Appendix: Robust Prediction Intervals from Three Neural Networks Trained by the MSE Loss

Anonymous Author(s)

1 High sensitivity of MPIW and RMSE with respect to the hyperparameters in QD's loss function

We include additional analysis (additional to Figure 2 in the main text) on the QD method (Pearce et al., 2018) to further illustrate the high sensitivity of QD with respect to the two hyperparameters in its loss function. Our PI3NN method uses the standard MSE loss, so this sensitivity analysis applies to the QD approach only and helps illustrate the robustness and general applicability of our method.

We investigate the sensitivity of three prediction interval criteria—PICP, MPIW, and RMSE—to the two hyperparameters—soften parameter s and weight parameter λ—introduced in QD for the 10 UCI benchmark datasets. Based on the recommended values in the QD paper (Pearce et al., 2018) where s=160.0 and λ=15.0 for most of the datasets, we designed 55 experiments where s has values from [100.0, 110.0, 120.0, 130.0, 140.0, 150.0, 160.0, 170.0, 180.0, 190.0, 200.0] and λ has values from [5.0, 10.0, 15.0, 20.0, 25.0]. For the two datasets—*boston* and *concrete*—where the QD paper provided the hyperparameter settings, we used the exactly same values; and for the remaining 8 datasets where the QD paper did not provide hyperparameter values, we used the values from the *boston* case. The data splitting seed and random seed are fixed at values of 1 and 10, respectively, for the experiments. The sensitivity results of PICP is presented in Figure 2 of the main text, and the results of MPIW and RMSE are presented below in Appendix Figure 1 and Figure 2, respectively. Figure 2 in the main text indicated that there is about 28% difference between the biggest and the smallest PICP of the target 0.95 value for the *boston* dataset, demonstrating a high sensitivity of QD performance to its two hyperparameters. We also observe the similar high sensitivity with respect to MPIW criterion in Appendix Figure 1, where for the *boston* dataset, the maximum MPIW value of the 55 experiments is about twice of the minimum value and for the *yacht* dataset, the maximum value is almost five times larger than its minimum MPIW. The prediction accuracy of QD is also sensitive to the two hyperparameters as measured by RMSE and demonstrated in Appendix Figure 2. For example, in the *yacht* case, the maximum value of RMSE is 4.47 and the minimum value is 0.97. Above analysis demonstrated that the performance of QD is sensitive to the choice of its two hyperparameters, which makes QD less robust in practice where we usually do not know the suitable hyperparamters for a new dataset. In contrast, our PI3NN uses standard MSE loss and introduces no extra hyperparameters making it robust and reliable for application.

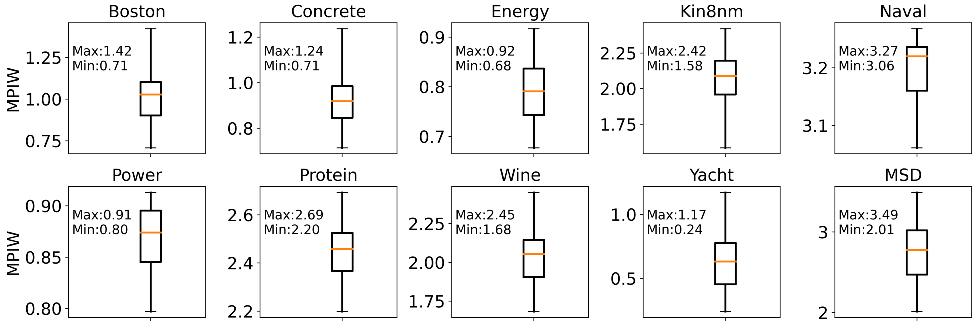

Figure 1: Sensitivity of MPIW with respect to the two hyperparameters in QD's loss function

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

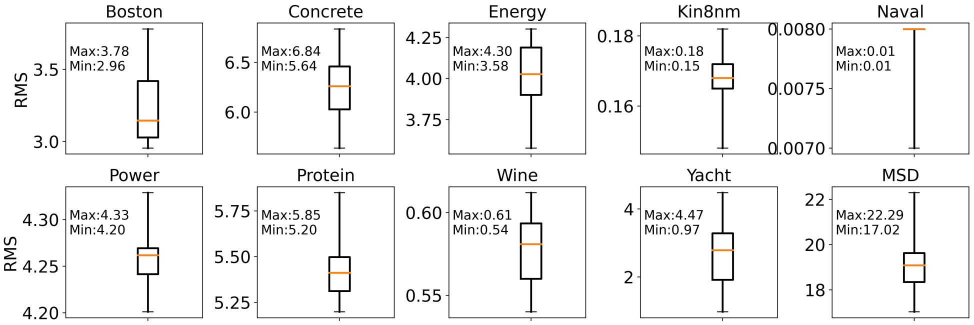

Figure 2: Sensitivity of RMSE with respect to the two hyperparameters in QD's loss function

2 Code and data availability

We provide code and data used in this study for reproducing the results we generated. We also include our pre-generated results and the plotting function for fast evaluation for each figure and table. The instructions on running the code are made available in the code folders and the comment section within the code. We encapsulated the code and pre-generated results in the supplemental material zip file. However, the relatively larger size UCI data sets and pre-split UCI data sets used in this study can be downloaded through the URLs:

https://figshare.com/s/53cdf79be5ba5d216ba8 for UCI data sets

https://figshare.com/s/e470b7131b55df1b074e for pre-split UCI data sets

Or follow the instructions provided in the supplemental material, and download the data by running the script (download_data.sh).

Meanwhile, the data splitting can be done by running our splitting code (UCI_data_splitting.py) once the original UCI data sets are ready (in UCI_datasets folder)

All of our experiments are done on a single UBUNTU 20.04 workstation with an Intel I9-10980xe CPU except the PI3NN for *MSD* case was executed on a single NVIDIA RTX 3090 GPU. All models are implemented with package environment with Python 3.8.3, and Google TensorFlow 2.4.1, except the PI3NN for OOD detection experiment was conducted with PyTorch 1.7.1. Note that the original QD code was built on TensorFlow 1. Thus, we added compatibility adjustment and made it run smoothly under the TensorFlow 2 environment. The detailed packages we used and the dependencies are provided in the *environment.yml* file within the supplemental material.

3 Data sets used in this study

The 10 data sets used in this study are open access data sets accessible through UCI machine learning data sets repository (Dua and Graff, 2017) , including Boston housing (boston) (Dua and Graff, 2017), Concrete compressive strength (concrete) (Yeh, 1998), Energy efficiency (energy) (Tsanas and Xifara, 2012), KINematics 8 inputs non-linear medium unpredictability/noise (kin8nm) (Corke, 1996), Combined Cycle Power Plant (power) (Tüfekci, 2014), Physicochemical Properties of Protein Tertiary Structure (Protein) (Dua and Graff, 2017), Wine quality (wine) (Cortez et al., 2009), Yacht Hydrodynamics (yacht) (Ortigosa et al., 2007), Year Prediction Million Song Dataset (MSD) (Bertin-Mahieux et al., 2011).

The data we are using does not contain personally identifiable information or offensive content.

References

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Corke, P. (1996). A robotics toolbox for matlab. *IEEE Robotics Automation Magazine*, 3(1):24–32.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553. Smart Business Networks: Concepts and Empirical Evidence.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Ortigosa, I., Lopez, R., and Garcia, J. (2007). A neural networks approach to residuary resistance of sailing yachts prediction. In *Proceedings of the international conference on marine engineering MARINE*, volume 2007, page 250.

Pearce, T., Brintrup, A., Zaki, M., and Neely, A. (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4075–4084. PMLR.

Tsanas, A. and Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings*, 49:560–567.

Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power Energy Systems*, 60:126–140.

Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808.