Neural SDE-based Epistemic Uncertainty Quantification in Deep Neural Networks

Aabila Tharzeen[1], Shweta Dahale[2] and Balasubramaniam Natarajan[1]

[1] Electrical and Computer Engineering, Kansas State University, KS, USA
{aabilatharzeen,bala}@ksu.edu
[2] Eaton Research Labs, CO, USA.
shwetaddahale@eaton.com

Abstract. Deep learning tools are now widely used across various areas due to the increasing interest in applied machine learning. While these tools demonstrate exceptional performance in prediction and classification tasks, they are often deployed as black-box inferencing entities without any precise measure of uncertainty associated with their outputs. Uncertainty quantification is essential for ensuring reliability and robustness, particularly in safety-critical applications. However, accurately quantifying model/epistemic uncertainty in machine learning-based regression and classification tasks is challenging. In this paper, we provide an analytical approach to quantify the epistemic uncertainty related to deep neural network models using neural stochastic differential equations. Through experiments carried out on synthetic data, we demonstrate that our proposed framework successfully addresses the challenge of representing uncertainty in deep neural network-based regression and classification without the computational complexity associated with the classic Monte Carlo dropout method.

Keywords: Uncertainty quantification · deep neural network · Neural stochastic differential equation

1 Introduction

Deep neural networks, or DNNs, have become highly effective models for handling challenging problems. They exhibit exceptional performance in areas including speech recognition, image classification, and natural language processing. However, despite their remarkable success, DNNs face several challenges that limit their reliability and interoperability [1]. These challenges include their vulnerability to adversarial attacks, their tendency to overfit the training data, and the lack of transparency in understanding their decision-making process. These limitations can have significant implications, particularly in safety-critical applications such as healthcare and autonomous vehicles, where incorrect predictions or unreliable decisions can lead to severe consequences [20]. Researchers have developed uncertainty quantification (UQ) methods for estimating and quantifying the uncertainty associated with DNN predictions. These methods aim to go beyond providing point estimates and instead provide measures of uncertainty, such

as confidence intervals or probability distributions over the predictions. Quantifying uncertainty in DNNs enables better decision-making, risk assessment, and model interpretability in real-world applications.

Despite the growing importance of uncertainty quantification for DNNs, the field is still in its early stages [8], and several challenges need to be addressed. One key challenge is developing scalable and efficient UQ methods that can handle large datasets and complex models. Additionally, there is a need for UQ methods that can provide interpretable uncertainty estimates, allowing users to understand and trust the predictions made by DNNs [20]. Furthermore, it is essential to integrate UQ methods seamlessly into existing DNN architectures to ensure practical applicability.

In this paper, we propose a novel framework that can analytically quantify the epistemic uncertainty of a neural network. Specifically, we use a surrogate neural stochastic differential equation (Neural SDE) framework that allows the derivation of the output mean and covariance along all neural network layers. We present a paradigm that can capture and measure epistemic uncertainty effectively, offering useful insights into the predictability of neural networks. This improved understanding of uncertainty will contribute to improved decision-making and foster trust in the application of neural networks in various domains.

The main contributions of this paper can be summarized as follows:

- We propose a generic framework that quantifies the epistemic uncertainty of deep neural networks with the help of a neural SDE framework as a surrogate model applicable across various tasks and domains.
- The framework allows for uncertainty propagation through all the layers of the target neural network model, and the analytical results capturing the uncertainty (output mean and covariance) along each neural network layer are derived.
- It can be applied to pre-trained networks using the analytical method, which eliminates the need for any computationally demanding uncertainty quantification procedures.
- The effectiveness of the proposed framework is demonstrated using synthetic data for classification and regression tasks and the results show up to 88.45% reduction in UQ computational complexity relative to the state-of-the-art Monte Carlo dropout method.

2 Background and Related work

Deep neural networks (DNNs) are a type of artificial neural network trained using large datasets, and optimization algorithms to learn the parameters that map input data to output predictions. However, a significant challenge in deploying DNNs lies in dealing with uncertainty. Various methods have been proposed to quantify uncertainty in DNNs. Existing uncertainty quantification techniques can be divided into (1) Bayesian approaches or (2) sampling-based techniques. A classic neural network model with input x and output y and network parameter θ

can be written probabilistically as $p(y|x, \theta)$. The posterior distribution obtained by applying Bayes' theorem can be written as follows:

$$p(\theta|x, y) = \frac{p(y|x, \theta)p(\theta)}{\int p(y|x, \theta)p(\theta)d\theta} \tag{1}$$

Existing approaches use Bayesian inference to estimate the posterior distribution over model parameters instead of treating the model parameters as fixed [3]. However, finding the posterior distribution over all possible model parameters can be intractable. There exists variation inference-based techniques that approximate the posterior probabilities rather than finding the exact value [6]. The process involves finding a simple distribution that is as close to the actual distribution. Using this technique training takes longer and inference is slower as approximating posterior probability requires several samples to be drawn from the posterior distribution. Laplace approximation is also used in Bayesian inference, focusing on the region around the maximum of the posterior distribution. By leveraging a Taylor-series expansion, it provides an approximation of the distribution in the vicinity of this maximum [15]. This method can only capture the local behavior of the distribution, which implies that the approximation may significantly differ from the true distribution in other areas. Among sampling-based techniques, another efficient technique for approximating inference is the Monte Carlo Markov chain (MCMC) method, which involves applying a stochastic transition to a random draw from a distribution. [10]. Despite the success of the MCMC method, the method convergences slowly. Monte Carlo dropout is yet another technique that uses dropout as a Bayesian Approximation to calculate the intractable posterior distribution [5]. Monte Carlo (MC) dropout, which involves running inference multiple times with different dropout masks during testing can be used to estimate the epistemic uncertainty. Although dropout is a widely used technique, it is an empirical approach that gets computationally expensive for larger networks. The Deep Ensembles method proposed in [11] involves training multiple DNNs on the same dataset and combining their predictions to obtain the mean and variance of the predictions and can be used as a metric for the uncertainty. However, the existing methods are time-consuming and require maintaining many copies of the model parameters to quantify uncertainty, which can be costly for large NNs. In [9], stochastic differential equations (SDEs)-based techniques were employed to train the SDE model using out-of-distribution (OOD) data for training in order to quantify total uncertainty. However, the method falls short in its ability to assess uncertainty in a rigorous and principled manner.

In this paper, we aim to address these challenges and propose a novel approach to uncertainty quantification in neural networks (UQ-Net) using a neural SDE framework. Our objective is to develop a method that provides a rigorous and principled way to measure epistemic uncertainty. By leveraging the neural SDE framework, our approach overcomes the limitations of existing methods and offers a more efficient and scalable solution for quantifying uncertainty in large neural networks.

3 Proposed UQ-Net approach

This section presents a detailed outline of the problem statement along with an introduction to the main elements of our suggested structure.

3.1 Problem Statement

Given a training dataset $D = (x_j, y_j)_{j=1}^{N}$, where x_j represents the input data and y_j corresponds to the corresponding ground truth labels, we can train a model $\mathbf{M}$ parameterized by θ to make predictions on new test cases. The predictive uncertainty originates from two sources as described earlier. Our goal is to analytically quantify the epistemic uncertainty of the neural network model with reduced computational time in comparison to the existing methods.

3.2 Uncertainty quantification with UQ-Net

Given a target neural network model that performs a regression or classification task on a set of data, the UQ-Net works as a surrogate network and helps to measure how uncertain the neural network model's predictions are. As shown in figure 1, the UQ-Net, which has an input, output, and a hidden layer, helps in propagating the uncertainty in the corresponding layers of the target neural network. As shown in figure 2, the UQ-Net architecture consists of input and output linear layers of deep neural networks at both the input and output stages. The neural SDE layer within UQ-Net corresponds to the hidden layer of the deep neural network. Since neural SDEs preserve dimensionality, the input and output layers of the network serve as reshaping layers before and after the hidden layer. These reshaping layers facilitate the seamless integration of the neural SDE layer into the overall network structure and help UQ-Net propagate the mean and covariance analytically from input to output. Each component of UQ-Net is further elaborated in the following sections, along with the analytical derivation of the output mean and covariance.

3.3 Neural SDE

Traditional neural networks typically consist of multiple stacked hidden layers that map input $\mathbf{x}$ to output $\mathbf{y}$. However, recent research has shown that these hidden representations can be interpreted as the states of a continuous dynamical system rather than discrete layers [4]. For instance, in the case of a residual neural network (a deep learning model in which each layer learns residual functions with respect to their input) [7], the transformation between layers can be expressed as

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, \theta) \tag{2}$$

Here, $\mathbf{h}_t$ represents the hidden features at depth t, and $\mathbf{f}$ is a neural network function approximation. The continuous limit of the residual neural network structure is expressed as

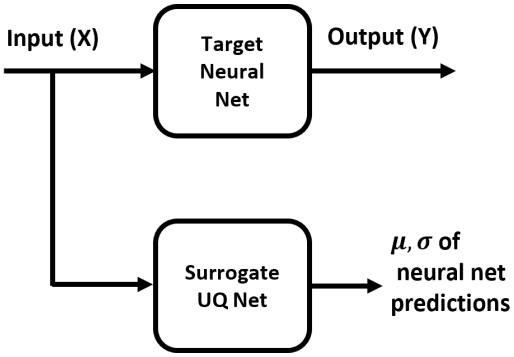

Fig. 1. Neural network with uncertainty quantification module (UQ-Net)

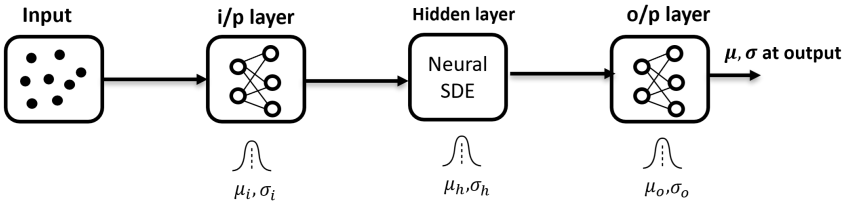

Fig. 2. UQ-Net

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \int_t^{t+1} \mathbf{f}(\mathbf{h}_\tau, \tau, \theta) d\tau \tag{3}$$

The equation (3) is the continuous approximation of the transformations in residual neural network architecture and also the solution to an ordinary differential equation (ODE) problem. The neural ordinary differential equation (Neural ODE) method [4] parameterizes $\mathbf{f}(\mathbf{h}_\tau, \tau, \theta)$ with a neural network and leverages an ODE solver to evaluate the hidden states during the continuous transformation. However, the deterministic nature of neural ODEs is not suitable to model epistemic uncertainty. To address this, we employ the neural SDE model [13], which augments a neural ODE with a stochastic term. Thus, neural SDE incorporates a diffusion term to model randomness using the Brownian motion component [13]. Thus equation (3) can be rewritten as follows:

$$d\mathbf{h}_t = \underbrace{\mathbf{f}(\mathbf{h}_t, t; w)dt}_{\text{drift}} + \underbrace{\mathbf{g}(\mathbf{h}_t, t; v)d\mathbf{B}_t}_{\text{diffusion}} \tag{4}$$

Here, $\mathbf{g}(\mathbf{h}_t, t; v)d\mathbf{B}_t$ represents the Brownian motion variance, which helps to capture the epistemic uncertainty of the hidden state $\mathbf{h}_t$ at depth t. The drift and diffusion terms from the equation are functions that are approximated via neural networks with w and v as the corresponding network parameters. Equation (4) is

a general form that can be modified to incorporate existing randomness models, such as dropout in neural networks.

Dropout using neural SDE: The dropout term randomly deactivates some neurons in the neural network and can be modeled using neural SDE [14]. Mathematically, the dropout rate can be incorporated into equation (3) as $\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) \odot \frac{\gamma_n}{p}$, where γ_n is drawn from i.i.d Bernoulli distribution with parameter p and $\odot$ indicates the Hadamard product. Furthermore, the above hidden state dynamic can be simplified as,

$$\begin{aligned}\mathbf{h}_{t+1} &= \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) + \mathbf{f}(\mathbf{h}_t, t; w) \odot (\frac{\gamma_n}{p} - \mathbf{I}), \\ &= \mathbf{h}_t + \mathbf{f}(\mathbf{h}_t, t; w) + \mathbf{f}(\mathbf{h}_t, t; w) \odot \sqrt{\frac{1-p}{p}} \boldsymbol{z}_n. \end{aligned} \tag{5}$$

where $\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. To compare the quantified epistemic uncertainty with the state-of-the-art Monte Carlo dropout method, we utilize equation (5), which unifies dropouts under the neural SDE framework. Here, $\sqrt{\frac{1-p}{p}}\mathbf{f}(\mathbf{h}_t, t; w)$ represents the diffusion term and $q = \sqrt{\frac{1-p}{p}}$ controls the strength of regularization. It determines the scaling factor of the diffusion term and affects the magnitude of the injected noise or uncertainty. By adjusting the value of q, one can balance the trade-off between regularization and the expressiveness of the model. q can be treated similarly to the dropout probability term in the MC dropout method [5].

3.4 UQ-Net

As discussed above, the mean and covariance are computed from the input layer and subsequently propagated through the hidden layer, effectively carrying the uncertainty measures to the output layer with the help of UQ-Net. The input and output layers can be linear or nonlinear functions depending on the presence of an activation function.

Uncertainty estimation in input/output layer: Quantification of uncertainty for a non-linear input/output layer can be achieved by computing the statistics of the linearized approximation of the nonlinear function [2]. Suppose that for a nonlinear function $\mathbf{f}$ and a random vector $\mathbf{x}$ with expected value $\hat{\mathbf{x}}$ and covariance $\mathbf{C}_{xx}$, the expected value $\hat{y}$ of the output $\mathbf{y} = \mathbf{f}(\mathbf{x})$ can be written as a Taylor series expansion around the mean as $\hat{\mathbf{y}} \approx \mathbf{f}(\hat{\mathbf{x}})$

Here $\nabla_x \mathbf{f}$ is the Jacobian of the function $\mathbf{f}$ at the operating point $\mathbf{x}$. Theorem 1 below can be used to evaluate the expected value of the mean and covariance of the output from the input/output layers of the neural network.

Theorem 1. Let $\mathbf{y} = \mathbf{f}(\mathbf{x}, \theta)$ be the output of the neural network layer denoted by $\mathbf{f}$, with the network layer parameter θ. The estimation of expected value ($\hat{\mathbf{y}}$) and the associated covariance can be calculated as,

$$\hat{\mathbf{y}} = \mathbf{f}(\hat{\mathbf{x}}, \theta),$$

$$\mathbf{C}_{yy} = (\nabla_x f)\mathbf{C}_{xx}(\nabla_x f)^\intercal \tag{6}$$

Here, the expected value of the input is $\mathbb{E}[\mathbf{x}] = \hat{\mathbf{x}}$ and its associated covariance is $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ is the covariance associated with the output of the neural network layer.

The uncertainty estimates obtained using Theorem 1 are propagated through the input layer and are passed to the hidden layer. Theorem 1 can also be applied to uncertainty estimation at the output layer.

Uncertainty estimation in hidden layer: Uncertainty quantification of the hidden layer of the neural network is achieved using Gaussian assumed density approximation of Neural SDE [18]. By utilizing linearization techniques, we can compute the statistics of nonlinear neural SDE. Specifically, the nonlinear neural SDE can be linearized via Taylor series approximation. The drift term can be approximated by linearizing around the mean $\mathbf{m}$ and at depth t as follows:

$$f(\mathbf{h}, t) \approx f(\mathbf{m}, t) + \mathbf{F}_{\mathbf{h}}(\mathbf{h}, t)(\mathbf{h} - \mathbf{m}) \tag{7}$$

And the diffusion term can be linearized as:

$$g(\mathbf{h}, t) \approx g(\mathbf{m}, t) + \mathbf{G}_{\mathbf{h}}(\mathbf{h}, t)(\mathbf{h} - \mathbf{m}) \tag{8}$$

Here $\mathbf{F}_{\mathbf{h}}$ and $\mathbf{G}_{\mathbf{h}}$ are the Jacobian of f and g with respect to the hidden state $\mathbf{h}$ and the statistics of the hidden state can be calculated using Theorem 2.

Theorem 2. Consider a neural SDE as shown in equation (4), a linearization-based approximation to neural SDE can be obtained by integrating the equation,

$$\frac{d\mathbf{m}}{dt} = f(\mathbf{m}, t) \tag{9}$$

$$\frac{d\mathbf{C}}{dt} = \mathbf{C}\mathbf{F}_{\mathbf{h}}^\intercal(\mathbf{m}, t) + \mathbf{F}_{\mathbf{h}}(\mathbf{m}, t)\mathbf{C} + \mathbf{G}_{\mathbf{h}}(\mathbf{m}, t)\mathbf{Q}\mathbf{G}_{\mathbf{h}}^\intercal(\mathbf{m}, t) \tag{10}$$

where $\mathbf{m}$ and $\mathbf{C}$ are the mean and covariance of the states of the neural SDE at depth t and $\mathbf{Q}$ is the diffusion matrix.

Equations (9) and (10) enable the analytical determination of the mean and covariance of the hidden layer output in the neural network. It is to be noted that the diffusion term, denoted as g, is proportional to the function $\mathbf{f}$ in our particular context and is defined as $g = \sqrt{\frac{1-p}{p}} f(h_t, t; w)$. Thus, the uncertainty quantification of a deep neural network can be achieved analytically using UQ-Net.

4 Experimental Results

In this section, we conduct a comprehensive evaluation of our proposed uncertainty quantification method through experiments on two synthetic datasets and two real-world datasets: two designed for regression tasks and the other two for classification tasks. Detailed descriptions of these datasets are provided below:

Synthetic Regression dataset: This dataset is generated synthetically and is intended for regression tasks. It comprises of a single feature $\mathbf{x}$ and consists of randomly generated homoscedastic data.

Half Moon dataset: This synthetic dataset consists of two features and is specifically designed for performing classification tasks. Each data point in the dataset can be classified into one of two distinct classes.

Housing dataset: The dataset is drawn from the 1990 U.S. Census reflecting real-world data. It consists 8 distinct features and is utilized to forecast the median house value within California districts [16]

Ionoshpehere dataset: This real-world dataset consists of 34 feature which classifies radar returns from the ionosphere into three different categories [19].

By utilizing these datasets, we aim to assess the performance and effectiveness of our proposed uncertainty quantification method. To assess the effectiveness of our proposed uncertainty quantification method, we compare its performance against the widely adopted Monte Carlo dropout technique. All training and evaluation experiments are performed on a computer with an Intel i7 processor running at 2.80 GHz with 12GB memory and 12GB RAM. The learning rate set was 0.001 and the Adam optimizer was used.

Regression task The regression task is performed on the single feature synthetic regression dataset. The uncertainty estimates obtained using UQ-Net is compared with the MC dropout method using the expected normalized calibration error (ENCE) metric [12]. The ENCE metric serves as an indicator of the reliability of the confidence scores provided by the methods. The ENCE metric is utilized to calibrate the regressor by aligning the mean square error (MSE), representing the expected error, with the predicted uncertainty, denoted by the standard deviation σ. To assess the calibration of the regressor, the standard deviation axis is divided into bins, and the ENCE metric is calculated using the following formula:

$$\mathrm{ENCE} = \frac{1}{N} \sum_{j=1}^{N} \frac{|\mathrm{RMV}(j) - \mathrm{RMSE}(j)|}{\mathrm{RMV}(j)} \tag{11}$$

Here, Bj is a bin that represents the standard deviation axis interval, root mean variance(RMV) and root mean square error (RMSE) are obtained using the following equations:

$$\mathrm{RMV}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} \sigma_t^2 \tag{12}$$

$$\mathrm{RMSE}(j) = \frac{1}{|B_j|} \sum_{t \in B_j} (y_t - \hat{y}_t)^2 \tag{13}$$

A lower ENCE value indicates a better-calibrated model. To assess the performance of UQ-Net, we calculate the ENCE score for various values of p, and the results are shown in Table 1. Similar experiments are conducted for the MC Dropout method, and the corresponding results are also presented in Table 1. From the table, it becomes evident that by adjusting the values of p, both methods effectively quantify the epistemic uncertainty and exhibit a similar trend. However, the advantage of UQ-Net lies in its analytical nature, resulting in significantly reduced computational time required to obtain uncertainty estimates compared to the empirical MC Dropout approach. The computational time required for both methods is summarized in Table 3.

For achieving optimal performance, the commonly chosen dropout value for hidden layers using MC Dropout is 0.5. However, the optimal dropout value may vary depending on various factors such as data size, model architecture, etc. [17]. Similarly, in the case of UQ-Net, there can be an optimal value of q that can be chosen to yield a lower ENCE score, making the choice of hyperparameters crucial for obtaining reliable uncertainty quantification results. We further asses the performance of our framework using real-world data from the California housing dataset. The results are detailed in Table 2.The trend observed in Table 1 for regression tasks is similarly evident in the evaluation using real-world datasets shown in Table 2. The computational time needed for assessing both UQ-Net and MC dropout is illustrated in Table 4.

Table 1. ENCE scores obtained using two methods for regression and classification task for synthetic dataset

p	Regression		Classification	
	UQ-Net	**MC dropout**	**UQ-Net**	**MC dropout**
0.1	0.760	0.804	0.779	2.685
0.2	0.758	1.608	0.632	1.981
0.3	0.767	0.656	0.693	0.977
0.4	0.747	2.423	0.916	0.992
0.5	0.77	0.300	0.990	0.426
0.6	0.718	0.401	0.607	0.439
0.7	0.246	0.184	0.692	0.638
0.8	0.150	0.492	0.755	0.338
0.9	0.607	0.516	1.343	0.476

Classification task In the classification task, we utilize the half-moon dataset with two features. Just like in the regression experiments, we compare the uncertainty estimates obtained using UQ-Net with those from the MC Dropout

Table 2. ENCE scores obtained using two methods for regression and classification task for real-world dataset

p	Regression		Classification	
	UQ-Net	**MC dropout**	**UQ-Net**	**MC dropout**
0.1	0.43	6.14	0.48	12.08
0.2	0.60	4.35	0.72	9.64
0.3	0.57	3.60	0.51	6.75
0.4	0.59	3.02	0.59	4.70
0.5	0.62	2.65	0.50	4.14
0.6	0.63	2.09	0.64	6.21
0.7	0.67	1.72	0.93	3.82
0.8	0.65	1.13	0.38	1.81
0.9	0.61	0.76	1.50	1.29

Table 3. Computation time in seconds required for regression and classification task for synthetic dataset

Regression		Classification	
UQ-Net	**MC dropout**	**UQ-Net**	**MC dropout**
0.83 sec	4.88 sec	0.41 sec	6.47 sec

Table 4. Computation time in seconds required for regression and classification task for real-world dataset

Regression		Classification	
UQ-Net	**MC dropout**	**UQ-Net**	**MC dropout**
8.79 sec	79.07 sec	0.37 sec	3.46 sec

method. While the ENCE score is used for evaluating uncertainty in regression tasks, it cannot be applied directly to classification tasks, as the RMSE loss is not relevant for classification. However, classifiers can predict continuous scores that are often transformed into class labels through a thresholding process during the final step of classification. To evaluate uncertainty estimates in our specific framework, we utilize the continuous output prior to this final step. Thus, the ENCE score is suitable for our specific case in evaluating uncertainty estimates for classification tasks.

Similar to the regression task, ENCE scores are calculated for various values of p for both UQ-Net and MC Dropout, and the results are depicted in Table 1. As previously discussed, an optimal value of p can be selected based on a lower ENCE score. Similarly, an optimal value of q can also be chosen. The computational time required for both methods is presented in Table 3 and similar to the case of regression task there is up to 94% reduction in computation time compared to MC dropout method.

We also evaluate our framework's performance for classification using real-world data from the ionosphere dataset. The results are presented in Table 2. The trend observed in Table 1 for classification tasks is also evident in the evaluation using real-world datasets, as shown in Table 2. The computational time required for assessing both UQ-Net and MC dropout is provided in Table 4.

5 Conclusions

Uncertainty quantification plays a critical role in safety-critical applications involving deep neural networks. This paper presents a theoretical approach to uncertainty quantification in neural networks using a neural stochastic differential equation framework. The proposed framework allows for the analytical quantification of epistemic uncertainty in DNNs for various tasks like classification and regression. The results were compared with the state of art MC dropout uncertainty quantification technique. The results obtained highlight the effectiveness of UQ-Net in accurately quantifying uncertainty while offering the advantage of reduced computational time compared to MC Dropout. This further demonstrates the practicality and efficiency of our proposed analytical approach in uncertainty quantification tasks. Future work includes exploring the use of parametric dynamic models instead of neural SDEs for uncertainty quantification, which can potentially improve the interpretability and scalability of the methods.

Acknowledgements The research was funded by National Science Foundation CNS 2039014

References

1. 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 **76**, 243–297 (2021)

2. Amini, A., Liu, G., Motee, N.: Robust learning of recurrent neural networks in presence of exogenous noise. In: 2021 60th IEEE Conference on Decision and Control (CDC). pp. 783–788. IEEE (2021)
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International conference on machine learning. pp. 1613–1622. PMLR (2015)
4. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in neural information processing systems **31** (2018)
5. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning [eb/ol]. arXiv preprint arxiv:1506.02142 (2015)
6. Graves, A.: Practical variational inference for neural networks. Advances in neural information processing systems **24** (2011)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. corr abs/1512.03385 (2015) (2015)
8. He, W., Jiang, Z.: A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective. arXiv preprint arXiv:2302.13425 (2023)
9. Kong, L., Sun, J., Zhang, C.: Sde-net: Equipping deep neural networks with uncertainty estimates. arXiv preprint arXiv:2008.10546 (2020)
10. Kupinski, M.A., Edwards, D.C., Giger, M.L., Metz, C.E.: Ideal observer approximation using bayesian classification neural networks. IEEE transactions on medical imaging **20**(9), 886–899 (2001)
11. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. arxiv. arXiv preprint arXiv:1612.01474 (2016)
12. Levi, D., Gispan, L., Giladi, N., Fetaya, E.: Evaluating and calibrating uncertainty prediction in regression tasks. Sensors **22**(15), 5540 (2022)
13. Liu, X., Xiao, T., Si, S., Cao, Q., Kumar, S., Hsieh, C.J.: Neural sde: Stabilizing neural ode networks with stochastic noise. arXiv preprint arXiv:1906.02355 (2019)
14. Liu, X., Xiao, T., Si, S., Cao, Q., Kumar, S., Hsieh, C.J.: How does noise help robustness? explanation and exploration under the neural sde framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 282–290 (2020)
15. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
16. Pace, R.K., Barry, R.: Sparse spatial autoregressions. Statistics & Probability Letters **33**(3), 291–297 (1997)
17. Pauls, A., Yoder, J.: Determining optimum drop-out rate for neural networks. In: Midwest Instructional Computing Symposium (MICS) (2018)
18. Särkkä, S., Solin, A.: Applied stochastic differential equations, vol. 10. Cambridge University Press (2019)
19. Sigillito, V., W.S.H.L., Baker, K.: Ionosphere. UCI Machine Learning Repository (1989), DOI: https://doi.org/10.24432/C5W01B
20. Wilkins, N., Johnson, M., Nwogu, I.: Regression with uncertainty quantification in large scale complex data. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 827–833. IEEE (2022)