



Neural Network based calibration of atmospheric density models

David Pérez, Riccardo Bevilacqua*

Mechanical and Aerospace Engineering Department, University of Florida, MAE-A Building, P.O. Box 116250, Gainesville, FL 32611-6250, USA

ARTICLE INFO

Article history:

Received 19 December 2014

Accepted 24 December 2014

Available online 19 January 2015

Keywords:

Thermospheric Density

Modeling

Neural Networks

ABSTRACT

Atmospheric density is the most important factor for accurate estimation of the drag force exerted on spacecraft at Low Earth orbits. Empirical models provide the most accurate estimation of the density currently available, although they still suffer from estimation errors. This work presents a novel approach based on Neural Networks for reducing the error in the density estimated by empirical models, along the orbit of a spacecraft. The Neural Networks take as inputs the density estimated by DTM-2013, NRLMSISE-00 and JB2008, three of the latest empirical atmospheric models available. Density estimated from the accelerometers of the CHAMP and GRACE missions are used as targets for the training, validation and testing of the Neural Networks. In addition, this work studies the use of the spacecraft's average speed as an input to the Neural Networks. The test results indicate that the Neural Networks produce density estimates with less error than the density from the three empirical models studied.

© 2015 IAA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Low Earth Orbits (LEO) contain the majority of artificial satellites currently in operation. At LEO, atmospheric drag force becomes a significant factor in the motion of spacecraft. For spacecraft below 700 km the drag force is the second most dominant force after gravity. Consequently, for accurate orbit propagation and determination, accurate estimation of the drag force is fundamental. The physical properties of the spacecraft (mass, crosswind area and drag coefficient) and the medium through which it flies (atmospheric winds and density of the medium) determine the drag force. However, as indicated in [1], the density is the parameter with the largest variations. Hence it is critical to properly model the

density in order to estimate the drag force accurately, and therefore to correctly model the motion of spacecraft in the thermosphere. The density variations are caused by the drastic and sudden changes that can occur in the thermosphere (80–640 km as defined in Ref. [2]). These changes are driven by the Sun's interaction with the thermosphere through solar radiation and solar wind. In turn, these phenomena are driven by solar activity which is still not fully understood. Therefore, accurately modeling the density in the thermosphere is a challenging problem that has received increasing interest in recent years. In addition, to improve orbit propagation and determination, a better density/drag estimation will be useful for further developing methods for maneuvering spacecraft at LEO using differential drag (see Refs. [3–8]). The doctoral dissertation of one of the authors [9] shows an example of how an accurate knowledge of the density/drag force can be used for creating realistic guidance trajectories for coplanar relative maneuvering of spacecraft using drag forces. Any drag/density estimation method used

* Corresponding author.

E-mail addresses: perezd4@ufl.edu (D. Pérez), bevilr@ufl.edu (R. Bevilacqua).

Nomenclature	
A	Spacecraft cross-wind section area for chaser and target spacecraft
a_d	Drag acceleration
ATMOP	Advanced Thermosphere Modeling for Orbit Prediction
B	Bias in the artificial neuron
CHAMP	CHALLENGING Minisatellite Payload
C_D	Spacecraft drag coefficient
DCA	Dynamic Calibration of the Atmosphere
D_{DTM}	Number of delays for the density estimated by DTM-2013
D_{JB}	Number of delays for the density estimated by JB2008
D_{MSISE}	Number of delays for the density estimated by NRLMSISE-00
D_S	Number of delays for the averaged speed
DTM	Drag Temperature Model
Dst	Disturbance storm index, geomagnetic activity index
FTDNN	Feed Forward Time-Delay Neural Network
$f(\cdot)$	Overall nonlinear function for the FTDNN
F10.7	10.7 solar radio flux, solar extreme ultraviolet radiation index
F_{20}	20-cm solar flux
F_d	Magnitude of the drag force
$g(\cdot)$	Overall nonlinear function for the RTDNN
GRACE	Gravity Recovery and Climate Experiment
HASDM	High Accuracy Satellite Drag Model
JB	Jacchia-Bowman
LEO	Low Earth orbit
m	Spacecraft's mass
MADM	Modified Atmospheric Density Model
M10.7	Middle solar ultraviolet radiation index
MgII	Ratio of the irradiance at the core of the Mg II feature, solar radiation index
MSE	Mean squared error
MSIS	Mass Spectrometer and Incoherent Scatter Radar
n	Number of samples in a data set
N	Number of inputs for the Neural Networks
R	Pearson correlation coefficient
RMSE	Root mean squared error
RTDNN	Recurrent Time-Delay Neural Network
SIP	Solar Irradiance Platform
S10.7	Solar extreme ultraviolet radiation index
$s_{\hat{\rho}}, s_{\rho}$	Standard deviations of the Neural Network outputs and targets
\bar{s}	Spacecraft's speed averaged over the previous orbital period
t	Time
t_s	Sampling period of the density data
v_s	Spacecraft velocity vector magnitude with respect to the Earth's atmosphere
W	Weight in the artificial neuron
x	Input of the artificial neuron
y	Output of the artificial neuron
Y10.7	Solar X-ray emission index
ρ	Atmospheric density
$\hat{\rho}$	Atmospheric density estimated by the Neural Networks
ρ_{DTM}	Atmospheric density estimated by DTM-2013
ρ_{JB}	Atmospheric density estimated by JB2008
ρ_{MSISE}	Atmospheric density estimated by NRLMSISE-00

for autonomous maneuvering using the drag forces, must be able to run on-board and to provide real time density estimations.

Since the early 1950s many different atmospheric models have been developed for calculating the main parameters of the thermosphere, including density. A summary of the different models available is presented by Vallado in [10], chapter 8.6.2. These models can be classified into empirical and physics-based models. Physics-based models, also known as global circulation models, solve the equations that govern the thermal, electromagnetic, chemical, and flow dynamics in different regions of the atmosphere. However, the physics-based models can suffer from bias caused by misrepresentation of the physics in the atmosphere (see Ref. [11]). In addition, physics-based models are computationally intensive since they calculate the state which often contains large numbers of variables and this is done over large portions of the atmosphere. For these reasons, physics-based models are not commonly used for orbit propagation and determination and are certainly not suitable for onboard implementation.

The alternative to the physics-based models are the empirical atmospheric models. As explained by Vallado

et al. [12], most of the recent efforts in this field are focused on improving existing models. This has resulted in classes of atmospheric models consisting of improvements of a previously developed model. The Jacchia models are one of the most used classes of models. The earliest of these empirical atmospheric models is the Jacchia 1960 model [13], which uses an empirical formula that estimates the density as a function of the geometric height, the 20-cm solar flux (F_{20}) and the angular distance to the center of the diurnal solar bulge. Further improvements on this model, include Jacchia models from 1970 [14], 1971 [15], 1977 [16], Jacchia-Roberts [17], up to Jacchia-Bowman 2006 (JB2006) [18] and 2008 (JB2008) [19]. The Mass Spectrometer and Incoherent Scatter Radar model (MSIS-77) [20] is the first model of the MSIS class of empirical models. MSIS-77 uses data from satellites as well as ground-based measurements from incoherent scatter radars to estimate density. Several improvements to the original MSIS from 1977 were made, including MSIS-86 [21], MSISE-90 ([22]), and NRLMSISE-00 developed by the U.S. Naval Research Laboratory [23]. Another class of empirical atmospheric models is the DTM class. The earliest of these models is the Drag Temperature Model (DTM-78) [24], which was developed in terms of spherical

harmonics, using data covering nearly two solar cycles. This model has been further developed as DTM-94 [25]; DTM-2000 [26], the first model to use the MgII index, which represents the solar UV and EUV emissions instead of the F10.7 index; DTM-2009 [27], used assimilated density data derived from the accelerometers onboard the CHALLENGING Minisatellite Payload (CHAMP) [28] and Gravity Recovery and Climate Experiment (GRACE) missions [29]; DTM-2012 [30], developed by the Advanced Thermosphere Modeling for Orbit Prediction (ATMOP) project and DTM-2013 [31], which is the latest of the DTM models.

There has also been some interest in developing methods for calibrating previously developed models. The Modified Atmospheric Density Model (MADM) [32] uses observations from one calibration satellite to generate a global calibration factor that allows for the improvement of the Jacchia 1970 model. A further refinement of these principles resulted in the High Accuracy Satellite Drag Model (HASDM) [33–35]. HASDM uses the Dynamic Calibration of the Atmosphere (DCA) [36] to calibrate the Jacchia 1970 model, based on observations from 75 calibration satellites, and thus produces an improved time varying density field. In addition, a method has been developed for exploiting precision orbit ephemeris (POE) of a spacecraft as observations to calibrate the density estimated by an empirical atmospheric density model along its orbit [37–39]. The models calibrated using this method were Jacchia 1971, Jacchia-Roberts, Committee on Space Research (COSPAR) International Reference Atmosphere (CIRA) 1972 [40], MSISE-90, and NRLMSISE-00. Shoemaker et al. developed a method for correcting (calibrating) a density model using a tomography based approach [41]. Measurements of the changes of the specific energy of spacecraft are used to solve for correction factors using Tikhonov regularization.

Neural Networks have been widely used for many different applications over the last couple of decades (see [42–45]). Over the last two years, there has been an increasing interest in utilizing Neural Networks for applications in the field of space weather. Yu et al. [46] presented a method for calculating the third adiabatic invariant L^* using a Neural Network instead of the computationally expensive line integration method used in the past. The Neural Networks were trained using empirical magnetic field models. This method was further extended in [47] to drift shells obtained from a physics-based magnetic field model. Choury et al. [48] presented a method for forecasting the exospheric temperature, which can be used for obtaining more accurate density predictions. Ref. [1] presents a method for predicting the density of a spacecraft along its future orbit using Neural Networks, provided that the value of the density at the present is known.

This work presents a method for calibrating empirical density models in order to obtain a more accurate estimate of the density. The method consists of combining the outputs of the three latest models of the MSIS, DTM and Jacchia classes of models (NRLMSISE-00, DTM-2013 and JB2008 respectively), along the orbits of a spacecraft. By using the densities estimated by the three models along

the orbit of the spacecraft, as inputs to Time Delay Neural Networks, a more accurate value for the density is produced. This method exploits the flexibility of Neural Networks to process diverse inputs, in order to calibrate multiple models at the same time instead of only one as previously done [32–39]. The Neural Networks are trained and validated using accelerometer-derived data from the GRACE [29] mission, during January 2003 as targets. In essence the networks combine the density estimates from the three models and improve them by assimilating the accelerometer density data. Hence, the networks calibrate the density estimates using the assimilated data. An evaluation of performances is obtained through tests using the accelerometer-derived data as targets, which were collected by the CHAMP [28] mission. In the first tests the Neural Networks and the models are tested along the orbit of CHAMP from January to September 2007. Two days from this time interval were selected to examine the performance of the Neural Networks and the models during days of high and low solar and geomagnetic activity. A final test was done using CHMAP density data from 2004 to 2008.

Two different Neural Networks architectures are tested: a Feed Forward Time Delay Neural Network (FTDNN), and a Recurrent Time Delay Neural Network (RTDNN). The FTDNN was previously used in [1] for forecasting the density along the orbit of a spacecraft. Additional tests are presented in which the average speed of the spacecraft over its previous orbital period is used as an additional input to the Neural Networks. Using this additional input takes advantage of the relationship between the behavior in time of the spacecraft speed and its acceleration. This in turn, provides information about the evolution of the orbit (which is affected by the density via the drag force) to the Neural Networks. The method presented here can be used for on-board implementation since it would only require running the three empirical models and the Neural Networks.

The foremost contributions of this work are as follows:

- 1) Development of a novel calibration method for combining the density estimated by different empirical models and assimilate density measurements to obtain an improved density estimate.
- 2) Testing of two different Neural Network structures: the FTDNN and a RTDNN. In the tests, different numbers of delays for each input for the two Neural Network structures were tested to find an appropriate combination.
- 3) The use of the average speed over the previous orbit of the spacecraft as an additional input to the Neural Network is proposed and tested.
- 4) Comparison of the results obtained in the estimation of the density between the three empirical models (DTM-2013, JB2008, NRLMSISE-00) and the Neural Networks. These results cover five years (from 2004 to 2008) of CHAMP density data which are used as targets.

The paper is organized as follows. Section 2 describes the atmospheric drag, density and the three empirical atmospheric models used. Section 3 describes the two

Neural Networks architectures used. Section 3 is dedicated to the data used for training, validating, and testing the Neural Networks. Section 4 presents the results of the different tests performed using the Neural Networks and evaluates their performances. Section 5 presents the conclusions.

The Neural Networks presented in this work are available for download at the Mathworks File Exchange webpage.¹

2. Calculating the drag force and atmospheric density

The drag equation can be used for calculating the drag force experienced by spacecraft at LEO. Even though this equation has limited applications at LEO, as pointed out by Vallado et al. [12], it is still useful for showing the main parameters that determine the drag force and it still widely used in the literature. The magnitude of the aerodynamic drag force as given by the drag equation is as follows

$$F_d = a_d m = \frac{1}{2} \rho C_D v_s^2 A \quad (1)$$

where F_d is the magnitude of the drag force, a_d is the magnitude of the drag acceleration, m is the spacecraft mass, ρ is the density of the local atmosphere, C_D is the drag coefficient, v_s is the speed of the spacecraft relative to the medium and A is the cross-sectional area perpendicular to the atmosphere's relative velocity. The mass and the cross-sectional area are known by design, provided that the attitude of the spacecraft is known, but the density, drag coefficient, and atmospheric wind must be estimated to calculate the drag force. The focus of this work is on the estimation of the density. Fig. 1 shows the profile of the density as measured by the CHAMP satellite along its orbit during May the 20th 2002.

The acceleration of the spacecraft is the sum of the gravitational acceleration, and the acceleration caused by all perturbations, including the drag force. Since the acceleration is the time derivative of velocity, there is a relationship between the speed (magnitude of the velocity vector) of the spacecraft, and the drag force and therefore the atmospheric density. The time evolution of the speed therefore is connected to the atmospheric density. The speed of the spacecraft is often available from its navigation systems, since the velocity of a spacecraft can be obtained by filtering the position of the spacecraft. An example of such method can be seen in [49], in which a Kalman filter and an Extended Kalman filter are used to obtain the velocity vector and angular velocity from the position vector and quaternions. In this work the use of the speed of the spacecraft as an additional input to obtain more accurate estimation for the density is proposed and successfully tested.

2.1. Empirical atmospheric models

This section present the three empirical atmospheric models selected as inputs for the Neural Networks. These three models are the latest versions of the Jacchia, MSIS and DTM models.

2.1.1. NRLMSISE-00

NRLMSISE-00 allows for the estimation of the temperature, composition and density of the thermosphere at a given position and time. As explained by Picone et al. [23], NRLMSISE-00 combines data from satellite accelerometers, and orbits (including the Jacchia and Barlier datasets); molecular oxygen density (obtained from the Solar Maximum Mission); and temperature obtained from incoherent scatter radar. NRLMSISE-00 takes as inputs: geodetic latitude and altitude, longitude, time, local apparent solar time, daily value of F10.7 solar extreme ultraviolet radiation index and its 81 day average, and the daily magnetic index (AP). NRLMSISE-00 FORTRAN code can be obtained from [50], while MATLAB and Simulink versions are part of the Aerospace Toolbox.

2.1.2. Jacchia-Bowman 2008

JB2008 estimates the temperature, total mass density at a given position and time and the exospheric temperature above that position. JB2008 combines data from satellites and accelerometers, including data from the CHAMP and GRACE missions. JB2008 combines solar radiation indices along with the Dst geomagnetic index. This combination results in an improvement in density estimations. According to Bowman et al. [19] during non-storm periods JB2008 provides a reduction on the density estimation error of over 5% in comparison with Jacchia 70 and NRLMSISE-00, while during major storms the error reduction increases up to 35% and 16% in comparison with Jacchia 70 and NRLMSISE-00, respectively. The inputs for JB2008 are right ascension, geocentric latitude, altitude, time, right ascension of the sun, declination of the sun, daily value of F10.7 solar extreme ultraviolet radiation index (with a one day lag) and its 81 day average, daily value of S10.7 solar extreme ultraviolet radiation index (with a one day lag) and its 81 day average, daily value of M10.7 middle solar ultraviolet radiation index (with a two day lag) and its 81 day average, daily value of Y10.7 solar X-ray emission index (with a five day lag) and its 81 day average, and the disturbance storm time index Dst. FORTRAN code for JB2008 can be obtained from [51]. The code includes a main driver code that reads an input file containing positions and corresponding times. Based on these inputs the main driver code reads the solar and geomagnetic indices from additional files and calculates all the other inputs (such as sun position). The main driver code calls the JB2008 code and calculates its outputs which are stored in an output file.

2.1.3. DTM-2013

DTM-2013 calculates the temperature, composition and density at a determined position and time, and the exospheric temperature above that position. DTM-2013 uses the F30 solar radiation index, scaled to the F10.7 index

¹ Link <http://www.mathworks.com/matlabcentral/fileexchange/49370-neural-network-calibrators-for-atmospheric-density-models>

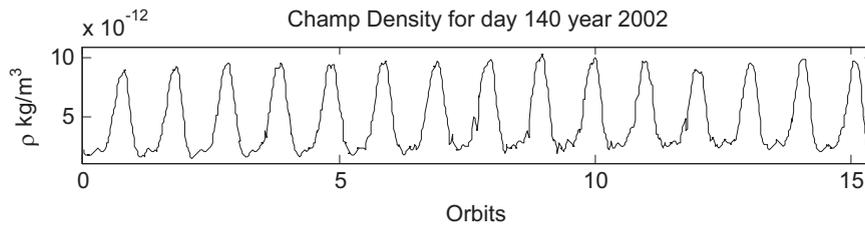


Fig. 1. Density from CHAMP day 140 of 2002 (obtained from [1]).

along with assimilated density data from the CHAMP, GRACE, Stella, Starlette and GOCE missions. The inputs for DTM-2013 are altitude, latitude, longitude, time, the F30 solar radiation index and the Kp geomagnetic index. FORTRAN code for DTM-2013 is available at the website of the Advanced Thermosphere Modeling for Orbit Prediction (ATMOP) project [52]. The code includes a wrapper routine that provides a time and a position, computes the remaining inputs (solar and geomagnetic inputs) and runs the main DTM-2013 routine.

The 81 day averages present a challenge for using the empirical models for estimating the current density (as well as for forecasting) since they are calculated using the value for the index over the past 40 days, the present day and the future 40 days. Forecasts for the indices can be obtained from models such as the Solar Irradiance Platform (SIP) (formerly known as SOLAR2000) [53,54]. HASDM uses these forecasted indices to be able to obtain current estimations of the density and also to forecast the density itself.

3. Neural Networks

Artificial Neural Networks are mathematical artifacts designed to resemble biological brains. Neural Networks are made up of artificial neurons as the one illustrated in Fig. 2. Artificial neurons are input output processing units, characterized by sets of inputs, outputs, biases, weights and a nonlinear transfer function. To build the Neural Network, the artificial neurons are connected in layers, which are structures of artificial neurons arranged in parallel. Each layer may have a different numbers of neurons, which allows for different structures that may work better for different applications. Neural Network can have loops feeding back data from the output or one of the hidden layers back to the input or other hidden layers.

The following equation is the mathematical expression for the neuron shown in Fig. 2

$$y = f\left(\sum_{j=1}^N W_j x_j + B\right) \quad (2)$$

where y is the output of the neuron, f is the transfer function, N is the number of inputs, W_j is the j th weight, x_j is the j th input, and B is the bias.

Neural Networks can store information that relates the inputs to the outputs in the biases and weights of their artificial neurons. During training, the weights and biases are adapted to minimize a performance function that measures how well the input–output relation of the specific problem is being represented by the neural

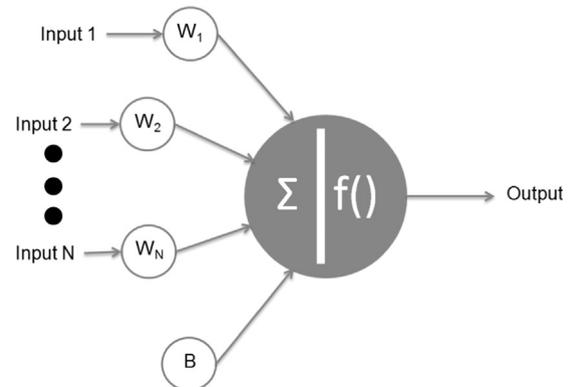


Fig. 2. Artificial neuron diagram (obtained from [1]).

network. A data set of inputs and outputs is selected specifically for training. The minimization of the performance function must be stopped at some point in order to prevent overtraining. The training process is stopped when one of the following conditions is satisfied: the performance function converges, the maximum number of epochs is reached, or the performance function of the neural network for a different data set, called a validation set, reaches a minimum. That way, the ability of the neural network to model new sets of inputs and outputs is preserved. As with any machine learning method, the selection of the training and validation data sets is very important; the better they represent the behaviors to be modeled by the Neural Network, the better the Neural Network will perform. After the Neural Network has been trained and validated, it is tested on a third group of data points (test set) to assess the quality of the Neural Network for modeling new data.

As shown by the universal approximation theorem (see [55,56]), any continuous function on a compact subset of \mathbf{R}^n can be approximated by a feed forward Neural Network with one hidden layer containing transfer functions satisfying certain conditions. Therefore, Neural Networks are good candidate for developing models for complex behaviors such as that one of value for the density. However, sufficient input output data, representing the behavior of the function to be modeled, must be available for training the Neural Network appropriately. Furthermore, overfitting (or overtraining), a condition in which the Neural Network has retained too much information about the training set, must be avoided since it causes the Neural Network to lose its generalization ability. Conversely, underfitting (or undertraining) is the condition in which

the Neural Network does not have enough complexity to properly model the input output relationship. Overfitting and underfitting can be avoided by choosing a number of neurons that provide enough complexity to learn the behavior, without actually memorizing it and by selecting a training data set large enough.

3.1. Feed Forward Time-Delay Neural Network (FTDNN)

The FTDNN architecture contains a set of delays at the input layer that allows retention of the evolution of the inputs in time, and enhances the ability of the network for time series applications. The number of delays can be different for each input, which allows the network to deal with the differences in the time evolution of the inputs. Fig. 3 shows a diagram for a FTDNN with two inputs, two delays for one input and one for the other input, two nonlinear neurons in the input layer and one linear neuron in the output layer.

3.2. Recurrent Time-Delay Neural Network (RTDNN)

In essence a Recurrent Neural Network contains one or more loops that route the outputs of the layers or of the network itself back to other layers or the input layer. The loops allow recurrent Neural Networks to use past context [57]. Similarly to the FTDNN, the RTDNN architecture contains a set of delays at the input layer, which can be different for each input. In this work, the RTDNN architecture chosen includes a loop with a delay that feeds the output of the network back to the input layer, and thus it is autoregressive. The RTDNN used in this work has memory of the inputs and outputs in its internal state, providing context for both the inputs and outputs. Fig. 4 shows a diagram for a RTDNN with two inputs, two delays for one input and one for the other input, two nonlinear neurons in the input layer, one linear neuron in the output layer, and one delay in the feedback loop.

in the input layer and one linear neuron in the output layer. This Neural Network architecture can be trained with the loop open or closed. If the loop is open during training, the RTDNN has access to the targets through the open loop. The loop is closed for testing the RTDNN, which is the configuration used for testing and implementing the RTDNNs. The advantage of training the network with the loop open is that feedback outputs are treated as an additional input during training, and thus the classical back propagation method for training can be used. On the contrary, if the loop is closed, the RTDNN is trained having the same configuration it will have during testing and therefore during implementation; however, more complex training algorithms must be used such as back propagation through time algorithms.

The FTDNNs and RTDNNs developed consist of two layers, an input layer with one nonlinear neuron with a sigmoid transfer function, and an output layer with one linear neuron. The number of delays for each input was determined by testing different configurations. The results of these experiments are included in Section 4. Both FTDNNs and RTDNNs are trained using density estimated from accelerometer data onboard spacecraft at LEO, as the targets, and hence learn to combine the inputs (the density estimated by the three models), and to calibrate the density based on the density targets. The FTDNNs and RTDNNs, take as inputs the natural logarithm of the density estimated by DTM-2013, JB2008, and NRLMSISE-00 in $\text{kg/m}^3 \times 10^{-12}$, and output a calibrated value of the natural logarithm of the density in $\text{kg/m}^3 \times 10^{-12}$. The natural logarithm of the density values was used since it helps stabilizing the variance of the density, as shown in Ref. [58]. The exponential of the output is taken to obtain the calibrated density in $\text{kg/m}^3 \times 10^{-12}$. This work also tests the idea of including the speed of the spacecraft as an additional input to both types of networks. The speed is

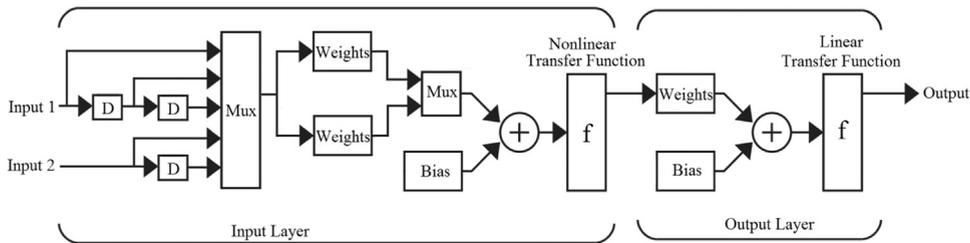


Fig. 3. Diagram for a FTDNN with two layers, two inputs (one with two delays and the second one with one), two nonlinear neurons in the input layer and one linear neuron in the output layer.

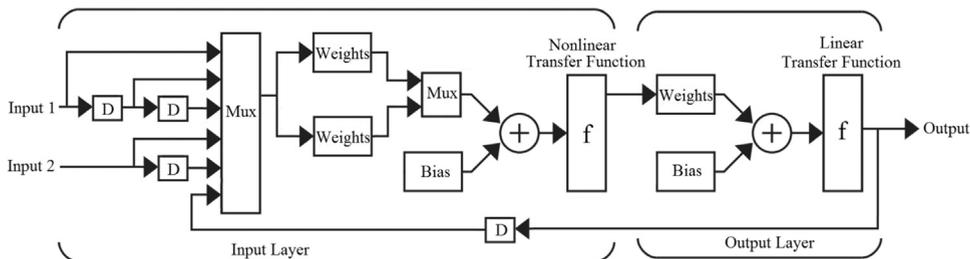


Fig. 4. Diagram for a RTDNN with two layers, two inputs (one with two delays and the second one with one), two nonlinear neurons in the input layer, one linear neuron in the output layer, and one delay in the feedback loop.

assumed to be available at the same instant as the densities estimated by the models. It is averaged over the previous orbital period with the intent to include some of the time evolution of velocity, which is related to the acceleration of the spacecraft. The delays allow the Neural Networks to access the density estimated by the three models along the past orbit of the spacecraft as well as its speed. The formulations for the FTDNN and the RTDNN, including the natural logarithm as part of the Neural Networks and the averaged speeds, are Eqs. (3) and (4) respectively.

$$\hat{\rho}(t) = g \left(\begin{array}{l} \rho_{DTM}(t), \dots, \rho_{DTM}(t - D_{DTM}t_s), \\ \rho_{JB}(t_s), \dots, \rho_{JB}(t - D_{JB}t_s), \\ \rho_{MSISE}(t), \dots, \rho_{MSISE}(t - D_{MSISE}t_s), \\ \bar{s}(t), \dots, \bar{s}(t - D_S t_s) \end{array} \right) \quad (3)$$

$$\hat{\rho}(t) = f \left(\begin{array}{l} \rho_{DTM}(t), \dots, \rho_{DTM}(t - D_{DTM}t_s), \\ \rho_{JB}(t_s), \dots, \rho_{JB}(t - D_{JB}t_s), \\ \rho_{MSISE}(t), \dots, \rho_{MSISE}(t - D_{MSISE}t_s), \\ \bar{s}(t), \dots, \bar{s}(t - D_S t_s), \\ \hat{\rho}(t - t_s) \end{array} \right) \quad (4)$$

where $\hat{\rho}$ is the calibrated density value (exponential of the Neural Network output); g and f are the overall nonlinear functions of the FTDNN and the RTDNN, respectively (including the natural logarithms at the input and the exponential at the output); ρ_{DTM} , ρ_{JB} and ρ_{MSISE} are the density estimated by DTM-2013, JB2008 and NRLMSISE-00, respectively; t is the time; t_s is the sampling period of the data; \bar{s} is the speed averaged over the previous orbital of the spacecraft; and D_{DTM} , D_{JB} , D_{MSISE} and D_S are the number of delays for ρ_{DTM} , ρ_{JB} , ρ_{MSISE} and \bar{s} , respectively.

The Levenberg–Marquardt algorithm ([59,60]), included in MATLAB's Neural Network Toolbox [61], was used to train the Neural Networks. This algorithm was chosen since it often has higher rates of convergence than the other algorithms provided in the Toolbox. The mean squared error (MSE), as explained in Eq. (5), was selected as the performance function because by minimizing it during training, both the variance and the bias of the error are also minimized.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{\rho}_i - \rho_i)^2 \quad (5)$$

where n is the number of samples and $\hat{\rho}_i$ and ρ_i are the i th calibrated value for the density and density target value, respectively.

4. Data used

Neural Networks require input–output data sets for training, validation and testing. The CHAMP [28] and GRACE [29] satellites have provided the research community with accurate density values with high temporal resolutions estimated by using their high precision accelerometers. Density data derived from accelerometers includes errors, as it is the case with any measurements of a physical quantity. However, these measurements have been widely used in the literature to validate empirical models as well as physical models. For example, JB2008, DTM2009, DTM2012 and DTM2013 have been validated using accelerometer-derived density data from the CHAMP and/or GRACE missions (see [19,27,30,31]). Similarly, efforts have also been made to assimilate CHAMP and GRACE densities into physical based models such as GITM in order to improve their accuracy (see IMPACT project at LANL [62,63]). Furthermore, many other papers in the literature (see a few examples [1,37,62–69]), assume those densities to be reference values for the density.

The GRACE mission consists of twin satellites (GRACE-A and GRACE-B), launched in 2002, with the objective of measuring Earth's gravitational field. The CHAMP satellite was launched in 2000 with the task of providing measurements of Earth's gravity and magnetic fields. The density along the orbit of the GRACE-A satellite and its speed during January 2003 were used for training and validation of the Neural Networks. Similarly, the density along the orbit of the CHAMP satellite and its velocity from day 2 to day 250 of 2007 (January 2nd to September 7th of 2007) were used for testing the Neural Networks (Test 1). This test data set was used to find the appropriate number of delays by following the method described in Section 5.1. This test data set comprising roughly seven months is comparable in time span to the test set used by Bowman et al. [35] for testing HASDM. A second test was performed using density along the orbit of the CHAMP satellite and its velocity covering years 2004 through 2008 (Test 2). This second test allows for evaluating the capacity of the Neural Network calibrators to operate under significantly different solar conditions. The altitude and inclination ranges for the GRACE-A and CHAMP satellites, and the average value for the F10.7 and Dst indices during the training/validation, Test 1 and 2 data sets are shown in Table 1.

The density data used for training, validation and testing of the Neural Networks was obtained from the Department of Aerospace Engineering Sciences data base at the University of Colorado [70]. The velocities used came from CHAMP Rapid Science Orbit data (CH-OG-3-RSO) and

Table 1

Altitude and inclination ranges of the CHAMP and GRACE-A satellites during the periods covered by the data used.

	GRACE-A	CHAMP	
Data Set	Training/validation	Test 1	Test 2
Time interval	01/01/2003–01/31/2003	01/02/2007–09/07/2007	01/02/2004–31/12/2008
Altitude (km)	473–526	341.7–380	416.9–320.8
Inclination (deg.)	89.0118–92.7957	87.22–90.89	87.22–90.94
Average F10.7 (sfu)	139.39	74.57	83.8
Average Dst (nT)	–11.97	–9.57	–11.1

GRACE-A GPS navigation data (GA-OG-1B-NAVSOL), which can be found at the website of the Information System and Data Center from the German Research Centre for Geosciences [71]. The GRACE-A and CHAMP satellites were launched into similar, nearly polar orbits; consequently, the time periods for the training/validation data set and the two test data sets were chosen so that there would be some differences in the altitudes of the orbits of the spacecraft (see Table 1). However, this table also shows that the inclination of the orbits of GRACE-A and CHAMP satellites are similar during the training/validation and the two test periods.

The information about the orbits of both satellites (altitude, latitude, longitude and time), during training/validation and test periods, was used to estimate the density, using the three models along their orbits. These are used as inputs to the Neural Networks. The training/validation data set was sampled randomly, 70% of its data was used for training, and the remaining 30% for validation. All the data was linearly interpolated between the

available data points at the times defined by a sampling rate of 60 s, to ensure that there was a constant difference in time between consecutive samples in all the data.

4.1. Solar and geomagnetic activity

Figs. 5 and 6 show the daily averaged values for the F10.7 and Dst indices (solar and geomagnetic activity indices) during the training/validation and Test 1 periods, respectively, while Fig. 7 contains the F10.7 and Dst indices averaged every 27 days during Test 2 period.

As it can be seen from Figs. 5 and 6, and the average value for the Dst index shown in Table 1, during the training/validation and Test 1 time intervals, there was mostly low geomagnetic activity (Dst above -20 nT) with some sporadic periods of medium geomagnetic activity (Dst between -20 and -50 nT). In contrast, the solar activity, as represented by the F10.7 index shown in Figs. 5 and 6 and its average values included in Table 1, is fairly different between these time intervals. The average value

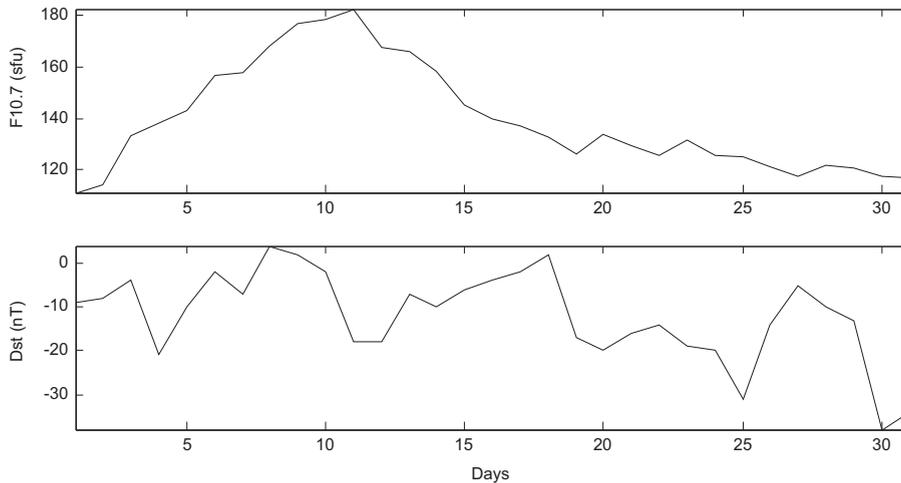


Fig. 5. Daily averaged F10.7 and Dst indices during the period covered by the training/validation data set.

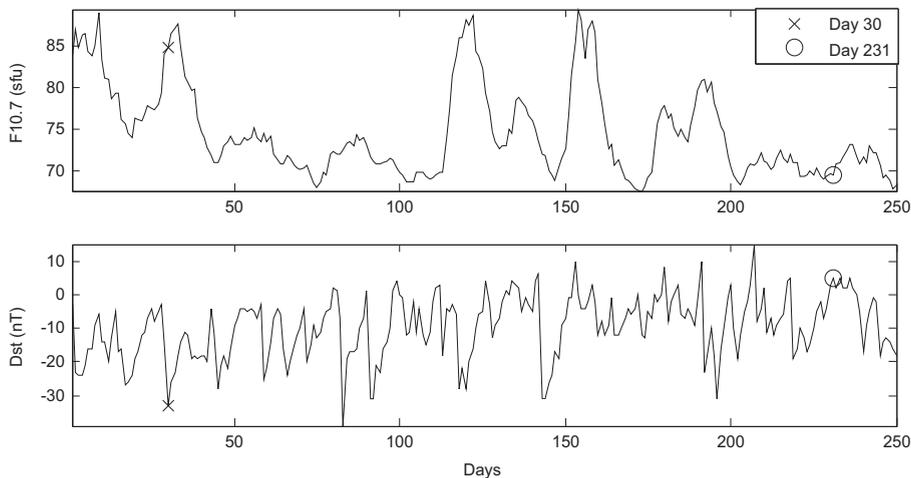


Fig. 6. Daily averaged F10.7 and Dst indices during the period covered by the Test 1 data set.

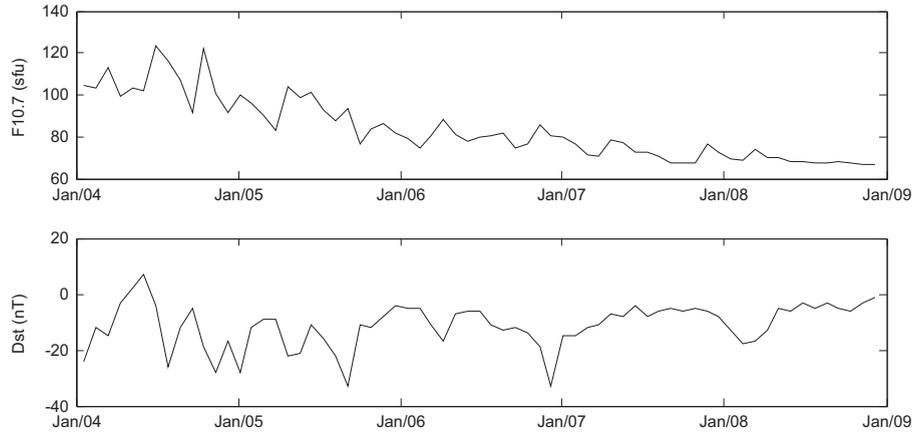


Fig. 7. F10.7 and Dst averaged every 27 days indices during the period covered by the Test 2 data set.

for F10.7 during training/validation is almost twice as large as it was for the test period. As shown in Figs. 6 and 7 the F10.7 index is much larger during years 2004 and 2005 than during Test 1 period.

To further observe the performance of the Neural Networks during periods of low and high solar and geomagnetic activity, two days within the test period were selected. The first of the two days chosen is day 30 (January the 30th) of 2007, which as can be seen in Fig. 6, corresponds to a region near a high peak of the F10.7 curve (F10.7=84.9 sfu) and a negative peak in the Dst curve (Dst=−33 nT, which according to [72] corresponds to a weak geomagnetic storm). The second day is day 231 (August the 19th) of 2007, which as can also be seen in Fig. 6, corresponds to region near a valley of the F10.7 curve (F10.7=69.4 sfu) and a high peak in the Dst curve (Dst=4 nT).

Fig. 7 shows a significant decrease in the F10.7 index during the five years, indicating a reduction in the solar radiation in the latter years. Likewise, it can be seen how the Dst index decreases in magnitude, indicating a similar reduction on geomagnetic activity. These changes are part of the 11 year solar cycle as Sun goes from solar maxima to minima.

5. Test results

To assess the performance of the Neural Networks, different metrics were used: the MSE (shown in Eq. (5)), and used as the performance parameter of the Neural Networks); the Pearson correlation coefficient of the targets to the model outputs (shown in Eq. (6), which measures the linear dependence between two values; the root mean squared error (RMSE) (shown in Eq. (7)), which is better for comparing the results from different models, since it has the same units of the quantity being modeled; and the mean of the ratios of the targets to the outputs ($\rho/\hat{\rho}$), which measures the bias in the estimated density.

$$R = \frac{1/(n-1) \sum_{i=1}^n (\rho_i - \bar{\rho})(\hat{\rho}_i - \bar{\hat{\rho}})}{s_{\hat{\rho}} s_{\rho}}, \quad (6)$$

where $s_{\hat{\rho}}$ and s_{ρ} are the standard deviations of the Neural Network outputs and targets, respectively.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\rho}_i - \rho_i)^2} \quad (7)$$

5.1. Method for determining the appropriate number of delays for the FTDNNs and RDTNNs

As explained in Section 3, the delays in both the FTDNNs and RDTNNs allow them to access the recent evolution of the inputs through time. Therefore the number of delays greatly influences the performance of the Neural Networks. The following method was used to find an appropriate number of delays in the FTDNNs for each input:

1. Set D_{JB} and D_{MSISE} to zero.
2. Vary D_{DTM} from zero to ten in increments of two while training, validating and testing the resulting FTDNN.
3. Pick the best value for D_{DTM} by selecting the FTDNN that had the lowest MSE (which was the performance function for the Networks) over the test data set.
4. Repeat same procedure for finding the best value for D_{JB} and D_{MSISE} in that order.
5. Repeat steps 2 through 4 but only for the two odd values adjacent to the best even value found before.
6. A FTDNN is trained, validated and tested using the best combination of delays, out of the six even numbers tried in steps 2 through 4 and two odd numbers tried in step 5.

To find an appropriate number of delays in the RTDNNs for each input the following slightly different method was followed:

1. Set D_{JB} , D_{MSISE} and D_S to zero.
2. Vary D_{DTM} from zero to ten in increments of two while training with the loop open, validating and testing the resulting RTDNN ten times.

3. Pick the best value for D_{DTM} by selecting the RTDNN that had the lowest MSE over the test data set.
4. Repeat same procedure for finding the best value for D_{JB} , D_{MSISE} and D_S in that order.
7. Repeat steps 2 through 4 but only for the two odd values adjacent to the best even value found before.
8. 100 RTDNNs are then trained with the loop open, validated and tested using the best combination of delays, out of the six even numbers tried in steps 2 through 4 and two odd numbers tried in step 5.
9. Repeats steps 1 through 8 with the loop closed and select the RTDNNs with the lowest MSE out of the 200 networks trained with the best combination of delays (100 trained with the loop open and 100 trained with the loop closed)

The characteristics of the learning algorithm (the weights and biases are initialized randomly) causes variation in the MSEs of the test for Neural Networks trained with the exact same architecture, the same number of delays. In the case of the FTDNNs, this variation was small (in the order of $0.001 \text{ (kg/m}^3 \times 10^{-12})^2$). However, in the case of the RTDNNs this variation was significant (in the order of $0.01 \text{ (kg/m}^3 \times 10^{-12})^2$). This is the reason for training the RTDNNs several times for the same exact architecture and number of delays, while the FTDNN was only trained once. These variations are likely caused by the learning algorithm converging to local minima. Therefore, this error can be fixed by changing the convergence condition for the learning algorithm. It should be noted that these two methods are brute force methods for finding an appropriate number of delays, and that there are other methods which should be computationally less taxing for finding the number of delays.

5.2. Test results for 01/02/2007 through 09/07/2007 (Test 1)

The two methods illustrated above were followed for four cases: 1) a FTDNN with three inputs (density estimated by the DTM-2013, JB2008 and NRLMSISE-00), 2) a RTDNN with the same three inputs, 3) a FTDNN with four inputs (density estimated by the DTM-2013, JB2008 and NRLMSISE-00; and the averaged of the spacecraft speed over the previous orbit), 4) and a RTDNN, with the same four inputs. As an additional bench mark the average of the densities estimated by the three models (average of models) was also calculated. The performance metrics of the resulting four Neural Networks, the three empirical atmospheric models and the average of models are compared in Table 2.

The resulting four Neural Networks produced values for the densities with less error than the three models and the average of models (see Table 2.), with reductions on the RMSE over DTM-2013 (the most accurate model) of 4.1%, 9.3%, 8.5% and 9.9% for the FTDNN and RTDNN with three inputs, and the RTDNN and the FTDNN with four inputs, respectively. The four Neural Networks also increased the Pearson correlation coefficient between the CHAMP densities and the estimated density, in comparison with the three models and the average of models (see Table 2). The average of the ratio of the targets to the outputs was closer

to a value of one for the four Neural Networks in comparison with the three models and the average of models. This means that the use of the Neural Networks allows for density estimation with less bias. In the case of the FTDNNs, the performance metrics also show that including the average velocity as an additional input, provides a significant error reduction of 4.6% in the RMSE. This reduction also occurs for the RTDNNs, but is much smaller (only 0.6% in the RMSE). Similarly, there is a significant error reduction between the FTDNN and RTDNN with three inputs (5.5% in the RMSE). This reduction also occurs for FTDNN and RTDNN with four inputs, but its value 1.6% in the RMSE, is much smaller. Interestingly, the best RTDNN with three inputs, trained with the loop open, had a lower MSE than its equivalent trained with the loop closed. In contrast the best RTDNN with four inputs trained with the loop open, had a higher MSE than its equivalent trained with the loop closed. The performance metrics of the average of models were better than those of NRLMSISE-00 and JB2008, but worse than DTM-2013 and all the Neural Networks.

As shown in Table 2 the RTDNN with four inputs provided the best results of all networks and models. Figs. 8–11 show the plots for the daily average values of the four performance metrics used over the whole test period for the RTDNN with four inputs.

Fig. 8 shows that the daily averaged values for the Pearson correlation coefficient for the three models and the RTDNN with four inputs shows a very similar behavior. However, significant differences can be seen in the plots for the daily averaged MSE and RMSE for the three models and the RTDNN with four inputs (see Figs. 9 and 10). More specifically, it is evident that the density estimated by DTM-2013 (red line) is the most accurate out of the three models, followed by the density estimated by JB2008 (green line). It can also be seen that the density estimated by the RTDNN with four inputs (black line) is the most accurate of all. The black line follows the red line closely, but it remains below it for most of the time. Fig. 11 shows that the ratio of the targets to the outputs of the RTDNN with four inputs remains closer to a value of one for most days than this same ratio for the three models. This indicates that the density estimated by the RTDNN has less bias than the densities estimated by the three models.

5.3. Test results for days 30 and 231 of 2007

As explained in Section 4.1 two days, included in the test period, were selected to observe the performance of the models and the Neural Networks. Day 30 of 2007 which presented high solar and geomagnetic activity (high in comparison with the other test days) and day 231 of 2007 which presented low solar and geomagnetic activity. The performance parameters for the three models, the average of the models and the four Neural Networks tested on these two days are shown in Table 3.

Table 3 shows that for both days, the four Neural Networks produced density estimations with error reductions and less bias in comparison with the three models and the average of models. For day 30, which included more solar activity (in relation to the other days in the test

Table 2

Performance metrics for the three models, the average of models and the Neural Networks during the whole test period (bold for best results, italics for worst).

Model	<i>R</i>	$MSE \left(\frac{kg}{m^3} \times 10^{-12} \right)^2$	$RMSE \left(\frac{kg}{m^3} \times 10^{-12} \right)$	$Mean(\rho/\hat{\rho})$	$D_{DTM}, D_{JB}, D_{MSE}, D_S$
NRLMSIS-00	0.9078	0.3283	0.5730	0.8235	
JB2008	0.9142	0.1794	0.4236	0.9368	
DTM-2013	0.9273	0.1372	0.3705	0.9524	
Average of models	0.9338	0.1622	0.4027	0.8933	
FTDNN3 Inputs	0.9332	0.1263	0.3554	0.9698	1,7,9,NA
RTDNN3 Inputs ^a	0.9358	0.1128	0.3359	0.9617	2,4,2,NA
FTDNN4 Inputs	0.9364	0.1150	0.3391	0.9651	0,0,0,5
RTDNN4 Inputs ^b	0.9357	0.1112	0.3338	0.9971	6,3,5,0

^a Trained with the loop open.

^b Trained with the loop closed.

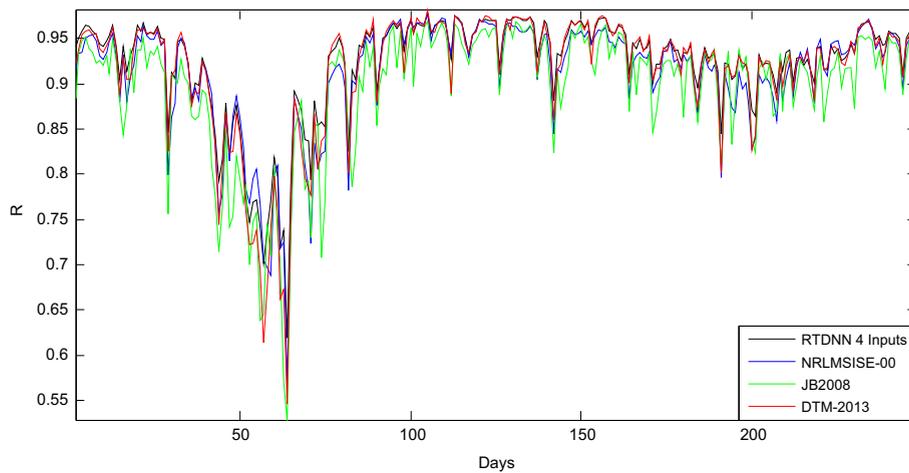


Fig. 8. Daily value for the Pearson correlation coefficient between the CHAMP density and the density estimated by DTM-2013, JB2008, NRLMSIS-00 and the RTDNN using four inputs, over the test period.

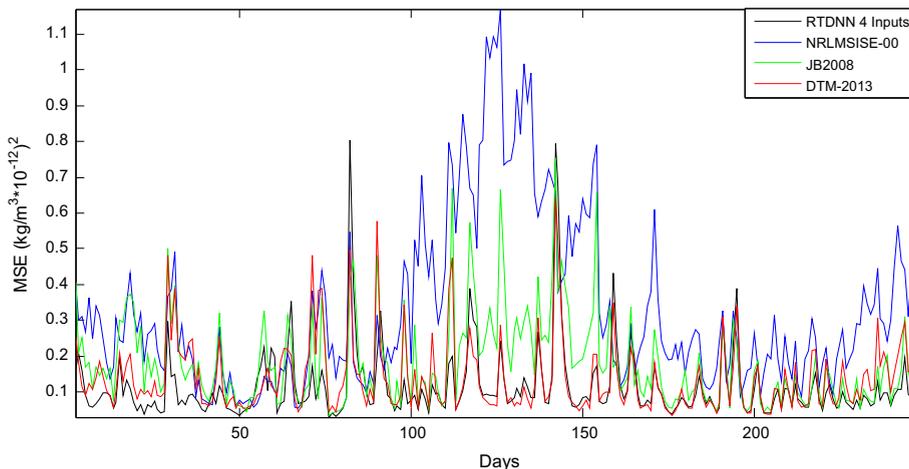


Fig. 9. Daily value for the MSE of the density estimated by DTM-2013, JB2008, NRLMSIS-00 and the RTDNN using four inputs, over the test period. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

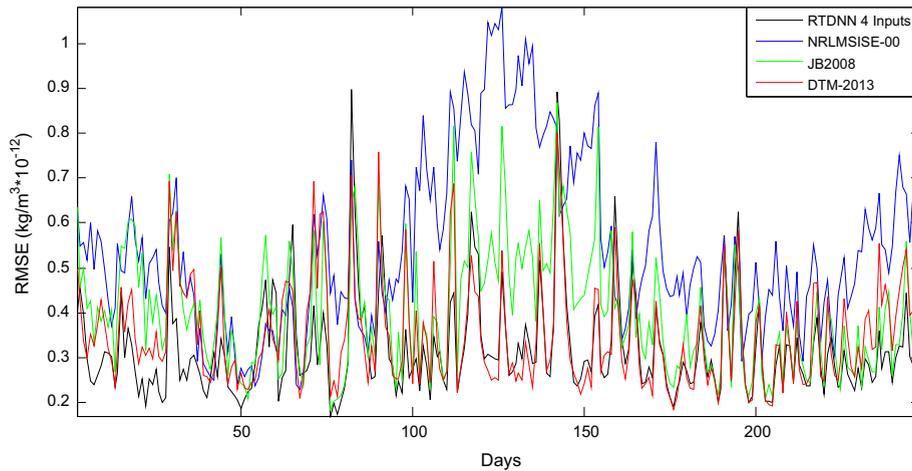


Fig. 10. Daily value for the RMSE of the density estimated by DTM-2013, JB2008, NRLMSISE-00 and the RTDNN using four inputs, over the test period. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

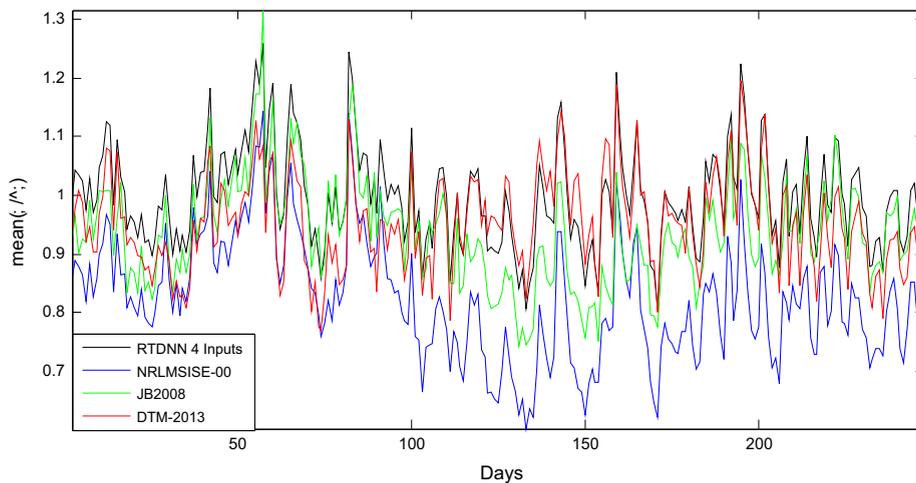


Fig. 11. Daily value for the average ratio of the targets to the outputs ($\rho/\hat{\rho}$) estimated by DTM-2013, JB2008, NRLMSISE-00 and the RTDNN using four inputs, over the test period.

data set) and a weak geomagnetic storm, the reduction in the RMSE provided by the FTDNN and RTDNN with three inputs, and the RTDNN and the FTDNN with four inputs, in comparison with DTM2013 (the most accurate model for this day) were of 11.2%, 15.8%, 17.9% and 21.1%, respectively. These error reductions are larger than those achieved for the whole test period. Similarly for day 231, which included the lower solar and geomagnetic activity, the error reductions in comparison with JB2008 (the most accurate model for this day) were of 1.4%, 4.5%, 3.4% and 11.4% for the FTDNN and RTDNN with three inputs, and the RTDNN and the FTDNN with four inputs, respectively. In this case the reductions in the error for all the Neural Networks, except the RTDNN with four inputs (which produces a slightly larger reduction in the error), were smaller than those achieved for the whole test period. The Pearson correlation coefficient between the CHAMP densities and the density estimated by JB2008 was the largest of the three models and the four Neural Networks for day

30, while for day 231 it was that of DTM-2013. The performance metrics for the average of models were better than two of the models but worse than the remaining one for both days.

Again, as indicated in Table 3, the RTDNN with four inputs provided the best results of all networks and models during these two days. The density as estimated by the three models, and the RTDNN with four inputs, is compared with the CHAMP accelerometer density, during days 30 and 231 of 2007, in Figs. 12 and 14, respectively. Zoom-in images of these density plots during the first five hours of days 30 and 231 of 2007 are included in Figs. 13 and 15, respectively.

A comparison between Figs. 12 and 13 and Figs. 14 and 15 shows that the CHAMP density had a more complex behavior (with more small oscillations) during day 30 than during day 231. This occurred because the higher solar and geomagnetic activity (including the weak geomagnetic storm), present during day 30 causes the density to behave

Table 3

Performance metrics for the three models, the average of models and the Neural Networks during days 30 and 231 of 2007 (bold for best results, italics for worst).

Day	Model	R	MSE ($\frac{\text{kg}}{\text{m}^3} \times 10^{-12}$) ²	RMSE ($\frac{\text{kg}}{\text{m}^3} \times 10^{-12}$)	Mean($\rho/\hat{\rho}$)
30 (F10.7 = 84.9 sfu, Dst = -33 nT)	NRLMSIS-00	0.8556	0.3683	0.6069	0.8685
	JB2008	0.9091	0.2866	0.5353	0.8878
	DTM-2013	0.8960	0.2415	0.4914	0.9038
	Average of models	0.9035	0.2669	0.5167	0.8837
	FTDNN 3 Inputs	0.8920	0.1906	0.4366	0.9839
	RTDNN 3 Inputs ^a	0.9015	0.1714	0.4140	0.9410
	FTDNN 4 Inputs	0.9031	0.1628	0.4035	0.9735
231 (F10.7 = 69.4 sfu, Dst = 4 nT)	RTDNN 4 Inputs ^b	0.9043	0.1503	0.3877	0.9778
	NRLMSIS-00	0.9565	0.2784	0.5276	0.7614
	JB2008	0.9496	0.0583	0.2414	0.9072
	DTM-2013	0.9629	0.0695	0.2636	0.9265
	Average of models	0.9605	0.1045	0.3232	0.8549
	FTDNN 3 Inputs	0.9495	0.0566	0.2380	0.9202
	RTDNN 3 Inputs ^a	0.9562	0.0532	0.2306	0.9329
FTDNN 4 Inputs	0.9582	0.0544	0.2332	0.9143	
RTDNN 4 Inputs ^b	0.9578	0.0458	0.2139	0.9683	

^a Trained with the loop open.

^b Trained with the loop closed.

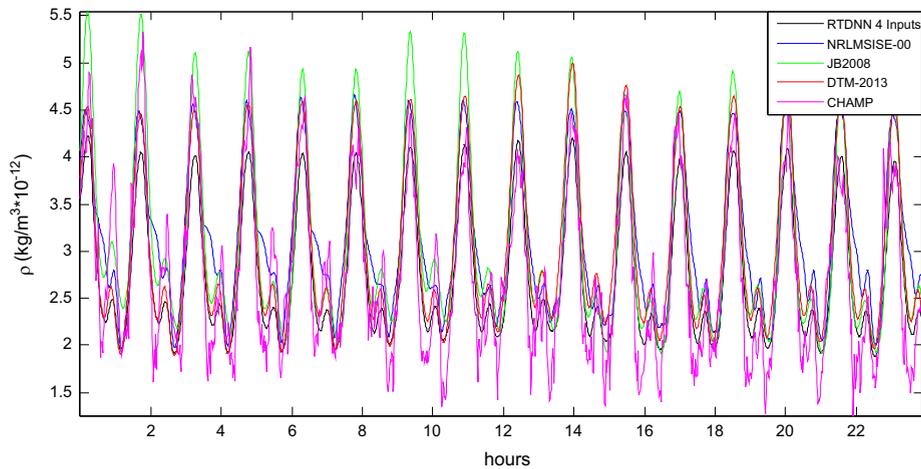


Fig. 12. Estimated density by the three models, and the RTDNN with four inputs and CHAMP accelerometer derived density during day 30 of 2007.

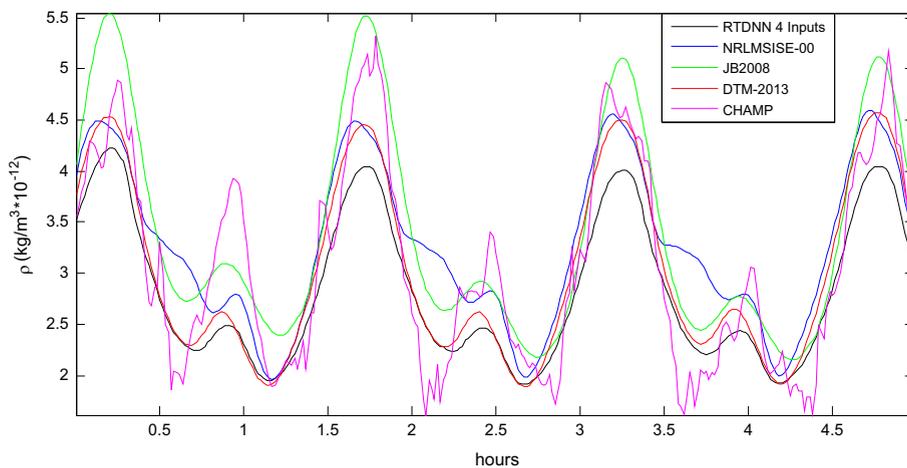


Fig. 13. Zoom-in of density plot during the first 5 h of day 30 of 2007.

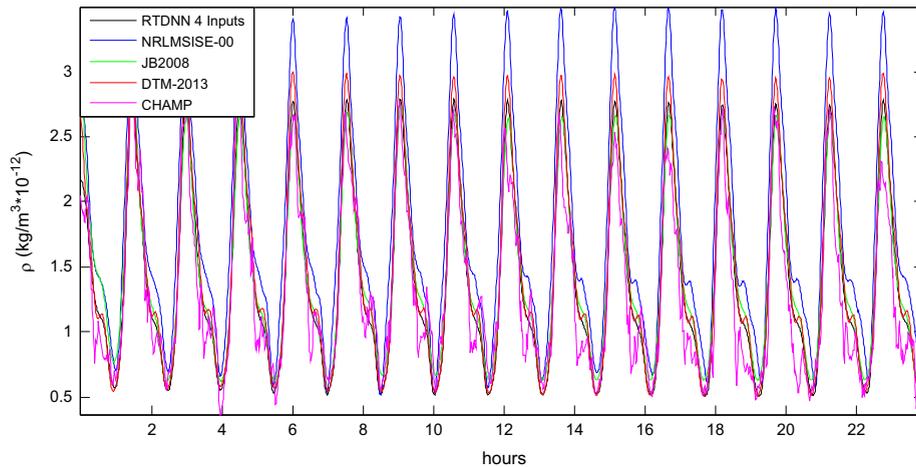


Fig. 14. Estimated density by the three models, and the RTDNN with four inputs and CHAMP accelerometer derived density during day 231 of 2007.

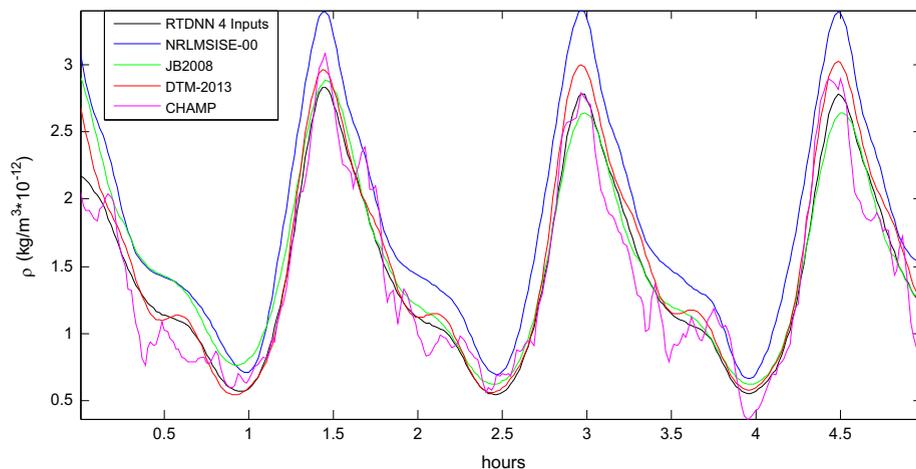


Fig. 15. Zoom-in of density plot during the first 5 h of day 231 of 2007.

in a more chaotic manner. This resulted in the larger estimation errors, for all models and Neural Networks, for day 30 in comparison with day 231 that can be observed in Table 3.

5.4. Test results for 2004 through 2008 (Test 2)

The Neural Networks and the models were tested using the Test 2 data set which covers the years from 2004 to 2008. This test allows to evaluate the performance of the Neural Networks over a wide temporal resolution covering different solar conditions. The performance parameters for the three models, the average of the models and the four Neural Networks tested on the five years are shown in Tables 4 and 5.

The results from Test 2 in Tables 4 and 5 show that the Neural Networks performance decreases for the years 2004 and 2005 which are the furthest away in time from the Test 1 time interval which was used to find the number of delays. During 2004, only the RTDNN with three inputs performs better than the models and during 2005, none of

the Neural Networks perform better than the three models. The reason for this is that at those years the solar cycle is at a different stage than during the Test 1 time interval (January to September 2007). Hence, the solar activity is different (both F10.7 and Dst are higher during 2004–2005 than during 2006–2008). On the preceding and following years to Test 1 interval (2006 and 2008) and during 2007 itself, the Neural Networks performed better than the models with the exception of the FTDNN with three inputs during 2006. Over all, during the whole Test 2 interval the RTDNNs with three and four inputs present the best results, with the former been the best overall (5.3% improvement in RMSE over DTM-2013), while the two FTDNNs fall behind DTM-2013 (which is the best of the three model during the Test 2 interval). This further confirms that the RTDNNs provide better results than the FTDNNs for this application. Interestingly, the average of the models provide better results overall than the three models (3.3% improvement in RMSE over DTM-2013) and all but one of the Neural Networks (the RTDNN with three

Table 4

Performance metrics for the three models, the average of models and the Neural Networks during years 2004, 2005 and 2006 (bold for best results, italics for worst).

Year	Model	R	MSE $\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$	RMSE $\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$	Mean($\rho/\hat{\rho}$)	
2004 (F10.7=106.0 sfu, Dst= – 12.0 nT)	NRLMSIS-00	<i>0.9038</i>	0.3348	0.5787	0.9696	
	JB2008	0.9313	0.2571	0.5070	1.0584	
	DTM-2013	0.9257	0.2981	0.5459	0.9642	
	Average of models	0.9463	0.1895	0.4353	0.9844	
	FTDNN 3 Inputs	0.939	<i>0.3753</i>	<i>0.6126</i>	1.0735	
	RTDNN 3 Inputs*	0.9429	0.2439	0.4938	1.0411	
	FTDNN 4 Inputs	0.9439	0.3179	0.5638	<i>1.0747</i>	
	RTDNN 4 Inputs**	0.9437	0.2722	0.5217	1.0694	
	2005 (F10.7=91.1 sfu, Dst= – 15.6 nT)	NRLMSIS-00	<i>0.8840</i>	0.4207	0.6486	0.9676
		JB2008	0.9030	0.3775	0.6144	1.0694
DTM-2013		0.9160	0.3103	0.5571	1.0289	
Average of models		0.9277	0.2699	0.5195	1.0109	
FTDNN 3 Inputs		0.9236	<i>0.5526</i>	<i>0.7434</i>	1.1171	
RTDNN 3 Inputs*		0.928	0.3726	0.6104	1.0797	
FTDNN 4 Inputs		0.9273	0.4776	0.6911	1.1117	
RTDNN 4 Inputs**		0.9259	0.4368	0.6609	<i>1.1174</i>	
2006 (F10.7=80.1 sfu, Dst= – 12.2 nT)		NRLMSIS-00	<i>0.9051</i>	<i>0.2788</i>	<i>0.528</i>	<i>0.8747</i>
		JB2008	0.9198	0.1843	0.4293	0.9758
	DTM-2013	0.9211	0.1667	0.4083	0.9793	
	Average of models	0.9341	0.1562	0.3952	0.9330	
	FTDNN 3 Inputs	0.9283	0.1784	0.4224	1.0143	
	RTDNN 3 Inputs*	0.9330	0.1422	0.3770	1.0004	
	FTDNN 4 Inputs	0.9337	0.1534	0.3917	1.0102	
	RTDNN 4 Inputs**	0.9329	0.1526	0.3906	1.0351	

* Trained with the loop open.

** Trained with the loop closed.

Table 5

Performance metrics for the three models, the average of models and the Neural Networks during years 2007, 2008 and all five years (2004 to 2008) (bold for best results, italics for worst).

Year	Model	R	MSE $\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$	RMSE $\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$	Mean($\rho/\hat{\rho}$)	
2007 (F10.7=73.1 sfu, Dst= – 8.3 nT)	NRLMSIS-00	<i>0.8988</i>	<i>0.4200</i>	<i>0.6480</i>	<i>0.8027</i>	
	JB2008	0.9076	0.1949	0.4415	0.9304	
	DTM-2013	0.9177	0.1417	0.3764	0.9625	
	Average of models	0.9276	0.1825	0.4272	0.8858	
	FTDNN 3 Inputs	0.9264	0.1266	0.3558	0.9665	
	RTDNN 3 Inputs*	0.9294	0.1172	0.3423	0.9579	
	FTDNN 4 Inputs	0.9303	0.1160	0.3406	0.9621	
	RTDNN 4 Inputs**	0.929	0.1132	0.3365	0.9924	
	2008 (F10.7=68.7 sfu, Dst= – 7.5 nT)	NRLMSIS-00	<i>0.8883</i>	<i>0.9980</i>	<i>0.8883</i>	<i>0.7403</i>
		JB2008	0.9004	0.2605	0.5104	0.9757
DTM-2013		0.9134	0.3255	0.5705	0.8841	
Average of models		0.9196	0.3646	0.6038	0.8461	
FTDNN 3 Inputs		0.9149	0.2586	0.5085	0.9344	
RTDNN 3 Inputs*		0.9204	0.2388	0.4886	0.9134	
FTDNN 4 Inputs		0.9194	0.2353	0.4851	0.9324	
RTDNN 4 Inputs**		0.9180	0.2232	0.4724	0.9527	
2004–2008 (F10.7=83.8 sfu, Dst= – 11.1 nT)		NRLMSIS-00	<i>0.8715</i>	<i>0.4896</i>	<i>0.6997</i>	<i>0.8711</i>
		JB2008	0.9134	0.2547	0.5047	1.0019
	DTM-2013	0.9198	0.2482	0.4982	0.9639	
	Average of models	0.9278	0.2322	0.4819	0.9321	
	FTDNN 3 Inputs	0.9231	0.2980	0.5459	1.0212	
	RTDNN 3 Inputs*	0.9287	0.2227	0.4719	0.9985	
	FTDNN 4 Inputs	0.9272	0.2598	0.5097	1.0182	
	RTDNN 4 Inputs**	0.9287	0.2394	0.4893	1.0334	

* Trained with the loop open.

** Trained with the loop closed.

inputs), and during years 2004 and 2005 it provides the best results; however, during the remaining years its performance drops below DTM-2013.

Since the RTDNN with three inputs presented the best performance overall, a new RTDNN with three inputs was created using the method described in Section 5.1 but with

Table 6

Performance metrics for DTM-2013, the average of models, and the RTDNNs with three inputs with number of delays determined using the Test 1 and data sets during the five years.

Year	Metric	RTDNN 3 inputs delays found using Test 1 data ($D_{DTM}=2$, $D_{JB}=4$, $D_{MSISE}=2$)	RTDNN 3 inputs delays found using Test 2 data ($D_{DTM}=4$, $D_{JB}=2$, $D_{MSISE}=10$)
2004 (F10.7=106.0 sfu, Dst=−12.0 nT)	R	0.9429	0.9410
	MSE	0.2439	0.2184
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
	RMSE	0.4938	0.4674
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
2005 (F10.7=91.1 sfu, Dst=−15.6 nT)	Mean($\rho/\hat{\rho}$)	1.0411	1.0462
	R	0.9280	0.9260
	MSE	0.3726	0.3294
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
	RMSE	0.6104	0.5739
2006 (F10.7=80.1 sfu, Dst=−12.2 nT)	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
	Mean($\rho/\hat{\rho}$)	1.0797	1.0817
	R	0.9330	0.9279
	MSE	0.1422	0.1497
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
2007 (F10.7=73.1 sfu, Dst=−8.3 nT)	RMSE	0.3770	0.3870
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
	Mean($\rho/\hat{\rho}$)	1.0004	1.0159
	R	0.9294	0.9243
	MSE	0.1172	0.1250
2008 (F10.7=68.7 sfu, Dst=−7.5 nT)	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
	RMSE	0.3423	0.3535
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
	Mean($\rho/\hat{\rho}$)	0.9579	0.9759
	R	0.9204	0.9174
2004–2008 (F10.7=83.8 sfu, Dst=−11.1 nT)	MSE	0.2388	0.2461
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
	RMSE	0.4886	0.4961
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
	Mean($\rho/\hat{\rho}$)	0.9134	0.9233
2004–2008 (F10.7=83.8 sfu, Dst=−11.1 nT)	R	0.9287	0.9272
	MSE	0.2227	0.2135
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)^2$		
	RMSE	0.4719	0.4621
	$\left(\frac{\text{kg}}{\text{m}^3} \times 10^{-12}\right)$		
	Mean($\rho/\hat{\rho}$)	0.9985	1.0087

the Test 2 data (the full five years) set instead of the Test 1 data set. In order to reduce the computation time of testing on the Test 1 data set instead of the Test 1 dataset, only five and ten iterations were made in steps 1 and 8, of the method in Section 5.1. The performance parameters for the three models, the average of the models and the four Neural Networks tested on these two days are shown in Table 6.

The results in Table 6 indicate that by determining the number of delays using the data from the whole five years (Test 2 data set) for the RTDNN with three inputs, its performance can be increased over the whole five years in comparison with the other RTDNN with three inputs (7.3% vs 5.3% improvement in RMSE over DTM-2013 respectively). The latter was the best performing over the five years, and which had its number of delays determined using the Test 1 data set covering from January to

September 2007. This improvement occurs over the years 2004 and 2005 at the cost of a slight decrease in performance during years 2005–2008. This indicates that a Neural Network can be created to calibrate the models over a large time span covering a wide range of solar conditions; however, individual networks created for specific solar conditions can achieve better performance during those specific conditions, at the cost of poorer performance under different solar conditions.

The results presented show that Neural Networks provide a valuable method for combining the densities estimated by the three models and calibrating them based on accelerometer data. This method captures the benefits of the individual models, and further reduces the error and bias in the estimated density by assimilating accelerometer data.

6. Conclusions

All the Neural Networks presented (FTDNNs and RTDNNs with three or four inputs) provide a significant error reduction in the estimation of the atmospheric density along the orbit of CHAMP, during the Test 1 period (from January to September 2007, which was used to choose the number of delays), in comparison with DTM-2013, JB2008 and NRLMSISE-00. Similarly, the Neural Networks reduce the bias in the estimated density for the same period in comparison with the three models. Furthermore, the FTDNNs and RTDNNs also increase the value of the correlation factor between the estimated density and the CHAMP density, again in comparison with the three empirical models, during the Test 1 period.

The tests done for the two days of interest, show that the FTDNNs and RTDNNs can also provide better results than the three empirical models during periods with both higher and lower solar and geomagnetic activity (in comparison with the other days in the test data set). Moreover, the results of the FTDNNs and RTDNNs were considerably better than those of the three models during day 30 of 2007, when a weak geomagnetic storm occurred.

During the Test 2 period (the five years) the RTDNNs provide better performance than the three models in terms of the correlation factor, the error and the bias, while the FTDNNs failed to do so. This suggests that the RTDNNs provide more robustness against changes in the solar activity, and are therefore better suited for calibrating the density. Similarly, for Test 1 the RTDNNs provide better results than the FTDNNs, further confirming that the autoregressive nature does allow for a better estimation of the density. Nonetheless, there was a large variation on the MSE for the tests done with the RTDNNs for a given number of delays, due to the local minima in the training of the Neural Networks. This issue was mitigated by training several RTDNNs for a given number of delays, and picking the one with the lowest MSE, which is a very computationally costly solution (as described in Section 5.1). Therefore, further study for determining the appropriate learning algorithm for the RTDNNs and its convergence conditions must be done. As an alternative to Neural Networks, kernel methods could be used, given that they do not suffer from the local minima issue.

A second RTDNN with three inputs was created by repeating the method described in Section 5.1 (which is used to define the number of delays) using the Test 2 data (the full five years) set instead of the Test 1 data set. This RTDNN provided better results over the five years, but it had poorer performance than the original RTDNN with 3 inputs during the Test 1 period (January to September 2007). This result highlights the importance of the number of delays for the inputs in the performance of the networks. This suggests that a group of networks with the number of delays chosen for the different solar conditions over a time interval, might perform better than just one networks with delays set for the whole interval covering those same solar conditions. Such a group of networks could be combined through a committee of machines to create a calibrator valid for different solar conditions. Another possibility would be to create an

adaptive calibrator which is retrained as the solar conditions evolve.

The use of the speed averaged over the previous orbits of the spacecraft as an additional input provided an error reduction during the Test 1 period, but not during the Test 2 period. Consequently, more tests must be performed to fully comprehend its effect on the performance of the networks.

Surprisingly, the average of models provided better results over all than the three models over the five years. This improvement was very significant during the years 2004 and 2005 during which the solar activity was higher. The good performance of the average of models suggests that it may be used as an additional input to the Neural Networks to improve their performance. The good performance of the average of models also confirms that combining different atmospheric models can lead to better density estimations.

This work verifies the feasibility of using Time Delay Neural Networks, for calibrating the density estimated by atmospheric models, while assimilating accelerometer derived densities, and using accurate navigation data, if available. This method can be further developed to create an atmospheric density model, specifically designed for orbit propagation and determination. Furthermore, this method can be run onboard since the Neural Networks have a low computational cost and the three models, and a method for obtaining the spacecraft's speed from the navigation systems can both be run onboard.

Future work will be the development of a group of calibrators defined for different solar conditions and the creation of a committee machine calibrator to combine the individual networks. Moreover, the possibility of implementing adaptive neural networks to create a dynamic calibration method is also of interest, since it may provide the methodology presented here, with the ability to exploit real time data to calibrate the models, in a similar fashion as HASDM does.

Acknowledgments

The authors wish to acknowledge the United States Office of Naval Research (ONR) for sponsoring this investigation under the Young Investigator Program (Award no. N00014-13-1-0536).

References

- [1] D. Pérez, B. Wohlberg, T. Lovell, M. Shoemaker, R. Bevilacqua, Orbit-centered atmospheric density prediction using artificial neural networks, *Acta Astronaut.* 98C (2014) 9–23, <http://dx.doi.org/10.1016/j.actaastro.2014.01.007>.
- [2] S.Ray, The layers of Earth's atmosphere, Jet Propulsion Laboratory [government website], URL: (http://airs.jpl.nasa.gov/maps/satellite_feed/atmosphere_layers/), 2013 (accessed 19.09.13).
- [3] C.L. Leonard, W.M. Hollister, E.V. Bergmann, Orbital formation-keeping with differential drag, *AIAA J. Guid. Control Dyn.* 12 (1) (1989) 108–113, <http://dx.doi.org/10.2514/3.20374>.
- [4] T.D. Maclay, C. Tuttle, Satellite station-keeping of the ORBCOMM constellation via active control of atmospheric drag: operations, constraints, and performance, *Adv. Astronaut. Sci.* 120 (Part I) (2005) 763–773. (AAS 05-152).
- [5] B.S.Kumar, A.Ng, A bang-bang control approach to maneuver spacecraft in a formation with differential drag, in: *Proceedings of the AIAA*

- Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, August 2008, <http://dx.doi.org/10.2514/6.2008-6469>.
- [6] D. Pérez, R. Bevilacqua, Differential drag spacecraft rendezvous using an adaptive Lyapunov control strategy, *Acta Astronaut.* 83 (2013) 196–207, <http://dx.doi.org/10.1016/j.actaastro.2012.09.005>.
- [7] D. Pérez, R. Bevilacqua, Spacecraft maneuvering via atmospheric differential drag using an adaptive Lyapunov controller, paper AAS 13-440, in: Proceedings of the 23rd AAS/AIAA Spaceflight Mechanics Meeting, Kauai, Hawaii, February 2013.
- [8] D. Pérez, R. Bevilacqua, Lyapunov-based adaptive feedback for spacecraft planar relative maneuvering via differential drag, *AIAA J. Guid. Control Navig.* (2014) 1678–1684, <http://dx.doi.org/10.2514/1.G000191>.
- [9] D. Pérez (Ph.D. dissertation), Adaptive Lyapunov Control and Artificial Neural Networks for Spacecraft Relative Maneuvering Using Atmospheric Differential Drag, Mechanical, Aerospace, and Nuclear Engineering Dept, Rensselaer Polytechnic Institute, Troy, NY, 2013.
- [10] D.A. Vallado, *Fundamentals of Astrodynamics and Applications* 3rd ed. Springer, New York, 2001 (Chap 8).
- [11] Y. Deng, A.J. Ridley, Possible reasons for underestimating Joule heating in global models: E field variability, spatial resolution, and vertical velocity, *J. Geophys. Res.: Space Phys.* 112 (A9) (2007) 1–20.
- [12] D.A. Vallado, D. Finkleman, A critical assessment of satellite drag and atmospheric density modeling, *Acta Astronaut.* 0094-5765(2013) 141–165, <http://dx.doi.org/10.1016/j.actaastro.2013.10.005>. (Available online 25 October).
- [13] L.G. Jacchia, A variable atmospheric-density model from satellite accelerations, *J. Geophys. Res.* 65 (1960) 2775–2782, <http://dx.doi.org/10.1029/JZ065i009p02775>.
- [14] L.G. Jacchia, *New Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles* (Smithsonian Astrophysical Observatory Special Report, No. 313), 1970.
- [15] L.G. Jacchia, *Revised Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles* (Smithsonian Astrophysical Observatory Special Report, No. 332), 1971.
- [16] L.G. Jacchia, *Thermospheric Temperature, Density, and Composition: New Models* (Smithson. Astrophys. Obs. Spec. Rept. No. 375), 1977.
- [17] C.E. Roberts Jr., An analytic model for upper atmosphere densities based upon Jacchia's 1970 models, *Celest. Mech.* 4 (3–4) (1971) 368–377.
- [18] B.R. Bowman, W.K. Kent Tobiska, F.A. Marcos, C. Valladares, The JB2006 empirical thermospheric density model, *J. Atmos. Sol. Terr. Phys.* 70 (5) (2008) 774–793.
- [19] B.R. Bowman, W.K. Tobiska, F.A. Marcos, C.Y. Huang, C.S. Lin, and W. J. Burke, A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices, in: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, AIAA, Honolulu, HI, August 2008b, pp. 1–19.
- [20] A.E. Hedin, J.E. Salah, J.V. Evans, C.A. Reber, G.P. Newton, N.W. Spencer, D.C. Kayser, D. Alcaydè, P. Bauer, J.P. McClure, A global thermospheric model based on mass spectrometer and incoherent scatter data MSIS, 1. N2 density and temperature, *J. Geophys. Res.* 82 (16) (1977) 2139–2147.
- [21] A.E. Hedin, MSIS-86 thermospheric model, *J. Geophys. Res.* 92 (A5) (1987) 4649–4662, <http://dx.doi.org/10.1029/JA082i016p04649>.
- [22] A.E. Hedin, Extension of the MSIS thermospheric model into the middle and lower atmosphere, *J. Geophys. Res.* 96 (1991) 1159, <http://dx.doi.org/10.1029/JA092iA05p04649>.
- [23] J.M. Picone, A.E. Hedin, D.P. Drob, A.C. Aikin, NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues, *J. Geophys. Res.* 107 (A12) (2002) 1468, <http://dx.doi.org/10.1029/2002JA009430>.
- [24] F. Barlier, C. Berger, J.L. Falin, G. Kockarts, G. Thuillier, A thermospheric model based on satellite drag data 34 (1978) 9–24, [http://dx.doi.org/10.1016/S1364-6826\(03\)00137-8](http://dx.doi.org/10.1016/S1364-6826(03)00137-8) *Ann. Geophys.* 34 (1978) 9–24, [http://dx.doi.org/10.1016/S1364-6826\(03\)00137-8](http://dx.doi.org/10.1016/S1364-6826(03)00137-8).
- [25] C. Berger, R. Biancale, M. Ill, F. Barlier, Improvement of the empirical thermospheric model DTM: DTM94 a comparative review of various temporal variations and prospects in space geodesy applications, *J. Geod.* 72 (3) (1998) 161–178, <http://dx.doi.org/10.1007/s001900050158>.
- [26] S. Bruinsma, G. Thuillier, F. Barlier, The DTM-2000 empirical thermosphere model with new data assimilation and constraints at lower boundary: accuracy and properties, *J. Atmos. Sol.-Terr. Phys.* 65 (9) (2003) 1053–1070, [http://dx.doi.org/10.1016/S1364-6826\(03\)00137-8](http://dx.doi.org/10.1016/S1364-6826(03)00137-8).
- [27] S.L. Bruinsma, N. Sánchez-Ortiz, E. Olmedo, N. Guijarro, Evaluation of the DTM-2009 thermosphere model for benchmarking purposes, *J. Space Weather Space Clim.* (2012), <http://dx.doi.org/10.1051/swsc/2012005>.
- [28] C. Reigber, H. Lühr, P. Schwintzer, CHAMP mission status, *Adv. Space Res.* 30 (2) (2002) 129–134, [http://dx.doi.org/10.1016/S0273-1177\(02\)00276-4](http://dx.doi.org/10.1016/S0273-1177(02)00276-4).
- [29] B.D. Tapley, S. Bettadpur, M. Watkins, C. Reigber, The gravity recovery and climate experiment: mission overview and early results, *Geophys. Res. Lett.* 30 (L09607) (2004) 1–4, <http://dx.doi.org/10.1029/2004GL019920>.
- [30] S.L. Bruinsma, DTM2012 evaluation report, advanced thermosphere modelling for orbit prediction [government website], URL: (http://www.atmop.eu/public-documents/evaluation_report_dtm2012.pdf), 2014 (accessed 20.10.14).
- [31] S.L. Bruinsma, DTM2013 evaluation report, advanced thermosphere modelling for orbit prediction [government website], URL: (http://www.atmop.eu/public-documents/evaluation_report_dtm2013.pdf), 2014 (accessed 20.10.14).
- [32] F.A. Marcos, M.J. Kendra, J.M. Griffin, J.N. Bass, J.F. Liu, D.R. Larson, Precision low earth orbit determination using atmospheric density calibration, *Adv. Astronaut. Sci.* 97 (1) (1998) 515–527. (AAS).
- [33] M.F. Storz, B.R. Bowman, M.J.I. Branson, S.J. Casali, W.K. Tobiska, High accuracy satellite drag model (HASDM), *Adv. Space Res.* 36 (12) (2005) 2497–2505.
- [34] M.F. Storz, B.R. Bowman & J.I. Branson, High accuracy satellite drag model (HASDM), in: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Monterey, California, 5–8 August 2002.
- [35] B.R. Bowman, M.F. Storz, High accuracy satellite drag model (HASDM) review, *Adv. Astronaut. Sci.* 116 (2003) 1943–1952.
- [36] S.J. Casali, & W.N. Barker (August). Dynamic calibration atmosphere (DCA) for the high accuracy satellite drag model (HASDM), In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2002, pp. 5–8.
- [37] E.D. Fattig (M.S. thesis), Comparison of Precision Orbit Derived Density Estimates for CHAMP and GRACE Satellites, Aerospace Engineering and the Graduate Faculty of the University of Kansas, Lawrence, KS, 2011.
- [38] C.A. McLaughlin, A. Hiatt, D.M. Krishna, T. Lechtenberg, E. Fattig, P.M. Metha, Precision orbit derived atmospheric density: development and performance, in: Proceeding of the Advanced Maui Optical and Space Surveillance Technologies Conference, 2012.
- [39] C.A. McLaughlin, T. Lechtenberg, E. Fattig, D.M. Krishna, Estimating density using precision satellite orbits from multiple satellites, *J. Astronaut. Sci.* 59 (1–2) (2014) 84–100.
- [40] COSPAR Working Group IV, COSPAR International Reference Atmosphere, Akademie-Verlag, Berlin, 1972.
- [41] M.A. Shoemaker, B. Wohlberg, J. Koller, Atmospheric density reconstruction using satellite orbit tomography, in: Proceedings of the 23rd AAS/AIAA Spaceflight Mechanics Meeting, Lihue, HI, USA, February 2013, pp. 1235–1250.
- [42] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, M. Embrechts, Network-based intrusion detection using neural networks, *Intell. Eng. Syst. Artif. Neural Netw.* 12 (1) (2002) 579–584.
- [43] Y. Bengio, J. Buhmann, M.J. Embrechts, J.M. Zurada, Introduction to the special issue on neural networks for data mining and knowledge discovery, *IEEE Trans. Neural Netw.* 11 (3) (2000) 545–549.
- [44] M.J. Embrechts, S. Benedek, Hybrid identification of nuclear power plant transients with artificial neural networks, *IEEE Trans. Ind. Electron.* 51 (3) (2004) 686–693.
- [45] J. Xin, and M.J. Embrechts, Supervised learning with spiking neural networks, in: Proceedings of the IEEE International Joint Conference on Neural Networks, Vol. 3, IEEE, Washington, DC, July 2001, pp. 1772–1777.
- [46] Y. Yu, J. Koller, S.G. Zaharia, L* neural networks from different magnetic field models and their applicability, *Space Weather* 10 (2) (2012) 1–13.
- [47] Y. Yu, J. Koller, V.K. Jordanova, S.G. Zaharia, R.W. Friedel, S.K. Morley, H.E. Spence, Application and testing of the L* neural network with the self-consistent magnetic field model of RAM-SCB, *J. Geophys. Res.: Space Phys.* 119 (3) (2014) 1683–1692.
- [48] A. Choury, S. Bruinsma, P. Schaeffer, Neural networks to predict exosphere temperature corrections, *Space Weather* 11 (10) (2013) 592–602.
- [49] K. Saulnier, D. Pérez, G. Tilton, D. Gallardo, C. Shake, R. Huang, and R. Bevilacqua, Operational capabilities of a six degrees of freedom spacecraft simulator, in: Proceedings of the AIAA GNC Conference, Boston, MA, 2013, (AIAA 2013–5253).
- [50] J.M. Picone, A.E. Hedin, Drob, D.P., NRLMSISE-00 Model 2001, Community Coordinated Modeling Center (CCMC), Goddard Space Flight Center [government website], URL: (<http://ccmc.gsfc.nasa.gov/modelweb/atmos/nrlmsise00.html>), 2014 (accessed 15.07.14).

- [51] B.R. Bowman, W.K. Kent, Tobiska, F.A. Marcos, The Jacchia-Bowman 2008 empirical thermospheric density model, Space Environment Technologies [commercial website], URL: (<http://sol.spacenvironment.net/~jb2008/index.html>), 2014 (accessed 15.07.14).
- [52] M. Menvielle, Advanced thermosphere modelling for orbit prediction [government website], URL: (<http://www.atmop.eu/>), 2014 (accessed 15.07.14).
- [53] Solar Irradiance Platform, Space Environment Technologies [commercial website], URL: (<http://www.spacewx.com/solar2000.html>), 2014 (accessed 23.10.14).
- [54] W. Tobiaka, SOLAR2000 v2.30 and SOLA RFLARE v1.01: new capabilities for space system operations, AIAA-2007-495, in: Proceedings of the 45th AIAA Meeting on Aerospace Sciences, Reno, NV, January 10, 2007.
- [55] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.* 2 (4) (1989) 303–314.
- [56] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* 4 (2) (1991) 251–257.
- [57] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, A. Novel, Connectionist system for improved unconstrained handwriting recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (5) (2009).
- [58] H. Lütkepohl, F. Xu, The role of the log transformation in forecasting economic variables, *Empir. Econ.* 42 (3) (2012) 619–638, <http://dx.doi.org/10.1007/s00181-010-0440-1>.
- [59] K. Levenberg, A method for the solution of certain problems in least squares, *Q. Appl. Math.* 2 (1944) 164–168.
- [60] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* 11 (1963) 431–441, <http://dx.doi.org/10.1137/0111030>.
- [61] The MathWorks, Levenberg-Marquardt backpropagation, Documentation center [commercial website], URL: (<http://www.mathworks.com/help/nnet/ref/trainlm.html>), 2013 (accessed 4.09.13).
- [62] H.C. Godinez, Data assimilation with GITM, IMPACT project review [government website], URL: (http://www.impact.lanl.gov/docs/workshop_slides/6-HCGodinez_DataAssimGITM.pdf), 2014 (accessed 20.10.14).
- [63] H.C. Godinez, B. Nadiga, A.J. Ridley, J. Koller, E. Lawrence & D. Higdon, Atmospheric density specification with the global ionosphere-thermosphere model (GITM) using the Ensemble Kalman filter, In: Proceedings of the AGU Fall Meeting Abstracts, Vol. 1, 2012 December, p. 2145.
- [64] R. Linares, A. Klimenko, S. Brennan, H.C. Godinez Vazquez, D.M. Higdon, J. Koller, & B.E. Wohlberg, A framework for integrated modeling of perturbations in atmospheres for conjunction tracking (IMPACT) (No. LA-UR-14-23694). Los Alamos National Laboratory (LANL), 2014.
- [65] N.B. Stastny, F.R. Chavez, C. Lin, T.A. Lovell, R.A. Bettinger & J. Luck, Localized density/drag prediction for improved onboard orbit propagation. Air Force Research Lab Kirtland AFB NM Space Vehicles Directorate, 2009.
- [66] R. Liu, S.Y. Ma, H. Lühr, Predicting storm-time thermospheric mass density variations at CHAMP and GRACE altitudes, *Ann. Geophys.* 29 (3) (2011) 443–453. (Copernicus GmbH).
- [67] S.L. Bruinsma, J.M. Forbes, Storm-time equatorial density enhancements observed by CHAMP and GRACE, *J. Spacecr. Rockets* 44 (6) (2007) 1154–1159.
- [68] Eelco Doornbos, *Thermospheric Density and Wind Determination From Satellite Dynamics*, Springer, Berlin, Heidelberg, 2012.
- [69] S.L. Bruinsma, & M. Fedrizzi, Simultaneous observations of TADs in GOCE, CHAMP and GRACE density data compared with CTIPE. In AGU Fall Meeting Abstracts, Vol. 1, 2012, December, p. 05).
- [70] J.M. Forbes, R.S. Nerem, E.K. Sutton, X. Zhang, S. Bruinsma, Thermosphere Studies Using Accelerometer Measurements from the CHAMP and GRACE Satellites, Department of Aerospace Engineering Sciences University of Colorado, 2014 (<http://sisko.colorado.edu/sutton/>) [online database], URL: [retrieved July 15th].
- [71] Information System and Data Center for geoscientific data, GFZ [online database], URL: (<http://isdg.gfz-potsdam.de/>), 2014 (accessed 15.07.14).
- [72] C.A. Loewe, G.W. Pröls, Classification and mean behavior of magnetic storms, *J. Geophys. Res.: Space Phys.* 102 (A7) (1997) 14209–14213.



laer Polytechnic Institute, Troy, New York.



Madrid, Spain, during 2003. Dr. Bevilacqua's research interests focus on Guidance, Navigation, and Control of multiple spacecraft systems and multiple robot systems.

Dr. David Pérez is a Postdoctoral Research Associate in the ADvanced Autonomous Multiple Spacecraft (ADAMUS) Laboratory at the Mechanical and Aerospace Engineering Department at the University of Florida (UFL). David's main research interests are Guidance, Navigation and Control of autonomous Spacecraft and Robotics. He received his B.Sc. (2008) in Mechanical Engineering from St. Cloud State University, St. Cloud, Minnesota and his M.Eng (2009) and Ph.D. (2013) in Mechanical Engineering and Aeronautical Engineering both from the Rensselaer Polytechnic Institute, Troy, New York.

Dr. Riccardo Bevilacqua is an Associate Professor of the Mechanical and Aerospace Engineering Department at the University of Florida (UFL). He holds a M. Sc. in Aerospace Engineering (2002), and a Ph.D. in Applied Mathematics (2007), both earned at the University of Rome, "Sapienza", Italy. He was a US National Research Council Post-Doctoral Fellow from 2007 to 2010 and an Assistant Professor at Rensselaer Polytechnic Institute from 2010 to 2014, before joining UFL. He also worked as project engineer in Mission Analysis at Grupo Mecanica del Vuelo, in