MDPI

Article

Applying Monte Carlo Dropout to Quantify the Uncertainty of Skip Connection-based Convolutional Neural Networks Optimized by Big Data

Abouzar Choubineh [1,2,*], Jie Chen [2,*], Frans Coenen [1] and Fei Ma [2]

[1] Department of Computer Science, University of Liverpool, Liverpool, L69 7ZX, UK
[2] Department of Applied Mathematics, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China
* Correspondence: a.choubineh@liverpool.ac.uk or a.choubineh20@student.xjtlu.edu.cn (A.C.); jie.chen01@xjtlu.edu.cn (J.C.)

Abstract: Although Deep Learning (DL) models have been introduced in various fields as effective prediction tools, they often do not care about uncertainty. This can be a barrier to their adoption in real-world applications. The current paper aims to apply and evaluate Monte Carlo (MC) dropout, a computationally efficient approach, to investigate the reliability of several skip connection-based Convolutional Neural Network (CNN) models while keeping their high accuracy. To do so, a high-dimensional regression problem is considered in the context of subterranean fluid flow modeling using 376,250 generated samples. The results demonstrate the effectiveness of MC dropout in terms of reliability with a Standard Deviation (SD) of 0.012-0.174, and of accuracy with a coefficient of determination (R^2) of 0.7881-0.9584 and Mean Squared Error (MSE) of 0.0113-0.0508, respectively. The findings of this study may contribute to the distribution of pressure in the development of oil/gas fields.

Keywords: deep learning; Monte Carlo dropout; reliability; regression; fluid flow modeling; mixed GMsFEM; standard deviation

Citation: Choubineh, A.; Chen, J.; Coenen, F.; Ma, F. Applying Monte Carlo Dropout to Quantify the Uncertainty of Skip Connection-based Convolutional Neural Networks Optimized by Big Data. *Electronics* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

The terms Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are frequently used interchangeably. From a holistic perspective, DL is a subcategory of ML which, in turn, is a subdivision of AI. Artificial intelligence is a far-reaching branch of computer science in which a range of tools and techniques are used to make machines (namely computers and robots) more intelligent and consequently more effective and efficient. Computational methods of ML such as Neural Networks (NNs), support vector machines, and decision trees are employed to find relevant patterns within a dataset. DL methods represent more sophisticated extensions of classical ML techniques, and are generally superior to them. There are various DL algorithms such as Convolutional Neural Networks (CNNs), deep auto-encoders, and generative adversarial networks.

ML and DL models have been introduced in various fields [1–8] to make decisions using available data and domain knowledge. It is crucial to consider both accuracy and reliability when evaluating such models. These models are typically assessed based on their accuracy using statistical error metrics such as: (i) for regression: the coefficient of determination (R^2), Mean Squared Error (MSE), and relative error, and (ii) for classification: precision, F1 score, and confusion matrix.

In terms of reliability, ML and DL deal with two main types of uncertainty: (i) aleatoric uncertainty (also called irreducible uncertainty/data uncertainty/inherent randomness) and (ii) epistemic uncertainty (also called knowledge uncertainty/subjective uncertainty) [9]. Aleatoric uncertainty arises from an inherent property of the data and cannot be reduced even with a higher volume of samples. The data used to develop a model can be sourced from experimental measurements, collected from other resources, or produced via simulation/programming. This data always contains noise, which refers to the data distribution and errors made while measuring, collecting, or generating data. A related problem is incomplete coverage of the domain. That is why most models are constructed based on a limited range of data and cannot be generalized. Epistemic uncertainty is a

property of a model caused by factors such as the selection of very simple or complex structures, the stochastic nature of optimization algorithms, or the type of statistical error metrics. This uncertainty is reducible by feeding enough training samples into the model.

Uncertainty Quantification (UQ) techniques are beneficial to limit the effect of uncertainties on decision-making processes. According to [9], there are three main types of UQ: (i) Bayesian methods such as Monte Carlo (MC) dropout, Markov chain Monte Carlo, variational inference, Bayesian active learning, Bayes by backprop, variational autoencoders, Laplacian approximations, and UQ in reinforcement learning like Bayes-adaptive Markov decision process, (ii) ensemble techniques such as deep ensemble, deep ensemble Bayesian/Bayesian deep ensemble, and uncertainty in Dirichlet deep networks like information-aware Dirichlet networks, and (iii) other methods such as deep Gaussian Process (GP) and UQ in the traditional ML domain using ensemble techniques like support vector machine with Gaussian sample uncertainty.

Typically, researchers tend to apply their techniques or methods to existing datasets. Even when using new data, it may still face limitations such as a restricted sample size or low dimensionality. Moreover, while both classification and regression algorithms are supervised learning techniques, previous studies on DL have mostly focused on classification, and regression has received much less attention. Additionally, despite much research into the accuracy of DL models, their reliability analysis remains inadequate. Finally, MC dropout is a computationally efficient method that uses dropout as a regularization term to estimate uncertainty. Putting these points together, the main contribution of this paper is to demonstrate the potential of using MC dropout in skip connection-based CNN models based on big data.

A high-dimensional regression problem from the domain of petroleum engineering is included as a case study because subsurface flow problems usually involve some degree of uncertainty due to the lack of data with which models are constructed. Moreover, despite extensive efforts towards renewable energy, the oil/gas sector still supplies a significant proportion of global energy consumption, so this research has real-world applications.

The rest of this paper is arranged as follows. Section 2 provides an overview of MC dropout and references a number of relevant publications. In Section 3, the mixed Generalized Multiscale Finite Element Method (GMsFEM) is briefly explained as a case study. Section 4 presents the characteristics of the skip connection-based CNN models used along with MC dropout. The results are given in Section 5. Section 6 provides some discussion and the limitation of the research. The conclusions and future study are given in Section 7.

2. MC dropout and its related work

Standard deterministic deep NNs operate on a one input-one output basis. Unlike single-point predictions of such models, Bayesian methods such as Bayesian Neural Networks (BNNs) and (GPs) give predictive distributions. The weights of BNNs are incorporated with priors distribution, whereas GPs introduce priors over functions. A drawback of BNNs and GPs is the computational cost, which becomes prohibitive given a very large network, as in the case of deep networks. BNNs need to get the posterior distribution across the network's parameters, in which all possible events are obtained at the output. Gaussian processes require to sample prior functions from multivariate Gaussian distribution, wherein the dimension of Gaussian distribution increases proportionally with the number of training points involving the whole dataset during predictions.

A computationally more efficient method called MC dropout has been recently developed [10]. A NN with any depth and non-linearities accompanying dropout before weight layers might be interpreted as a Bayesian approximation of the probabilistic deep GP. Additionally, the dropout objective minimizes Kullback-Leibler (KL) divergence between an approximate distribution and the posterior of a deep GP.

Dropout basically serves as a regularization technique within the training process to reduce over-fitting in NNs. For the testing samples, the dropout is not applied, but weights are adjusted, e.g. multiplied by '1 – dropout ratio'. With regards to MC dropout, the dropout is applied at both training and test time. So, the prediction is no longer deterministic at test time.

Given that $\hat{y}$ is an output of a NN model with hidden layers L. Also, $w = \{W_1, \ldots, W_L\}$ represents the NN's weight matrices, and y^* is the observed output corresponding to input x^*.

By defining $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_N\}$ as the input and output sets, the predictive distribution is expressed as:

$$p(y^* \mid x^*, X, Y) = \int p(y^* \mid x^*, w) p(w \mid X, Y) dw \tag{1}$$

here, $p(y^* \mid x^*, w)$ and $p(w \mid X, Y)$ are the NN model's likelihood and the posterior over the weights.

The predictive mean and variance are used in the predictive distribution to estimate uncertainty. The posterior distribution is, however, analytically intractable. As a replacement, an approximation of variational distribution $q(w)$ can be obtained from the GP such that it is as close to $p(w \mid X, Y)$ as possible, in which the optimization process happens through the minimization of the KL divergence between the preceded distributions as shown below:

$$KL(q(w) \mid p(w \mid X, Y)) \tag{2}$$

With variational inference, the predictive distribution can be described as follows:

$$q(y^* \mid x^*) = \int p(y^* \mid x^*, w) q(w) dw \tag{3}$$

According to [10], $q(w)$ is selected to be the matrices distribution whose columns are randomly set to zero given a Bernouli distribution specified as:

$$W_i = M_i . diag([z_{i,j}]_{j=1}^{K_i}) \tag{4}$$

where $z_{i,j} \sim Bernoulli(p_i)$ for $i = 1, \ldots, L$ and $j = 1, \ldots, K_{i-1}$ with $K_i \times K_{i-1}$ as the dimension of matrix W_i. Also p_i represents the probability of dropout and M_i is a matrix of variational parameters. Therefore, drawing T sets of vectors of samples from Bernoulli distribution gives $(W_1^t, \ldots, W_L^t)_{t=1}^T$, and consequently, the predictive mean will be:

$$E_{q(y^* \mid x^*)(y^*)} \approx \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, W_1^t, \ldots, W_L^t) = p_{MC}(y^* \mid x^*) \tag{5}$$

Where $\hat{y}^*$ is the output obtained by the given NN for input x^*, and p_{MC} is the predictive mean of MC dropout, equivalent to doing T stochastic forward passes over the network during the testing process with dropout and then averaging the results. It is useful to view this method as an ensemble of approximated functions with shared parameters, which approximates the probabilistic Bayesian method known as deep GP. In this method, there are several outputs (considered 30, 50, 100, and 200 in this research) for a given input. Subsequently, uncertainty could be examined in terms of factors such as variance, entropy, and mutual information.

In the following, four examples are given to show the application of MC dropout in modelling subsurface fluid flow. The researchers in [11] investigated the uncertainty involved in ML seismic image segmentation models. A salt body detection was considered as an example. They used MC dropout and concluded the developed models were reliable.

The researchers in [12] used the dropout method for a classification problem to quantify the fault model uncertainty of a reservoir in Netherlands. The networks were trained with dropout ratios of 0.25 and 0.5. The researchers concluded that the model variance increased by increasing the dropout ratio. Also, they suggested training with more data is needed.

The MC dropout approach and a bootstrap aggregating method were used to quantify uncertainties of CO_2 saturation based on seismic data in [13]. The researchers carried out DL inversion experiments using noise-free and noisy data. The results showed that the model can estimate 2D distributions of CO_2 moderately well, and UQ can be done in real time.

A semi-supervised learning workflow was used to effectively integrate seismic data and well logs and simultaneously predicting subsurface characteristics in [14]. It had three distinct benefits: (i) using 3D seismic patterns for developing an optimal nonlinear mapping function with 1D logs, (ii) being capable of automatically filling the gap of vertical resolution between seismic and well logs, and (iii) having a MC dropout-based epistemic uncertainty analysis. The results of two examples

showed reliable seismic and well integration, and robust estimation of properties like density and porosity obtained by this procedure.

3. Case Study

Fluid flow in petroleum reservoirs is typically governed by: (i) the equation of mass conservation, (ii) momentum law (Darcy's law), (iii) energy equation, (iv) fluid phase behavior equations (also known as equations of state) and certain rock property relationships (such as compressibility) [15]. To solve this system of equations, it is necessary to specify boundary and initial conditions. Analytical (exact) solutions can be determined for relatively simple reservoirs (i.e., by making several assumptions). An alternative is to apply numerical (approximate) solutions, such as finite difference method, finite element method, finite volume method, spectral method, and meshless method.

A mixed GMsFEM framework, as a numerical method, has recently been proposed for a single-phase fluid in 2D heterogeneous (matrix composition and fracture distribution) porous media [16]. The model approximates reservoir pressure in multiscale space. It does so by applying several multiscale basis functions to a single coarse grid of the reservoir volume. The fluid velocity is directly estimated across a fine grid space. Generally, the number of Partial Differential Equations (PDEs) requiring solutions to enable multiscale basis functions to be derived is dependent on the number of local cell and local eigenvalue problems involved, which necessitates a substantial overhead. Therefore, it is reasonable to replace PDE solvers with ML/DL approaches, given their exceptional abilities and general acceptance in recent years. Readers are referred to [16] for additional information, especially what the original flow problem is and how the mixed GMsFEM works.

For the configuration defined in this paper, the computational domain was set to be $\Omega = [0,1] \times [0,1]$ (Figure 1). The fine grid system adopted involves a uniform 30×30 mesh. On the other hand, a sparser, uniform 10×10 mesh was applied to represent the coarse grid network. This configuration consists of 1300 separate PDEs, made up of 1200 (100×12) PDEs addressing the local cell problems plus 100 (100×1) local eigenvalue problems. There were five multiscale basis functions, identified as Basis 1, 2, 3, 4, and 5 for each generated permeability field (as the only input). A range of values for the permeability of the matrix was chosen from 1 to 5 milliDarcies (mD) incrementing in steps of 1 (i.e., 1, 2, 3, 4, and 5 mD); and for the permeability of the fracture from 500 to 2000 mD incrementing in steps of 250 (i.e., 500, 750, 1000, 1250, 1500, 1750, and 2000 mD). The number of fractures was set to 1, 2, 3, ..., 23, 24, and 25 (25 cases). Basis 1 is a piecewise constant, with binary values of -1 and +1. Basis 1 is defined as part of the finite element method, it hence requires no training for DL modeling. However, Basis 2, 3, 4, and 5 take values distributed across the range (-1, +1), and therefore require training for DL modeling.

In terms of supervised learning, our problem was mapping an input of 100×9 to an output of 900×1. Because there were four different basis functions, we had four distinct mappings. In this regard, 376,250 samples were produced in the MatLab software including 306,250 examples for the training, 35,000 for the validation, and 35,000 for the testing. Due to the random generation of the permeability fields, duplicates might have been present. Consequently, the generated dataset was filtered to remove any duplicate data records. This is necessary to remove the risk of introducing bias towards specific model configurations in the DL analysis. For our data, 1739 training, 579 validation, and 6121 testing samples were kept out. This reduced the training, validation and testing samples to 304,511, 34,421, and 28,879, respectively.

4. Skip Connection-based CNN Model Architecture

Depending on the way in which an algorithm learns from data sets, DL (and also ML) algorithms fall into four categories: (i) supervised, (ii) unsupervised, (iii) semi-supervised, and (iv) reinforcement. Our problem is a supervised learning task. There are several approaches that can be adopted in this category such as Recurrent Neural Networks (RNNs) and CNNs. RNNs are often applied to process video, sound, and text data. On the other side, CNNs are particularly designed for problems invovling 2D arrays like the regression case study in this research, where an input of 100×9 is mapped to an output of 900×1. The format defined for the permeability field was as a

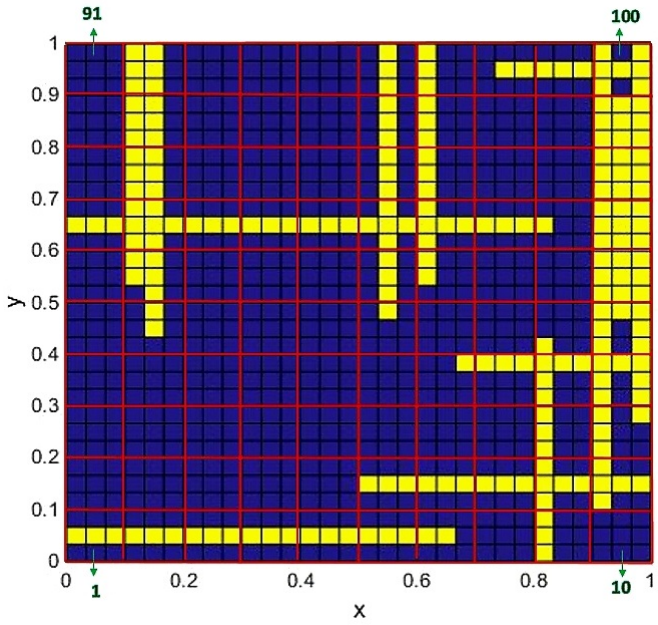

Figure 1. A typical permeability field of a fractured porous medium. The matrix permeability is assumed to be 4 mD. The fracture permeability is assumed to be 2000 mD. The fine grid squares represent the formation matrix (blue) in some cases and fractures (yellow) in other cases (selected randomly). The red lines define the coarse grid. Each coarse grid square contains of nine fine grid squares. There are fifteen fractures assigned to this porous medium.

vector (900×1), subsequently adjusted to be expressed as a 2D tensor (100×9), in which, coarse grid units=100 and each coarse grid contains 9 fine grids. Each row in the array therefore represents a coarse grid. Such a configuration enables the use of 2D CNN kernels. Furthermore, there was a logical and convincing mathematical procedure behind convolutional filters. Convolutional neural networks also automatically and adaptively learn the spatial hierarchies of features. Lastly, it can reduce the number of parameters without sacrificing model quality. With regards to the output, it was necessary to maintain the five basis functions as 900×1 vectors, so that they could be evaluated in the Fully Connected (FC) layers (dense layers) forming the final section of the CNN network.

A classic CNN model is normally composed of alternate convolutional and pooling layers, followed by one or more FC layers at the end. In some situations, it is sensible to replace an FC layer with a global average pooling layer. The convolutional and pooling layers perform feature extraction, while the FC layers map the extracted features into an output layer.

Distinct CNN model configurations, involving various combinations of convolutional, pooling, FC, Batch Normalization (BN), regularization, and dropout filtering were tested separately for each basis function requiring training (Basis 2, 3, 4, and 5). A similar optimal CNN configuration was obtained for each of those four basis functions (Figure 2). It consists of five convolutional layers, two FC layers but does not include any pooling layers. Each convolutional layer is followed by a single BN layer of the same dimensions. Typically, neural network models are able to apply higher learning rates and converge more quickly when the input to each layer is normalized; hence the value of adding the BN layers. Normalized input data tends to generate average (normalized) dependent variable prediction values that approximate zero with (normalized) standard deviations approximating one. The two FC layers contain 2000 neurons with a dropout rate of 0.05.

The gradient of the loss function might quickly approach zero when a deep NN back propagates the gradient from the final layers to earlier layers close to the input layer. This refers to the 'vanishing gradient problem', that makes the earlier layers not benefit from additional training. Using the skip connection (shortcut) strategy, which enables the gradient to be directly back propagated to earlier layers of a network, is one of the most effective ways to tackle this problem. After testing different

cases, we found out it would be better to add simultaneously two shortcut schemes to the main CNN structure: (i) from the middle to the last layer and (ii) from the middle to the second-to-last layer.

The model, treated as a Bayesian approach, produces a different output each time called with the same input. This is because each time a new set of weights is sampled from the distributions to develop the network and produce an output. Here, we arranged to have 30 outputs for a given input. The models employ the activation function of 'Rectified Linear Unit (ReLU)' for the convolutional layers, 'sigmoid' for the FC layers, and 'linear' for the output.

5. Evaluation

To understand the role of MC dropout in the developed CNN models based on accuracy, two statistical error metrics of R^2 and MSE are included:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2} \tag{6}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{7}$$

where y_i, $\bar{y}$, and $\hat{y}_i$ are the actual basis function of the i-th data point, the average of actual basis function for all samples, and the predicted basis function for the i-th data point, respectively. Also, N is the number of data points. As mentioned earlier, each basis function is in the form of a 900×1 tensor and R^2 of all outputs are averaged, weighted by the variances of each individual output. The R^2 value lies between $-\infty$ and 1 [17]. The closer the value is to 1, the more accurate the predictions produced by the model. The error metric of MSE measures the average of squares of errors (i.e., the difference between predicted and real values). It is basically non-negative, where values closer to zero indicate more-accurate performance. The models without dropout yield promising results when evaluated on the training subset using R^2 and MSE metrics. Except for Basis 5, the R^2 of others is above 0.9. The values obtained for MSE lie within the range of 0.0075 to 0.0243. The constructed models perform suitably for the validation subset, with an R^2 of 0.7900 to 0.8811 and MSE 0.0128 of to 0.0512. Because the validation and testing subsets were selected from the similar distribution of data, we can see almost the same results over the testing samples: an R^2 of 0.7857 to 0.8809, and MSE 0.0126 of to 0.0513.

According to Table 1, the dropout after two FC layers enhances performance over all subsets for all multiscale basis functions. For the training subset, it has the maximum effect on the model for Basis 3 and the minimum effect for Basis 4. For basis 3, R^2 increases from 0.9327 to 0.9584, and MSE decreases from 0.0141 to 0.0113. There is an R^2 increase from 0.9283 to 0.9326 and MSE decrease from 0.0107 to 0.0101 for Basis 4.

Adding dropout to the initial architecture has generally a marginal positive effect on the validation and testing samples. The range of R^2 and MSE is 0.7919-0.8858 and 0.0120-0.0507 for validation. The R^2 and MSE lie in the range of 0.7881-0.8839 and 0.0121-0.0508 for testing.

As a general result, it is evident that the use of dropout has a positive impact on the performance of the developed models across the training subset, regardless of the basis functions used. Furthermore, it also demonstrates a similar positive impact over the validation and testing subsets for Basis 3 and Basis 5. However, for Basis 2 and Basis 4, there is only a marginal difference between the performance of $CNN_{initial}$ and $CNN_{dropout}$ models. The probable reasons for this could be attributed to the high-dimensional regression problem considered in this paper and the complexity and non-linear nature of DL models. Nonetheless, even this slight improvement in the models' performance could help reduce overfitting and enhance generalization in the constructed $CNN_{dropout}$ models. Additionally, it can significantly affect the pressure distribution obtained through the basis functions.

Depending on the input/output dimensions, type (classification/regression), and approach applied to a problem, the magnitude of uncertainty can be analyzed statistically and graphically. Standard deviation measures the dispersion of a data set relative to its average. It is the square root of the variance. The closer the value of SD is to zero, the values of data are closer to the average. A high SD indicates that the values are spread out over a broad range. Basically, the variance and SD

Table 1. Performance of the developed models without dropout for Basis 2, 3, 4, and 5 in terms of R^2 and MSE.

Subset	Model	R^2				MSE			
		Basis 2	**Basis 3**	**Basis 4**	**Basis 5**	**Basis 2**	**Basis 3**	**Basis 4**	**Basis 5**
Training	$CNN_{initial}$	0.9002	0.9327	0.9283	0.8847	0.0243	0.0141	0.0107	0.0075
	$CNN_{dropout}$	**0.9113**	**0.9584**	**0.9326**	**0.9089**	**0.0211**	**0.0113**	**0.0101**	**0.0058**
Validation	$CNN_{initial}$	0.7900	0.8434	0.8811	0.8038	0.0512	0.0329	0.0176	0.0128
	$CNN_{dropout}$	**0.7919**	**0.8620**	**0.8858**	**0.8155**	**0.0507**	**0.0290**	**0.0170**	**0.0120**
Testing	$CNN_{initial}$	0.7857	0.8422	0.8809	0.8044	0.0513	0.0332	0.0177	0.0126
	$CNN_{dropout}$	**0.7881**	**0.8622**	**0.8839**	**0.8132**	**0.0508**	**0.0290**	**0.0173**	**0.0121**

is defined for a single-point data set (there is only one output). On the other hand, the output (basis functions) in this study is in the form of a 900×1 vector. While dealing with a vector, it is necessary to calculate the variance of each element of the vector separately. Then, the obtained variances are averaged to reach the total variance. Finally, the SD is obtained as the square root of the variance for each case. Standard CNNs (without dropout) give only one output for a given input. That is why the SD is not defined for such models (it is always zero).

According to Table 2, the SD values lie within 0.0181-0.158, 0.0179-0.152, 0.0169-0.104, and 0.0121-0.086 for the CNN models with dropout developed for Basis 2, 3, 4, and 5 based on the training subset. For all basis functions, most samples have an SD lower than 0.05. For instance, 221006 out of 304511 samples for Basis 3 are in the range of 0-0.05. In general, SD exceeds 0.15 only for 547 samples. The SD obtained for Basis 4 and 5 is lower than that for Basis 2 and 3.

With regards to the validation subset, the developed models for Basis 2, 3, 4, and 5 have an SD of 0.0268-0.174, 0.0237-0.124, 0.019-0.171, and 0.012-0.097, respectively. Generally, only 27 out of 34421 samples have an SD higher than 0.15. The model built for Basis 5 has the best performance in terms of uncertainty, 24276 samples with an SD of lower than 0.05 and 10145 samples with an SD of 0.05-0.1. After that, the models developed for Basis 4 and 3 come. The model designed for Basis 2 has the worst performance because only 2577 samples have an SD of 0-0.05.

For the testing subset, the SD values lie within 0.025-0.169, 0.024-0.142, 0.020-0.113, and 0.012-0.098 for the CNN models with dropout developed for Basis 2, 3, 4, and 5. The trend is the same as the validation subset. In other words, the model for Basis 5 has the best, and the one for Basis 2 has the worst performance, respectively. Also, there is no sample with an SD higher than 0.15, except with 24 cases for Basis 2.

As mentioned earlier, the output is in the form of a 900×1 vector, which is too big to show in a graph. Additionally, basis functions in the mixed GMsFEM are defined in one coarse grid element, which includes 9 fine grids. Figure 3 gives the 30 values obtained for each of nine points using MC dropout for a coarse grid with the matrix permeability of 1 mD (as a representative sample). The average of 30 outputs (for each point) is considered as the model's output. The figure demonstrates that the values are close to each other (some overlap) and have a very low SD.

In order to visualize the pressure changes over the defined computational domain, three examples are illustrated for selected training (Figure 4a), validation (Figure 4b), and testing (Figure 4c) subsets. The plots in the left-side columns display the permeability fields, for representative sample grids. The plots in the central columns display the pressure distribution derived by FEM (considered to be true distribution). The plots in the right-side columns display the predicted pressure distributions using the skip connection-based CNN models developed in this study. In fact, the pressure is obtained through the multiscale basis functions. Generally, there is a better match for the training sample in comparison with the validation and testing cases.

Table 2. Reliability of the developed models for Basis 2, 3, 4, and 5 using MC dropout in terms of SD.

Subset	Range of SD	Basis 2	Basis 3	Basis 4	Basis 5
Training	[0-0.05)	197572(0.649)	221006(0.726)	238107(0.782)	261537(0.859)
	[0.05-0.1)	99364(0.326)	81227(0.267)	63189(0.208)	42974(0.141)
	[0.1-0.15)	7143(0.023)	2163(0.007)	3215(0.01)	-
	≥ 0.15	432	115	-	-
Validation	[0-0.05)	2577(0.075)	4679(0.136)	7475(0.217)	24276(0.705)
	[0.05-0.1)	19296(0.561)	29395(0.854)	26937(0.783)	10145(0.295)
	[0.1-0.15)	12522(0.364)	347(0.01)	8	-
	≥ 0.15	26	1	1	-
Testing	[0-0.05)	2245(0.079)	3984(0.138)	6232(0.216)	20725(0.718)
	[0.05-0.1)	16321(0.565)	24599(0.852)	22641(0.784)	8154(0.282)
	[0.1-0.15)	10289(0.356)	296	6	-
	≥ 0.15	24	-	-	-

6. Discussion

In terms of accuracy, it was perceived that considering high initial sets of weights does not influence the accuracy of the models. More specifically, no meaningful improvement was observed by defining 50, 100, and 200 sets. Hence, the number of 30 sets considered here seems almost optimal given the developed models' high accuracy and low SD for multiscale basis functions.

In a standard deterministic NN, a single prediction is obtained for a given input, with no information about the uncertainty of the used data or the model fitness. This is because only one initial set of weights/biases is used/updated in such models. The Bayesian methods can be applied to tackle this issue somewhat, taking a positive step towards the reliability of NN models. Bayesian neural networks are different from standard NNs in that their weights are assigned a probability distribution rather than a single value or point estimate. These probability distributions describe the uncertainty in weights and can be used to estimate uncertainty in predictions. In this research, we used the MC dropout only for the FC layers of the CNN structures. In other words, the dropout technique was not used regarding the convolutional layers because it negatively affected on the accuracy of the models. Moreover, although multiple techniques were used to quantify the data uncertainty, we got some errors. So, it would be better to consider both uncertainty sources to construct more reliable CNN models.

In terms of UQ statistical investigation, we defined several indices for uncertainty such as entropy, Negative Log Likelihood (NLL), and SD for the statistical measures. However, the values obtained for entropy and NLL were meaningless. Therefore, it would be helpful to use more applicable statistical measures to convey the information about the uncertainty more meaningfully.

7. Conclusions

Standard deterministic deep NNs converge on a one input-one output basis, with no information about the uncertainty of the data or model fitness. Bayesian approaches are effective in uncertainty estimations. However, they face a high computational cost when applied to large datasets. That is why MC dropout, a computationally more efficient method, was used in this study as a positive step

towards the reliability of skip connection-based CNN models based on 376,250 samples from the oil/gas domain. The SD values obtained confirm the robustness of MC dropout in terms of epistemic uncertainty, in addition to the high degree of accuracy. There are two suggestions for mitigating the limitations of the present study: (i) quantifying the aleatoric uncertainty for the developed models, and (ii) using more dropout ratios and comparing it with the ratio of 0.05 considered here.

Author Contributions: Conceptualization, A.C. and J.C.; methodology, A.C. and F.C.; formal analysis, A.C., and F.C.; data curation, A.C. and J.C.; writing original draft, A.C.; writing—review and editing by A.C., and F.C.; visualization, A.C.; supervision, J.C., F.C. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Key Program Special Fund in XJTLU (KSF-E-50), XJTLU Postgraduate Research Scholarship (PGRS1912009), and XJTLU Research Development Funding (RDF-19-01-15).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and the codes can be obtained from the corresponding authors upon reasonable request.

Acknowledgments: We appreciate the University of Liverpool for providing us with a powerful computer system to run our codes. Also, we are grateful to Ms. Roslyn Joy Irving for her precious comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Daryasafar, A.; Daryasafar, N.; Madani, M.; Kalantari Meybodi, M.; Joukar, M. Connectionist approaches for solubility prediction of n-alkanes in supercritical carbon dioxide. *Neural Computing and Applications* **2018**, *29*, 295–305.
2. Wood, D.A.; Choubineh, A. Transparent machine learning provides insightful estimates of natural gas density based on pressure, temperature and compositional variables. *Journal of Natural Gas Geoscience* **2020**, *5*, 33–43.
3. Wood, D.A. Trend decomposition aids forecasts of air particulate matter (PM2. 5) assisted by machine and deep learning without recourse to exogenous data. *Atmospheric Pollution Research* **2022**, *13*, 101352.
4. Suleymanov, V.; Gamal, H.; Elkatatny, S.; Glatz, G.; Abdulraheem, A. Machine Learning Models for Acoustic Data Prediction During Drilling Composite Lithology Formations. *Journal of Energy Resources Technology* **2022**, *144*, 103201.
5. Abdollahi, A.; Amini, A.; Hariri-Ardebili, M.A. An uncertainty-aware dynamic shape optimization framework: Gravity dam design. *Reliability Engineering & System Safety* **2022**, *222*, 108402.
6. Xu, C.; Nait Amar, M.; Ghriga, M.A.; Ouaer, H.; Zhang, X.; Hasanipanah, M. Evolving support vector regression using Grey Wolf optimization; forecasting the geomechanical properties of rock. *Engineering with Computers* **2020**, pp. 1–15.
7. Cao, L.; Zheng, X.; Fang, L. The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm. *Electronics* **2023**, *12*, 929.
8. Shanmugavel, A.B.; Ellappan, V.; Mahendran, A.; Subramanian, M.; Lakshmanan, R.; Mazzara, M. A Novel Ensemble Based Reduced Overfitting Model with Convolutional Neural Network for Traffic Sign Recognition System. *Electronics* **2023**, *12*, 926.
9. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* **2021**, *76*, 243–297.
10. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the international conference on machine learning. PMLR, 2016, pp. 1050–1059.
11. Zhao, T.; Chen, X. Enrich the interpretation of seismic image segmentation by estimating epistemic uncertainty. In *SEG Technical Program Expanded Abstracts 2020*; Society of Exploration Geophysicists, 2020; pp. 1444–1448.

12. Feng, R.; Grana, D.; Balling, N. Uncertainty quantification in fault detection using convolutional neural networks. *Geophysics* **2021**, *86*, M41–M48.
13. Um, E.S.; Alumbaugh, D.; Lin, Y.; Feng, S. Real-time deep-learning inversion of seismic full waveform data for CO2 saturation and uncertainty in geological carbon storage monitoring. *Geophysical Prospecting* **2022**.
14. Di, H.; Abubakar, A. Estimating subsurface properties using a semisupervised neural network approach. *Geophysics* **2022**, *87*, IM1–IM10.
15. Fanchi, J.R. *Principles of applied reservoir simulation*; Elsevier, 2005.
16. Chen, J.; Chung, E.T.; He, Z.; Sun, S. Generalized multiscale approximation of mixed finite elements with velocity elimination for subsurface flow. *Journal of Computational Physics* **2020**, *404*, 109133.
17. Neter, J.; Kutner, M.H.; Nachtsheim, C.J.; Wasserman, W.; et al. Applied linear statistical models **1996**.

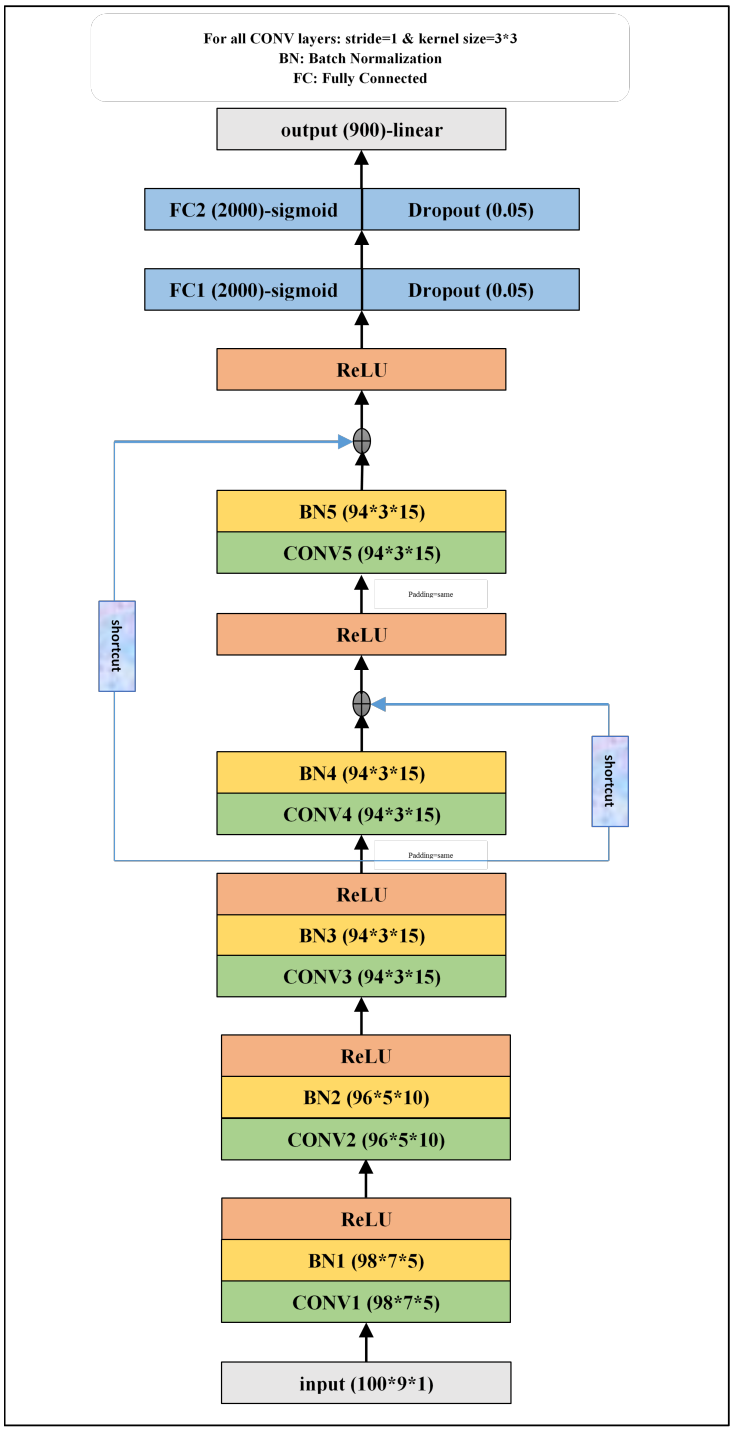

Figure 2. Structure of the skip connection-based CNN model used in this study.

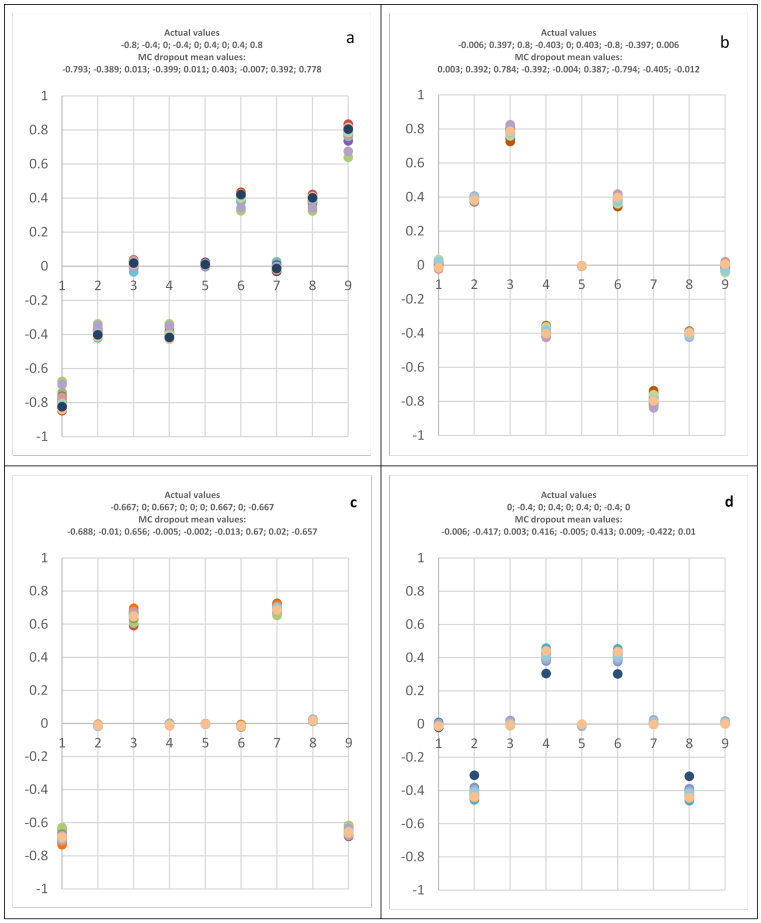

Figure 3. Values dispersion of a representative coarse grid: (a) for Basis 2, (b) for Basis 3, (c) for Basis 4, and (d) for Basis 5.

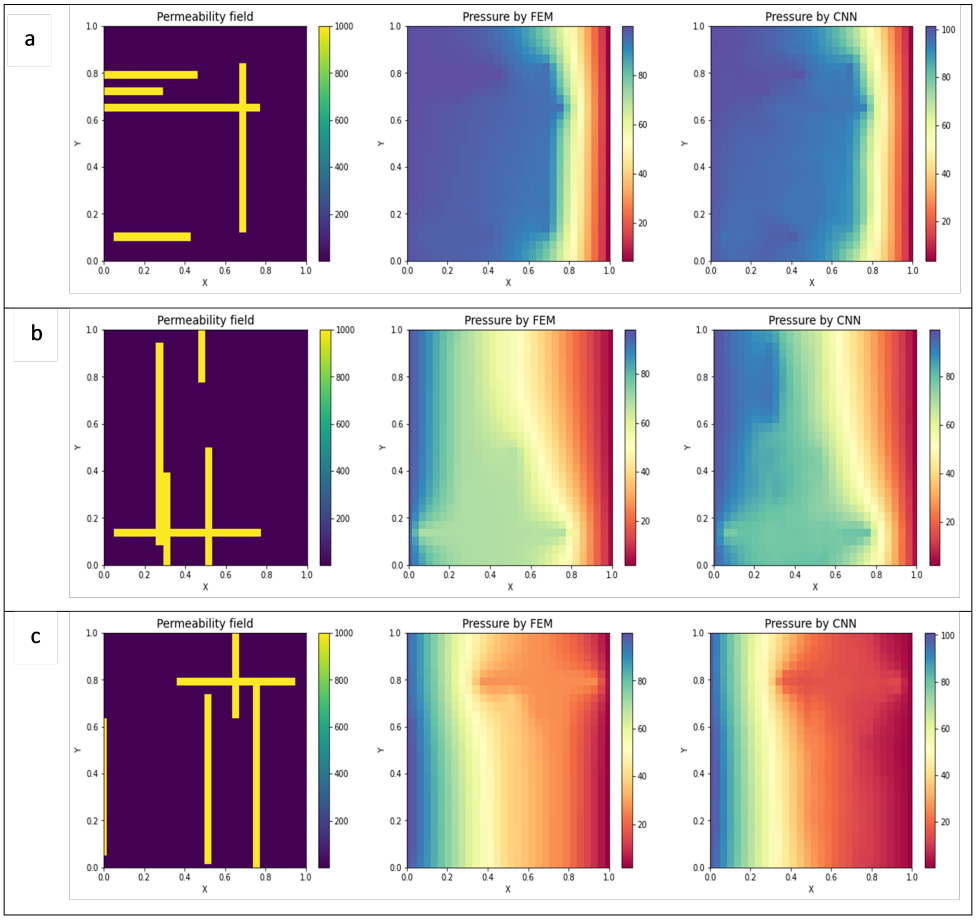

Figure 4. A comparison between the actual pressure distributions and those obtained by the skip connection-based CNN models: (a) training sample, (b) validation sample, and (c) testing sample.