FORECASTING ELECTRICITY DEMAND OF CHARGINGS STATIONS USING DEEP GAUSSIAN PROCESSES (DGP)

Villads Stokbro s175548, Johannes Boe Reiche s175549, Julian Róin Skovhus s174045

ABSTRACT

Understanding electricity demand of charging stations is an increasingly important task as the amount of electrical vehicles skyrocketed in the last decade. Time series forecasting is essential to optimize the electrical grid with many potential DL methodologies applicable. DGP offer a non-parametric approach which even works in scenarios with scarce data. We argue that using 100 inducing points initialized using K-means sampling is a good model configuration for the DGP for both short and long term forecasting on the Palo Alto dataset. The predictive quality is not state-of-the-art, but the uncertainty measures could be useful in a production setup.

1. INTRODUCTION

In the last couple of years, there has been a major increase in the accuracy of machine learning and deep learning models. As a consequence, software solutions based on these types of models are rapidly increasing and are being implemented across all sectors. However, the current state-of-the-art models within most fields are based on varieties of neural networks. A neural network such as the widely used convolutional neural network (CNN) can often achieve a high prediction accuracy, but gives no uncertainty measure of the predictions. Since machine learning and deep learning models have to make very important decisions e.g. in a hospital, it is critical that there is some kind of uncertainty measure for how confident the model is. Consequently, the interest in Bayesian models has grown since they can model the uncertainty of predictions[1].

1.1. Purpose of the project

This study utilizes deep Gaussian processes (DGP), which combines the advantages of deep learning such as minibatch training and stochastic gradient with Bayesian training allowing for an uncertainty measure of the predictions. The main focus of this paper is the use of DGP for time series modelling, with the focus on the number of inducing points used and the quality of uncertainty measures. The objective is not necessarily finding the most optimal model for the given problem, and the use of alternative machine learning models is therefore not investigated as in the paper by Hüttel et al. [2].

The data used in our work is the Palo Alto data set, which will be introduced in the following section. In short, the Palo Alto data set is a time series of the electricity demand for electric vehicles in Palo Alto (city in the US). In this paper, three research questions will lay the foundation of the experiments carried out and the discussion of the results obtained. The three research questions are:

1. *Which forecasting capabilities does the DGP have on both long and short term forecast horizons?*
2. *Are the uncertainty measures made by the DGP useful in a time series setting?*
3. *What effect does the number of inducing points and the initialization method for selection have on the performance?*

2. DATA

The Palo Alto data set is a spatio-temporal data set containing information on electricity demand of 82 charging stations in Palo Alto from 2011 to 2020. For the purpose of this project we aggregated the electricity demand on a daily basis for all charging stations, and then across the charging stations yielding a univariate time series, see figure 1. Initially, only a few charging stations were active, and charging stations were set up in new locations on a continuous basis. The data set was split into a training set and a test set: the training set consisted of all data up until 2019, and the data from 2019 constituted the test set which corresponds to 12,5 % of the data. In figure 1 it appears that the time series is non-stationary since the mean and variance are time-varying as supported by table 1. In mid 2017 there is a big drop in electricity demand, which could be due to a competitor entering the Palo Alto area. We also see that in 2018 and on wards there are huge fluctuations in the aggregated electricity demand.

	Train	Test
Count	2712	365
Mean	582.78	1198.9
Standard deviation	467.5	295.7

Table 1. Descriptive statistics on the Palo Alto data set.

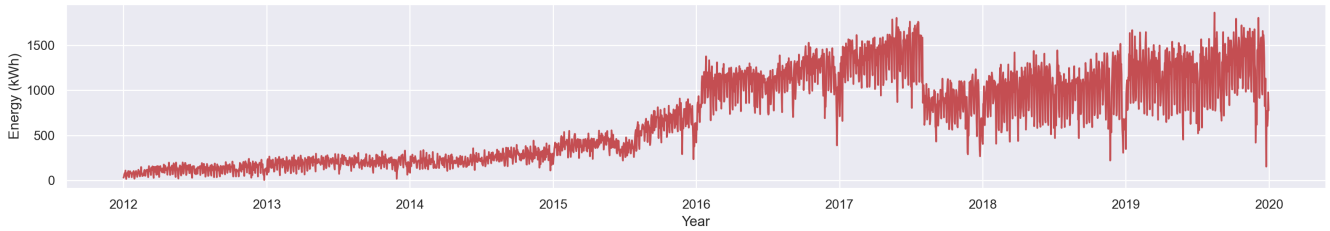

Fig. 1. Aggregated electricity demand for the entire period.

3. METHOD

This section will briefly describe how each of the research questions were investigated. We also present methodologies of preprocessing, finding hyperparameters and performance metrics.

Preprocessing: The preprocessing are inspired by the work of Hüttel et al. [2]. We have defined short-term as a 7-day period and long-term as a 30-day period. Consequently, we created two data sets. The data set for 7-day predictions had input from the previous 30 days electricity demand, yielding a 30-dimensional input and 7-dimensional target. For the data set for 30-day predictions the input was the previous 120 days and targets were the following 30 days. To enable the model to train on all different lags for both the input and target, we made the data set in a sliding window fashion using a shift of one day (see figure 2). This method was used for both data sets, and for the test set as well. These two data set was used for all experiments carried out. The input was normalized so that every dimension of the input it lies in the range [-1,1]. The targets were normalized to have zero mean and unit variance.

Fig. 2. Illustration of the sliding window used for the training set.

Hyperparameters: All models was trained locally on a CPU with 10 epochs, a learning rate of 0.01 and batch size 10. For 7-day predictions initial experiments showed that 3 hidden DGP layers was suitable, while a 2-layer DGP was suitable for 30-day predictions. After finding a reasonable configuration for the models, no further work was put into finding the optimal hyperparameters.

Performance metrics: To evaluate the models, two different metrics were used. The first one is the root mean square error (RMSE) as a measure for how accurate the point estimates are. To evaluate the quality of the uncertainty measures, we find the average percentage of observations that lies within the 95 % CI from the posterior predictive distribution of the DGP's for all forecasting horizons.

Research question 1: As a baseline model we choose a simple AR(7) for the 7-day predictions and an AR(30) for the 30-day predictions. We then initialized two DGP's, one for each data set. Predictive power of DGP and baseline models were evaluated using the RMSE for the two forecasting horizons.

Research question 2: To investigate the quality of the uncertainty measures made by the DGP's, we evaluated the 95 % CI on all forecasting horizons for all models using the performance metric described above.

Research question 3: To test the effect of the number of inducing points, we trained 3 different DGP's for each data set with an increasing amount of inducing points, namely 50, 100 and 200. Each model was then evaluated by both performance metrics. Since the models trained for 30-day predictions was much slower to train, we focused on the model for 7-day predictions when investigating the effect of methodology for choosing the inducing points. In total 6 models were trained on 7-day predictions. Three models were trained with random inducing points (50, 100 and 200) and three models were trained with the inducing points being equal to the cluster means of the converged k-means algorithm applied to the training data. This would ensure that we use areas of high density in the input space as a representation of the full input space.

4. THEORY

4.1. Gaussian Processes

To commence, we will introduce Gaussian processes (GPs) as they are central elements in DGPs and give the intuition to understand the functionalities of DGPs. We are in a multivariate regression setting where we seek to predict target values $\mathbf{Y} \in \mathbb{R}^{NxQ}$ from observations $\mathbf{X} \in \mathbb{R}^{NxD}$. In a traditional linear regression setting like neural networks, we would train a set of weights $\theta(\mathbf{X})$ that map our observations to the targets. For GPs, we assume that each datapoint $\mathbf{y}_n$ is generated from a latent function $f(\mathbf{x}_n)$ which we can determine. Assuming

we have noisy targets we have [3]:

$$\mathbf{y}_n = f(\mathbf{x_n}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon \mathbf{I}) \tag{1}$$

where f is drawn from a GP which is described only by a mean function $\mu(\mathbf{X})$ and a kernel function $k(\mathbf{X}, \mathbf{X}')$, $f \sim \mathcal{GP}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X}'))$ where $\mathbf{X}$ denotes the training set and $\mathbf{X}'$ denotes the test as outlined in section 3. A zero mean function is often used and therefore the form of the function f is general as it is determined purely by the choice of covariance matrix (kernel) which allows flexibility within our model. We will discuss the choice of kernel function k in section 4.2. It is the hyperparameters related to the kernel that are optimized during training. We seek to maximize the probability of getting targets $\mathbf{Y}$ given input $\mathbf{X}$. This corresponds to maximize the marginal likelihood of targets $\mathbf{Y}$ given the input $\mathbf{X}$:

$$p(\mathbf{Y}|\mathbf{X}) = \int_{\mathbf{dF}} \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{f}_n) p(\mathbf{f}_n|\mathbf{x}_n) \mathbf{dF} \tag{2}$$

where $\mathbf{F}$ is the collection of latent functions $\mathbf{F} = \{\mathbf{f}_n\}_n^N$. For GPs, $\mathbf{F}$ is normally distributed and thus (2) can be computed analytically [1]. In a DGP setting, non-linearities will arise and the integral in (2) will be intractable and different measures are needed to compute the marginal likelihood as we will discuss in section 4.2.

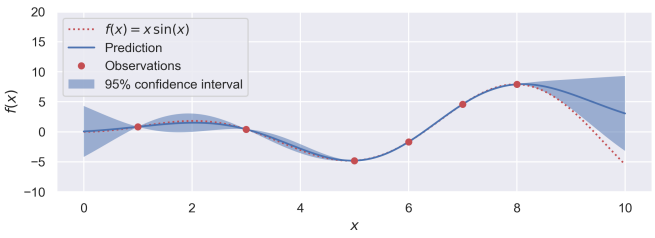

Fig. 3. Example of posterior distribution

In figure 3 the prediction of a GP is shown for a noise-free toy example. It appears how the model is uncertain where few observations are present and the prediction goes exactly through the training points with no uncertainty. The confidence interval related to the prediction is one of the powerful advantages for GPs, DGPs and Bayesian neural networks over traditional neural networks.

4.2. Deep Gaussian Processes

When we expand the framework to DGPs we essentially stack GPs vertically and/or horizontally to create multiple layers of GPs. Our implementation is made using `GPyTorch`, which is based on the paper by Salimbeni et al. [4]. In this implementation, we stack GDPs vertically to create a hierarchical structure. While for GPs the output is given as a mapping from the input, the output of a DGP is given by a mapping from the latent space $\mathbf{Z} \in \mathbb{R}^{NxH}$ where H denotes the number of hidden layers. In such a structure, the output of one GP which constitute a latent layer $\mathbf{Z}_i$ is the input to the following GP latent layer $\mathbf{Z}_{i+1}$. Rather than using a naive prior with mean and covariance zero, it is estimated in the initial latent layer and propagated through the network to obtain an optimal prior and thereby improving our predictions. We found two and three layer DGP to be sufficient for our purposes.
The structure of a DGP is depicted in a simplified diagram in figure 4. $\mathbf{F}_Z$ and $\mathbf{F}_Y$ constitute the collection of latent functions from the input space to the latent space and the latent space to the output space respectively.

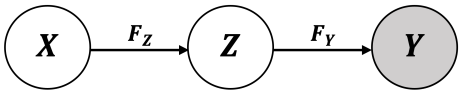

Fig. 4. Simplified illustration of a two-layer DGP with input, latent and output space

4.2.1. *Training*

We employ variational inference to approximate the integral in (1) using the methodology outlined in Salimbeni et al. [4]. We refer to the Salimbeni-paper for a detailed walkthrough. Essentially, we evaluate the ELBO loss:

$$\mathcal{L} = \mathrm{E}_q[\log p_\theta(\mathbf{Y}|\mathbf{F}_Z, \mathbf{F}_Y)] - \mathrm{KL}(q||p) \tag{3}$$

where the first term seeks to minimize the loss between the predicted mean and the observed values $\mathbf{Y}$. θ denotes the hyperparameters of the kernel which are optimized when minimizing the ELBO. The actual posterior distribution p is estimated using the approximate, sparse posterior q. It is approximated using inducing points which is a selected subset of the training data hence the term sparse. In terms of the GP, this mean that the kernel function is evaluated only for a subset of the training data. The approximated posterior q is assumed to be normally distributed and we can therefore sample from this distribution to obtain the predictions, which is not the case for p. The second term is the regularizing Kullback-Leibner divergence which ensures that q approximates p. Stochastic gradient descent is used to compute the maximized ELBO loss with respect to the kernel function parameters [5].
Inference with q does not only allow us to work with datasets with non-Gaussian likelihoods, but also reduces the runtime complexity of GP from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2K)$ where N is number of training points and K is the number of inducing points where $N \gg K$ [6].

We employed the automatic relevance determination (ARD) kernel [3]:

$$k(\mathbf{X}, \mathbf{X}') = \sigma_{ard}^2 e^{\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{i,q} - x_{j,q})^2} \tag{4}$$

where σ_{ard}^2 is the length scale denoting the similarity between training points and w_q is a weight for each of the latent layers. These are the hyperparameters related to the kernel which we optimize.

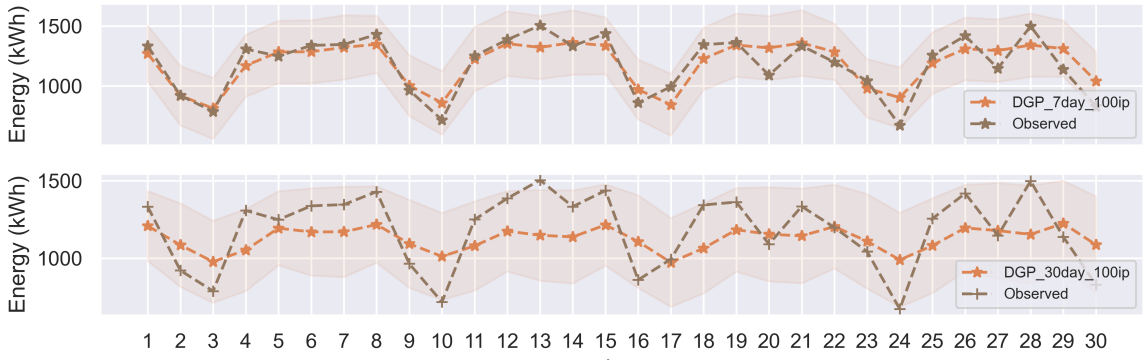

Fig. 5. 30 days forecast for March 2019 (part of the test set). **Upper**: The best 7-day model. Five predictions are made with the first prediction covering the first seven days in the plot etc. No overlapping of predictions is used. **Lower**: The prediction of best 30-day model, corresponding to one prediction.

5. RESULTS

The forecast window of the models are indicated by the first number, e.g. 7day. Inducing points are denoted by the second number, e.g. 50ip. If the inducing points are selected randomly, _R is included in the end of the model name while no label indicates selection with K-means.

5.1. 7-day forecast

In figure 6 the total RSME over all 7 forecast horizons can be seen. All DGP models outperform the baseline, with the worst performing model being DGP_7day_50ip_R. The best model is DGP_7day_100ip, with the difference in RSME to the baseline being 43, however the performance is similar to that of the other DGP's except DGP_7day_50ip_R. All models using K-means to initialise inducing points outperform the corresponding randomly initialised models. This effect seems to decrease as the number of inducing points increases.

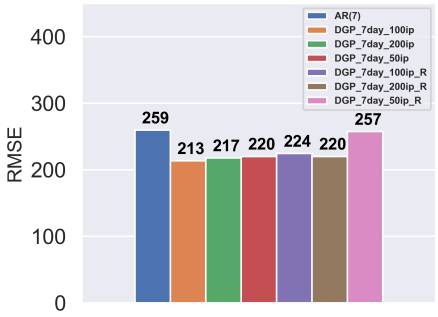

Fig. 6. RSME of 7 day forecast models, with the mean taken over all 7 forecast horizons.

In figure 7 the RSME on each of the 7 forecast horizons can be seen. The RSME of all DGP models seem to increase over the 7 horizons, since the predictions are bound to become more uncertain further into the future. The model with lowest total RSME described in section 5.1 outperforms the other models after day 2. DGP_7day_200ip has a slightly lower RSME for the first two days. All DGP models outperform the baseline for all days.

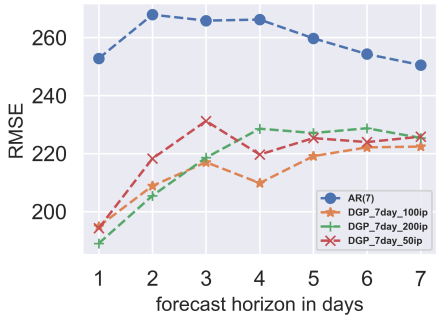

Fig. 7. RSME of each forecast horizon for the 7-day models.

In figure 9 the percentage of the observations within the confidence intervals from posterior predictive distribution of the DGP models can be seen. It is clear that the confidence intervals are too narrow, since only around 85 % of the observations are within. It also seems that the quality of the confidence intervals are similar for the different DGP models.

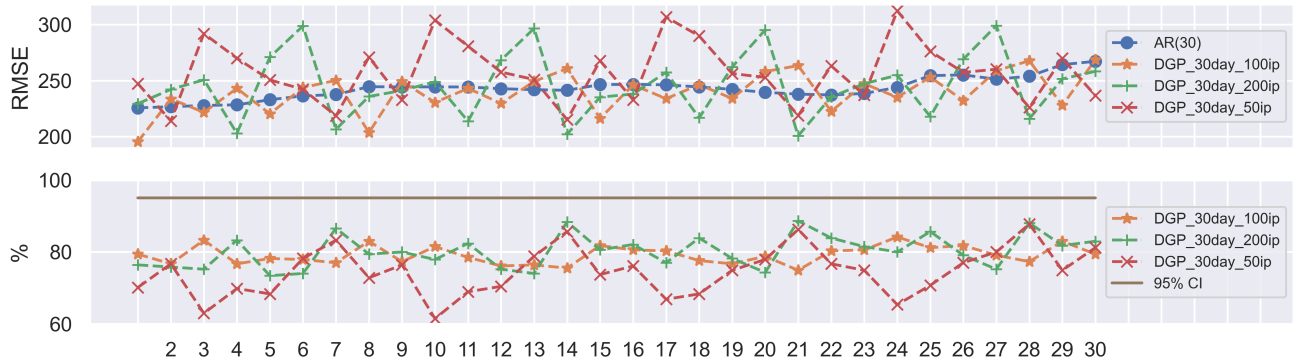

Fig. 8. **Upper**: RSME of each forecast horizon for the 30-day models. **Lower**: percentage of observations within the 95 CI stemming from the posterior predictive distribution of the probabilistic 30-day models

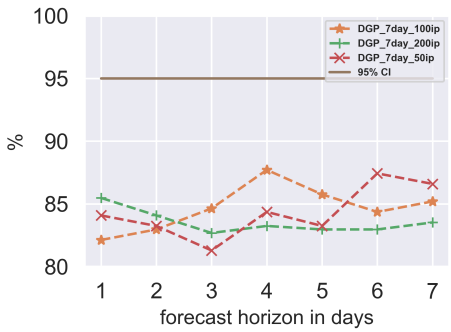

Fig. 9. Percentage of observations within the 95 CI stemming from the posterior predictive distribution of the probabilistic 7-day models.

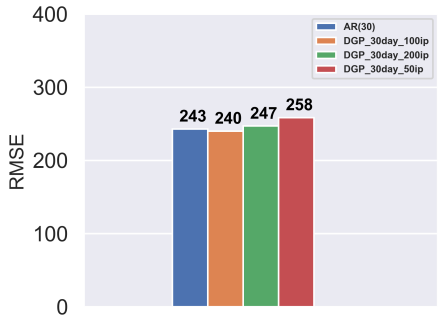

Fig. 10. RSME of 30 day forecast models, with the mean taken over all 30 forecast horizons.

5.2. 30-day forecast

From figure 10 it can be seen that the baseline AR(30) has similar performance to the DGP-models. The best model is DGP_30day_100ip, however the difference in RSME to the baseline is only 3. The ranking of the different DGP model is the same as for the seven day forecast models, with the second best being DGP_30day_200ip and the worst being DGP_30day_50ip. The difference between the performance of the DGP models is larger for the 30-day forecast models compared to the 7-day forecast models.

Looking at the upper subfigure of figure 8 it can be seen that the variance of the RSME of the different forecast horizons is large and periodic for the DGP models especially with 100 and 200 inducing points. It indicates that there are some periodic trends that the models do not learn. This is also the impression of the lower subfigure in figure 5, showing the predictions on March 2019 of DGP_30day_100ip. It seems that the predictions of the size of the weekly fluctuations are too small.

As can be seen from the lower subfigure of figure 8 the estimation of the size of the confidence intervals is too narrow similarly to what was seen for the 7-day models. However the fluctuations are larger especially for DGP_30day_50ip coinciding with the fluctuations of the RSME visualized in the upper subfigure.

Predictions on March 2019 of the best 30-day model and the best 7-day model can be seen in figure 5. The 30-day model has learned a periodic pattern of seven days, that is repeated for all 30 days. In general it is underestimating the weekly fluctuations. The 7-day model better predicts the size of the weekly fluctuations however the weekly patterns seems to be closely related with only minor adaptations to the local fluctuations in the individual weeks.

6. DISCUSSION

In this section, we will discuss the research questions stated initially in section 1.

6.1. Forecasting capabilities

For the 7-day forecast horizon the DGP models were able to significantly outperform the baseline model. If we compare to the best model (T-GCN) of [2] it achieves a RSME of 184 on the same forecast horizon, which is significantly better than the best DGP model (RSME 213). Therefore, it does not seem that the DGP is optimal in terms of the accuracy of predictions compared to other deep learning models. However, in their study no spatial aggregation of energy consumption was made and a different test set was considered thus the results are not directly comparable. In figure 5 we see that the predictions seem reasonable, and it does not appear that there are any systematic deviations from the observations. This is also the impression from figure 7, in which the RSME increases rather smoothly with the forecast horizon.
For the 30-day forecast horizon there is not a significant difference in the performance of the best DGP (RSME: 240) compared to the baseline (RSME: 243). This is in contrast to what was found in [2], in which the best model achieves a RSME of 161. In the results section we saw that the models seem to underestimate the periodic trends. It is also clear that the RSME for the first week of the 30-day forecast horizon is higher than for the 7-days model. It seems that the increased output space makes it difficult for the models to obtain a good fit. To counter this problem the hyperparameters of the training procedure could be changed, perhaps using a lower learning rate, more epochs and/or larger mini-batches. It could also be that increasing the complexity of the model architectures with more layers and hidden units could solve the problem, although we did not see immediate improvements by adding more layers.

6.2. Uncertainty measurements

From the results obtained it seems that the 95 % CI from the posterior predictive distribution are not in fact a true 95 % CI as they are too narrow. However for both 7-day and 30-day predictions at least one of the models seems to be fairly stable across the forecasting horizons. This implies that the CI could just be made wider to obtain a CI closer to % 95, although it is not clear how much wider it should be. If the CI becomes too wide it the uncertainty measure are less useful since the range becomes large.

We also saw only a slight increase in uncertainty for both 7-day and 30-day predictions on longer forecasting horizons in figure 5 and no specific lags that seems to have large uncertainty. This aligns with the lack of local fluctations in the predictive mean function as the models seems to find a more general weekly pattern. However especially the DGP for 7-day predictions, the CI are in general quite good. This indicates that the uncertainty measure from the posterior predictive distribution does have potential for being applied in a time series setting if the company wants to know the expected interval which the electricity demand lies within on a weekly basis.

6.3. Inducing points

We sought to investigate the effect of inducing points and their method of initial selection. For both 7 and 30 day forecasts, models with inducing points selected with K-means clustering outperformed models with inducing points initialized randomly. Thus the models benefit from having a guess on the locations of the inducing points based on the variance of the training points rather than arbitrary selection. This is in line with results obtained in previous works investigating the effect of selecting inducing points [7] [8].
Intuitively, incrementing the amount of inducing points will increase the predictive power simultaneously at the cost of runtime, since we approximate the training set with a larger subset. For both 7 and 30 day predictions, the results show that models with 50 inducing points were the worst performers and models with 100 inducing points were the best performers. The result indicates that 50 inducing points is too few to approximate the training set properly and we gain no additional information by using 200 inducing points. Models with 100 inducing points would even slightly outperform models with more inducing points, but this can possibly be attributed to stochasticity within the optimization of the models. For a larger training set, previous works have shown that more inducing points improve predictive power [7].

7. CONCLUSION

We found that a DGP with 100 inducing points initialized with a K-means methodology outperformed the baseline models on both short and long term. However the RMSE obtained by the models were not state of the art compared to other deep learning models. The DGP models where able to catch some weekly trends in the data, yet for long term predictions the model failed to catch the weekly fluctuations. In general it seemed that the CI for the posterior predictive distribution were too narrow, as most models had around 80 % of the test observations within the CI. It was evident that the inducing points heavily influenced the resutls, as the model with fewest inducing points performed worst on both short and long term forecasting. When comparing the initialization method we saw that choosing the inducing points randomly had worse performance than all of the models with K-means initialization, most evident when using only 50 inducing points.

8. REFERENCES

[1] Federico Bergamin, "Lecture 1: Gaussian processes," in *02463 Active Machine Learning and Agency*. DTU, 2020.

[2] Frederik Boe Hüttel, Inon Peled, Filipe Rodrigues, and Francisco C. Pereira, "Deep spatio-temporal forecasting of electrical vehicle charging demand," 2021.

[3] Andreas C. Damianou and Neil D. Lawrence, "Deep gaussian processes," arXiv, 2013.

[4] Hugh Salimbeni and Marc Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," 2017.

[5] Michael Nielsen, "Neural networks and deep learning," 2019.

[6] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai, "When gaussian process meets big data: A review of scalable gps," 2019.

[7] Anders Kirk Uhrenholt, Valentin Charvet, and Bjørn Sand Jensen, "Probabilistic selection of inducing points in sparse gaussian processes," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, Cassio de Campos and Marloes H. Maathuis, Eds., 2021, vol. 161 of *Proceedings of Machine Learning Research*, pp. 1035–1044.

[8] Thang D. Bui, José Miguel Hernández-Lobato, Yingzhen Li, Daniel Hernández-Lobato, and Richard E. Turner, "Training deep gaussian processes using stochastic expectation propagation and probabilistic backpropagation," 2015.