydrocarbon sand and others using synthetic dataset and identify two facies classes: shale and sand using seismic dataset.

This work is presented as follows; methodology is described in section II, application of the model is in section III the application of the model is done and the findings are shown; section IV consists of the blind test results and comparison between the proposed model and other models; in Section V the overall research is discussed and section VI concludes the paper.

II. METHODOLOGY

A. DROPOUT IN A NEURAL NETWORK

Here is a quick summary of the Srivastava et al. [25] neural net dropout model for the situation of a single hidden layer NN. This is done for convenience of notation, and it is simple to generalise to additional levels. Let, $\boldsymbol{W}_1, \boldsymbol{W}_2$ be the weight matrices that, respectively, connect the input layer to the hidden layer and hidden layer to the output layer. These weight matrices linearly transform the layers' inputs before employing some element-wise non-linearity $\sigma(.)$. The symbol $\boldsymbol{b}$ is the biases that shifts the input of the non-linearity. Let us assume the model inputs are G dimensional vectors and the outputs are C dimensional vectors and it has H hidden units. Hence, $\boldsymbol{W}_1$ is a $G \times H$ matrix, $\boldsymbol{W}_2$ is a $H \times C$ matrix, and $\boldsymbol{b}$ is a H dimensional vector. A standard neural net would output, $\boldsymbol{y}_p = \sigma(\boldsymbol{x}_o \boldsymbol{W}_1 + \boldsymbol{b})\boldsymbol{W}_2$ for some input $\boldsymbol{x}_o$ [24].

By sampling a pair of binary vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ of G and H dimensions respectively, dropout is implemented. The elements of the vectors follow a Bernoulli distribution with a probability parameter $P_i \in [0, 1]$ for $i = 1, 2$. Hence, $v_{1,q}$ ~Bernoulli(P_1) for $q = 1, \ldots, G$, and $\boldsymbol{v}_{2,h}$ ~Bernoulli(P_2) for $h = 1, \ldots, h$. Given a particular input $\boldsymbol{x}_o$, $1-P_1$ portion of the objects of the input set are set to zero: $\boldsymbol{x}_o \; o \; \boldsymbol{v}_1$ where o signifies the Hadamard product [24]. The first layer's output is, $\sigma((\boldsymbol{x}_o \; o \; \boldsymbol{v}_1)\boldsymbol{W}_1 + \boldsymbol{b}) \; o \; \boldsymbol{v}_2$, that is linearly transformed to the output of the dropout model $\boldsymbol{y}_p = (\sigma((\boldsymbol{x}_o \; o \; \boldsymbol{v}_1)\boldsymbol{W}_1 + \boldsymbol{b})) \; o \; \boldsymbol{v}_2\boldsymbol{W}_2$. Which is equivalent to multiplying $\boldsymbol{W}_1, \boldsymbol{W}_2$ by the binary vectors to nullify the entire rows:

$$\boldsymbol{y}_p = \sigma(\boldsymbol{x}_o(\boldsymbol{v}_1 \boldsymbol{W}_1) + \boldsymbol{b}(\boldsymbol{v}_2 \boldsymbol{W}_2)) \tag{1}$$

The same workflow is repeated for all the layers. To keep the notation neat, $\boldsymbol{v}_1$ is used as the symbol for $diag(\boldsymbol{v}_1)$ with the $diag(.)$ operator that maps a vector to a diagonal matrix whose diagonal is the elements of the vector. For using the NN model in regression problem Euclidean loss (square loss) is used [25]:

$$E = \frac{1}{2N}\sum_{n=1}^{N} ||\boldsymbol{y}_n - \boldsymbol{y}_{p_n}||_2^2 \tag{2}$$

where $\boldsymbol{x}_{o_1}, \ldots, \boldsymbol{x}_{o_N}$ are N inputs (observed), $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are the corresponding outputs (observed), and $\boldsymbol{y}_{p_1}, \ldots, \boldsymbol{y}_{p_N}$ are the model predicted outputs.

For classification model, predicting the probability of an object $\boldsymbol{x}_o$ being classified with label $1, \ldots, C$,output of the model $\boldsymbol{y}_p$ is passed through a softmax function to get normalised values: $\hat{p}_{nd} = \exp(\boldsymbol{y}_{p_{nd}})/(\sum_{d'} \exp(\boldsymbol{y}_{p_{nd'}}))$. Taking the log of this function results in a *softmax* loss [24],

$$E = -\frac{1}{N}\sum_{n=1}^{N} \log(\hat{p}_{n,c_n}) \tag{3}$$

where, $c_n \in [1, 2, \ldots, C]$ denotes the observed class for an input n. Often a regularisation term is added at the time of optimization. An $L2$ regularisation is usually used that is weighted by some weight-decay λ. Weight-decay is used to scale the derivatives of the loss function and by doing this the complexity of the model is reduced [24]. The weight-decay for the dropped-out weights is scaled by the probability of the weights to not be dropped [24]. Dropout, $L_{dropout}$ can be explained as,

$$L_{dropout} := E + \lambda_1||\boldsymbol{W}_1||_2^2 + \lambda_2||\boldsymbol{W}_2||_2^2 + \lambda_3||b||_2^2 \tag{4}$$

For every input point and forward pass in the model (evaluating the model's output), new realisations are sampled for the binary vectors $\boldsymbol{v}_i$. The derivatives that are found from the loss function are propagated to the parameters in the backward pass using the same values [24].

To keep a consistent output magnitude, the dropped weights $\boldsymbol{v}_1\boldsymbol{W}_1$ and $\boldsymbol{v}_2\boldsymbol{W}_2$ are scaled by $\frac{1}{P_i}$. This is comparable to scaling the weights $\boldsymbol{W}_i$ by P_i at test time and initialising the weights $\boldsymbol{W}_i$ with scale $\frac{1}{P_i}$ during training [24].

For general conditions or NN with more than one layer; Let $\boldsymbol{y}_p$ denote the output of the model with L layers and E(.,.) to denote the loss function e.g. softmax loss or the Euclidean loss (square loss). $\boldsymbol{W}_i$ denotes the NN's weight matrices of dimensions $H_i \times H_i - 1$, and b_i denotes the bias vectors of dimensions H_i for each layer $i = 1, \ldots, L$. $\boldsymbol{y}_{o_i}$ denotes the observed output corresponding to input $\boldsymbol{x}_{o_i}$ for $1 \leq i \leq$ N data points. Considering an $L2$ regularisation weighted by weight decay λ and T as the number of stochastic forward passes, dropout can be defined as,

$$L_{dropout} := \frac{1}{N}\sum_{t=1}^{T} E(\boldsymbol{y}_i, \boldsymbol{y}_{p_i}) + \lambda \sum_{i=1}^{L}(||\boldsymbol{W}_i||_2^2 + \boldsymbol{b}_i||_2^2||) \tag{5}$$

In dropout, binary variables are sampled for each input point and each network unit in each of the layers (except from the final one). For layer i, each binary variable has a chance of P_i of taking the value 1. If a unit's equivalent binary variable takes the value 0, it is considered dropped (i.e., its value is set to zero) for the specified input. In the backward pass propagating the derivatives to the parameters, the same values are utilised.

B. OBTAINING MODEL UNCERTAINTY

MC Dropout based Deep Learning methods may be viewed as a probabilistic approach that can quantify the posterior uncertainty [see Fig. 1]. Given the training data d_{tr}, a prediction network, $MCDL(.)$ and a new data point $\hat{\boldsymbol{x}}$, we can calculate the probability of possible output values $\hat{\boldsymbol{y}}$.

Let $p\left(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, d_{tr}\right)$, be the predictive distribution, with target, $\hat{\boldsymbol{y}}$, input $\hat{\boldsymbol{x}}$; and some training examples, $d_{tr} = (\boldsymbol{x}_{o_i}, \boldsymbol{y}_{o_i})_{i=1}^{N}$. After obtaining a predictive distribution, it is possible to inspect the variance among the predictions and uncover the uncertainty in the predictions. To learn a predictive distribution, an approach is to learn a distribution over the functions, or, correspondingly, a distribution over the parameters (i.e. the parametric posterior distribution $p(\Phi|d_{tr})$ where $\boldsymbol{W}, \boldsymbol{b} \in \Phi$.

The described Monte carlo dropout technique [24] offers a scalable way to learn a predictive distribution. As discussed in the previoun section, the purpose of Dropout is to switch off some neurons during the training time. MC dropout performs to randomly switch-off the neurons in a deep learning model, which regularizes the network. Each dropout configuration corresponds to a slightly different sample from the approximate parametric posterior distribution $q(\Phi|d_{tr})$:

$$\Phi_t \sim q(\Phi|d_{tr}) \tag{6}$$

where Φ_t is a particular dropout configuration, or, correspondingly, a simulation $\sim$, sampled from the approximate parametric posterior $q(\Phi|d_{tr})$.

Sampling from the approximate posterior $p(\Phi|d_{tr})$ enables Monte Carlo integration of the model's likelihood, $\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \Phi)$ which uncovers the predictive distribution, as follows:

$$\begin{aligned} p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}) &\approx \int_{\sigma} p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \Phi)q(\Phi|d_{tr})d\Phi \\ &\approx \frac{1}{T}\sum_{t=1}^{T} p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \Phi_t), \, such\ that, \, \Phi_t \sim q(\Phi|d_{tr}) \end{aligned} \tag{7}$$

The likelihood can be expressed to be normally distributed for coherence of understanding:

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \Phi) \approx \Lambda(f(\hat{\boldsymbol{x}}, \Phi), s^2(\hat{\boldsymbol{x}}, \Phi)), \tag{8}$$

with the Gaussian function Λ specified by the mean $f(\hat{\boldsymbol{x}}, \Phi)$ and variance $s^2(\hat{\boldsymbol{x}}, \Phi)$ parameters, which are outputs from the MCDL model:

$$f(\hat{\boldsymbol{x}}, \Phi), s^2(\hat{\boldsymbol{x}}, \Phi) \sim MCDL(\hat{\boldsymbol{x}}) \tag{9}$$

Figure 1 shows how MC dropout performs; each of the dropout configurations results a slightly different output by

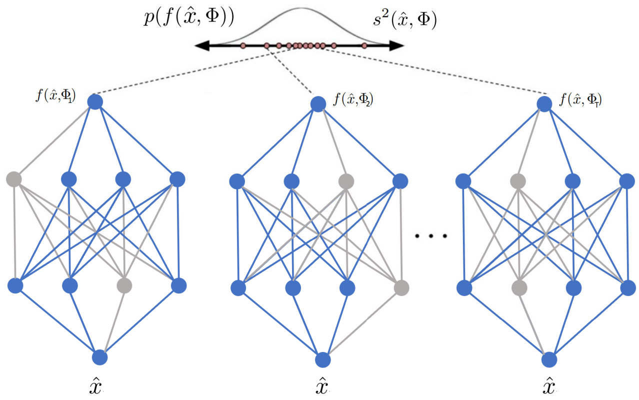


FIGURE 1. Prediction in a NN with dropout.

randomly turning neurons on (blue circles) and off (grey circles) with each forward propagation. Multiple forward passes with different dropout configurations result a predictive distribution over the mean $p(f(\hat{\mathbf{x}}, \Phi))$. The variance between the predictions, $s^2(\hat{\mathbf{x}}, \Phi)$ is the epistemic uncertainty of the model. The overview of the methodology is shown in Figure 1.

```
Algorithm 1 Prediction in Monte Carlo Dropout Deep Learning
Input: New data $\hat{\mathbf{x}}$, prediction network, $MCDL(\Phi)$, number
       of realizations, $T$.
Output: predictive mean, $E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})$, epistemic uncertainty or
        variance, $Var_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})$
for $t = 1$ to $T$ do
    $\mathbf{y}_p^t \leftarrow MCDL(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)$
end
// predictive mean
$E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}}) \leftarrow \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_p^t(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)$
// epistemic uncertainty
$Var_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}}) \leftarrow \tau^{-1}I_D +$
  $\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_p(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)^T\mathbf{y}_p^t(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)$
  $- E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})^T E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})(\hat{\mathbf{y}})}$
return $E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})$, $Var_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})$
```

To estimate the first two moments of the predictive distribution empirically, T sets of vectors of realisations are sampled here from the Bernoulli distribution $\{\mathbf{v}_1^t, \ldots, \mathbf{v}_L^t\}_{t=1}^{T}$ with $\mathbf{v}_i^t = [v_{i,j}^t]_{j=1}^{H_i}$. Mathematically, MC dropout may be approximated by using $\{\Phi_1^t, \ldots., \Phi_L^t\}_{t=1}^{T} \in \Phi$ at test time) and the mean prediction and the variance can be calculated. This variance among the T stochastic forward passes tells us how much the predictions are varying due to the slight variation of NN parameters each time and hence the variance is the epistemic uncertainty of the model. giving $\{\Phi_1^t, \ldots., \Phi_L^t\}_{t=1}^{T} \in \Phi$. We estimate,

$$E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}}) \approx \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_p^t(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t) \tag{10}$$

TABLE 1. Facies class No. and their names.

Facies No.	*Facies (Synthetic Case)*	*Facies (Seismic Case)*
0	Others	Shale
1	Hydrocarbon-Sand	Sand
2	Brine-Sand	N/A
3	Shale	N/A

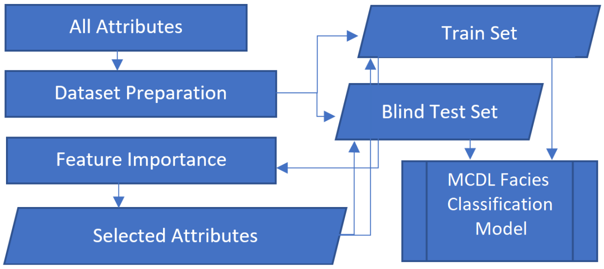


FIGURE 2. MCDL facies classification workflow.

Performing T stochastic forward passes across the network and get the mean prediction results is also equivalent to this. Moreover, the model uncertainty (epistemic uncertainty) can be calculated by taking into account all the individual realization results.

Given some precision hyper-parameter τ we estimate the second raw moment in the same way [24]:

$$E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}((\hat{\mathbf{y}})^T(\hat{\mathbf{y}})) \approx \tau^{-1}I_D + \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_p(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)^T \times \mathbf{y}_p^t(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t) \tag{11}$$

To obtain the model's predictive variance we have [24],

$$Var_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}}) \approx \tau^{-1}I_D + \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_p(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t)^T\mathbf{y}_p^t(\hat{\mathbf{x}}, \Phi_1^t, \ldots, \Phi_L^t) - E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})}(\hat{\mathbf{y}})^T E_{q(\hat{\mathbf{y}}|\hat{\mathbf{x}})(\hat{\mathbf{y}})} \tag{12}$$

which is equivalent of calculating the variance of T stochastic forward passes through the neural net. Here, $\hat{\mathbf{y}}$ is a row vector and hence, the sum is over the outer-products. Predictive variance gives an actual value to how much the predictions vary from the mean prediction.

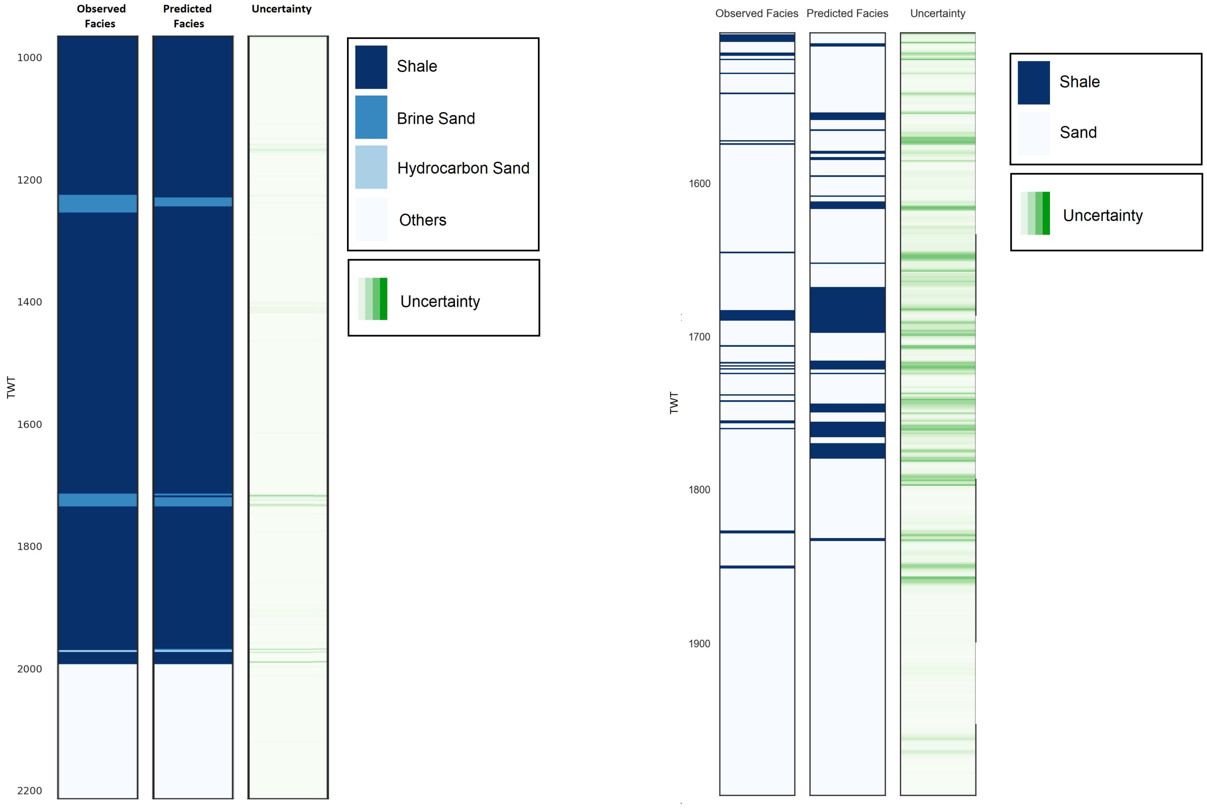
III. APPLICATION OF METHODOLOGY

A. DATASET PREPARATION

Facies information from the core description of the Synthetic datasets showed that the corresponding wells are composed of three major lithofacies; Shale, Brine Sand and Hydrocarbon Sand; Accordingly, 4 classes are considered where the 'other' class consists of the facies types other than the described three major facies classes. A total of 5 wells (Well-1, Well-2,..,Well 5) along with their facies class information are used in this research. Among them, all the wells except Well-3 are used to train the model, since Well-3 is selected for a blind test to see how the model performs in an unknown well. The name of the facies and their corresponding facies numbers are shown in the table 1.

TABLE 2. Data Overview.

Dataset Name	*Training Well Size*	*Blind Test Well Size*	*Total No. of Attributes*	*No. of Selected Attributes*	*Selected Attribute Sets*
Synthetic	5004	1251	33	6	Two way time, Instantaneous frequency, Integrated absolute amplitude, 5hz Lower frequency model, Seismic amplitude frequency, Enveloped instantaneous amplitude derivative
Seismic	2114	498	15	6	Two way time, Spectral decomposition (40Hz), Envelope, Integrate, Spectral decomposition (30Hz), derivative



(a) Synthetic Marmousi

(b) Seismic

FIGURE 3. Observed vs predicted and the associated epistemic uncertainty (blind test illustration).

TABLE 3. Classification accuracy (F1-Score) for different methods.

Method	*MCDL*	*DL*	*SVM*
Accuracy (Synthetic case)	.98	.96	.95
Accuracy (Seismic case)	.82	.78	.80

On the other hand, according to the core description of the Seismic datasets it is found that the corresponding wells consist of two major facies; Shale and Sand. A total of 7 wells (Seismic-Well1, Seismic-Well2…Seismic-Well6) along with their facies class information are used in this case and all the wells except Seismic-Well2 are used to train the model, since Seismic-Well2 is selected for a blind test. The name of the facies and their corresponding facies numbers are shown in the table below:

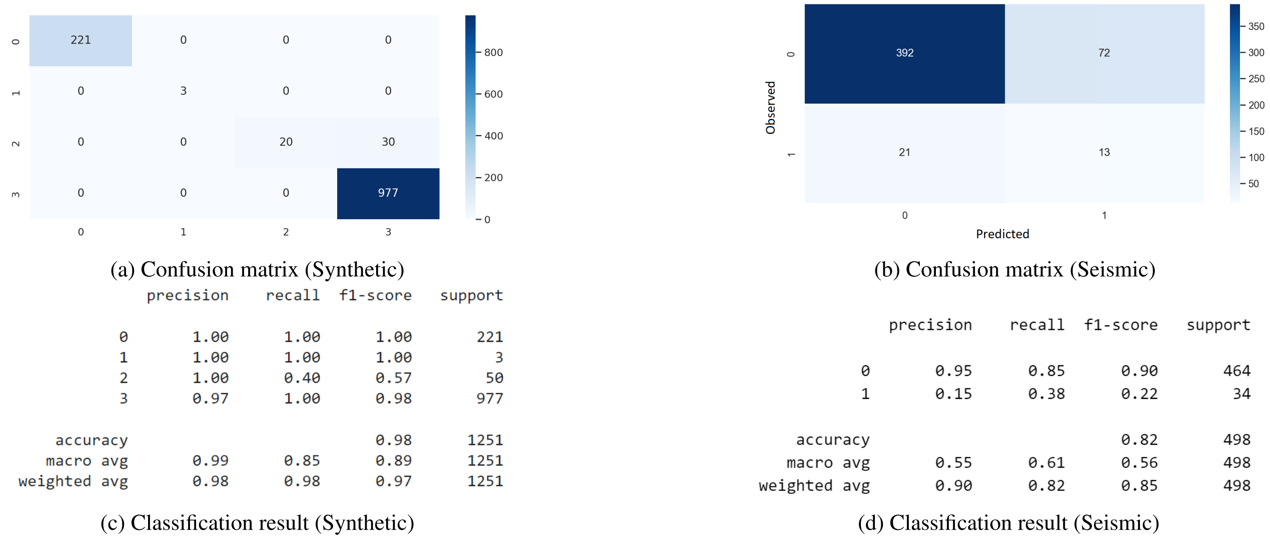
The training attributes are normalised using min-max normalisation to make the experiment computationally faster. The feature importance is calculated (see section iii.C) on the original dataset to reduce the size of the conditional attribute set and to choose the most crucial ones. Figure 2 below displays the stages for application.

B. IMPLEMENTATION DETAILS

The MCDL model was trained in CPU mode @ 2.10GHz Intel(R) Xeon(R) Silver 4110 processor with a RAM of 32 GB.

C. FEATURE SELECTION

In order to anticipate a target characteristic, feature selection refers to approaches that may compute the scores for all the input attributes and output the order of the most crucial ones. A higher score means that the particular characteristic will have a bigger impact on the model being used to forecast that particular variable. Extreme gradient boosted trees (XGBoost) approach is employed for our investigation. A collection of decision trees, XGBoost is an improved



(a) Confusion matrix (Synthetic)

(b) Confusion matrix (Seismic)

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       221
           1       1.00      1.00      1.00         3
           2       1.00      0.40      0.57        50
           3       0.97      1.00      0.98       977

    accuracy                           0.98      1251
   macro avg       0.99      0.85      0.89      1251
weighted avg       0.98      0.98      0.97      1251
```

(c) Classification result (Synthetic)

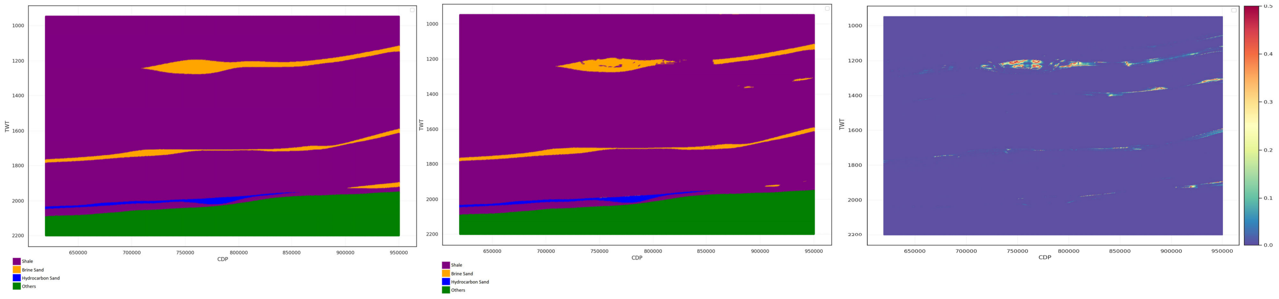
```
              precision    recall  f1-score   support

           0       0.95      0.85      0.90       464
           1       0.15      0.38      0.22        34

    accuracy                           0.82       498
   macro avg       0.55      0.61      0.56       498
weighted avg       0.90      0.82      0.85       498
```

(d) Classification result (Seismic)

FIGURE 4. Blind test performance summary of MCDL.

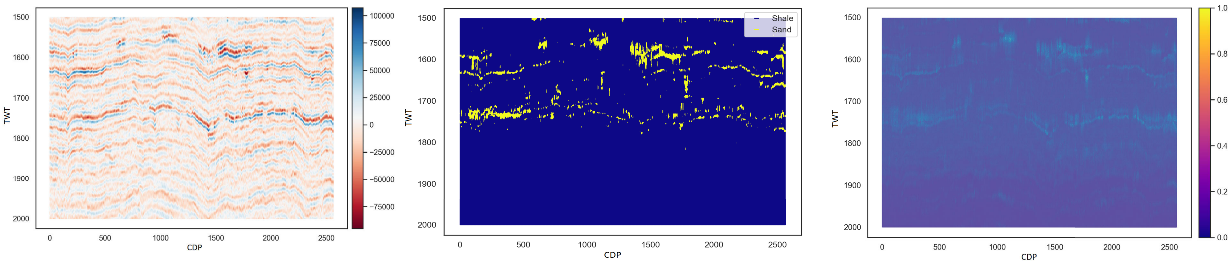


(a) Observed Facies

(b) Predicted Facies (MCDL)

(c) Epistemic Uncertainty

FIGURE 5. Visualization of the actual vs predicted classes and the associated epistemic uncertainty in the whole seismic section prediction.



(a) Observed Seismic response

(b) Predicted Facies (MCDL)

(c) Epistemic Uncertainty

FIGURE 6. Visualization of the actual vs predicted classes and the associated epistemic uncertainty in the whole seismic section prediction.

version of Gradient Boosting. An F-score (feature score), which indicates the frequency of features in the ensemble, is used by XGBoost to determine the importances of the features. A feature's significance in the classification process increases with its F-score. The overview of the dataset before and after the feature selection is shown in Table 2 below.

D. MONTE CARLO DROPOUT BASED DEEP LEARNING IMPLEMENTATION

Depending on the way in which an algorithm learns from data sets, DL (and also ML) algorithms fall into four categories: supervised, unsupervised, semi-supervised, and reinforcement. Our problem is a supervised learning task. For this facies classification problem, a neural network architecture is implemented where, the deep layers are customized into monte carlo dropout deep layers. In this architecture, the inputs are the selected features and accordingly 4 or 2 nodes are used in the output-layer to get the softmax probability of the 4 or 2 facies classes (for the synthetic and seismic datasets respectively). 128 nodes has been used for each of the 5 fully connected hidden layers. The hidden layers use the ReLU activation function with batch normalization [26]. To get the model error in the MCDL network, a categorical-cross-entropy loss function is used in the output.

To get predictive distribution from the output, the model randomly switches off neurons in the NN based on the dropout rate of 0.05. A total of 100 dropout configuration Φ_t or realisations are considered in this application. Each Φ_t corresponds to a slightly different sample from the approximate parametric posterior distribution $p(\Phi|d_{tr})$ which intuitively produces slightly different softmax outputs in each of the output nodes. To conduct the classification for a sample, for each of the 4 output nodes, the mean is calculated. The highest mean value (among the four means) decides the class for the particular sample, and the uncertainty is calculated from the standard deviation among the node outputs of the selected class.

Figure 3 illustrates the observed vs predicted classification results along with the uncertainty information the blind test well and Figure 4 visualises the detailed performance summary

E. IMPLEMENTATION OF THE MCDL MODEL IN 2D ARBITRARY SECTION

The trained model is implemented in 2D arbitrary sections, as shown in Figure 5 (synthetic case) and Figure 6 (seicmic case). In this implementation the result can be explained as a set of blind test result, where, each of the the common depth point (CDP) along the X axis are individual blind well (see Figure 5 and 6). For example, Figure 3(a) in section iii(D) is basically the result of one of these CDP's (CDP=908000 in particular) in Figure 5.

For both of the synthetic and seismic cases, prediction is done for each of the CDP points (see Figure 6b and Figure 6b) along with the associated epistemic uncertainty (see Figure 6c and Figure 6c).

IV. BLIND TEST PERFORMANCE COMPARISON WITH OTHER MODELS

For comparison we have adapted two other techniques to obtain the facies classification results: Deep Learning (DL) and XGBoost (XGB).

A. DEEP LEARNING (DL)

Numerous classification models have been built using Artificial Neural Nets and their variations throughout the years [28], [29]. Because of the robustness and remarkable capacity of locating a complicated mapping between nonlinearly linked with input and output data, they are quite effective for predictions [30], [31]. To analyse DL model performance, the same dataset is used. The same training and blind testing sets are considered to calculate the model performance. In this DL implementiation, the same MCDL architecture (excluding the dropouts in each layer) is considered. In another words, Five dense layers with 128 nodes each with ReLU activation function is used as hidden layers. For the output layer, softmax probability activation function is used in 4 or 2 nodes to classify the 4 or 2 facies classes (for the Synthetic and Seicmic cases respectively). The classification result of the DL model is shown in Table 3.

B. SUPPORT VECTOR MACHINE (SVM)

SVM is a machine learning tool proposed by Vladmir Vapnik, has been used for 20 years to solve several problems, including facies prediction [32].SVM works by mapping non-linear inputs to a very high dimension feature space and creates a linear decision surface to predict its target [33]. To examine how SVM performs, the same training and blind test datasets are used. For implementing the SVM model, radial basis function kernel has been used. During the SVM model training, to find the optimal hyperparameter values for C and gamma value, a random search method is implemented [33]. C is a hyperparameter which is used to control error and Gamma is also a hyperparameter which used to give curvature weight of the decision boundary [33]. By using the optimal setting, blind test is perform and accuracy is calculated.

C. COMPARISON RESULT

Table 3 below compares the facies classification accuracy among MCDL, DL and XGB.

From Table 3 it is clear that MCDL performs slightly better than both DL and XGB.

D. 2D ARBITRARY SEISMIC SECTION COMPARISON RESULT

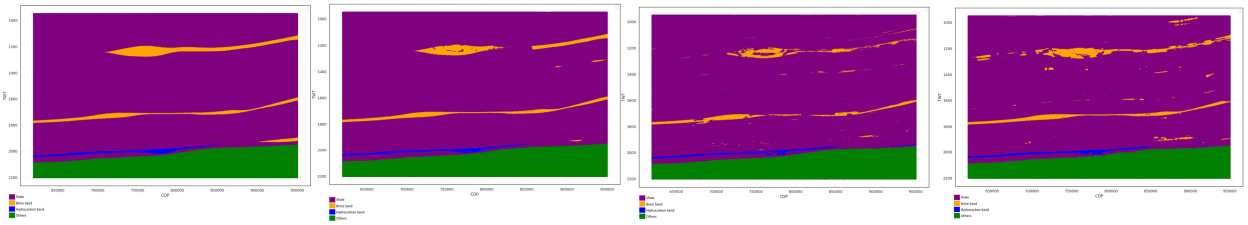
Similarly the 2D arbitrary section results are achieved for the deterministic soft computing methods in comparison (DL and SVM). Figure 7 and Figure 8 illustrate the prediction results with these methods for synthetic case and seismic case respectively.

V. DISCUSSION

To generate the most-likely model in the facies classification model, MCDL method is adapted [24], [25]. Under this architecture, in each MCDL iteration, a separate group of neurons are randomly sampled to be dropped out in each layer (in accordance with that layer's dropout rate). As a result, the architecture of the model changes significantly from one iteration to the other, leading to an ensemble of several distinct neural network configurations, each trained on a single batch of data. All of the pre-trained neurons and connections are utilised at inference time, or when predicting using the new data. Considering all the outputs from the realizations, the uncertainty is calculated.

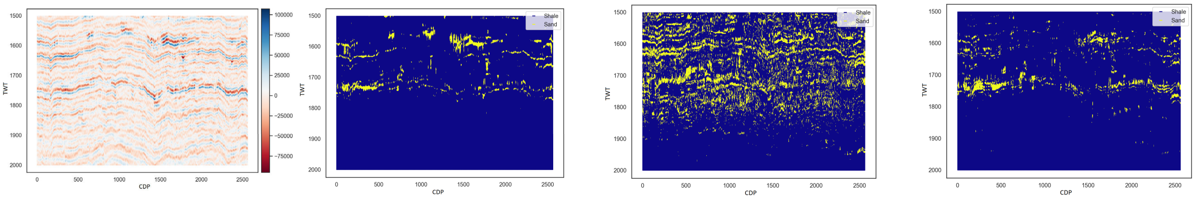
In terms of blind test accuracy, it is perceived that MCDL performs better than the other two methods in comparison. Morever, in the standard deterministic models in comparison (DL and SVM), a single prediction is obtained for a given input, with no information about the uncertainty of the used data or the model fitness.

To distinguish the facies classes, the proposed method has been extended to predict facies in the complete 2D arbitrary section. The result in Figures 5 and Figure 6 signifies that the MCDL model can classify the facies classes quite



(a) Observed Facies (b) Predicted Facies (MCDL) (c) Predicted Facies (DL) (d) Predicted Facies (SVM)

FIGURE 7. Visualization of the observed vs predicted classes for the two methods in comparison (DL and SVM).



(a) Observed Facies (b) Predicted Facies (MCDL) (c) Predicted Facies (DL) (d) Predicted Facies (SVM)

FIGURE 8. Visualization of the observed vs predicted classes for the two methods in comparison (DL and SVM).

efficiently. From Figure 5 and figure 6 it is also visible that the uncertainty is observed mostly in the misclassified regions or in some points in the boundary regions between two different classes. Hence, the measured uncertainty can give us the indication of how much trustworthy the prediction is. On the other hand, the 2D results for DL and SVM (as shown in Figure 7 and Figure 8) show that a plain deep learning network fails to predict the facies in several regions. Although, SVM performs a bit better than plain DL, the prediction is not as good as MCDL. Therefore, the MCDL model's robustness is not only in its prediction accuracy but also in finding the region where the prediction is uncertain. In another words, MCDL can express the trustworthiness of a particular prediction.

However, by altering the output layer parameters (a sigmoid function with a single activation unit instead of a softmax function with four or two units in our model), a similar model can be constructed to predict continuous attributes, such as reservoir properties or elastic properties.

VI. CONCLUSION

The presented facies classification method using monte carlo dropout based deep learning allows to generate multiple stochastic realizations for each prediction. By utilizing all the realizations and taking the mean of them leads to a more precise prediction. Additionally, MCDL can identify the out of distribution input noise by expressing the model's uncertainty by using the stochasticity among the realizations. For example, when the input data is outside the distribution that the model was trained on, the model shows more epistemic uncertainty in that particular prediction. The computation cost in MCDL is very low comparing to bayesian deep learning. The measured epistemic uncertainty in a particular point can give the proper indication of the trustworthiness of that prediction. However, although MCDL can quantify the epistemic uncertainty efficiently, to build a more reliable model, both aleatoric and epistemic uncertainty should be considered. To implement both epistemic and aleatoric uncertainty under the same network, in future, work on Evidential Deep Learning methodology.

ACKNOWLEDGMENT

The authors would like to thank PETRONAS, Malaysia, for providing the data for this study. They also like to thank all the colleagues and students in the Centre for Subsurface Imaging (CSI) and the Department of Geosciences, Universiti Teknologi PETRONAS, for their continuous support.

REFERENCES

[1] D. A. Wood and A. Choubineh, "Transparent machine learning provides insightful estimates of natural gas density based on pressure, temperature and compositional variables," *J. Natural Gas Geosci.*, vol. 5, no. 1, pp. 33–43, Feb. 2020.

[2] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[3] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," 2020, *arXiv:2011.06225*.

[4] T. Siddique and M. S. Mahmud. *Classification of fNIRS Data Under Uncertainty: A Bayesian Neural Network Approach 2021*. Accessed: Nov. 20, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9398971

[5] F. Li, H. Zhou, Z. Wang, and X. Wu, "ADDCNN: An attention-based deep dilated convolutional neural network for seismic facies analysis with interpretable spatial–spectral maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1733–1744, Feb. 2021.

[6] M. Pang, J. Ba, L.-Y. Fu, J. M. Carcione, U. I. Markus, and L. Zhang, "Estimation of microfracture porosity in deep carbonate reservoirs based on 3D rock-physics templates," *Interpretation*, vol. 8, no. 4, pp. 43–52, Nov. 2020.

[7] W. L. Martinez and A. Martinez, *Computational Statistics Handbook With MATLAB*, 3rd ed. New York, NY, USa: Chapman & Hall, 2007.

[8] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2009.

[9] D. Grana, L. Azevedo, and M. Liu, "A comparison of deep machine learning and Monte Carlo methods for facies classification from seismic data," *Geophysics*, vol. 85, no. 4, pp. 41–52, Jul. 2020.

[10] D. Grana, M. Pirrone, and T. Mukerji, "Quantitative log interpretation and uncertainty propagation of petrophysical properties and facies classification from rock-physics modeling and formation evaluation analysis," *Geophysics*, vol. 77, no. 3, pp. 45–63, May 2012.

[11] Z. Liu, J. Cao, Y. Lu, S. Chen, and J. Liu, "A seismic facies classification method based on the convolutional neural network and the probabilistic framework for seismic attributes and spatial classification," *Interpretation*, vol. 7, no. 3, pp. 225–236, Aug. 2019.

[12] J. S. Dramsch and M. Lthje, "Deep-learning seismic facies on state-of-the-art CNN architectures," in *Proc. 88th Annu. Int. Meeting SEG Expanded Abstr.*, Oct. 2018, pp. 1–5.

[13] M. Liu, M. Jervis, W. Li, and P. Nivlet, "Seismic facies classification using supervised convolutional neural networks and semisupervised generative adversarial networks," *Geophysics*, vol. 85, no. 4, pp. 47–58, Jul. 2020.

[14] F. Qian, M. Yin, X.-Y. Liu, Y.-J. Wang, C. Lu, and G.-M. Hu, "Unsupervised seismic facies analysis via deep convolutional autoencoders," *Geophysics*, vol. 83, no. 3, pp. 39–43, May 2018.

[15] T. M. Hossain, J. Watada, I. A. Aziz, and M. Hermana, "Machine learning in electrofacies classification and subsurface lithology interpretation: A rough set theory approach," *Appl. Sci.*, vol. 10, no. 17, p. 5940, Aug. 2020, doi: 10.3390/app10175940.

[16] J. Mitros and B. Mac Namee, "On the validity of Bayesian neural networks for uncertainty estimation," 2019, *arXiv:1912.01530*.

[17] V. Das, A. Pollack, U. Wollner, and T. Mukerji, "Convolutional neural network for seismic impedance inversion," *Geophysics*, vol. 84, no. 6, pp. 869–880, Nov. 2019.

[18] T. M. Hossain, J. Watada, I. A. Aziz, M. Hermana, S. T. Meraj, and H. Sakai, "Missing well log data handling in complex lithology prediction: An NIS apriori algorithm approach," *Int. J. Innov. Comput., Inf. Control*, vol. 17, no. 1, pp. 1077–1091, 2021.

[19] M. Liu and D. Grana, "Accelerating geostatistical seismic inversion using TensorFlow: A heterogeneous distributed deep learning framework," *Comput. Geosci.*, vol. 124, pp. 37–45, Mar. 2019.

[20] R. Feng, N. Balling, D. Grana, J. S. Dramsch, and T. M. Hansen, "Bayesian convolutional neural networks for seismic facies classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8933–8940, Oct. 2021, doi: 10.1109/TGRS.2020.3049012.

[21] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015.

[22] T. M. Hossain, M. Hermana, M. S. Jaya, H. Sakai, and S. J. Abdulkadir, "Uncertainty quantification in classifying complex geological facies using Bayesian deep learning," *IEEE Access*, vol. 10, pp. 113767–113777, 2022, doi: 10.1109/ACCESS.2022.3218331.

[23] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa, "A review on Bayesian deep learning in healthcare: Applications and challenges," *IEEE Access*, vol. 10, pp. 36538–36562, 2022, doi: 10.1109/ACCESS.2022.3163384.

[24] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 1050–1059.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] M. Y. Avci, Z. Li, Q. Fan, S. Huang, B. Bilgic, and Q. Tian, "Quantifying the uncertainty of neural networks using Monte Carlo dropout for deep learning based quantitative MRI," 2021, *arXiv:2112.01587*.

[27] A. Choubineh, J. Chen, F. Coenen, and F. Ma, "Applying Monte Carlo dropout to quantify the uncertainty of skip connection-based convolutional neural networks optimized by big data," *Electronics*, vol. 12, no. 6, p. 1453, 2023, doi: 10.3390/electronics12061453.

[28] C. P. Ross and D. M. Cole, "A comparison of popular neural network facies-classification schemes," *Lead. Edge*, vol. 36, no. 4, pp. 340–349, Apr. 2017, doi: 10.1190/tle36040340.1.

[29] S. Chikhi and M. Batouche, "Using probabilistic unsupervised neural method for lithofacies identification," *Int. Arab J. Inf. Technol.*, vol. 2, no. 1, pp. 58–66, 2005.

[30] S. Sahoo and M. K. Jha, "Pattern recognition in lithology classification: Modeling using neural networks, self-organizing maps and genetic algorithms," *Hydrogeol. J.*, vol. 25, no. 2, pp. 311–330, Mar. 2017.

[31] Y. Imamverdiyev and L. Sukhostat, "Lithological facies classification using deep convolutional neural network," *J. Petroleum Sci. Eng.*, vol. 174, pp. 216–228, Mar. 2019.

[32] X.-Y. Liu, L. Zhou, X.-H. Chen, and J.-Y. Li, "Lithofacies identification using support vector machine based on local deep multi-kernel learning," *Petroleum Sci.*, vol. 17, no. 4, pp. 954–966, Aug. 2020, doi: 10.1007/s12182-020-00474-6.

[33] T. Horvath, R. G. Mantovani, and A. C. de Carvalho, "Effects of random sampling on SVM hyper-parameter tuning," in *Proc. Int. Conf. Intell. Syst. Design Appl.*, 2017, pp. 268–278, doi: 10.1007/978-3-319-53480-0_27.

TOUHID MOHAMMAD HOSSAIN received the B.Sc. degree (Hons.) (cum laude) in computer engineering from Independent University, Bangladesh (IUB), in 2016, and the M.Sc. degree in information technology from Universiti Teknologi PETRONAS (UTP), Malaysia, in 2021.

He was a Teaching Assistant of mathematics with IUB, from 2013 to 2014. He is currently a Research Scientist with UTP. His research interests include soft computing, explainable machine learning, data science, and knowledge computing. He received the Best Student Paper Award in International Conference on Unconventional Modeling, Simulation and Optimization-Soft Computing and Meta Heuristics (UMSO 2018, Kitakyushu, Fukuoka, Japan). He was a recipient of UTP CIS Academic Excellence Award, in 2021.

MAMAN HERMANA received the B.Sc. degree in physics, majoring in geophysics from the University of Indonesia, in 1998, the M.Sc. degree in applied geophysics from the Institute Technology of Bandung, in 2002, and the Ph.D. degree in petroleum geosciences, majoring in hydrocarbon prediction from UTP, in 2015. He has been appointed as an Associate Professor with UTP. With 15 years of experience in teaching and research in geophysics, he has published several papers in international journals and conferences, and holds a patent in Q.I. He is an Active Member of SEG, SPE, and EAGE organizations.

SAID JADID ABDULKADIR (Senior Member, IEEE) received the B.Sc. degree in computer science from Moi University, the M.Sc. degree in computer science from Universiti Teknologi Malaysia, and the Ph.D. degree in information technology from Universiti Teknologi PETRONAS. He is currently an Associate Professor with the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS. His research interests include machine learning, deep learning architectures, and applications in predictive and streaming analytics. He is serving as a Executive Committee Member for IEEE Computational Intelligence Society Malaysia chapter and the Editor-in-Chief for platform journal.