Uncertainty in Non-Intrusive Load Monitoring

Akshat Mangal [†*], Gaurav Sonkusle [†*], Mohamed Shamir [†*], Mohmmad Aslam [†*], Keyur Unadkat [†*], Hetvi Shastri [†], Nipun Batra [†]
Indian Institute of Technology Gandhinagar, India

1 ABSTRACT

Non-intrusive load monitoring(NILM) is the task of separating the total household energy measured into its constituent appliances. An open-source NILM toolkit called NILMTK was developed in 2014 to make NILM research reproducible and to ensure easy comparison of benchmark NILM algorithms. With the addition of more algorithms like Seq2Seq, Seq2Point models and improvements in both the experiment and model interface, an improved version of the toolkit NILM-contrib was released in 2019. However, the existing neural network models in the toolkit do not capture the model uncertainty. This project aims to add the uncertainty prediction component using MC dropout method to the existing state-of-the-art Seq2Point model [3] and then to the Seq2Seq model.

2 INTRODUCTION

Non-intrusive load monitoring, or energy disaggregation, aims to solve the problem of separating the total household energy measured into its constituent appliances. Solving this problem would help the household occupants to make informed decisions regarding their energy consumption by regulating the respective appliances. NILM Toolkit (NILMTK) released in 2014, is an open source toolkit to make NILM research reproducible and to compare various energy disaggregation algorithms. Various disaggregation algorithms like mean, edge detection, combinatorial optimisation, Seq2Point, Seq2Seq, etc have been developed and released as part of the improved version of the toolkit NILM-contrib. Seq2Point and Seq2Seq algorithms have performed well with Seq2Point being verified as the state-of-the-art NILM model [8].

Model uncertainty is indispensable for deep learning practitioners. With model confidence at hand, we can treat uncertain inputs and special cases explicitly. There can be certain critical infrastructure that employs a model, which might return a result with high uncertainty. If not accounted for, it can lead to bad consequences.

However, the existing algorithms and neural network models in the NILM toolkit do not capture the model uncertainty. Incorporating uncertainty into the existing NILM models would increase the usability of NILM models. This project aims to implement probabilistic deep learning by using MC dropout on Seq2Seq and Seq2Point NILM models.

3 LITERATURE REVIEW

The updated NILM-contrib repository contains various algorithms like mean, edge detection, combinatorial optimisation, Seq2Point, Seq2Seq models etc [3]. Seq2Point and Seq2Seq algorithms have minimum MAE or have achieved good performance. Advanced neural networks models BiLSTM, ResNet, BERT, etc. were also implemented as part of NILM toolkit and the newly implemented algorithms performed better than the existing baseline algorithms for a subset of household appliances. Even though many models have been implemented to solve the problem, it fails to capture the uncertainties in the model.

Bayesian Neural Networks (BNNs) are one of the most popular approaches for uncertainty quantification which learns a distribution over weights. However these require significant modifications to the training procedure and are computationally expensive compared to non-Bayesian NNs[7].

Lakshminarayanan et al.[7] proposed an alternative to Bayesian Neural networks using deep ensembles and produced uncertainty estimates which are as good or better than Bayesian NNs. This method requires very few modifications to standard NNs, and is well suited for distributed computation. And thereby making it useful for large-scale deep learning applications like NILM.

Gal and Ghahramani[5] built a probabilistic interpretation of dropout which helped to get model uncertainty from existing deep learning models. Monte Carlo integration is used to estimate how well the model fits the mean and uncertainty. A similar approach is employed by Uber for some reliable time series prediction tasks[9].

4 METHODOLOGY

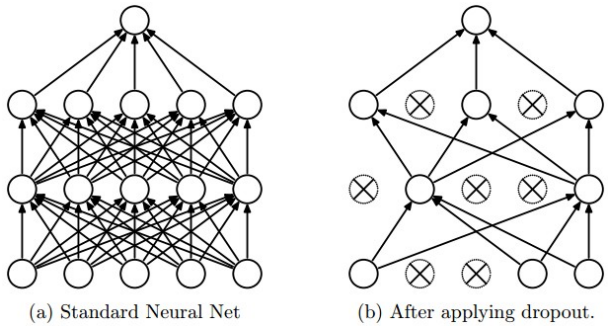
In this section, we will discuss two methods, MC Dropout and Deep Ensemble, that can be used to incorporate uncertainty components into deep learning models.

4.1 MC Dropout

Dropout layers in deep learning models simply mean to switch off some neurons at each training step. Dropout layers help to generalize the model better and hence reducing variance or overfitting of the model. For example, in a deep learning network, for a single neuron let's call it the current neuron, there would be a couple of other neurons that provide it with inputs. By introducing dropouts, the current neuron cannot be dependent solely on one or two neurons as they can disappear at any time while training. The dependency would be spread out evenly and hence helps the model in generalizing better. In normal dropout, the neurons are switched off only while training while in Monte Carlo Dropout (MC Dropout), it is dropped both during the training as well as testing time. By doing this way, the prediction is no longer deterministic instead it depends on which nodes or links we randomly chose to keep. Therefore, given the same data point, our model could predict different values each time. So the primary goal of Monte Carlo dropout is to generate random predictions and interpret them as samples from a probabilistic distribution.

To calculate uncertainty using MC dropout, the below algorithm is used.

[†]{akshat.mangal,gaurav.sonkusle,mohamed.shamir,mohmmad.aslam,unadkatkeyur, hetvi.shastri,nipun.batra}@iitgn.ac.in
[*] Undergraduate authors with equal contribution.



(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Layers[1]

Algorithm 1 Calculating Mean and Standard deviation using MC dropout method

```
procedure UNCERTAINTYCALCULATION(test_Data)
    results = []                ▷ initializing a list to store predictions
    for i in range(n_iterations) do
        y_pred = model.predict(test_data)
        results.append(y_pred)
    mean ← column wise average of results list
    std_deviation ← column wise std deviation of the results
    return mean, std_deviation
```

4.2 MC Dropout on a toy dataset

In this section, testing of the MC dropout algorithm is done on a toy dataset. We have used the function , $y = x^2 + 5x + noise$.

Figure 10 shows the prediction with variance. The hue in the plots represents the uncertainty for the prediction in the dataset.

4.3 Deep Ensemble

Deep Ensemble is a non Bayesian method for estimating uncertainties. The speciality of deep ensemble models is that for each point the mean and the variance is predicted from with in the model. i.e, at the output layer, we will get a mean and variance for each row of data points (Figure 2). Since we are estimating distributional variance along with the mean, we will have to account for it in the loss function. This can be achieved by using negative log-likelihood function of the normal distribution as the loss function as shown below.

$$L(x, y) = -\log \phi(x, y) = \log \frac{\hat{\sigma}^2(x)}{2} + \frac{(y - \hat{\mu}(x)^2)}{2\hat{\sigma}^2(x)}$$

where $\hat{\sigma}^2(x)$ is the predicted variance, $\hat{\mu}(x)^2$ is predicted mean and y is the ground truth label.

Instead of training a single neural network, an ensemble of M networks with random initialization is used. All the ensemble models would behave similarly in the areas of sufficient training data is available. The results will be different when there is less to no data is available. We combine the results from M ensembles as follows,

$$\hat{\mu}_c(x) = \frac{1}{M} \sum_{i=1}^{M} \hat{\mu}_i(x)$$

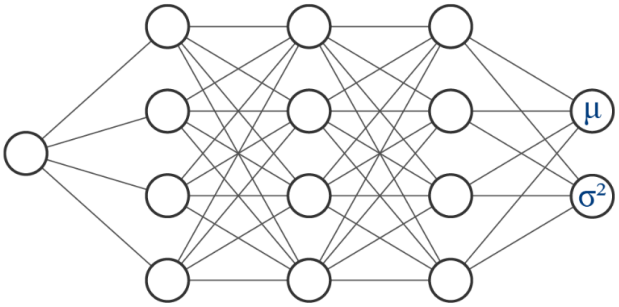


Figure 2: A sample neural network architecture for single model in Deep Ensemble method [2]

$$\hat{\sigma}_c^2(x) = \frac{1}{M} \sum_{i=1}^{M} \hat{\sigma}_i^2(x) + \left[\frac{1}{M} \sum_{i=1}^{M} \hat{\mu}_i^2(x) - \hat{\mu}_c^2(x) \right]$$

4.4 Deep Ensemble on a toy dataset

We experimented deep ensemble method on a toy dataset.We have used the function, $y = x^3 + noise$ for generating the dataset.We haved used a simple neural architecture with μ and σ^2 in the output prediction layer (Figure 12).

4.5 Evaluation Metrics

We used Mean Absolute Error (MAE) and calibration plots as evaluation metrics. Calibration plots are used to quantify the predictive uncertainty of a model. When making a prediction, we can make an α prediction interval that aims to capture observed values $\alpha\%$ of time. For different values of α, we can calculate the proportion of the test data that actually fall within the prediction interval[4]. The calibration plot is plotted by taking the predicted proportion of the test data we expect to lie inside the interval on x axis and the actual observed proportion of the test data in the interval on y axis. Let's take, $f(x)$ as Observed Proportion in the Interval. For a good calibrated model, the plot will have the line $f(x) = x$. If the curve $f(x)$ is below the line $f(x) = x$, then we can say the model is under confident (too much uncertainty) and if it is above the line $f(x) = x$, then the model is over confident (too little uncertainty) in its predictions (Figure 8). The area between the line $f(x) = x$ and the curve $f(x)$ can be considered as metric to evaluate how miscalibrated our model is. Let's call this miscalibrated area. The lesser the area, the better the model is calibrated.

5 EXPERIMENTS

We considered seq2seq and seq2Point as our baseline implementation and then we implemented MC dropout method on the top of both Seq2Seq and Seq2Point. We further tuned the hyper parameters for the models using reliability plots and Mean Absolute Error as an evaluation metric.

5.1 Dataset Details

We have used REDD dataset [6] for the experiments. It contains aggregate and sub-metered power data of 6 houses for 3 - 19 days of duration with 3 seconds appliance frequency.

5.2 Baseline Implementation

We considered seq2seq and seq2point as baseline and reproduced the results from the paper[8] using the APIs available in the nilm-contrib library. Buildings 1,2,3 and 5 were used for appliance Fridge, buildings 1, 2 and 3 were for appliance Dish washer, buildings 1,2 and 3 were used for appliance microwave and buildings 1 and 2 were used for the appliance Washer Dryer. MAE was used as an evaluation metric and cross validation was done during testing. We had used a sample rate of 60, epochs of 50 and a batch size of 32 as parameters same as given in the baseline paper.

5.3 Hyper parameter tuning

We have tuned the hyper parameters dropout rate and the number of iterations which is used during the prediction phase to estimate uncertainty in the model. Seq2Seq Algorithm and the appliance Dish Washer is used for the tuning purpose. Hyper parameter tuning experiments were done on the buildings 1,2 and 3 in similar to the experiments in the paper[8].

5.4 Computational Resources

All the experiments are performed on Virtual Machine (VM) instance of google cloud platform. We used multiple instances of machine type E2-Standard-2 with 8 GB ram and without GPU. A total of 34 experiments were performed which took approximately 100 hours to complete.

6 RESULTS

The following results are obtained from the experiments. At first, we show the results which we reproduced from the baseline paper [8]. In the second section, we did experiments for tuning the hyper parameters drop rate and number of iterations. In the third section, we predict the results using Seq2Seq and Seq2Point after introducing MC dropout. And in the fourth section, we compare the quality of uncertainty of Seq2Seq and Seq2Point implementation.

6.1 Baseline Implementation

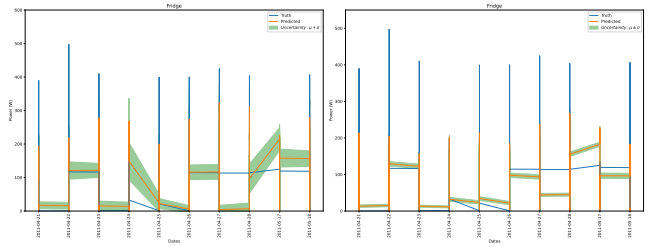
In the baseline implementation results, similar results are obtained for dish washer and microwave in Seq2Point as well as Fridge and Dish washer in Seq2Seq. The difference in the results can be attributed to the difference in initialization of weights, or difference in the machine used to run the experiments or it can also be due to the difference in the time interval of the data which was used for experiments.

6.2 Hyper parameter Tuning

Experiments are done for the dropout rates 0.2, 0.4 and 0.7. And for each dropout rate the parameter number of iterations is varied from 50 to 200 as shown in table 2. We observe that the parameter number of iterations do not have much effect on the predictions as the RMSE and MAE does not change much. However, it has considerable effect on the variance. The RMSE and MAE values increased while increasing the dropout rates. Therefore, we chose the optimum dropout rate as 0.2 and the number of iterations as 100 for all the uncertainty experiments.

Appliances	Baseline Result Seq2Point	Our Result Seq2Point	Baseline Result Seq2Seq	Our Result Seq2Seq
Fridge	33.84	22.78	31.33	30.43
Dishwasher	14.43	13.78	13.98	13.09
Microwave	16.11	12.05	14.05	20.1
Washer Dryer	38.06	13.66	44.13	15.6

Table 1: Base line results. All the values reported are Mean Absolute Errors (lower is better)



(a) Seq2Point **(b) Seq2Seq**

Figure 3: Prediction with uncertainty for the appliance Fridge for Seq2Point and Seq2Seq

Hyper Parameter Tuning			
No. of iters	**Dropout Rate**	**RMSE**	**MAE**
50	0.2	57.6 ± 0.098	7.68 ± 0.195
100	0.2	57.61 ± 0.106	7.68 ± 0.203
200	0.2	57.58 ± 0.115	7.68 ± 0.203
50	0.4	58.13 ± 0.221	7.9 ± 0.454
100	0.4	60.58 ± 0.108	8.3 ± 0.234
200	0.4	60.58 ± 0.113	8.29 ± 0.236
50	0.7	61.76 ± 0.348	8.87 ± 0.918
100	0.7	61.83 ± 0.354	8.88 ± 0.926
200	0.7	61.77 ± 0.358	8.87 ± 0.939

Table 2: Results of hyper parameter tuning for different dropout rates and number of iterations during prediction phase.

6.3 MC Dropout Predictions

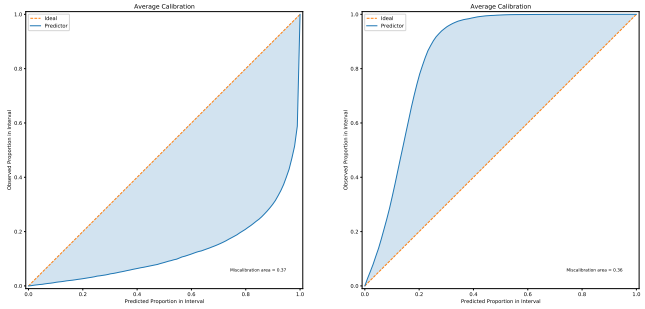
Experiments are done for appliances Fridge, Dish washer, Microwave and Washer Dryer for Seq2Point and Seq2Seq with MC Dropout. The RMSE and MAE are shown in the table 3 along with the mean and the variance. Seq2Point model predicted uncertainty well in comparison with Seq2Seq model as we can observe from figure 3.

Appliances		Algorithms	
		Seq2Point	Seq2Seq
Fridge	RMSE	47.13 ± 0.372	51.47 ± 0.191
	MAE	30.58 ± 9.803	36.03 ± 2.511
Dishwasher	RMSE	33.99 ± 0.337	71.52 ± 0.350
	MAE	5.48 ± 2.833	9.37 ± 0.417
Microwave	RMSE	96.28 ± 0.910	85.71 ± 0.315
	MAE	24.44 ± 13.306	21.79 ± 1.491
Washer Dryer	RMSE	0.91 ± 0.013	301.41 ± 0.332
	MAE	0.76 ± 4.035	40.56 ± 0.014

Table 3: MC Dropout prediction results for the appliances Fridge, Dishwasher, Microwave and Washer Dryer

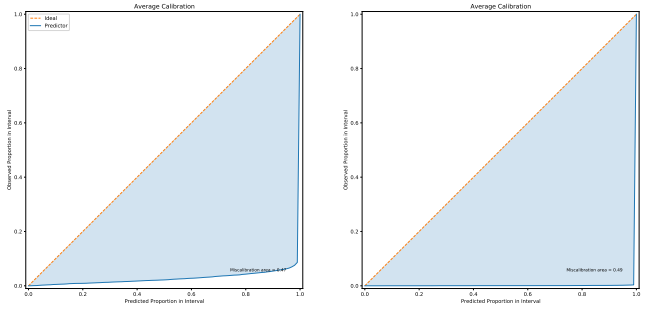
6.4 Comparison between Seq2Seq and Seq2Point

We compared the quality of uncertainty predictions of Seq2Seq and Seq2Point using calibration plots. The quality of the uncertainty predictions can be estimated by the area (miscalibrated area) between the curve f(x) and the line f(x)=x in the calibration plots, figure 4 shows calibration curve for Fridge and Washer Dryer of Seq2Seq and Seq2Point respectively. We observed that the Seq2Point is under confident in its predictions for Washer Dryer and for the rest, it is over confident. While in the case of Seq2Seq, for all the appliances, the model is over confident about the predictions. The miscalibrated area of Seq2Point is less than Seq2Seq for all the appliance and hence we can infer that Seq2Point is predicting the results with better uncertainty than Seq2Seq.



(a) Seq2Point : Fridge

(b) Seq2Point : Washer Dryer



(c) Seq2Seq : Fridge

(d) Seq2Seq : Washer Dryer

Figure 4: Calibration plots of Seq2Seq and Seq2Point

7 CHALLENGES AND FUTURE WORKS

We tried our best to include Deep Ensemble to Seq2Seq and Seq2Point. To incorporate Deep Ensemble uncertainty method to the existing model, we have to add a custom loss function (Negative log likelihood (NLL). The custom loss function has both mean and variance. Including mean was easy but the real trouble came when including variance in the loss function. The custom function takes only two inputs y_pred and y_true and therefore we cannot directly add variance to the loss function. When we tried indirect ways, we faced multiple errors which we were not able to debug till the end. In the future, Deep Ensemble can be incorporated into the Seq2Seq and Seq2Point as it performs better than MC Dropout Method [7]. Uncertainty implementation can further be explored to other deep learning algorithms in the NILM-contrib repository.

8 SUMMARY

In this project, we implemented MC Dropout method on the top of Seq2Seq and Seq2Point algorithm. We tuned the hyper parameters dropout rate and the number of iterations used during the prediction phase. We observed a drop out rate of 0.2 and number of iteration of 100 is optimum for the models. Then we quantified the uncertainties of Seq2Seq and Seq2Point models (our implementation) using calibration plots. Seq2Point model performed better than Seq2Seq model in uncertainty predictions since miscalibrated area in calibration plot is less for all appliances for Seq2Point in comparison to Seq2Seq.

9 CODE REPOSITORY

All the codes and the results of the implementation is available at **github.com/Akshat10101998/NILM_ML_project**

REFERENCES

[1] [n.d.]. Normal Dropout vs MC Dropout. https://d33wubrfki0l68.cloudfront.net/7c69679510220b220fb38d5affdb85ed60bd2575/2fa65/images/model-uncertainty-in-deep-learning-with-monte-carlo-dropout_files/dropout.jpeg

[2] [n.d.]. A Sample Deep Ensemble Architecture. https://www.inovex.de/wp-content/uploads/2019/09/Distribution-Estimation.png

[3] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo C. M. Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019).

[4] Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. 2021. Uncertainty Toolbox: an Open-Source Library for Assessing, Visualizing, and Improving Uncertainty Quantification. *arXiv preprint arXiv:2109.10254* (2021).

[5] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *ArXiv* abs/1506.02142 (2016).

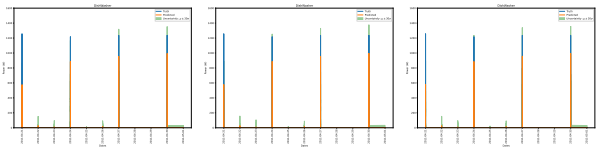
[6] J. Zico Kolter. 2011. REDD : A Public Data Set for Energy Disaggregation Research.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NIPS*.

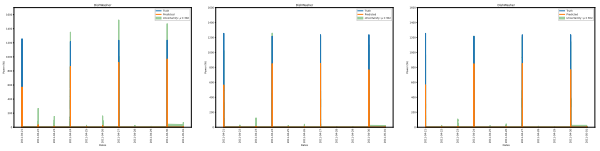
[8] Hetvi Shastri and Nipun Batra. 2021. Neural network approaches and dataset parser for NILM toolkit. *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2021).

[9] Lingxue Zhu and Nikolay Pavlovich Laptev. 2017. Deep and Confident Prediction for Time Series at Uber. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (2017), 103–110.

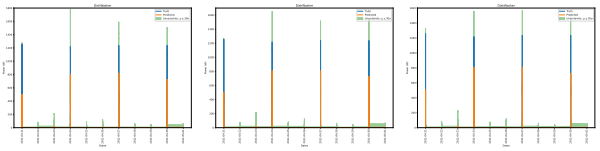
Appendix



(a) Dropout 0.2, Iter 50 (b) Dropout 0.2, Iter 100 (c) Dropout 0.2, Iter 200

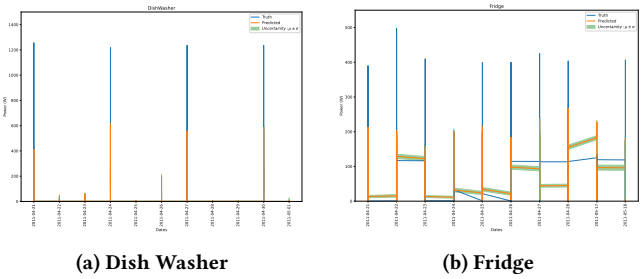


(d) Dropout 0.4, Iter 50 (e) Dropout 0.4, Iter 100 (f) Dropout 0.4, Iter 200

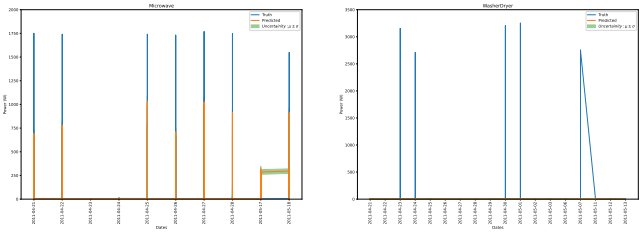


(g) Dropout 0.7, Iter 50 (h) Dropout 0.7, Iter 100 (i) Dropout 0.7, Iter 200

Figure 5: Hyperparameter Tuning done for Seq2Seq Dishwasher

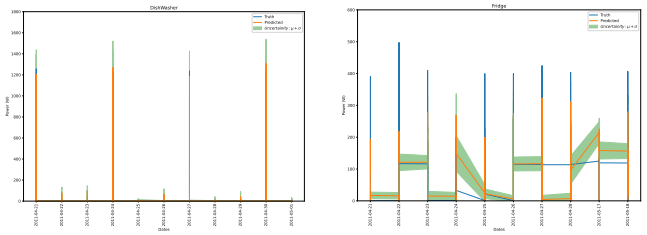


(a) Dish Washer (b) Fridge

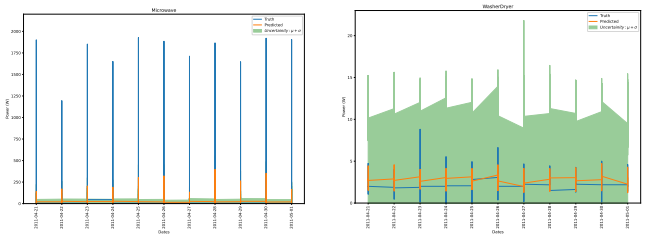


(c) Microwave (d) Washer Dryer

Figure 6: Seq2Seq experiments for different appliances keeping MC dropout 0.2 and Number of iteration 100

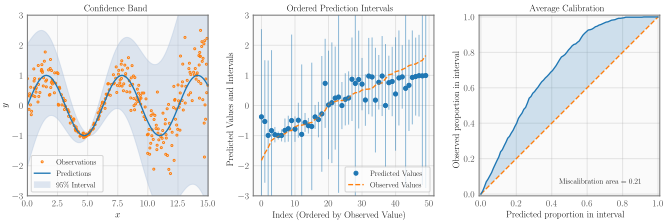


(a) Dish Washer (b) Fridge

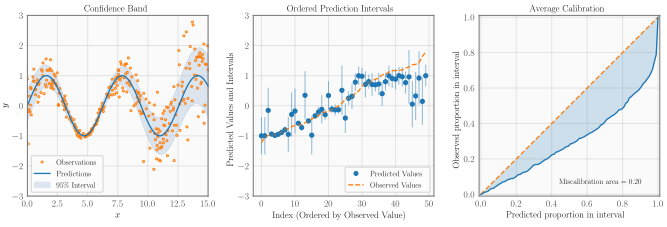


(c) Microwave (d) Washer Dryer

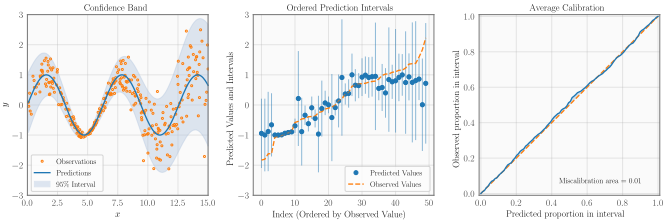
Figure 7: Seq2Point experiments for different appliances keeping MC dropout 0.2 and Number of iteration 100



(a) Under Confident or too much uncertainty

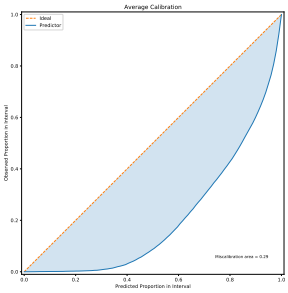


(b) Over confident or too little uncertainty

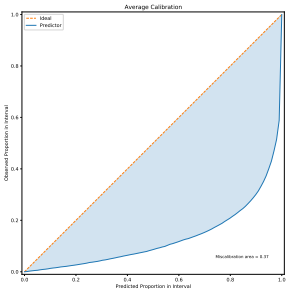


(c) Well Calibrated

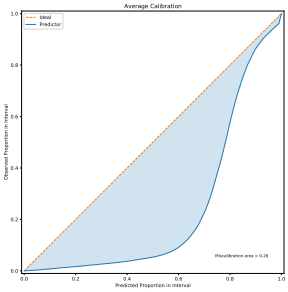
Figure 8: Examples of Overconfident, Underconfident and Perfectly calibrated plots [4]



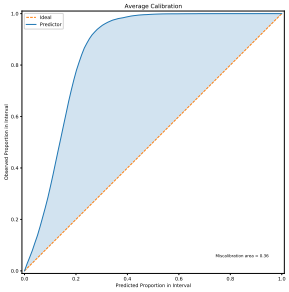
(a) Dish Washer



(b) Fridge

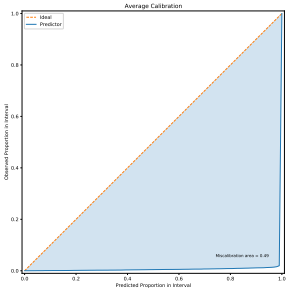


(c) Microwave

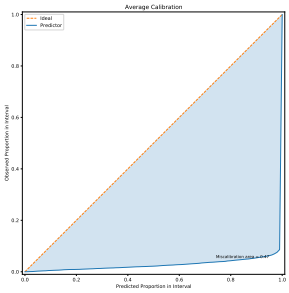


(d) Washer Dryer

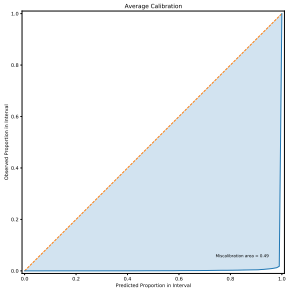
Figure 9: Reliability Curve for Seq2Point experiments for different appliances keeping MC dropout 0.2 and Number of iteration 100



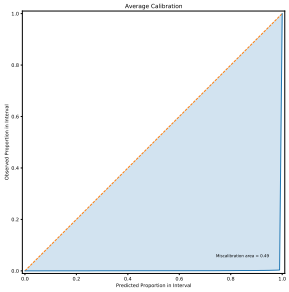
(a) Dish Washer



(b) Fridge



(c) Microwave



(d) Washer Dryer

Figure 11: Reliability Curve for Seq2Seq experiments for different appliances keeping MC dropout 0.2 and Number of iteration 100

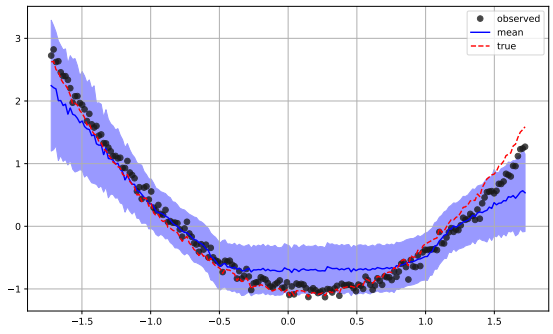


Figure 10: MC Dropout prediction on a toy dataset

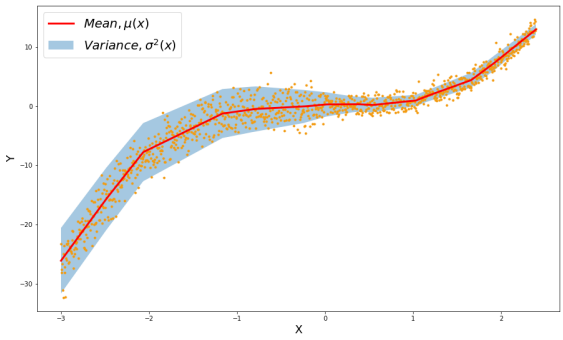


Figure 12: Predictions on a toy dataset using Deep Ensemble with $\mu(x)$ and $\sigma^2(x)$