AAS 22-571 pre-print

COMPARISON OF GAUSSIAN PROCESSES AND NEURAL NETWORKS FOR THERMOSPHERIC DENSITY PREDICTIONS DURING QUIET TIME AND GEOMAGNETIC STORMS

Yiran Wang[*], Xiaoli Bai[†]

ABSTRACT

Atmospheric density is the predominately uncertain term among all factors affecting the drag force. Air drag force on satellites is affected by changes in thermospheric density and composition during magnetic storms. The current empirical models and physics-based models to predict the density often generate large errors because of the assumed parametric formulas or the uncertain input. This paper presents a data-driven density prediction framework based on two machine learning (ML) methods that integrates satellite accelerometer data, solar and geomagnetic indices, and two empirical models. The first ML method is based on a heteroscedastic and sparse Gaussian Processes (GPs) model. The second ML method uses neural networks (NN) to build the model. This paper investigates the density during quiet time in 2007, and storm conditions from 2003 to 2005. Based on the experiment results, we demonstrate that the two models can achieve high accuracy with reliable uncertainty estimations for both quiet time and storm time.

INTRODUCTION

For satellites at low altitudes, atmospheric drag is the dominant perturbation force and neutral mass density is presently the predominantly uncertain term among all the factors affecting the drag. The current best density prediction performances are often achieved by empirical models such as the Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Extended (NRLMSISE)[1] and Jacchia-Bowman (JB)[2] models, which however can generate very large errors during periods of high solar and geomagnetic activities and the uncertainties can reach beyond 100% under extreme conditions.[3] Although principled, physics-based models including the Thermospheric General Circulation Models[4] and Global Ionosphere Thermosphere Model[5] can overcome the limitation of the empirical models and are critical for advancing our knowledge of atmospheric physics, they currently suffer from largely uncertain input and boundary conditions and have not outperformed the empirical models.

Studies indicate that thermospheric density and composition change around the globe during magnetic storms.[6,7] The changes in thermospheric mass density will lead to fluctuation of air drag force on satellites. The density changes are one of the principal uncertainties for orbit determination

[*]Graduate Student, Department of Mechanical and Aerospace Engineering, Rutgers, The State University of New Jersey, NJ, 08854, yw619@rutgers.edu

[†]Associate Professor, Department of Mechanical and Aerospace Engineering, Rutgers, The State University of New Jersey, NJ, 08854, xiaoli.bai@rutgers.edu

and predictions of low-altitude satellites. The magnetic activity index Disturbance Storm Time (Dst) is often used to classify whether or not a storm has occurred, and to define the duration of a storm. The Dst values are also used to distinguish between quiet and disturbed geomagnetic conditions. Commonly, a minor storm is defined when the Dst range is between -30 nT to -50 nT, a moderate storm is defined when the Dst range is between -50 nT to -100 nT, an intense storm is defined when the Dst is smaller than -100 nT and the great storm is defined when the Dst is smaller than -250 nT.[8] In our experiments, we study the intense storm conditions when the Dst values are smaller than -100 nT.

Over the past years, many researchers have study thermospheric density with novel methods. Perez et al.[9] use neural networks for reducing the error in the density estimated by three empirical models. The inputs are from DTM2013, NRLMSISE00 and JB2008. To train the model the output data are the density values from CHAMP and GRACE satellites. The test results indicate that neural networks produce density estimates with fewer errors than the density from the three empirical models. Xiong et al.[10] establish an empirical model named CH-Therm-2018 using 9 years of accelerometer measurements from the CHAMP satellite with seven key parameters including height, solar flux index, day of the year, local magnetic time, geographic latitude, longitude, and the magnetic activities represented by the solar wind merging electric field. The performance of the model agrees well with the CHAMP satellite observations, and the model can predict the atmospheric density better than the NRLMSISE-00 model. Zhou et al.[11] introduce a multiple linear regression analysis with proper time shifts to study the mass density during the storm time. The input includes the total global Joule heating power and the Sym-H index from 2001 to 2004. Their results show the corrected mass density can reproduce the storm-time mass density better than the NRLMSISE-00 model. Chen et al.[12] model the storm-time atmospheric density using neural networks. The density data used for training the neural network models are derived from the measurements of the satellites CHAMP and GRACE. They use Dst, $F_{10.7}$, $F_{10.7A}$, local time, season, latitude, and altitude as the network input variables. They also explore different lengths of Dst and Ap that affect the performance. The results based on the EUVE satellite show the optimal performance of the mean ratio is 1.04. Oliveira et al.[13] investigate the satellite orbital drag effects at low earth orbit during magnetic storms. They use the GRACE and CHAMP data to estimate drag from the historical events. By analyzing several extreme storm cases in historical data, the results point out that the time duration of the storm is strongly associated with storm time orbital drag effects, and the orbital degradation is more severe for the most intense storms. Bonasera et al.[14] use the Monte Carlo method and deep ensembles to estimate the thermospheric mass density and the uncertainty from 2002 to 2021. The network work is trained using density data from CHAMP, GRACE, GOCE, SWARM-A, and SWARM-B, the orbital information, solar and geomagnetic indices as input. Richard et al.[15] use Principal Component Analysis (PCA) to build the model and test the density along the satellite orbit from 2002 to 2010.

In this paper, we extend one study about a machine learning (ML)-based, data-driven density prediction framework that integrates satellite accelerometer data, solar and geomagnetic indices, and empirical models including NRLMSISE-00 and JB2008. Two types of ML methods are explored. The first ML method is a heteroscedastic and sparse Gaussian Processes (GPs) model which has been demonstrated to be efficient and capable of handing input-dependent noise in our physics-based learning approach for density prediction.[16,17] The second ML method used in this paper is based on Neural Networks (NN). It is used as a comparison of the GPs model, and is explored to have a better performance.

We study both quiet time and storm time in this paper. A quiet time of 2007 is selected for the experiment, and this case is the same as the paper.[16] The training data is collected from 04/14/2007 to 06/30/2007, and the test data is from 07/01/2007 to 07/31/2007, which is in the future of the training section. For the storm situations, we test several cases in 2003, including the Halloween storm. We study a storm in 2004 which has been studied by Liu et al.,[18] and compare the results from the NN model with Liu's results. The test sections of the two cases in 2003 and 2004 are within the range of the training section. We also test two cases in 2005 in which the test sections are not included in the training section and are in the future time.

This paper makes the following contributions. First, we design a new frame work based on the previous model proposed in paper.[16] The new model can enhance the predicted performance of the GPs model during quiet time. Second, we explore the effect of Dst and SymH including the time delays on the results. Third, the GPs model and the NN model show great potential in predicting the thermospheric density during storm time. Both models are investigated and demonstrate to lead to results with high accuracy and high reliability in both the quiet time and storm time Third, we explore the effect of Dst and SymH including the time delays on the results.

The rest of this paper is organized as follows. Section 2 describes the methodologies of the sparse Gaussian processes model and the neural network model. We also introduce the metrics used to evaluate the model performance. Section 3 introduces the training data and test data used during the quiet time and discusses the performances of the two models. Section 4 studies some storm cases and investigates the performances of the models. Conclusions are presented in the last section.

METHODOLOGY

Heteroscedastic sparse Gaussian Processes Regression

Gaussian processes (GPs) regression is a non-parametric, Bayesian approach regression model. The full GPs uses all the samples to perform the prediction. The computational complexity to train a full GP is $O\left(n^3\right)$ due to the inversion of the $n \times n$ covariance matrix at each iteration, where n is the number of training data.

To reduce the computational complexity of training GPs, in this paper, we use a sparse Gaussian Process Regression method developed by Almosallam et al.[19,20] The method defines semi-parametric basis function models (BFM) to reduce the complexity to $O\left(mn^2\right)$ via a set of weights, where m is the number of basis functions.

Assuming there are given input variables $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^{n \times d}$ and the corresponding output $\mathbf{y} = \{y_i\}_{i=1}^n \in \mathbb{R}^n$, where n is the number of samples in the data set and d is the dimension of the input. The observed target y_i is assumed to be generated by a linear combination of m non-linear basis functions $\phi\left(\boldsymbol{x}_i\right) = \left[\phi_1\left(\boldsymbol{x}_i\right), \ldots, \phi_m\left(\boldsymbol{x}_i\right)\right] \in \mathbb{R}^m$ plus additional noise $\epsilon_i \sim \mathcal{N}\left(0, \beta^{-1}\right)$. This is expressed as:

$$y_i = \phi\left(\boldsymbol{x}_i\right) \boldsymbol{w} + \epsilon_i \tag{1}$$

where $\boldsymbol{w}$ is a vector of length m of real-valued coefficients.

In this paper, we choose the radial basis function (RBF) kernel as the basis function, which is defined as:

$$\phi_j\left(\boldsymbol{x}_i\right) = \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{p}_j\right)^T \Gamma_j^T \Gamma_j \left(\boldsymbol{x}_i - \boldsymbol{p}_j\right)\right) \tag{2}$$

where $\{p_j\}_{j=1}^{m} \in \mathbb{R}^{m \times d}$ are defined be the set of basis vectors associated with the basis functions, and $\Gamma_j \in \mathbb{R}^{d \times d}$ are the covariance matrices associated with each basis function. We refer to the model with such basis functions as Gaussian processes with Variable Length-scales (VL) covariance function. The VL covariance metrics is a bespoke isotropic covariance for each basis function with the scaler γ_j. The covariance metrics can be represented as $\Gamma_j = \mathbf{I}\gamma_j$. The obtained length-scales represent the importance of each variables.

Neural Network Model

In this work, we also investigate a model using neural networks (NNs) as the ML model in the proposed data-driven prediction framework. Neural networks are comprised of an input layer, one or more hidden layers, and an output layer. Each neuron connects to another and has an associated weight.

Standard neural networks for regression and classification does not capture model uncertainty. To overcome this shortage, Gal et al.[21] propose the Monte Carlo (MC) dropout method and prove the use of dropout in NNs can be interpreted as the Bayesian approximation of the Gaussian Processes. Dropout is originally used as a method for preventing overfitting. The dropout layer stochastically turns off some of the neurons each step during training with some probability. The probability is a parameter of the neural network called the "dropout rate". The MC dropout can be approximated by averaging the weights of the network during the update iteration. Mean and variance values are calculated using the MC method to approximate the predictive distribution.

The neural network model in this paper is implemented using Keras library from TensorFlow with Python 3.9. After several experiments, we choose a NN structure that can obtain the best performance for both quiet time and storm time, and the structure is as Figure 1 shows. The input is fully connected to the first hidden layer, which has 128 neurons. The second hidden is connected to the first hidden layer with 256 neurons and the dropout rate in this layer is set as 0.25. The output layer is followed by the second hidden layer containing one neuron that output the prediction. In the figure, the red lines represent the neurons are turned off because of the dropout setting.

The activation function in this neural network model is set as an exponential linear unit (ELU) function, which is defined as Equation. 3

$$ELU(z) = \begin{cases} \alpha, z \geq 0 \\ \alpha\left(e^{z} - 1\right), z < 0 \end{cases} \tag{3}$$

where z is the input from the last layer, and α is set as the default value 1. We choose this activation function because it avoids the zero feedback of its deviation if the input value is negative. The loss function is chosen as the mean square error (MSE), which is defined as Equation 4.

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{4}$$

.

Performance Metrics

To evaluate the performance of the proposed model, four metrics are used, including the Pearson correlation coefficient (R), root mean squared error (RMSE), mean ratio (Mean), and the coverage

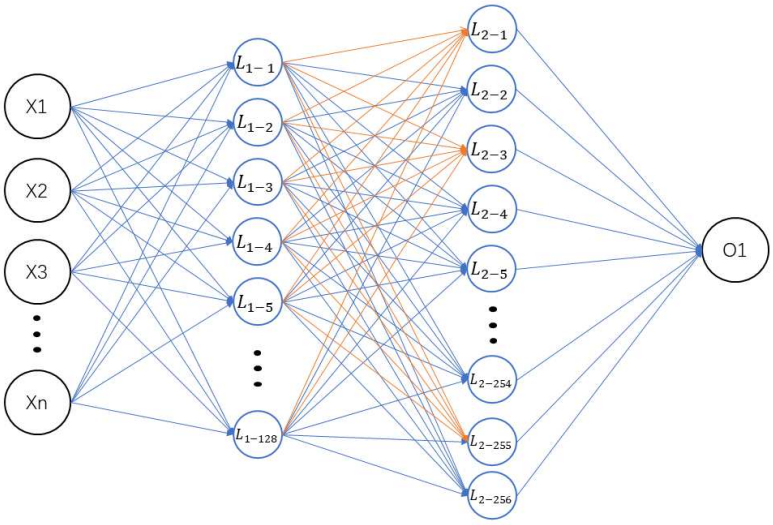

Figure 1: The NN model structure

rate of 2σ area (Cov Rate). These metrics can be mathematically expressed as:

$$R = \frac{\sum_{i=1}^{n} (\rho_i - \bar{\rho}) \left(\hat{\rho}_i - \bar{\hat{\rho}}\right)}{(n-1)\sigma_\rho \sigma_{\hat{\rho}}} \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{\rho}_i - \rho_i)^2} \tag{6}$$

$$\text{Cov Rate } = \frac{k}{n} \times 100\% \tag{7}$$

$$\text{Mean ratio } = \frac{\sum_{i=1}^{n} (\rho_i / \hat{\rho}_i)}{n} \tag{8}$$

Where ρ_i and $\hat{\rho}_i$ are the true density and predicted density. $\bar{\rho}$ represents the mean value of the density. σ_ρ and $\sigma_{\hat{\rho}}$ are the standard deviations of the truth and the predictions. n is the size of the data that are used for evaluation, and k is the number of the true density that is within the 2σ uncertainty boundaries estimated by the GP model or the NN model. A good performance shall have the R and Mean Ratio close to one, a Coverage Rate close to 100%, and the RMSE as small as possible.

QUIET TIME

Model Design

The model is a data-driven predictive framework that combines empirical models with high-resolution, accelerometer-inferred densities from the CHAMP satellite and other geomagnetic and

solar indices. The model we designed can be described as Equation 9 with the input variables used for the quiet time prediction task.

$$\lg\left(\hat{\rho}\left(t\right)\right) = f\begin{pmatrix} \lg\left(\rho_{JB}(t)\right), \ldots \lg\left(\rho_{JB}\left(t - D_{JB}t_s\right)\right) \\ \lg\left(\rho_{NRL}(t)\right), \ldots \lg\left(\rho_{NRL}\left(t - D_{NRL}t_s\right)\right) \\ F_{10.7}(t-1d), F_{10.7A}(t-1d) \\ Ap(t), F_{30}(t), Dst(t), SymH(t) \\ \rho_{CHAMP}(t - t_D) \end{pmatrix} \tag{9}$$

The $\hat{\rho}$ on the left side of the equation is the predicted density from GPs or the NN model at time t. ρ_{JB} and ρ_{NRL} are the densities estimated by the two empirical models JB2008 and NRLMSISE-00. t_s is the time delay step in density estimation , which is set as 60 sec. D_{JB} and D_{NRl} are the numbers of delays in JB2008 or NRLMSISE-00, which are set as 16 based. The value t_s and the delays of the empirical models are decided based on the experiments results. $1d$ is equal to 24 hours, here $F_{10.7}(t-1d)$ and $F_{10.7A}(t-1d)$ mean the daily value of $F_{10.7}$ solar flux and its 81-day averaged value with one-day lag. $Ap(t)$ is derived from the 3-hour geomagnetic index K_p. $F_{30}(t)$ is the daily value of F_{30} solar index. $Dst(t)$ is the value of magnetic activity index measuring the intensity of the globally symmetrical equatorial electrical current. $SymH(t)$ is the one-minute resolution version of the Dst index.[22] We also add $\rho_{CHAMP}(t - t_D)$, the density from CHAMP measurement with a time delay t_D. In this case, we set it as 300 sec.

The methods to pre-process the data for the GPs model and the NN model are different. For the NN model, to get better performances, we normalize the training data along each dimension separately into the range $[0, 1]$, as Equation 10 defines, and use the maximum and minimum value from the training section to process the test data.

$$N_i = \frac{x_i - x_{i_{min}}}{x_{i_{max}} - x_{i_{min}}} \tag{10}$$

N_i is the normalized value of each group of the variables, x_i is the original sample. $x_{i_{min}}$ and $x_{i_{max}}$ are the minimum value and maximum value of the corresponding variables x_i. i represents the input variables including the JB2008 model and the NRLMSISE-00 density model, $F_{10.7}$, $F_{10.7A}$, F_{30}, Ap, Dst, $SymH$ and density information from CHAMP. For the GPs model, standardizing the data, as Equation 11 defines, leads to better performances.

$$S_i = \frac{x_i - \mu_i}{\sigma_i} \tag{11}$$

S_i is the standard score of each dimension, x_i is the original sample. μ_i is the mean value of each dimension, and σ_i is the standard deviation value of the variables.

Database

The information of data in quiet time is as Table 1 shows, including the start and end date of the training and test section, the sample period, and the minimum Dst value during the training and test section. The test data is in the future time of the training data in this case.

Table 1: Quiet Time Data Information

Data Set	Training/Validation	Test
Time interval	04/14/2007 - 06/30/2007	07/01/2007-07/31/2007
Sampling period	1 min	1 min
Min Dst (nT)	-63	-46

There are two reasons we investigate this period. Firstly, in this database, the smallest Dst value in the training section is -64 nT, which is larger than the intense storm standard (-100 nT). It can be considered most of the time in the training section is in quiet time. The smallest Dst value in the test section is larger than -50 nT, which can be considered as a quiet time. Additionally, our previous paper[16] has studied this period, and proved the GPs model can obtain an accurate and robust density prediction than the empirical models with quality uncertainty estimations. Different from the previous model, we add SymH index in this study. Using this new model we can explore if the variable can help to enhance the GPs model during the quiet time.

During training process, 10% of the data is chosen by random seeds as the validation part. The test section is in the future time of the training section.

Result

Firstly we present the results from the GPs model with different basis functions. To evaluate model performance, we set 10 random seeds to train the model and test, then calculate the averaged results. The results using the GPs method to predict the density at quiet time with different basis function are presented in Table 2.

Table 2: Quiet Time: results from the GPs model

Basis function	3	5	8	10	15
R	0.9005	0.9066	0.9061	**0.9019**	0.9049
RMSE($\times 10^{-12}$)	0.2717	0.2713	0.2704	**0.2688**	0.2650
Cov Rate	0.9234	0.9194	0.9035	**0.9161**	0.9070
MeanRatio	0.9592	0.9487	0.9337	**0.9571**	0.9491

From the numerical results we can see all the R values are beyond 0.9, the RMSE values are around 0.27^{-12}, and the coverage rates are all beyond 0.90. We bold the optimal results with 10 basis functions. It has a relatively larger R value, and the coverage rate is beyond 91%. The corresponding RMSE value is the second smallest. The mean ratio value is the second-best among all the GPs models.

Next we investigate whether the NN model can accomplish the density prediction task with accurate results. Similar to the GPs model, we also set 10 random seeds to train the NN model. For every random seed, we use the MC dropout method to have a well-trained model and experiment with the training section and test section 500 times to get the standard deviation and mean value of the predictions. Finally, we calculate the average result from 10 random seeds as the result to compare with the true density value.

Table 3: Quiet Time: results from the NN model

	Training Data	Test Data
R	0.9892	0.9779
RMSE ($\times 10^{-12}$)	0.1420	0.1551
Coverage Rate	0.9737	0.9632
Mean Ratio	0.9952	1.0592

Comparing the results between the NN model in Table 3 and the optimal results from the GPs model bolded in Table 2, all the metrics from the NN model are better than the results from the GPs model except the mean ratio value. The deviations between the two mean ratio values and 1 are close to 0.05. The mean ratio values for the GPs model are all smaller than 1, while the mean ratio value from the NN model is larger than 1. This indicates the predicted values from the GPs model are larger than the true values, while the output from the NN model is smaller than the truth. We can see this in the following figures.

The whole test section is plotted in Figure 2, in which we plot the output of GPs model (red) with 10 basis functions with its uncertainty boundaries (red shadow), the predictions from the NN model (blue) with its uncertainties (blue shadow) and the truth (black). Figures 3 and 4 shows two sections of the test data. The y-axis is the density processed by the logarithm function. The x-axis is the corresponding time. Figure 3 shows a section of the test period of the two models. They can both capture the small fluctuations when the density values change, and the predicted values are very close to the true data. The shadowed boundaries can also cover the truth within 2σ uncertainties. Figure 4 shows the situation the true density is out of the uncertainty boundaries. At around July 12, 2007, 02:00, the bottom of the truth is out of the two boundaries. At around July 12, 2007, 05:00, the truth is out of the GPs uncertainties, but within the boundaries of the NN model. There exist other sections similar to this situation, which lead smaller coverage rate of the GPs model.

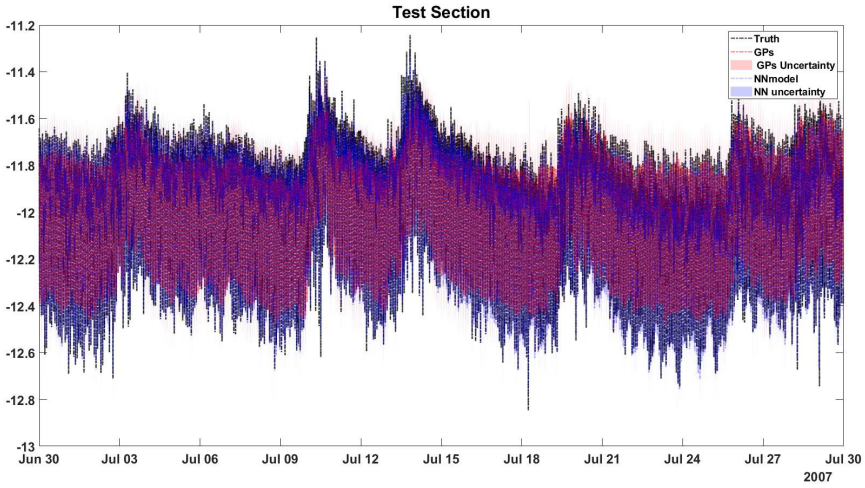

Figure 2: Quiet time: Test section

In general, both models can capture the periodic density change during quiet time. Evaluated by the R value, both models have a large R value which indicates agreement between the predictions and the truth. The corresponding RMSE values indicate the error between the truth and the pre-

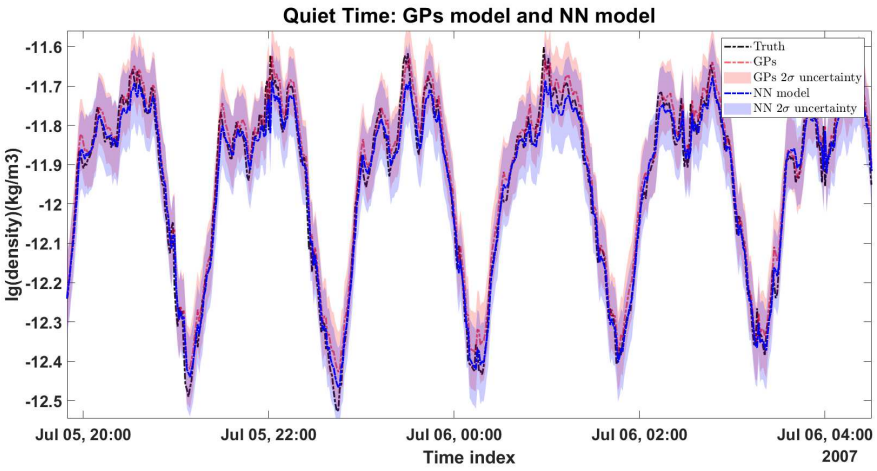

Figure 3: Quiet time: Test Section-1

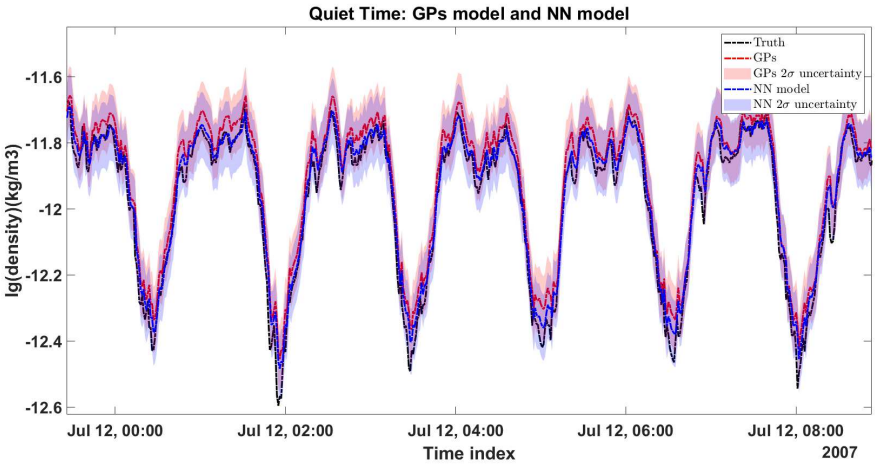

Figure 4: Quiet time: Test Section-2

diction is small. From the coverage rate, the predicted results from both models lie within the 2σ boundaries over 92% of cases in the test section. The NN model can capture the density distribution better, which presents a higher R value and a smaller RMSE value. The NN model also provides more reliable uncertainty boundaries, which leads a higher coverage rate value.

In summary, the NN model shows a more accurate prediction than the GPs model with better uncertainty estimation during quiet time.

STORM TIME

Database

We select the storm period when the Dst value is smaller than -100 nT from 2003/01/01 to 2005/12/31, and the information of the storms is presented in Table 4, including the duration and the minimum Dst values during the storms. The training data is from the beginning of 2003 to the middle of 2005, which covers the Halloween storm and contains enough data for training.

Table 4: Storm Period

Year	Date	Min Dst	Year	Date	Min Dst
2003	5.29-6.2	-164	2004	11.8-11.11	-397
	6.18-6.19	-165	2005	1.7-1.8	-112
	8.18-8.19	-140		1.18-1.19	-1.07
	10.29-10.31	**-432**		5.8-5.9	-117
	11.20-11.21	**-490**		5.15-5.16	-305
2004	1.22-1.23	-137		5.30-5.31	-127
	2.11-2.12	-107		6.13-6.14	-113
	3.10-3.11	-101		6.23-6.24	-101
	4.3-4.4	-149		7.10-7.11	-114
	7.22-7.28	**-208**		**8.24-8.25**	**-179**
	8.30-8.31	-128		**8.31-9.1**	**-119**

To prevents the model from being over-fitted, we set 10% of the training data as the validation part. The test cases we studied are bolded in Table 4. We define five test cases in Table 5.

Table 5: Test case list

Index	Period
Case-A	11/20/2003-11/21/2003
Case-B	10/29/2003-10/31/2003
Case-C	07/22/2004-07/28/2004
Case-D	08/24/2005-08/25/2005
Case-E	08/31/2005-09/01/2005

When the test section is during the period 11/20/2003-11/21/2003, we label this situation as "Case-A", which contains the largest storm because the Dst is the smallest during this period. In Case-A we study the effect of variables Dst and SymH, including the time delay of the two variables. Based on the results from Case-A, we then study the following test cases.

We label the situation as Case-B when test data is during the period 10/29/2003-10/31/2003, which covers the 2003 Halloween storm and contains the second biggest storm in 2003.

Case-C is when the test data is from 07/22/2004 to 07/28/2004. We will compare our results with Liu's study.[18] The training data in our experiment is as Table 4 defines, from the beginning of 2003 to the middle of 2005. The training period in our experiment is shorter than in Liu's settings.

Case-D is when the test section is from 08/24/2005 to 08/25/2005. The training data is defined as Table 4, from the beginning of 2003, until 07/11/2005. The test section in Case-D is not included in the training section, and it is in the future time of the training section.

Case-E is when the test section is from 08/31/2005 to 09/01/2005. It is similar to Case-D. The test section is not included in the training data, but the test section is in a further future time of the training section.

TEST ON 11/20/2003-11/21/2003

The storm cases are more complicated than the quiet time, so we introduce additional variables into the input to explore a better model. After several experiments, now the design of the model is described as Equation 12.

$$\lg\left(\hat{\rho}\left(t\right)\right) = f\left(\begin{array}{l} \lg\left(\rho_{JB}(t)\right), \ldots \lg\left(\rho_{JB}\left(t - D_{JB}t_s\right)\right) \\ \lg\left(\rho_{NRL}(t)\right), \ldots \lg\left(\rho_{NRL}\left(t - D_{NRL}t_s\right)\right) \\ F_{10.7}(t-1d), F_{10.7A}(t-1d) \\ Ap(t), F_{30}(t), \rho_{CHAMP}(t - t_D) \\ Dst(t - T_{Dst}), ..., Dst(t-1hr), Dst(t) \\ SymH(t - T_{SymH}), ..., SymH(t-1min), SymH(t) \end{array}\right) \tag{12}$$

The set of D_{JB}, D_{NRL} and t_D are the same as Equation 9. The differences between this model with the previous one defined in Equation 9 are the additional time delays of the Dst and SymH. To explore the time delay of Dst and SymH, we design several models with different inputs, as Table 6 shows. The unit of the Dst time delay is one hour, which is the same as its resolution. The unit of the SymH time delay is one minute. After several experiments, we choose the time delay of SymH in this model as 15 minutes and study the effect.

Table 6: Model definition

Model	1	2	3	4	5	6
T_{Dst}/hr	0	0	No Dst	3	0	3
T_{SymH}/min	0	No SymH	0	0	15	15
Total variable	41	40	40	44	56	59

Model-1 is with both SymH and Dst at the current time, and the other variables. The total number of input variables is 41. To explore the importance of the variable Dst and SymH, we design Model-2 and Model-3. Model-2 is with Dst at the current time and other variables, but without SymH data. The total number of the variables is 40. Model-3 is with SymH at the current time and other variables, but without Dst data. There are 40 variables in this model. Then to study the time delay of the important variables, we design Model-4 and Model-5. Model-4 contains Dst from the past three hours to the current time, and Symh at the current time. There are 44 variables in total. Model-5 contains Dst at the current time, and SymH from the past 15 minutes to current time. The number of the variables in this model is 56. Model-6 contains Dst from past three hours to the current time, SymH from past 15 minutes to current time and other variables. There are 59 variables in this model.

We use 10 different random seeds, and calculate the averaged results of the test section as Table 7.

Table 7: Case-A: results from the NN models

Model	1	2	3	4	5	6
R	0.8716	0.8844	0.8712	0.8956	0.8903	0.9034
RMSE($\times 10^{-12}$)	2.0364	2.0815	2.0952	1.6510	1.7641	1.5883
Cov Rate	0.8756	0.8764	0.9107	0.8977	0.9099	0.9256
Mean Ratio	1.0866	0.9644	1.0359	0.8980	1.0373	1.0231

Compared Model-1, Model-2 with Model-3, the R and RMSE values indicate that if the model does not include Dst (Model-2) or SymH (Model-3), the performance is worse than the model that contains both Dst and SymH (Model-1). Adding time delay of the important variables helps improve the model performances, as the R values increase from 0.87 to 0.89, and the RMSE values reduce from 2.03×10^{-12} to 1.65×10^{-12}. Model-6 contains the time delay of both Dst and SymH, and the result is the most optimal among all the designs.

Based on the results above, we decide on Model-6 with both Dst and Symh delays, and the other variables. Now the model can be defined as Equation 13.

$$\lg\left(\hat{\rho}\left(t\right)\right) = f\left(\begin{array}{l} \lg\left(\rho_{JB}(t)\right), \ldots \lg\left(\rho_{JB}\left(t - D_{JB}t_s\right)\right) \\ \lg\left(\rho_{NRL}(t)\right), \ldots \lg\left(\rho_{NRL}\left(t - D_{NRL}t_s\right)\right) \\ F_{10.7}(t-1d), F_{10.7A}(t-1d) \\ Ap(t), F_{30}(t), \rho_{CHAMP}(t - t_D) \\ Dst(t-3hr), ..., Dst(t-1hr), Dst(t) \\ SymH(t-15min), ..., SymH(t-1min), SymH(t) \end{array}\right) \tag{13}$$

Based on this model we will study the remaining cases.

TEST ON 10/29/2003-10/31/2003

In this case, the training data is selected as Table 4 defines, and we will show two test conditions: The first case is when the test section is included in the training section, which we label "Case-B1". The other case is when the test section is not included in the training section, which means the training section is the selected storms from the beginning of 2003 to the middle of 2005, except the period 10/29/2003-10/31/2003. To show the difference from Case-B1, we label this condition "Case-B2". We first show the results of Case-B1. The test section can be considered as a validation for the trained model.

Case-B1

The averaged GPs results with different basis functions are presented in Table 8.

Table 8: Case-B1: results from the GPs model

Basis function	2	3	5	8	10	15	20
R	0.8304	0.8175	0.8134	0.8010	0.8108	0.8063	**0.8051**
RMSE($\times 10^{-12}$)	2.0440	2.0303	2.0336	2.0157	2.0071	2.0142	**1.8755**
Cov	0.9132	0.9122	0.9121	0.9115	0.9122	0.9101	**0.9139**
MeanRatio	1.0713	1.0760	1.0515	1.0531	1.0391	1.0262	**1.0317**

From the numerical results, we cannot see which model is the most optimal with a specific basis function number. The GPs model with 2 basis functions has the largest R value, but the other three metrics are not the best compared with the others. The model with 20 basis functions has the smallest RMSE and highest coverage rate, but the R value is not the biggest value compared to the others. Considering the GPs model with small number of basis functions ignores more information, we choose the model with 20 basis functions to compare with the NN model.

The averaged results from the NN model are as Table 9 shows.

Table 9: Case-B1: results from the NN model

	Training Result	Test Result
R	0.9199	0.8123
RMSE ($\times 10^{-12}$)	0.7570	2.1130
Coverage Rate	0.9013	0.9199
Mean Ratio	1.0167	1.0192

The R value from the NN model is smaller than the GPs model with small basis functions (2, 3, and 5), but larger than the other GPs models. The RMSE value from the NN model is a bit larger than the GPs models, while the coverage rate and the mean ratio value are better than in the GPs models.

We plot the test results for the GPs with 20 basis function (the results are bolded in Table 8) in the first subplot, and the NN model in the second subplot in Figure 5. The black dash line represents the true density. The red marks represent the predicted results from the GP model. The uncertainties from the GPs model cannot be restored to the original data magnitude in a linear relationship, so we keep the data in a standardized format. The blue marks represent the output from the NN model. The reproduced density and the uncertainty can be restored to the logarithm format, so the y axis of the NN model is the value of the truth after the logarithm function processing. The third subplot shows the error values of the two models on the actual physical value. The red point represents the error between the GPs prediction and the true density. The blue point represents the error value from the NN model compared with true density. Both outputs are restored to the data with the original physical meaning, so the y-axis unit of this plot is $10^{-12}(kg/m^3)$. The subplot at the bottom shows the Dst values along with the time series.

There are two storms that happened as the Dst values go below -100 nT, one storm is around Oct.29, 12:00 to Oct.30, 12:00, and the other storm is around Oct.31, 00:00 to Oct.31, 12:00. From the figure, we can see both the GPs model and the NN model can capture most of the density data. Though around Oct.29, 12:00 and Oct.31, 00:00, as the true density changes dramatically, the predicted results from both models show relative larger errors around the two instance.

Case-B2

In Case-B2, the test section is not included in the training section. The averaged results from the GPs model are as Table 10 shows.

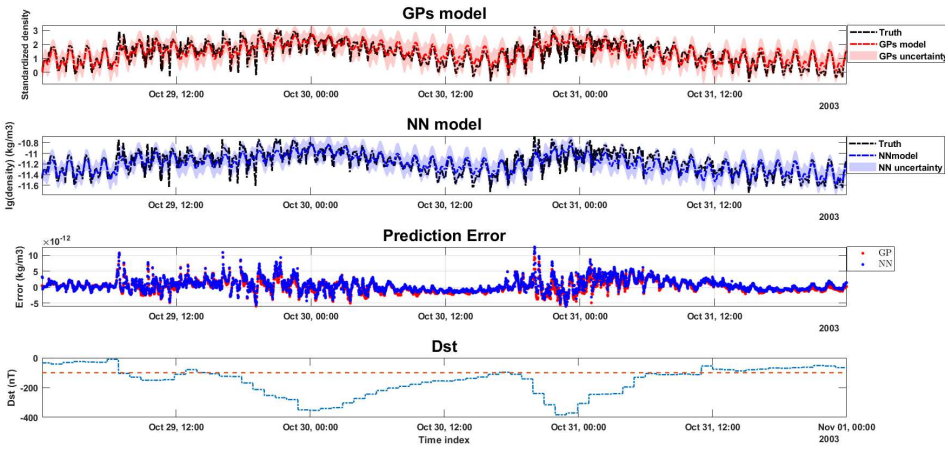

Figure 5: Test on 10/29/2003-10/31/2003: Case-B1

Table 10: Case-B2: results from the GPs model

Basis function	2	3	5	8	10	15	20
R	0.8217	0.8023	0.7649	0.7523	0.7792	**0.7840**	0.7870
RMSE($\times 10^{-12}$)	2.7222	2.4268	2.2538	2.2888	2.1976	**2.1393**	2.6134
Cov	0.7917	0.8428	0.8704	0.8423	0.8742	**0.8913**	0.7692
MeanRatio	1.3191	1.2358	1.1051	0.9639	1.0541	**1.0261**	0.8764

The optimal results of the GPs model in Case-B2 is with 15 basis functions. Compared the performance to Case-B1, the performance becomes worse when the test section is not included in the training section. The best R value in Case-B1 is 0.83, from the model with 2 basis functions, reduced to 0.82 in Case-B2, and the smallest RMSE increased from 1.8755×10^{-12} in Case-B1 from the model with 20 basis functions to 2.1393×10^{-12} in Case-B2 from the model with 15 basis functions. The coverage rate is reduced from 0.9139 in Case-B1 from the model with 20 basis functions to 0.8913 in Case-B2 from the model with 15 basis functions.

The results from the NN model are presented in Table 11.

Table 11: Case-B2: results from the NN model

	Training Result	Test Result
R	0.9106	0.7986
RMSE ($\times 10^{-12}$)	0.8012	2.1181
Coverage Rate	0.9618	0.8950
Mean Ratio	1.0216	1.0244

The performance of Case-B2 is worse than the results of Case-B1 for the NN model. The R in the test section is reduced from 0.8123 to 0.7986, and the coverage rate is from 0.9199 to 0.8950. The corresponding RMSE value is increased from 2.1130×10^{-12} to 2.1181×10^{-12}.

Compared to the two models in Case-B2, the NN model shows a better performance than the GPs model. Although the averaged R value from the NN model is not always larger than the GPs model, it is better than the GPs model with large basis functions (basis function number is larger than 3). The averaged RMSE value from the NN model is smaller than all the GPs averaged RMSE values. The coverage rate is also larger than all the GPs models. The mean ratio from the NN model is closer to 1 than in the GPs model.

In general, the performance of the test section is better when the test data is included in the training section. The two proposed models can also achieve accurate predictions with quality uncertainties when the test data is not included in the training data.

TEST ON 07/22/2004-07/28/2004

The test section, in this case, is from 07/22/2004 to 07/28/2004, and it is included in the training data. We label this situation as Case-C. We study and compare the results with Liu's[18] results, as Liu also studied the same test section.

In Liu's paper, they investigate the thermospheric density of the merging electric field during magnetic storms. The mass density is influenced by the latitude and the local time, so they analyzed the test section from 6 panels, including 3 latitude ranges(30° to 60°, -30° to 30°, and -30° to -60°), and the corresponding day-side and night-side. The training data is selected from 2005 to 2005 from the CHAMP satellite, and test the case from 07/22/2004 to 07/28/2004. The metrics they use are the corresponding coefficient R, the mean $E(\epsilon)$ and standard deviation values $\sigma(\epsilon)$ of the relative error(ϵ). The results from the 3 latitudes and 2 local time zones in their study can be summarized in Table 12. Cases 1, 2, and 3 correspond to the day-side of the three latitude ranges. Cases 4, 5, and 6 correspond to the night-side of the three latitude ranges.

Table 12: Case-C: Liu's result

Case	1	2	3	4	5	6
R	0.87	0.90	0.83	0.93	0.95	0.93
$E(\epsilon)$	8%	4%	15%	10%	14%	40%
$\sigma(\epsilon)$	17%	17%	28%	21%	18%	32%

Our experiment is also based on the CHAMP satellite along the same orbits. The results from our model is the averaged performance from the whole test section, but without the division of the latitude or the time zone. The averaged performance from the NN model is as Table 13 shows.

Table 13: Case-C: results from the NN model

	Training Result	Test Result
R	0.9107	0.8996
RMSE ($\times 10^{-12}$)	0.8006	0.8218
Coverage Rate	0.9616	0.9337
Mean Ratio	1.0266	1.0984
$E(\epsilon)$	2.16%	3.97%
$\sigma(\epsilon)$	22.78%	24.44%

We calculate the mean and standard deviation of the relative error values for the proposed NN models. Liu's results show the largest R value is 0.95 in case 5, which is higher than our R value 0.8996. But the corresponding mean value of the relative error (E) from Liu's method is 14%, which is much larger than 3.97% in our results. This indicates the errors between the truth and the predicted results from our model are smaller than their model. Our model has another advantage: the training section of our model is from the beginning of 2003 to the middle of 2005, which is smaller than their results. The NN model uses less data but the predicted error is smaller.

TEST ON 08/24/2005-08/25/2005

The test section is 08/24/2005-08/25/2005, which is not included in the training section, and we label it as "Case-D". Different from Case-A and Case-B, the test section in Case-D is in the future time of the training section. The averaged results from the GPs model are presented in Table 14.

Table 14: Case-D: results from the GPs model

Basis func	2	3	5	8	10	15	20
R	0.8979	0.8958	0.8978	**0.8985**	0.8991	0.9035	0.8989
RMSE($\times 10^{-12}$)	1.4524	1.4357	1.4143	**1.3926**	1.4327	1.4704	1.4408
Cov	0.9061	0.9012	0.9034	**0.9057**	0.8994	0.8959	0.8962
MeanRatio	0.9980	0.9981	0.9995	**0.9991**	0.9988	0.9988	0.9998

The R values are around 0.89, and when the basis function is equal to 15, the R value reaches to 0.90, which indicates a good agreement between the predicted results and the truth. Most of the RMSE values are larger than 1.4×10^{-12}, and the coverage rate is around 90%.

The results from the NN model of Case-D are presented in Table 15.

Table 15: Case-D: results from the NN model

	Training Result	Test Result
R	0.9251	0.8764
RMSE ($\times 10^{-12}$)	0.7965	1.0483
Coverage Rate	0.9634	0.9112
Mean Ratio	1.0262	1.0105

The results from the NN model are as Table 15 shows. The numerical results show a better RMSE value and a higher coverage rate than the GPs model. The averaged RMSE value is 1.0483×10^{-12}, smaller than all the RMSE values from the GPs model. The coverage rate is 0.9112, which is higher than all the results from the GPs model. The mean ratio values from both models are close to 1, while the GPs model is closer.

In Figure 6 we plot the predicted results from the GPs model with 8 basis functions because it has the smallest RMSE value and the R value and coverage rate are higher than most of the other GPs models in the first subplot. We also plot the results of the NN in the second subplot the error values in the third subplot, and the Dst value in the last subplot along with the time series.

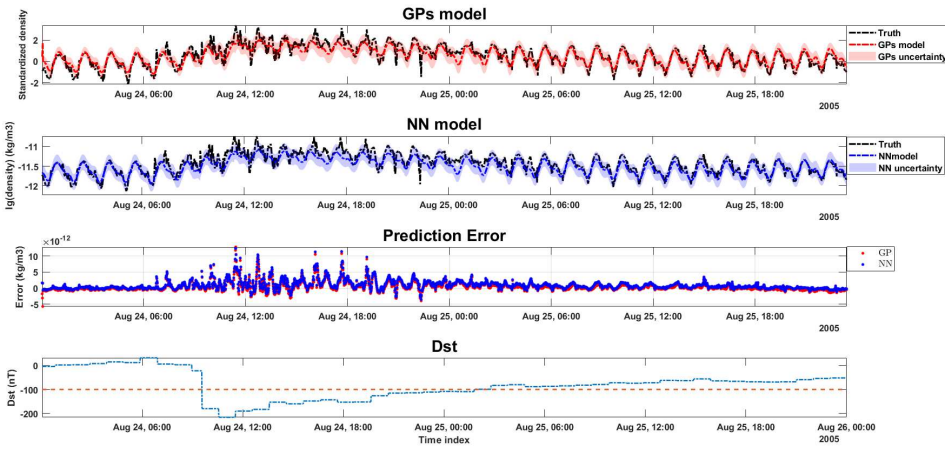

Figure 6: Test on 08/24/2005-08/25/2005

From the plot we can see the performances of the two models are quite close. But the predictions from the NN model are closer to the truth data at the bottom of each periodical change before and after the storm period, and this leads to a smaller RMSE value than the GPs model. The density changes along with the Dst value changes. We plot the prediction error which is the deviation of the true density in the third subplot. The large error values occur around Aug. 24, 08:00, and end around Aug. 25, 00:00. From the first two subplots, we can see both models cannot predict very accurately during this time. From the last subplot, we can see the Dst value changes over time. Around Aug.24, 08:00, the Dst value quickly dropped below -100 nT, which corresponds to the moment when a large error value appears. The Dst values keep under -100 nT until Aug. 25, 02:00, which is close to the moment when the large error values disappear.

In general, the NN model shows a smaller RMSE value and higher coverage rate than the GPs model. The mean ratio values of the two models are comparable.

TEST ON 08/31/2005-09/01/2005

The test section is from 08/31/2005 to 09/01/2005. In comparison to Case-D, the test section in Case-E is not included in the training section either, and the test is scheduled for a future time in the training section. The averaged results from the GPs model are as Table 16 shows.

Table 16: Case-E: results from the GPs model

Basis func	2	3	5	8	10	15	20
R	0.8836	0.8837	0.8782	0.8797	**0.8840**	0.8803	0.8782
RMSE($\times 10^{-12}$)	0.6908	0.7304	0.7228	0.7167	**0.7020**	0.7061	0.7178
Cov	0.9722	0.9589	0.9647	0.9659	**0.9659**	0.9594	0.9578
MeanRatio	1.0399	1.0853	1.0426	1.0285	**1.0328**	1.0110	1.0111

The R values from all the GPs models are around 0.88. The optimal results are from the GPs model when the GPs model has 2 basis functions. It has the smallest RMSE value and largest R

value and coverage rate. The GPs model with 10 basis function is the second best among all the GPs models. Considering the GPs model with small basis functions ignores more details than the model with larger basis functions, we choose the GPs model with 10 basis functions to compare with the NN model. It has the largest R value which is 0.8840, and the RMSE value is 0.7020 $\times 10^{-12}$. The coverage rate reaches 0.9659, which is the second highest.

The results from the NN model are as Table 17 presented.

Table 17: Case-E: results from the NN model

	Training Result	Test Result
R	0.9104	0.8889
RMSE ($\times 10^{-12}$)	0.7992	0.7014
Coverage Rate	0.9618	0.9661
Mean Ratio	1.0250	1.0266

The R value of the NN model is 0.8886, which is larger than the R value of all the GPs model. The RMSE value is 0.7014^{-12}, which is better than the model with 10 basis functions. The coverage rate for the NN model is 0.9661, slightly higher than the GPs model. Although the performances of the two models are very close, the NN model shows better results than the GPs model with 10 basis functions in Case-E. We plot the predicted results from the GPs model with 10 basis functions and the NN model, the predicted error of the two models on the actual physical value, and the distributions of the Dst values along the time in Figure 7.

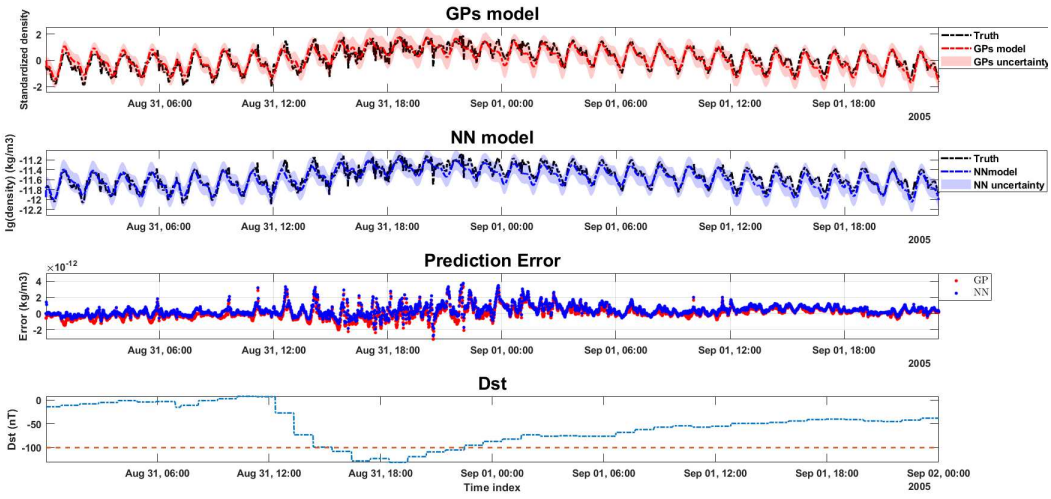

Figure 7: Test on 08/31/2005-09/01/2005

From Figure 7 we can see the predictions of the two models and the corresponding truth values, as well as the uncertainty boundaries. The prediction error in the third subplot indicates the predictions of the two models are very similar. The large error values occur from Aug. 31, 12:00 to Sep. 01, 02:00. The Dst value during this period indicates there is a storm starting around Aug. 31, 12:00, as

the Dst value is below -50 nT. Around Aug. 31, 14:00, the Dst value goes below - 100 nT. Around Sep. 01, 00:00, the Dst value goes up and becomes larger than -100 nT around this time. As Dst values become larger, the error becomes smaller. From the first two subplots, the true density points are within the uncertainty boundaries mostly.

CONCLUSION

This paper presents a Gaussian Processes model and a neural network model to estimate the thermospheric density with quality uncertainty estimations during both quiet and storm time.

The experiment of the quiet time is selected on 2007. The training data is from 04/14/27 to 06/30/2007, and test from 07/01/2007 to 07/31/2007. The optimal averaged R value for the GPs model is 0.9019, with the RMSE value 0.2688 $\times 10^{-12} kg/m^3$, the coverage rate reaches to 0.9161, and the mean ratio is 0.9571. The NN model in this paper, obtains better performances than the GPs model. The R value from the NN model is 0.9779, with a smaller RMSE values $0.1551 \times 10^{-12} kg/m^3$. The coverage rate is increased to 0.9632. In general, during the quiet time the two models can obtain good results, with high coverage rates of 2sigma uncertainty boundaries.

Some storm cases are also studied in this paper. The storm data are collected from 2003 to 2005. We first explore the effects of the variables Dst and SymH, and investigate different time delays of them to find an optimal model, based on the test case from 11/20/2003 to 11/21/2003. The model contains not only the densities from the empirical models, and the geomagnetic indices but also the Dst with a 3-hour delay and SymH with a 15-minute delay. We test the famous Halloween storm case in 2003 with two conditions based on the model. We try two situations during the Halloween storm. The first situation is when the test section is included in the training section. The other situation is when the test section is not included in the training section. The GPs model presents smaller RMSE values than the NN model for the condition the test section is included in the training section. On the second situation, the NN model shows a smaller RMSE value than the GPs in the condition the test section is not included in the training section. Besides, the NN model shows better performances on metrics R, coverage rate, and mean ratio for the two conditions.

Then we test the storm case from 07/22/2004 to 07/28/2004 and compare the performances with Liu's[18] study. Our NN model uses fewer data but obtains a smaller relative error value in this case. We also test two storm cases in 2005. One case is from 08/24/2005 to 08/25/2005, the other case is from 08/31/2005 to 09/01/2005. Both cases are not included in the training section, but are in the future time of the training section. The two models show very close performances in the two test cases, but there are some differences. The GPs model with small basis functions shows larger R values and a better mean ratio value than the NN model when tested on the period 08/24/2005 - 08/25/2005. The NN model in the storm cases shows smaller RMSE values and a higher coverage rate for the two test cases. The NN model also shows a larger R value and a better mean ratio than the GPs model on the case testing on 08/31/2005-09/01/2005.

Overall, the two models show great potential in predicting density with high-quality uncertainty estimations during both quiet time and storm time.

ACKNOWLEDGEMENTS

The research has been supported by National Science Foundation under award number 2149747.

REFERENCES

[1] J. Picone, A. Hedin, D. P. Drob, and A. Aikin, "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues," *Journal of Geophysical Research: Space Physics*, Vol. 107, No. A12, 2002, pp. SIA–15.

[2] B. Bowman, W. K. Tobiska, F. Marcos, C. Huang, C. Lin, and W. Burke, "A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices," *AIAA/AAS astrodynamics specialist conference and exhibit*, 2008, p. 6438.

[3] C. W. Group, "Cospar international reference atmosphere–2012," tech. rep., Technical report, The Committee on Space Research, 2012.

[4] "The Thermospheric General Circulation Models (TGCM's)," http://www.hao.ucar.edu/modeling/tgcm.

[5] "Global Ionosphere Thermosphere Model (GITM)," https://ccmc.gsfc.nasa.gov/models/modelinfo.php?model=GITM.

[6] A. Burns, T. Killeen, W. Deng, G. Carignan, and R. Roble, "Geomagnetic storm effects in the low- to middle-latitude upper thermosphere," *Journal of Geophysical Research: Space Physics*, Vol. 100, No. A8, 1995, pp. 14673–14691.

[7] A. Burns, T. Killeen, W. Wang, and R. Roble, "The solar-cycle-dependent response of the thermosphere to geomagnetic storms," *Journal of atmospheric and solar-terrestrial physics*, Vol. 66, No. 1, 2004, pp. 1–14.

[8] W. D. Gonzalez, B. T. Tsurutani, and A. L. C. De Gonzalez, "Interplanetary origin of geomagnetic storms," *Space Science Reviews*, Vol. 88, No. 3, 1999, pp. 529–562.

[9] D. Pérez and R. Bevilacqua, "Neural Network based calibration of atmospheric density models," *Acta Astronautica*, Vol. 110, 2015, pp. 58–76.

[10] C. Xiong, H. Lühr, M. Schmidt, M. Bloßfeld, and S. Rudenko, "An empirical model of the thermospheric mass density derived from CHAMP satellite," *Annales Geophysicae*, Vol. 36, Copernicus GmbH, 2018, pp. 1141–1152.

[11] Y. Zhou, S. Ma, H. Lühr, C. Xiong, and C. Reigber, "An empirical relation to correct storm-time thermospheric mass density modeled by NRLMSISE-00 with CHAMP satellite air drag data," *Advances in Space Research*, Vol. 43, No. 5, 2009, pp. 819–828.

[12] H. Chen, H. Liu, and T. Hanada, "Storm-time atmospheric density modeling using neural networks and its application in orbit propagation," *Advances in Space Research*, Vol. 53, No. 3, 2014, pp. 558–567.

[13] D. M. Oliveira, E. Zesta, H. Hayakawa, and A. Bhaskar, "Estimating satellite orbital drag during historical magnetic superstorms," *Space Weather*, Vol. 18, No. 11, 2020, p. e2020SW002472.

[14] S. Bonasera, G. Acciarini, J. A. Pérez-Hernández, B. Benson, E. Brown, E. Sutton, M. K. Jah, C. Bridges, and A. G. Baydin, "Dropout and Ensemble Networks for Thermospheric Density Uncertainty Estimation,"

[15] R. J. Licata, P. M. Mehta, W. K. Tobiska, and S. Huzurbazar, "Machine-Learned HASDM Thermospheric Mass Density Model With Uncertainty Quantification," *Space Weather*, Vol. 20, No. 4, 2022, p. e2021SW002915.

[16] T. Gao, H. Peng, and X. Bai, "Calibration of atmospheric density model based on Gaussian Processes," *Acta Astronautica*, Vol. 168, 2020, pp. 273–281.

[17] Y. Wang, H. Peng, X. Bai, J. T. Wang, and H. Wang, "Advance Thermospheric Density Predictions through Forecasting Geomagnetic and Solar Indices Based on Gaussian Processes," *AAS/AIAA Astrodynamics Specialist Conference*, 2021.

[18] R. Liu, H. Lühr, E. Doornbos, and S.-Y. Ma, "Thermospheric mass density variations during geomagnetic storms and a prediction model based on the merging electric field," *Annales Geophysicae*, Vol. 28, Copernicus GmbH, 2010, pp. 1633–1645.

[19] I. A. Almosallam, M. J. Jarvis, and S. J. Roberts, "GPz: Non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts," *Monthly Notices of the Royal Astronomical Society*, Vol. 462, No. 1, 2016, pp. 726–739.

[20] I. Almosallam, *Heteroscedastic Gaussian processes for uncertain and incomplete data*. PhD thesis, University of Oxford, 2017.

[21] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[22] J. A. Wanliss and K. M. Showalter, "High-resolution global storm index: Dst versus SYM-H," *Journal of Geophysical Research: Space Physics*, Vol. 111, No. A2, 2006.