12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Physics-informed neural network uncertainty assessment through Bayesian inference.

Erbet Almeida Costa * **Carine Menezes Rebello** *
Vinícius Viena Santana * **Idelfonso B. R. Nogueira** *

* *Department of Chemical Engineering, Norwegian University of Science and Technology, Gløshaugen, Trondeim, Norway (e-mail: erbet.a.costa@ntnu.no, carine.m.rebello@ntnu.no, vinicius.v.santana@ntnu.no, idelfonso.b.d.r.nogueira@ntnu.no).*

Abstract: This work presents a Bayesian approach to evaluating the uncertainty of physics-informed neural network models. The proposed strategy uses a hybrid methodology for training and assessing the uncertainty of model parameters. In the first part of the training, a gradient-based algorithm is used to train and obtain the weights. In the second stage, a Markov Chain Monte Carlo algorithm is used to evaluate the uncertainty of the network weights. The developed method was used to solve Burger's equation, and the results show that it was possible to characterize the uncertainty region of the PINNs' prediction.

Keywords: Artificial Intelligence, Modeling, Uncertain Dynamic Systems, Partial Differential Equations, Markov Chain Monte Carlo.

1. INTRODUCTION

The development of computing resources allied to the growth of data availability provided significant advances in new research fields, such as Scientific Machine Learning (SciML). It boosted studies aimed at developing Machine Learning (ML) tools in the most varied areas of knowledge (Sengupta, 2013; Sarker, 2021). In this context, data-driven modeling contributes to in-depth analysis of complex processes (George, 2021). However, using machine learning techniques to build surrogate models can become prohibitive due to the volume of data needed and the cost associated with the experimental acquisition (Najafabadi et al., 2015). In this sense, small data sets can lead to the generation of underspecified models, which are not robust and are unreliable for real systems.

On the other hand, physical, chemical, and biological laws and correlations can provide valuable information about the dynamics of systems and, when combined with data, allow the construction of models with an excellent capacity for representing systems (Raissi et al., 2017a,b). An example of such a strategy is the physics-informed neural networks (PINNs). The PINNs are supervised learning algorithms that combine data and phenomenological laws described by nonlinear partial differential equations (PDEs) (Huang and Wang, 2023). In summary, this tool aims to encode fundamental laws of physics and knowledge already consolidated in the scientific environment in learning algorithms to perform more reliable modeling. Their reliability is inherited from physics knowledge. PINNs perform well since they restrict the set of possible solutions and direct the algorithm to an optimized solution (Huang and Wang, 2023). On the other hand, PINNs have restricted applications in scenarios with noisy data, which requires a robust approach capable of quantifying the parametric uncertainty (Yang et al., 2021). The quantification of uncertainty, in this case, still favors the development of efficient online learning strategies or adaptive sampling (Meng et al., 2022).

Several approaches to uncertainty quantification have been explored recently in the literature for PDE models (Li and Marzouk, 2014; Yan and Zhou, 2019a; Zhang and Garikipati, 2021; Bharadwaja et al., 2022) and others for surrogate models from data (Yan and Zhou, 2019b; Zhu et al., 2019). However, only a few recent works present uncertainty calculation associated with the PINNs approach, a topic still little explored in the literature.

Yang et al. (2020) quantify the parametric uncertainty of physics-informed generative adversarial networks. Zhang et al. (2019) addresses the polynomial chaos and dropout expressions to define the model uncertainty. Yang et al. (2021) proposed a Bayesian physics-informed neural network (B-PINN) to solve PDEs with noisy data, and the uncertainty quantification through the Hamiltonian Monte Carlo (HMC) methodology, the tool is effective for scenarios with high noise and avoids overfitting. Meng et al. (2022) developed a new Bayesian framework for neural networks that can extrapolate space-time from historical data and quantify uncertainty arising from noisy and irregular data from physical problems. In this work, the authors show that the tool can learn flexible, functional priors and can be extended to big data problems. Another approach can be found in Bai et al. (2021), who propose using a Lavengin Markov Chain Monte Carlo (LMCMC) algorithm that uses the trained PINNs model and noise

* This paper has been sponsored by the Norwegian Research Council.

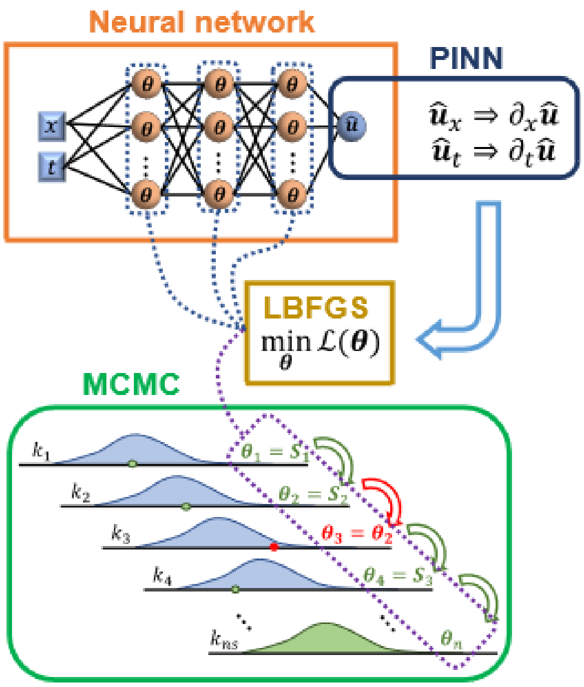

Fig. 1. Diagram of the PINNs-MCMC methodology.

measurement agents to obtain the posterior distribution of unknown parameters.

This article evaluates the uncertainty of the solution of partial differential equations using the Physics Informed Neural Network through Bayesian inference. It begins with a basic definition of PINNs, based on Bayesian formalism, and presents the results of the numerical solution of a PDE system.

2. METHODOLOGY

The methodology used in this work is based on the PINN training and uncertainty assessment. The first step is to train a PINN model using a standard algorithm to find a region of probable solution. The second step evaluates the uncertainty of the parameters found in the first step. Fig. 1 presents the overall description of the methodology proposed here. The following items will describe each step of the proposed method in detail.

2.1 PINNs definitions

PINNs are an artificial intelligence technique that has shown promise in solving partial differential equations of direct and inverse problems (Cuomo et al., 2022). This technique has been applied to solve physical and engineering problems, such as fluid mechanics (Cai et al., 2021a), among others (Sahli Costabal et al., 2020; Cai et al., 2021b; Haghighat et al., 2021). One of the advantages of using PINNs is the great capacity for extrapolation, which differs significantly from traditional data-driven models. This characteristic comes from the network learning being directed towards the problem's solution and not fitting the data. In other words, PINNs are artificial intelligence models that have learned to solve a rigorous model numerically. Hence, their great capacity for extrapolation (Yang et al., 2021).

To define the PINNs approach, consider a partial differential equation (PDE) system is defined mathematically Cai et al. (2021a):

$$f(x,t,\hat{u},\partial_x\hat{u},\partial_t\hat{u},\ldots,\lambda)=0,\quad x\in\Omega, t\in[0,T], \tag{1}$$
$$\hat{u}(x,t_0)=g_0(x),\quad x\in\Omega, \tag{2}$$
$$\hat{u}(x,t)=g_\Gamma(t),\quad x\in\partial\Omega, t\in[0,T], \tag{3}$$

In which f is the residual of the PDE, $\partial_{\mathbf{x}}\hat{u}$ and $\partial_t\hat{u}$ are the differential operators of f, $\mathbf{x}\in\mathbf{R}^d$ is the spatial coordinate, and t is the time. The parameters of the PDE system are λ. The initial and boundary conditions are represented by $g_0(\mathbf{x})$ and $g_\Gamma(t)$, respectively. Ω and $\partial\Omega$ represent the spatial and boundary domains.

In Fig. 1, the loss function is obtained using the Neural Network's derivatives, residual, and for the boundary and initial conditions in the PDE system. Then, the loss function is composed of three other losses and the respective weights.

$$\mathcal{L}=w_f\mathcal{L}_f+w_{bc}\mathcal{L}_{bc}+w_{ic}\mathcal{L}_{ic} \tag{4}$$

The first is the loss of the residual that is obtained by:

$$\mathcal{L}_f=\frac{1}{N_f}\sum_{i=1}^{N_f}\left|f\left(x_i^f,t_i^f\right)\right|^2 \tag{5}$$

N_f is the number of points sorted inside the spatial domain Ω. On the other hand, the second loss is written for the boundary and initial condition by:

$$\mathcal{L}_{bc}=\frac{1}{N_{bc}}\sum_{i=1}^{N_{bc}}|g(\hat{u},x,t)|^2 \tag{6}$$

$$\mathcal{L}_{ic}=\frac{1}{N_{ic}}\sum_{i=1}^{N_{ic}}\left|u\left(x_i^{ic},0\right)-u_i^{ic}\right|^2 \tag{7}$$

Thus, to find the solution to the PDE problem, it is necessary to minimize the loss function using the weights of the neural network layers as a decision variable. This solution can be performed through algorithms based on gradient descent.

2.2 Bayesian approach for PINNs uncertainty assessment

Bayesian inference is a powerful tool that solves different inference and statistical problems. It is based initially on Bayes' Theorem, which describes the probability of occurrence of an event based on existing prior knowledge. In this sense, Bayesian inference can be used to obtain the later PDF (h_θ) of a set of θ parameters, using information from experimental data (D) and existing information (I) (Gelman et al., 2013).

Considering the case of PINNs, it is proposed that the joint PDF of the parameters of the PINNs model be obtained through the Bayesian approach. In this sense, the problem can be written as (Gamerman and Lopes, 2006):

$$h(\theta\mid D,I)\propto L(\theta\mid D)h(\theta\mid I) \tag{4}$$

Where L is the likelihood, which is the relationship between the data and the model parameters, η are random values of θ.

Specifically, data are not used to train the network in building PINN models, so using a least-squares likelihood function is impossible. On the other hand, the loss function (4) presents a good relationship capable of functioning as a link between the network parameters and a model. In this way, in possession of the likelihood, it is possible to obtain the posterior marginal PDF through the solution of the following integral (Gamerman and Lopes, 2006):

$$h_{\theta_i}(\theta \mid D, I) \propto \int_{\eta} L(\theta \mid D) h(\theta \mid I) d\theta_{\eta-i} \tag{5}$$

In which $\theta_{\eta-i}$ is the θ vector without i-th element. With the posterior marginal, it is possible to obtain the most probable value $\hat{\theta}$ and the variance $V_{\theta\theta}$ of θ by:

$$\hat{\theta} = \int_{-\infty}^{\infty} \eta h(\theta \mid D, I) d\theta \tag{6}$$

$$V_{\theta\theta} = \int_{-\infty}^{\infty} (\eta - \hat{\theta})^T (\eta - \hat{\theta}) h(\theta \mid D, I) d\theta \tag{7}$$

The way to find a solution to PINN problems is by using numerical solutions through Monte Carlo samplers. Thus, this paper proposes using the Markov Chain Monte Carlo (MCMC) algorithm proposed by Haario et al. (2001, 2006) to build the Markov chain and get the posterior marginal PDF.

3. RESULTS

3.1 Case study

In this work, the case study is the Burgers' equation. This equation represents the nonlinear wave motion and the linear diffusion. Also, it is the simplest model for analyzing the effect of nonlinear advection and diffusion interaction. Consider the PDE problem composed of Burger's equation with boundary conditions of the Dirichlet type (Raissi et al., 2017a):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \frac{0.01}{\pi}\frac{\partial^2 u}{\partial x^2} = 0, \quad x \in [-1, 1], \tag{8}$$

$$u(0, x) = -\sin(\pi x), \tag{9}$$

$$u(t, -1) = u(t, 1) = 0. \tag{10}$$

This paper uses a network containing nine fully connected layers and 20 neurons with an activation function *tanh*. The first layer corresponds to inputs x and t, and the last layer has only one output related to $u(x, t)$. The network has 3021 trainable parameters.

In building PINNs, selecting points to impose the PDE solution is necessary. Thus, 25 points were set equally spaced for the boundary conditions between $u(x = -1, t) = 0$ and $u(x = 1, t) = 0$. 50 initial conditions were equally spaced at $u(x, t = 0) = -sin(pix)$. 10,000 quasirandom points of internal collocations were used to impose the neural network output to adhere to the output of the PDF.

The solution to the proposed inference problem was carried out in two steps. The first step was to solve the optimization problem using the Limited-memory BFGS method. This first step is necessary to speed up the convergence process and to an optimal solution used as an initial estimate for the second step of the solution through the MCMC algorithm. Fig. 2 presents a two-dimensional comparison between the prediction made by PINNs and the Burger equation's numerical solution. The figure also presents the confidence region of the PINN prediction for a 95% coverage probability. This region was generated by calculating the prediction with PINNs considering 1E4 random samples of parameters from the posterior PDF h_θ. Also, Fig. 2 presents the error calculated between the curves obtained through the following:

$$e = \frac{|u_{PINN} - u_{num}|}{|u_{num}|} \tag{11}$$

It is considered, however, that the evaluation of the output predicted by a model constructed from the Bayesian inference needs to consider the different trajectories projected from the parameters of the PDF. With the different trajectories in hand, the most probable value (median) and the upper and lower limits of the prediction are calculated, as shown in Fig. 1.

Burger's equation is a multivariable function with two dependent variables, t, and x In this sense, the solution can be represented in surface form, as shown in Fig. 3. In this figure, the most probable value of the solution (median) is represented by the surface in red. In turn, the blue surfaces represent the regions of uncertainty. By comparing Fig. 2 and 3, it is possible to observe that when $t > 0.4$, the uncertainty reduces, and the two surfaces converge to the mean values. In this way, it is possible to infer that the PINNs can replicate the expected behavior of the solved PDE model.

In general, a model's parameters are uncertain, and they must be evaluated. PINN models are no different. Although few studies address this issue directly, the presented results show that it is possible to assess the parametric uncertainty of these models through Bayesian inference.

The analysis of training convergence can be performed based on the behavior of the loss function. Fig. 4 shows the first 500 loss assessments during the first stage of the training. It is possible to observe that the LBFGS algorithm produces a solution that monotonically decreases towards a minimum point. However, it is necessary to point out that in this step, the algorithm tested the objective function 30372 times before reaching the tolerance of 1E-5. The final loss value in this step was 2.058E-6.

The MCMC was configured to build a later PDF with 1E5 samples in the uncertainty assessment step. This means the accounting does not consider rejected samples during the chain evolution. Fig. 5 shows a histogram of the loss values obtained during this process step. The graph shows that the loss distribution has a bimodal behavior, which indicates that the loss function possibly has two equilibrium points for the parameter values drawn by the MCMC. On the other hand, this feature does not

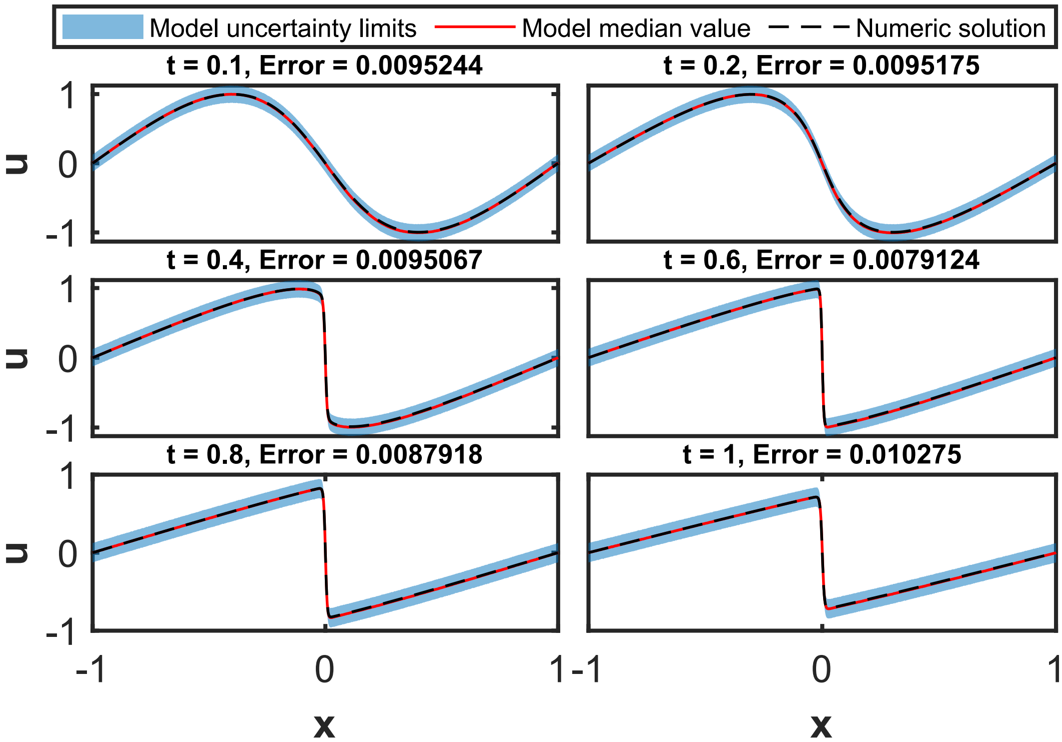

Fig. 2. Prediction of PINNs and coverage regions compared with the numerical solution.

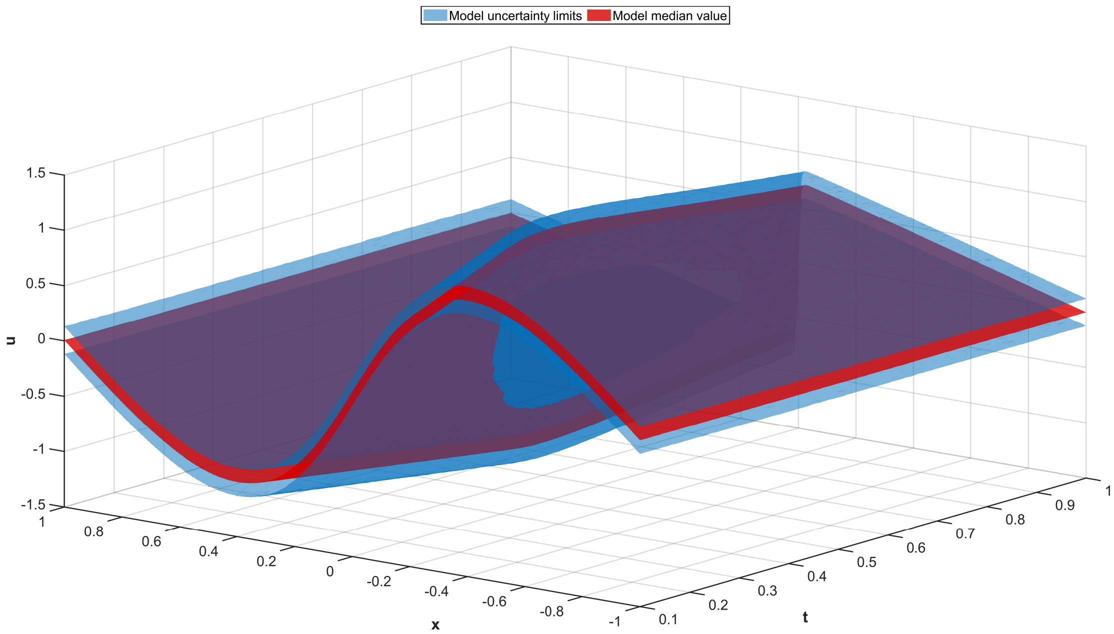

Fig. 3. Prediction surface of PINNs and coverage surface regions.

significantly interfere with the model's predictive ability, as seen in Fig. 2 and 3.

4. CONCLUSION

This paper presented a methodology for evaluating the parametric uncertainty of PINNs models. The methodology is based on a hybrid method to obtain the weights of PINNs. Then, the LGBFS method was used to obtain an initial solution to the problem, followed by the MCMC method used to analyze the uncertainty of the weights of PINNs. The results indicated that the methodology used was able to evaluate the parametric uncertainty and obtain the PDF of the weights, and, through propagation,

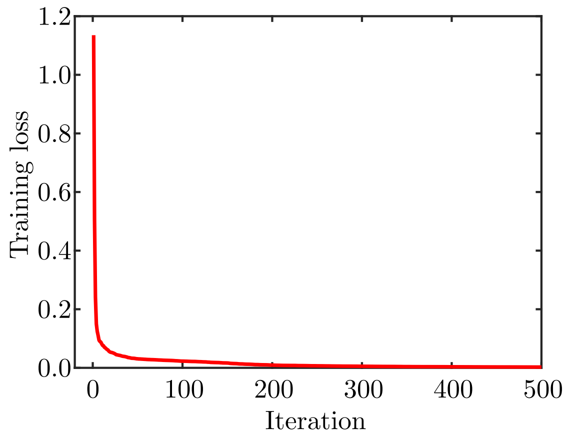

Fig. 4. First 500 training loss value during the solution with the LBFGS algorithm.

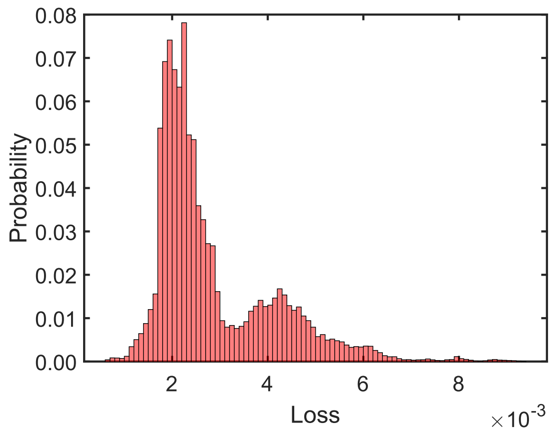

Fig. 5. Training loss value histogram obtained by the MCMC algorithm..

obtain a confidence region for the prediction of the PINNs model. Therefore, the main contribution of this work is to show that, through Bayesian inference, it is possible to understand how the parametric uncertainty of PINNs models affects the model's predictive capacity. This paper's results allow directing efforts toward using PINNs in general engineering applications where uncertainty is an important factor, such as model building and validation, robust control, and optimization.

ACKNOWLEDGEMENTS

The present work contributes to completing a sub-project at SUBPRO, a research-based innovation center within Subsea Production and Processing at the Norwegian University of Science and Technology. The authors would like to express their gratitude for the financial support from SUBPRO, funded by the Research Council of Norway through grant number 237893, major industry partners, and NTNU.

REFERENCES

Bai, H., Bhar, K., George, J., and Busart, C. (2021). Distributed bayesian parameter inference for physics-informed neural networks. In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. doi: 10.1109/cdc45484.2021. 9683353.

Bharadwaja, B.V.S.S., Nabian, M.A., Sharma, B., Choudhry, S., and Alankar, A. (2022). Physics-informed machine learning and uncertainty quantification for mechanics of heterogeneous materials. *Integrating Materials and Manufacturing Innovation*, 11(4), 607–627. doi:10.1007/s40192-022-00283-2. URL `http://dx.doi.org/10.1007/s40192-022-00283-2`.

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G.E. (2021a). Physics-informed neural networks (pinns) for fluid mechanics: a review. *Acta Mechanica Sinica*, 37(12), 1727–1738. doi:10.1007/s10409-021-01148-1. URL `http://dx.doi.org/10.1007/s10409-021-01148-1`.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., and Karniadakis, G.E. (2021b). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6). doi:10.1115/1.4050542. URL `http://dx.doi.org/10.1115/1.4050542`.

Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3). doi:10.1007/s10915-022-01939-z. URL `http://dx.doi.org/10.1007/s10915-022-01939-z`.

Gamerman, D. and Lopes, H.F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and Hall/CRC, 2 edition.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (2013). *Bayesian Data Analysis Third edition (with errors fixed as of 13 February 2020)*. CRC Press.

George, J. (2021). *Distributed Bayesian Parameter Inference for Physics-Informed Neural Networks*. IEEE.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16(4), 339–354. doi:10.1007/s11222-006-9438-0. URL `http://dx.doi.org/10.1007/s11222-006-9438-0`.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2), 223. doi:10.2307/3318737. URL `http://dx.doi.org/10.2307/3318737`.

Haghighat, E., Raissi, M., Moure, A., Gomez, H., and Juanes, R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379, 113741. doi:10.1016/j.cma.2021.113741. URL `http://dx.doi.org/10.1016/j.cma.2021.113741`.

Huang, B. and Wang, J. (2023). Applications of physics-informed neural networks in power systems - a review. *IEEE Transactions on Power Systems*, 38(1), 572–588. doi:10.1109/tpwrs.2022.3162473. URL `http://dx.doi.org/10.1109/TPWRS.2022.3162473`.

Li, J. and Marzouk, Y.M. (2014). Adaptive construction of surrogates for the bayesian solution of inverse problems. *SIAM Journal on Scientific Computing*, 36(3), A1163–A1186. doi:10.1137/130938189. URL `http://dx.doi.org/10.1137/130938189`.

Meng, X., Yang, L., Mao, Z., del Águila Ferrandis, J., and Karniadakis, G.E. (2022). Learning functional priors and posteriors from data and physics. *Journal of Computational Physics*, 457, 111073. doi:10.1016/j.jcp.2022.111073. URL http://dx.doi.org/10.1016/j.jcp.2022.111073.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1). doi:10.1186/s40537-014-0007-7. URL http://dx.doi.org/10.1186/s40537-014-0007-7.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2017a). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. doi:10.48550/ARXIV.1711.10561. URL https://arxiv.org/abs/1711.10561.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2017b). Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. doi:10.48550/ARXIV.1711.10566. URL https://arxiv.org/abs/1711.10566.

Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D.E., and Kuhl, E. (2020). Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8. doi:10.3389/fphy.2020.00042. URL http://dx.doi.org/10.3389/fphy.2020.00042.

Sarker, I.H. (2021). Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective. *SN Computer Science*, 2(5). doi:10.1007/s42979-021-00765-8. URL http://dx.doi.org/10.1007/s42979-021-00765-8.

Sengupta, P.P. (2013). Intelligent platforms for disease assessment. *JACC: Cardiovascular Imaging*, 6(11), 1206–1211. doi:10.1016/j.jcmg.2013.09.003. URL http://dx.doi.org/10.1016/j.jcmg.2013.09.003.

Yan, L. and Zhou, T. (2019a). Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems. *Journal of Computational Physics*, 381, 110–128. doi:10.1016/j.jcp.2018.12.025. URL http://dx.doi.org/10.1016/j.jcp.2018.12.025.

Yan, L. and Zhou, T. (2019b). An adaptive surrogate modeling based on deep neural networks for large-scale bayesian inverse problems. doi:10.48550/ARXIV.1911.08926. URL https://arxiv.org/abs/1911.08926.

Yang, L., Meng, X., and Karniadakis, G.E. (2021). B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425, 109913. doi:10.1016/j.jcp.2020.109913. URL http://dx.doi.org/10.1016/j.jcp.2020.109913.

Yang, L., Zhang, D., and Karniadakis, G.E. (2020). Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1), A292–A317. doi:10.1137/18m1225409. URL http://dx.doi.org/10.1137/18M1225409.

Zhang, D., Lu, L., Guo, L., and Karniadakis, G.E. (2019). Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397, 108850. doi:10.1016/j.jcp.2019.07.048. URL http://dx.doi.org/10.1016/j.jcp.2019.07.048.

Zhang, X. and Garikipati, K. (2021). Bayesian neural networks for weak solution of pdes with uncertainty quantification. doi:10.48550/ARXIV.2101.04879. URL https://arxiv.org/abs/2101.04879.

Zhu, Y., Zabaras, N., Koutsourelakis, P.S., and Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394, 56–81. doi:10.1016/j.jcp.2019.05.024. URL http://dx.doi.org/10.1016/j.jcp.2019.05.024.