

Space Weather®



RESEARCH ARTICLE

10.1029/2023SW003652

Giacomo Acciarini and Edward Brown
contributed equally to this work.

Key Points:

- Machine learning (ML) models can significantly outperform existing physics-based thermospheric neutral density models on exactly the same inputs
- ML models can improve over NRLMSISE-00 and JB-08 empirical density models by 61% and 39% respectively in the mean absolute percentage error
- The software allows the creation of ML-ready data for training and benchmarking new models, supporting solar irradiance and geomagnetic data

Correspondence to:

G. Acciarini and E. Brown,
giacomo.acciarini@gmail.com;
edward.j.e.brown@gmail.com

Citation:

Acciarini, G., Brown, E., Berger, T.,
Guhathakurta, M., Parr, J., Bridges,
C., & Baydin, A. G. (2024). Improving
thermospheric density predictions in
low-Earth orbit with machine learning.
Space Weather, 22, e2023SW003652.
<https://doi.org/10.1029/2023SW003652>

Received 22 JUL 2023

Accepted 21 JAN 2024

Author Contributions:

Conceptualization: Giacomo Acciarini,
Edward Brown, Madhulika Guhathakurta,
Atılım Güneş Baydin

Funding acquisition: Madhulika
Guhathakurta, James Parr

Investigation: Giacomo Acciarini,
Edward Brown

Methodology: Giacomo Acciarini,
Edward Brown

Project Administration: James Parr,
Christopher Bridges




Resources: Madhulika Guhathakurta

Software: Giacomo Acciarini, Edward
Brown, Atılım Güneş Baydin

© 2024. The Authors.

This is an open access article under
the terms of the [Creative Commons
Attribution License](#), which permits use,
distribution and reproduction in any
medium, provided the original work is
properly cited.

Improving Thermospheric Density Predictions in Low-Earth Orbit With Machine Learning

Giacomo Acciarini^{1,2} , Edward Brown^{1,3,4} , Tom Berger⁵ , Madhulika Guhathakurta⁶, James Parr¹,
Christopher Bridges^{1,2}, and Atılım Güneş Baydin^{1,7,8}

¹Trillium Technologies Inc., London, UK, ²Surrey Space Centre, University of Surrey, Guildford, UK, ³Computer Science Department, University of Cambridge, Cambridge, UK, ⁴Space Weather and Atmosphere Team, British Antarctic Survey, Cambridge, UK, ⁵Space Weather Center, CU Boulder, Boulder, CO, USA, ⁶NASA Headquarters, Washington DC, DC, USA, ⁷Department of Engineering Science, University of Oxford, Oxford, UK, ⁸Department of Computer Science, University of Oxford, Oxford, UK

Abstract Thermospheric density is one of the main sources of uncertainty in the estimation of satellites' position and velocity in low-Earth orbit. This has negative consequences in several space domains, including space traffic management, collision avoidance, re-entry predictions, orbital lifetime analysis, and space object cataloging. In this paper, we investigate the prediction accuracy of empirical density models (e.g., NRLMSISE-00 and JB-08) against black-box machine learning (ML) models trained on precise orbit determination-derived thermospheric density data (from CHAMP, GOCE, GRACE, SWARM-A/B satellites). We show that by using the same inputs, the ML models we designed are capable of consistently improving the predictions with respect to state-of-the-art empirical models by reducing the mean absolute percentage error (MAPE) in the thermospheric density estimation from the range of 40%–60% to approximately 20%. As a result of this work, we introduce Karman: an open-source Python software package developed during this study. Karman provides functionalities to ingest and preprocess thermospheric density, solar irradiance, and geomagnetic input data for ML readiness. Additionally, it facilitates developing and training ML models on the aforementioned data and benchmarking their performance at different altitudes, geographic locations, times, and solar activity conditions. Through this contribution, we offer the scientific community a comprehensive tool for comparing and enhancing thermospheric density models using ML techniques.

Plain Language Summary Accurately modeling the density of the thermosphere is pivotal for spacecraft operations such as collision avoidance, re-entry prediction, and orbital lifetime analysis. In this study, our aim is twofold. First, we want to study and compare the performance of data-driven machine learning (ML) models in predicting thermospheric density data against standard empirical models used in the field, which are used as baseline. By training ML models using precise orbit determination-derived satellite data, we show that they can achieve significant performance improvement compared to empirical models, with a reduction of 61% in the mean absolute percentage error. Second, we also provide the community with a shared software framework that supports the ingestion of solar irradiance, geomagnetic, and thermospheric density data, as well as a training and benchmarking framework to develop ML models. This framework allows researchers and operators to both train their ML models and to compare them at different periods of the solar cycle, geomagnetic storm conditions, geographical locations, and times.

1. Introduction

On 4 February 2022, SpaceX lost 40 of its 49 launched Starlink satellites, due to miscalculations in the predicted space weather indices, and the following unexpected increase in atmospheric drag. By further investigation of the event, it was found that the geomagnetic storm was only a minor one. Nevertheless, two consecutive coronal mass ejections hit the Earth on the 3–4 February, causing an average increase in thermospheric density of about 20%, with local peaks up to 60%, when compared with 2 February. This event, together with the fact that we are approaching the solar maximum of the 25th solar cycle when solar activity is expected to grow, show that an accurate prediction, modeling, and understanding of the Sun's influence on the thermospheric density is needed (Dang et al., 2022). In fact, atmospheric drag is the main perturbing force and the biggest source of uncertainty for space objects below 1,000 km (Berger et al., 2020). For this reason, its precise estimation is paramount to

Supervision: Tom Berger, Christopher Bridges, Atılım Güneş Baydin
Writing – original draft: Giacomo Acciarini, Edward Brown

many space applications, including but not limited to collision avoidance, re-entry, orbital lifetime estimation, orbit propagation, and tracking (Vallado, 2001).

The most common models used to forecast thermospheric density from solar irradiance and geomagnetic drivers are either physics-based or empirical models. The most widespread empirical models used in the operational community are NRLMSISE-00 (Picone et al., 2002), JB-08 (Bowman et al., 2008), and HASDM (Storz et al., 2005). The first two are publicly accessible, while the last model, maintained by Air Force Space, is not available open-source. Among the most used and publicly available ones, JB-08 is considered among the most accurate ones (Marcos et al., 2006). It is well known that these models present biases and do not always perform accurately, especially in periods of high and low solar activity (He et al., 2018; Licata et al., 2021). On the other hand, physics-based models also exist, which attempt to accurately model the underlying physics phenomena behind the Sun-Earth interaction in the atmosphere. These models, such as TIE-GCM (Richmond et al., 1992), were demonstrated to be more accurate than empirical models during geomagnetic storms (S. Bruinsma et al., 2021). However, results across all geomagnetic storm conditions, altitude ranges and irradiance conditions, do not always confirm their superiority with respect to empirical models. Moreover, their complexity and heavy computational burden prevent their use for operational applications (e.g., collision avoidance).

In addition, to improve the thermospheric density estimation problem, several researchers have discussed and investigated the inclusion of more geomagnetic and/or solar irradiance data in both physics-based and empirical models, for better modeling the Sun-Earth interaction, and provide better models. While some have proposed new solar irradiance and geomagnetic proxies (Bowman et al., 2008), others have advocated the necessity of including EUV irradiance inputs (instead of proxies) in the prediction models (Vourlidis & Bruinsma, 2018).

While semi-empirical approaches combine empirical mathematical equations with fitting data, machine learning (ML) models use very few prior assumptions about data to form relationships between variables. The hypothesis is that ML models, given enough data, can learn a representation of the thermospheric system without any physics-based assumptions and might be able to outperform semi-empirical existing models. Starting from a preliminary study performed by the authors (Bonasera et al., 2021), and with the aim to test this hypothesis, this paper develops ML models that can accurately predict thermospheric density variations. Furthermore, it also proposes and releases an open-source framework to train and analyze the performances of ML-based thermospheric density models and benchmark them against empirical models. The main findings and contributions of this study are:

1. ML models show a remarkable 60% reduction in mean absolute percentage error (MAPE) compared to existing semi-empirical models, showcasing their superior performance despite the absence of physics-based assumptions;
2. the release of Karman-ML Thermospheric Neutral Density Model (KML) (Acciarini, Brown, & Baydin, 2023). This model uses the same inputs as the JB08 model but with a significant performance improvement; combining multiple inputs into ML-ready data (or similar to this)
3. the release of a pipeline for combining multiple inputs from different sources, producing ML-ready data encompassing thermospheric density inputs, solar irradiance, and geomagnetic inputs. Through our work, researchers and operators now have a framework to handle solar irradiance and geomagnetic data (both raw measurements and proxies/indices), as well as empirical model predictions and precise orbit determination (POD)-derived thermospheric density from SWARM-A, SWARM-B, GRACE, GOCE, CHAMP satellites. This framework empowers the research community to develop new models by leveraging the same software infrastructure (as we showed with our proposed KML). A schematic illustration of our proposed workflow is shown in Figure 1. Furthermore, we have designed this framework in a modular fashion to encourage expansion and improvements by the open-source community, facilitating advancements through the inclusion of additional data sources;
4. the release of a benchmarking framework, where users can test their own models, as well as the provided ones, against POD-derived and empirical models of thermospheric density. This can be done at different geomagnetic storm conditions and altitudes, facilitating in-depth analysis of the results.

The concept of benchmarking physics-based, empirical, and ML models is not new and has been employed in several works (S. Bruinsma, 2015; S. L. Bruinsma et al., 2012; Jackson et al., 2020; Oliveira et al., 2020; Tobiska et al., 2021). However, none of these studies has focused on building an ML representation of the thermospheric density in low-Earth orbit from only accelerometer-derived density data, without the use of empirical

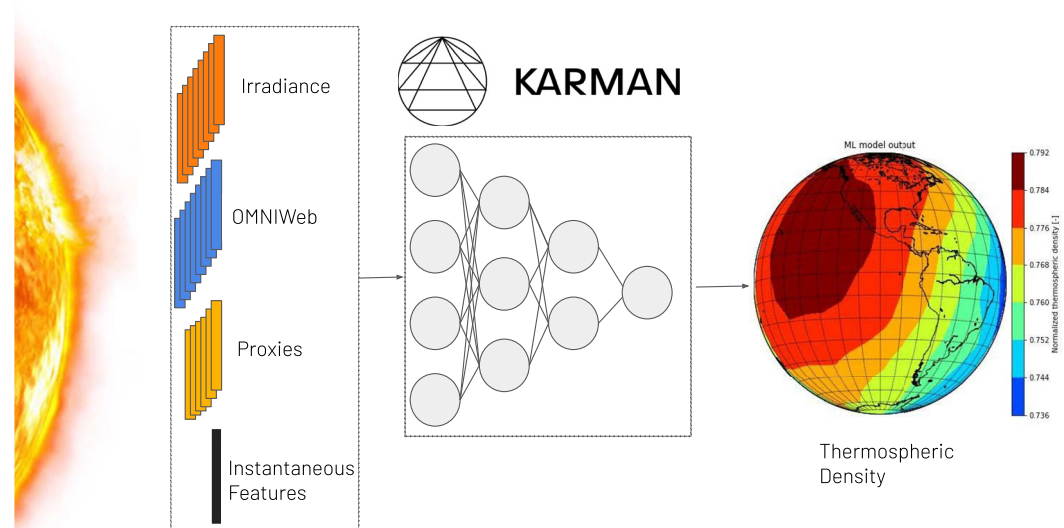


Figure 1. Schematic illustration of Karman machine learning workflow. Supported inputs and the expected output are shown.

models as training data, thereby providing an alternative model that can outperform state-of-the-art techniques. Furthermore, none has provided an open-source platform or shared data pipeline that facilitates the comparison of different models on the same data sets. Additionally, it remains hard to investigate the potential performance improvements that can be achieved by incorporating additional data into these models, due to the lack of a shared and open-source framework. As a result, progress in this field has been fragmented and lacks a unified approach. Our framework aims to address this gap by providing a centralized platform, enabling the research community to align their efforts and work toward common objectives.

In terms of ML models applied to the thermospheric density prediction task, several works have been published in recent years. Some have used artificial neural networks for predicting long-term thermospheric density trends (Weng et al., 2020), while others have produced ML models that use principal component analysis or reduced order modeling, together with density data from HASDM and JB-08 empirical models to produce an ML surrogate model for the thermospheric density (Licata & Mehta, 2022; Licata, Mehta, Tobiska, & Huzurbazar, 2022; Licata, Mehta, Weimer, et al., 2022; Mehta et al., 2018; Mehta & Linares, 2017; Turner et al., 2020). These works were not the only ones: there have been several attempts in both predicting short and long-term variations using ML (Chen et al., 2014; W. Li et al., 2023; Pérez & Bevilacqua, 2015; Pérez et al., 2014; Y. Wang & Bai, 2023; P. Wang et al., 2022, 2023; Zhang et al., 2021). All these works, however, have never included direct solar irradiance measurements (instead of proxies) and geomagnetic activity measurements besides the Ap index, Dst, and SYM-H: their only considered inputs were either proxies and/or indices, as also happens in empirical density models. Instead, it is believed that more information on the Sun and geomagnetic activity, from actual measurements of satellites and ground stations, could enhance our understanding and prediction of the Sun's influence on the thermosphere (Emmert, 2015; Vourlidis & Bruinsma, 2018).

One of the goals of this work is to propose a software framework that can help bridge the discussed gap and enable the community to devise ML-based thermospheric density models using geomagnetic and EUV irradiance measurements. This paper is organized as follows: in Section 2, we introduce the data pipeline and the supported data sources, which constitute the backbone on which ML models can be trained and evaluated, as well as the ML methodology and benchmarking framework. While in Section 3, we discuss the results of our analysis both globally, and at different altitude and geomagnetic storm conditions. Finally, in Section 4, we discuss the conclusions of our work.

2. Methods

2.1. Data Pipeline

One of the outcomes of this work is a pipeline to produce a unified data set that supports data from several different sources. These include high frequency (every 30 or 10 s) POD-derived thermospheric density data from five space missions (CHAMP, GOCE, GRACE, SWARM-A, SWARM-B), processed by the TU Delft precise orbit determination group (Doornbos, 2012; Siemes et al., 2016, 2023), as well as other data related to satellites' state and time (i.e., latitude, longitude, date). For more details on the thermospheric density data used in this work we refer the interested reader to the other publication of the authors (Bonasera et al., 2021). Empirical density data from NRLMSISE-00 and JB-08, as well as the needed inputs to run these models (e.g., sun declination and ascension, sidereal time, geomagnetic and irradiance indices). Geomagnetic inputs both in the form of measurements (from NASA's OmniWeb data service (King & Papitashvili, 2005; NASA, 2023), and indices [from Celestrack (Kelso & Sean, 2010)] and Space Environment Technologies, SET (Kent, 2023)). Some examples of geomagnetic measurements are interplanetary magnetic field magnitude, X, Y, Z-components of the magnetic field vector in GSM and GSE coordinates, SYM/D, SYM/H, ASY/D, ASY/H indices, also including some derived values such as the X, Y, Z-components of the bow shock nose, and many others (a full description of all the supported inputs can be found in the OMNI website). Solar irradiance inputs both in the form of measurements (from the Laboratory of Atmospheric and Space Physics, LASP, of CU Boulder (LASP, 2023)) and indices (from both the LASP datacenter as well as Space Environment Technologies [SET]). In particular, two different solar irradiance data products are supported. First, the daily (with 1-day cadence) and flare (with 1-min cadence) "stan bands" from FISM2 empirical model: this is an empirical model of the solar spectral irradiance from 0.01 to 190 nm at 0.1 nm spectral bins. It is derived from SORCE XPS L4, SDO-EVE, and SORCE SOLSTICE data, but it allows a continuous time and spectral coverage of solar irradiance data, which is very appealing for thermospheric density modeling purposes (Chamberlin et al., 2020). The "stan bands" are a set of 23 bands derived with a wavelength binning scheme covering the UV from 0.01 to 121 nm, that are used as inputs into Earth ionospheric and thermospheric models (Solomon & Qian, 2005). Second, a set of solar irradiance proxies that are commonly used in empirical thermospheric density models: these include F10.7, M10.7, S10.7, and Y10.7 (Tobiska et al., 2008).

Upon inputting the data into the aforementioned pipeline, a unified data set is generated, allowing for targeted queries at specific time points. This data set encompasses all or specific components of the aforementioned inputs at the desired time, accompanied by the corresponding POD-derived thermospheric density, serving as ground truth data. Additionally, users also have the option to access the temporal history of FISM2 and OMNIWeb data, at any given date (if these data are provided). In this way, temporal relationships between space weather and thermospheric density variations can be studied. In our experiments, in order to train our ML model, we used a data set of approximately 38 million POD-derived thermospheric density data points, where we only trained using the same inputs as the empirical models, in order to have a fair comparison. In Figure 2, we show an example of how four solar indices (F10.7, S10.7, M10.7, Y10.7) and two geomagnetic inputs (dDst/dT, Ap) varied during the 2015 St. Patrick Storm, and the resulting thermospheric density behavior, derived from SWARM-B satellite (shown in the fourth row in the figure). As can be seen, the severe geomagnetic storm (i.e., G4 class) that happened between 17 and 19 March 2015 has caused a steep increase in the thermospheric density. As it can be observed, while both geomagnetic indices reached a maximum in that period, however, not all the solar irradiance proxies have reflected this behavior (since they capture different regions of the EUV spectrum).

2.2. Machine Learning Methodology

2.2.1. Machine Learning Architecture

The study utilizes two distinct types of neural network models: NRLMSISE/JB08 inputs and JB08 + OMNI inputs. The NRLMSISE/JB08 input model comprises four layers, each containing 500 nodes. It takes empirical model inputs as its input and maps them to the outputted density. In particular, we use the following inputs for the NRLMSISE-00 ML model: day of the year, year, seconds in a day, right ascension and declination of the Sun, sidereal time, altitude, longitude, latitude, local solar time, Ap daily index, Ap 3-hourly index, F10.7 observed and F10.7 averaged. While for the ML model using the same inputs as JB-08, the same inputs as NRLMSISE-00 are used but instead of Ap indices, dDst/dT values (i.e., variations of the Dst index as a function of temperature variations) are used, and the solar proxies are F10.7, S10.7, M10.7 and Y10.7 (both observed and averaged). In

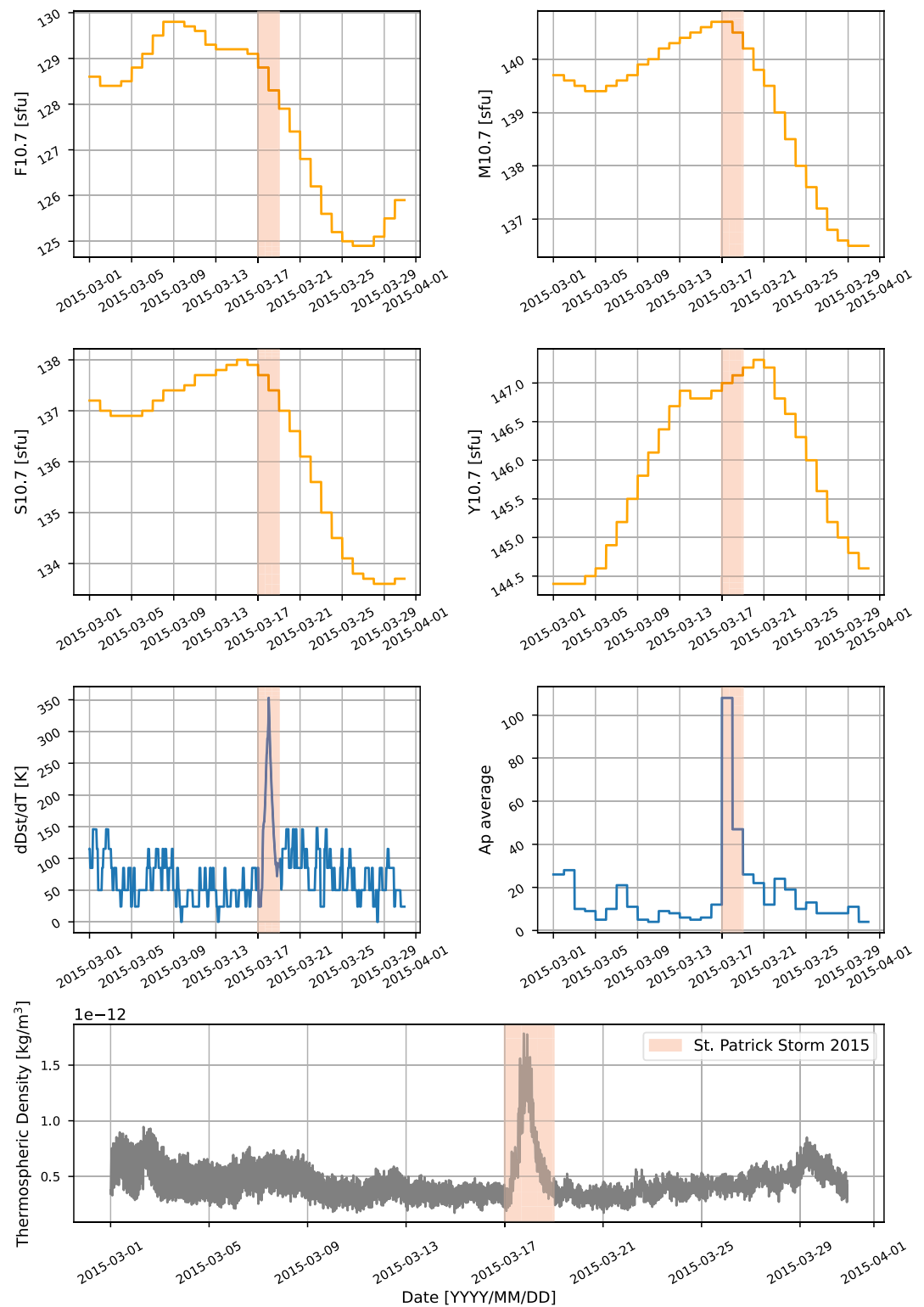


Figure 2. Solar irradiance proxies (i.e., F10.7, S10.7, M10.7, Y10.7), geomagnetic inputs (i.e., dD_{st}/dT , Ap average), and thermospheric density variations during the 2015 St. Patrick's Day geomagnetic storm.

the case in which OMNIWeb data is used, the following features are considered: the three Cartesian coordinates of the magnetic field in GSE and GSM frame, the three Cartesian coordinates of the plasma flow velocity in GSE coordinates, the three Cartesian coordinates of the bow shock nose in GSE coordinates, the AE, AL, AU, SYM/D, SYM/H, ASY/D, ASY/H, PCN indexes.

The JB08 + OMNI input model processes the NRLMSISE/JB08 inputs through a sequence of four layers with 500, 500, 500, and 175 nodes, respectively, resulting in 175 outputted features. In a similar manner, the OMNI inputs are also processed through four layers with 500, 500, 500, and 175 nodes, generating 175 additional outputted features. These 350 features (175 + 175) are concatenated and passed through a final processing four-layer network consisting of 500 nodes. This network maps all the inputs to the outputted density.

The NRLMSISE/JB08 inputs model architecture was chosen due to its ability to maintain a rather simple network, without sacrificing too much accuracy. As we will discuss in Section 3, we have compared this architecture with simpler ones (e.g., without hidden layers) and established the most performing one that was still maintaining a low level of complexity. This study is more concerned with demonstrating that a large volume of thermospheric density data allows data-driven methods to build accurate models of thermospheric density. The accurate study of the best architecture to perform thermospheric density prediction was not the main goal of this study, and it is therefore postponed for future studies. Second, concerning the JB08 + OMNI model, the reason why the above-mentioned architecture was chosen is modularity. This study looks at the effect of including extra input features, OMNI data, into the model to see how much model performance can be improved. The method of creating a separate network to process and encode different data sources was therefore employed where the processed features are simply concatenated and further processed in a final network. This allows the modularity to add new data sources without significantly increasing the model complexity.

Concerning the ML model training, we use mean squared error as a loss function, normalizing the output (i.e., target density) by taking the natural logarithm of its value multiplied by 10^{12} , which results in the following expression

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y - \log(\rho_{t,i} 10^{12}))^2, \quad (1)$$

where y is the neural network output, while ρ_t is the target thermospheric density value. To transform the neural network output into the desired predicted density, one can simply do:

$$\rho_p = \frac{e^y}{10^{12}}. \quad (2)$$

This is helpful during training because it has been observed that the logarithm of the empirical density and true density ratio is typically Gaussian (Bezděk, 2007; Doornbos, 2012). Hence, in our case, we are defining the loss as a function of the logarithm of the ratio of the predicted and true density (as it can be easily demonstrated via the properties of the logarithm): this gives the neural network an easier time learning the thermospheric density. Adam is used as the stochastic gradient descent optimization algorithm (Kingma & Ba, 2014), and Pytorch as ML framework (Paszke et al., 2019). We also performed Bayesian parameter search for the weight decay and learning rate and established a value of 1.37×10^{-8} for the weight decay parameter, and 5×10^{-5} for the learning rate parameter. Furthermore, due to the inherent randomness of the stochastic gradient descent algorithm, each run of the training process produces different weight initializations, leading to the optimization algorithm reaching different points of minimum loss in the neural network. To address this, we conducted each experiment five times, with five different seeds, and calculated the average of the metric score over those runs. Therefore, the reported model performance is always based on the average MAPE across these five runs.

2.2.2. Data Set Partitioning, Metrics, and Benchmarking Framework

The data set was divided into training, validation, and test sets. The partitioning algorithm organized the data set into months and allocated them to the respective sets. Each year was arranged into months, with the test month cyclically permuting by 2 for each year. The same process was applied to the validation set. The remaining months were added to the training set. This ensured that the ML models were trained on non-consecutive data, which would highly bias the model, and that the months used for training were not always the same, as this might cause the model to overlook seasonality events.

Table 1
Final Benchmarking Results, Expressed as Average Mean Absolute Percentage Error Over Five Runs

Category	NRLMSISE	JB08	ML (NRLMSISE inputs)	ML (JB08 inputs, single linear layer)	ML (JB08 inputs, 1 hidden layer)	ML (JB08 inputs, 2 hidden layers)	ML (JB08 inputs + OMNI)
Overall	63.18	40.71	24.63	54.20	28.92	24.81	21.32
200–250 km	10.24	14.75	8.17	18.21	8.50	7.60	6.36
250–300 km	26.22	22.30	11.00	25.53	10.65	9.32	7.01
300–350 km	65.61	29.72	11.38	26.42	14.44	12.53	10.43
350–400 km	35.07	30.38	14.05	42.29	15.57	13.36	11.09
400–450 km	69.33	39.99	22.24	76.30	27.38	24.19	21.27
450–500 km	48.99	27.14	19.06	44.08	22.07	18.20	15.17
500–550 km	150.48	97.29	65.01	122.36	78.63	67.94	60.27
Quiet (Ap 0–15)	69.77	43.47	25.84	57.35	31.15	26.60	23.01
Mild (Ap 15–30)	25.20	24.61	16.19	33.72	15.56	14.14	11.22
Minor (Ap 30–50)	20.97	23.81	19.38	34.95	15.37	13.94	11.33
Major (Ap 50+)	14.32	20.42	23.92	52.55	15.26	13.45	10.13

In terms of metrics, due to variations of thermospheric density values in the low-Earth orbit environment (which ranges from values of about 10^{-12} kg/m³ to values of about 10^{-16} kg/m³), reporting root mean squared error does not provide a useful metric to evaluate the models across different altitude ranges. Instead, we use the MAPE: this not only gives a more intuitive interpretation of the loss (since it represents the average percentage difference between predicted and actual values), but it is also a relative error metric, which makes it a scale-invariant. This latter property is very desirable for data sets like the thermospheric density one, where the magnitude of the predicted values can vary by several orders of magnitude.

To compare thermospheric density models, the proposed evaluation includes the following categories. Due to the significant variation in density at different altitudes, the data set is divided into 50 km increments, enabling a fair comparison between models. Additionally, the density can be greatly influenced by different geomagnetic conditions. Consequently, the overall performance of the model across the entire data set can vary based on these conditions. Hence, the model's performance is reported across various subcategories of the test data set. These subcategories are determined by altitude ranges: 200–250 km, 250–300 km, 300–350 km, 350–400 km, 400–450 km, 450–500 km, and 500–550 km. They are also categorized based on geomagnetic conditions, measured by the Ap index: Quiet (Ap 0–15), Mild (Ap 15–30), Minor storm (Ap 30–50), and Major storm (Ap 50+).

3. Results

Table 1 shows the MAPE from several density models for different subcategories of the test set, split by altitude and storm conditions based on the Ap index. The significance of the results is described below.

- *NRLMSISE*. The MAPE for the NRLMSISE model.
- *JB08*. The MAPE for the JB08 model. Notably, overall there is a significant reduction in MAPE versus the NRLMSISE model (40.71 vs. 63.18). This is not always the case, as the NRLMSISE model outperforms at the lowest altitude bracket for example, (200–250 km).
- *ML(NRLMSISE)*. This is a two-hidden layer neural network, with 500 nodes for each hidden layer, that is trained on exactly the same inputs as NRLMSISE. This model significantly outperforms the NRLMSISE model. Overall, the MAPE improves by 61% (24.63 compared to 63.18). This massive reduction in error is the first important result of the study. It shows that data-driven ML models can widely outperform semi-empirical physics-based model, even on exactly the same inputs. The drag on satellites is proportional to the density of the thermosphere, through which they travel. Therefore an improvement of 61% in the MAPE represents a 61% improvement in the calculation of the drag on the satellite, which is important for satellite operations.

The only case where the NRLMSISE model outperforms the ML(NRLMSISE) model is in the Major storm category ($A_p > 50$). This is likely due to the relatively small amount of extreme storm data points.

- *ML(JB08 inputs, single linear layer)*. This model is a simple model for a point of comparison. It is essentially a linear regression with linear combinations of the JB08 inputs. Unsurprisingly the model does not outperform the JB08 model.
- *ML(JB08, 1 hidden layer)*. This model includes a hidden layer of 500 nodes with a rectilinear activation for each transformation except the final layer. This simple model is able to outperform the JB08 model in every single sub-category. Again to reiterate- this is on exactly the same inputs as the JB08 model. It outperforms the ML(JB08 inputs, single linear layer) model significantly. This is because it has access to second order polynomial combinations of the JB08 inputs and so is much better able to model the density.
- *ML(JB08 inputs, 2 hidden layers)*. This model is a two-hidden layer model, of 500 nodes each, with rectilinear activations. It is trained on exactly the same inputs as JB08. Importantly, it significantly outperforms the JB08 model at all subcategories, with a reduction in overall MAPE of 39% (24.81 compared to 40.71). Similarly to the ML(NRLMSISE) analysis, this drop in the error on the density has significant implications for users of the JB08 model, because this model can widely outperform it on exactly the same inputs. Notably, it is slightly outperformed by the NRLMSISE model overall, but it has much lower errors in the higher storm categories. Compared to the previous two ML(JB08) models, it improves in all subcategories. Despite still being a fairly basic model, this shows that the mapping from the JB08 inputs to density is made more accurate by including increasingly non-linear transformations. It was noted during training that this improvement disappears once moving to 3 hidden layers or more. It is fully expected that there are better modeling choices for this task, but this is reserved for future studies.
- *ML(JB08 inputs + OMNI)*. The architecture for this model is described in Section 2.2.1. The idea is to add more measurements related to geomagnetic activity to see the improvement in model performance using this fuller picture. The inclusion of 48 hr (at 1-hr resolution) of these values results in further accuracy across all sub-categories compared to the ML(JB08 inputs, 2 hidden layers) model. This is an important result since it motivates the use of these measurements in future models. However, the overhead from having to use a significantly larger input data size might make this particular model less appealing. Obviously, learning exactly which extra geomagnetic parameters and at what time window back in time improves the density modeling will be a significant result, but this is saved for future studies.

An alternative way of comparing models is by looking at how the ratio between the observed and model density is distributed. As was observed in past works, this has typically a log-normal distribution, with values generally ranging from 0.5 to 2.0, and with varying variance, depending on the error made by the model (Doornbos, 2012). For a perfect model, one would expect this distribution to have zero variance and to be centered in one. However, this will not be the case, and any empirical, ML, or physics-based model will deviate from this behavior. In Figure 3, we display the distribution for the ML-based Karman model (KML), against NRLMSISE-00 and JB-08 (on the test set). As for the previous experiments, we ran the ML models with five different seeds and constructed a model ensemble, which we refer to as KML. The quantitative results of this model compared to NRLMSISE-00 and JB-08 are shown in Table 1. As can be observed, the ML model has the closest mean and mode to one, and it also has the smallest variance: both these characteristics confirm that the model is the most accurate and precise to predict the ground truth thermospheric density. Furthermore, between the empirical models, we see that JB-08 has a smaller variance than NRLMSISE-00, and it also more closely captures the average ground truth density, compared to the other empirical model. We report all quantitative data of the mean and standard deviation of these models for different altitude bands and geomagnetic storm conditions in Table 2.

Due to the significant results in terms of performance improvements over NRLMSISE-00 and JB-08 empirical models, as an outcome of this work, we release the KML model trained using JB-08 inputs. Furthermore, we also release Karman (Acciarini, Brown, & Baydin, 2023; Acciarini, Brown, Bridges, et al., 2023): an open-source Python package for data-driven thermospheric density modeling. This has been used to collect the data, train the ML models, and benchmark them against empirical models: our hope is that the framework can become a test bench for data-driven thermospheric density modeling, with shared data, metrics, and models. In Figure 4, we show a typical example of an output from the software. As we see from the plots of the first row, the model can return both the mean and standard deviation of the thermospheric density value, thereby providing a tool for quantifying the uncertainty of the prediction. This is achieved by combining an ensemble of neural networks (in our case five), trained with different weight initialization: more details on uncertainty quantification using an ensemble of models and their comparison with Monte Carlo dropout techniques were discussed by the authors in

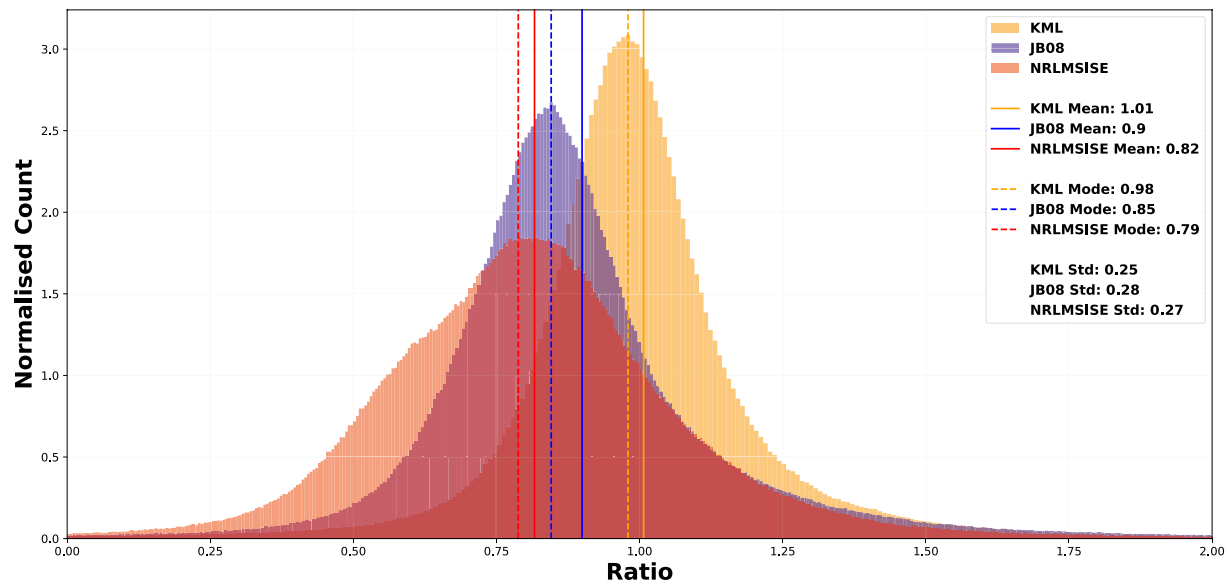


Figure 3. Ratio of Observed Density to Model Density for the KML, NRLMSISE-00, and JB08 thermospheric neutral density models.

a previous publication (Bonasera et al., 2021). On the left plots, we display the thermospheric density data associated with CHAMP satellite, while on the right with GOCE. As we observe, three models are compared against the POD-derived thermospheric density: JB-08, NRLMSISE-00, and KML model with JB-08 inputs. As displayed in the plots in the second row, the KML model is able to maintain the lowest MAPE over the tested period of time.

4. Conclusions and Future Work

Thermospheric density variations are the main source of uncertainty for the determination of resident space objects' position and velocity in low Earth orbit. Since the Sun is the main driver of changes in the atmosphere's upper levels, it is important to better model the Sun-Earth system to aid this issue. Traditionally, models based on empirical laws and fitted through data have been used in operational settings. Empirical models use solar proxies and geomagnetic indices to model the Sun's influence, due to the difficulties of obtaining EUV measurements across the spectral bands that affect the atmosphere, which can only be captured from orbit. However, it is well known that these proxies do not fully capture solar activity (Vourlidis & Bruinsma, 2018). In the last 20 years, there has been a growing amount of space missions that have gathered the Sun's EUV irradiance data from orbit (e.g., TIME/d, SOHO, SDO, GOES, etc.). Moreover, advances in ML have allowed the development of neural network models that can very well capture the nonlinear relationships between data, in a black-box and data-driven fashion.

Motivated by the above-mentioned aspects, in this work we develop Karman: an open-source ML and benchmarking framework for data-driven thermospheric density modeling. In our proposed software package, we enable users to develop and train thermospheric density ML models and benchmark their outputs against empirical models (e.g., JB-08 and NRLMSISE-00). In particular, we devise the library in order to support the ingestion of data from disparate sources, ranging from geomagnetic data of NASA OMNIWeb high-resolution service, to irradiance data from solar proxies (e.g., F10.7, M10.7, S10.7, Y10.7) and EUV irradiance empirical models (e.g., FISM2). The data can be queried with an associated temporal history and can be fed into ML models for forecasting the corresponding thermospheric density variations, where the POD-derived density from several satellites (e.g., CHAMP, SWARM-A, SWARM-B, GOCE, GRACE) is used as a target. We show that simple ML

Table 2
Mean and Standard Deviation of Ratio Between Observed and Model Densities

Category	NRLMSISE-00		JB08		KML	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Overall	0.82	0.27	0.90	0.28	1.01	0.25
200–250 km	0.97	0.11	0.88	0.08	1.00	0.08
250–300 km	0.82	0.12	0.83	0.10	0.97	0.10
300–350 km	0.63	0.12	0.80	0.14	0.99	0.14
350–400 km	0.79	0.18	0.80	0.15	0.97	0.15
400–450 km	0.70	0.24	0.85	0.25	1.00	0.24
450–500 km	0.83	0.29	0.94	0.25	1.01	0.21
500–550 km	0.89	0.40	1.05	0.49	1.07	0.44
Quiet (Ap 0–15)	0.80	0.27	0.90	0.29	1.00	0.25
Mild (Ap 15–30)	0.92	0.24	0.90	0.22	1.04	0.20
Minor (Ap 30–50)	0.97	0.27	0.90	0.23	1.08	0.20
Major (Ap 50+)	1.01	0.17	0.87	0.14	0.98	0.14

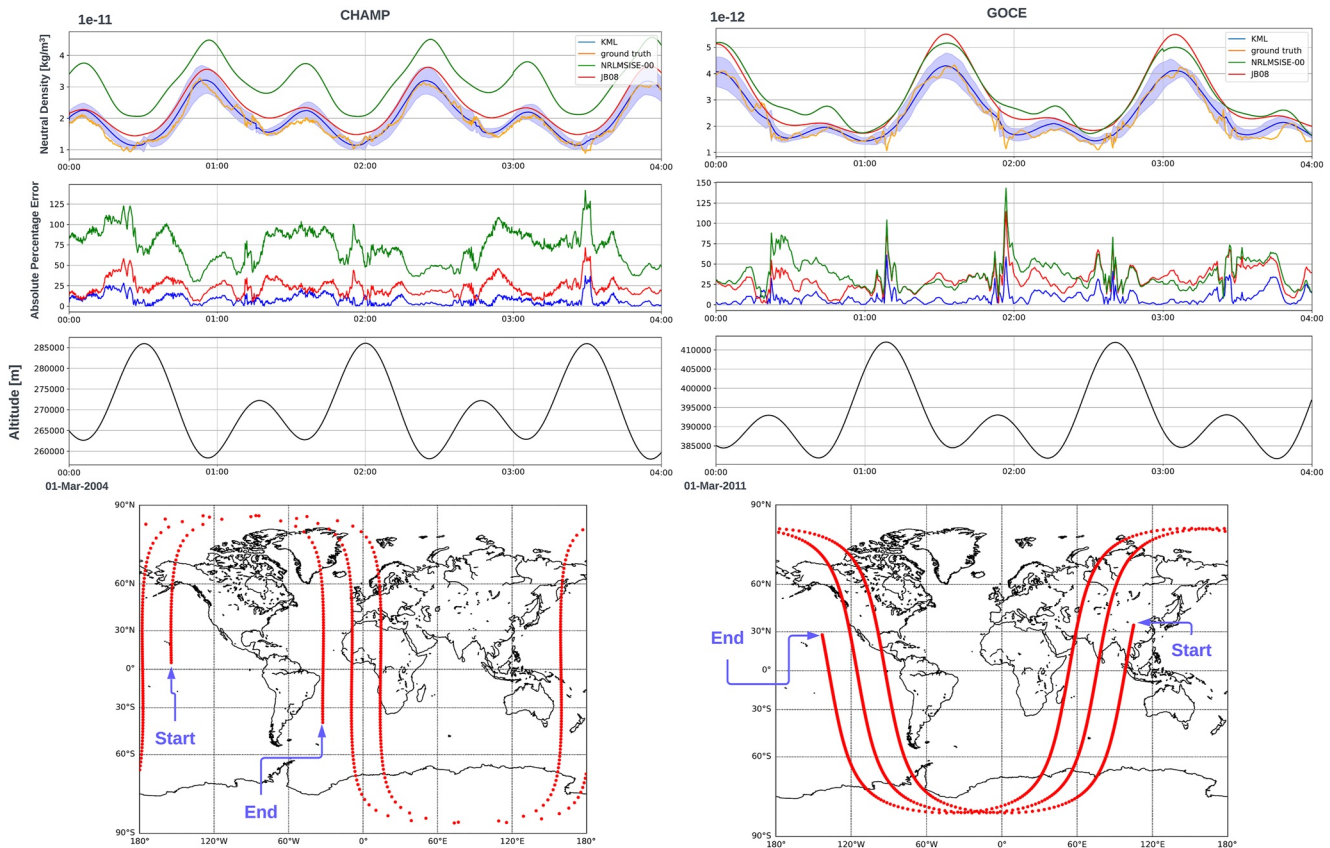


Figure 4. KML, JB08, and NRLMSISE-00 output for 1 March 2011 (GOCE) and 1 March 2004 (CHAMP).

models can easily outperform empirical models, using the same inputs, at different altitude ranges and geomagnetic storm conditions. However, it is crucial to ensure that the statistical properties of the training data closely resemble those of the validation and test data for the ML model to generalize successfully. In instances where this condition is not met, our analysis in the Appendix reveals that the ML method may experience a decline in performance. We also release one of these models, which uses the same inputs as JB-08 and nearly halves its accuracy, in terms of MAPE. Moreover, we also highlight that the ingestion of further data (in our case OMNIWeb data) can further reduce the error by a few percentage points, corroborating that a modular framework to support data from different sources such as the proposed one, can become an essential tool for the community.

In the future, we plan to use Karman for the direct ingestion of EUV spectral irradiance data and for investigating whether these data can significantly benefit the neutral density prediction task at high altitudes. Moreover, we also plan to use the framework to investigate the temporal relationship between the Sun activity and the Earth's thermosphere, to better clarify the open questions on how far back in time each phenomenon from the Sun affects the Earth's thermosphere.

Appendix A: Model Inference on Unseen Satellite Data

During the development of neural network models, it is crucial to have a train/validation/test split that avoids the model simply learns to interpolate data, and to overfit to them. In order to avoid that the model simply learns to interpolate the underlying data (which is acquired along satellites' orbit), in our work we used held-out months of data as test and validation. At the same time, we also permuted the held-out months across different years, to make sure that seasonal effects are well represented during training. In this appendix, the objective is to explore a different train/validation/test split, where the model is trained on four satellites, and then tested in a different and unseen satellite. While this is an interesting case to study, we believe that it is not ideal (and hence its inclusion in the appendix), since it will test the model on a set of data that might be sampled from a different

underlying statistical distribution, and neural networks are well known to perform poorly in out-of-distribution data. In fact, the held-out satellite might be at a completely different altitude regime, or it might observe the solar cycle in a different period or during a particular geomagnetic storm, which is not well or at all represented during training. Nonetheless, we decided to perform this experiment, carefully choosing the held-out satellite to avoid out-of-distribution testing: thus, we used SWARM-A data as a test satellite. The choice of SWARM-A was mostly driven by the fact that its density observations are at an altitude range that is covered by the other satellite data. We employ a similar training setup as the one used for the KML model. In Table A1 we show the MAPE results for the SWARM-A satellite, while Figure A1 shows a normalized histogram for the ratio between the observed density and the model density for the ML model (yellow), the JB08 model (blue) and their NRLMSISE-00 model (red). Furthermore, in Table A2, we also present the quantitative data for the density ratio results, in terms of both overall performance, as well as by different altitude and geomagnetic storm bins, reporting mean and standard deviation of the results. The ML model used was using the same inputs as JB-08. We display in the figure the distribution of the ratio of the modeled and observed data (on the test set). As it can be confirmed both qualitatively and quantitatively, the learned model still manages to outperform both empirical models. The overall MAPE achieved was about 31%, against 38% of JB-08 and 61.89% of NRLMSISE-00. This confirms that even in this train, validation, and test split setup, the ML-based model still manages to have an edge over the empirical models, even when the train, validation, and test split is made using a held-out satellite.

Table A1
Mean Absolute Percentage Error Results for the NRLMSISE, JB08, and ML(JB08) Models While Using SWARM A Density Observations as a Held-Out Test Set

Category	NRLMSISE	JB08	ML(JB08)
Overall	61.89	37.84	30.97
400–450 km	93.66	55.87	41.43
450–500 km	43.00	27.21	24.91
500–550 km	31.36	17.20	15.57
1. (0–15) Quiet	65.59	39.73	32.58
2. (15–30) Mild	42.65	27.65	22.01
3. (30–50) Minor	35.14	24.42	20.85
4. (50+) Major	34.47	30.27	24.76

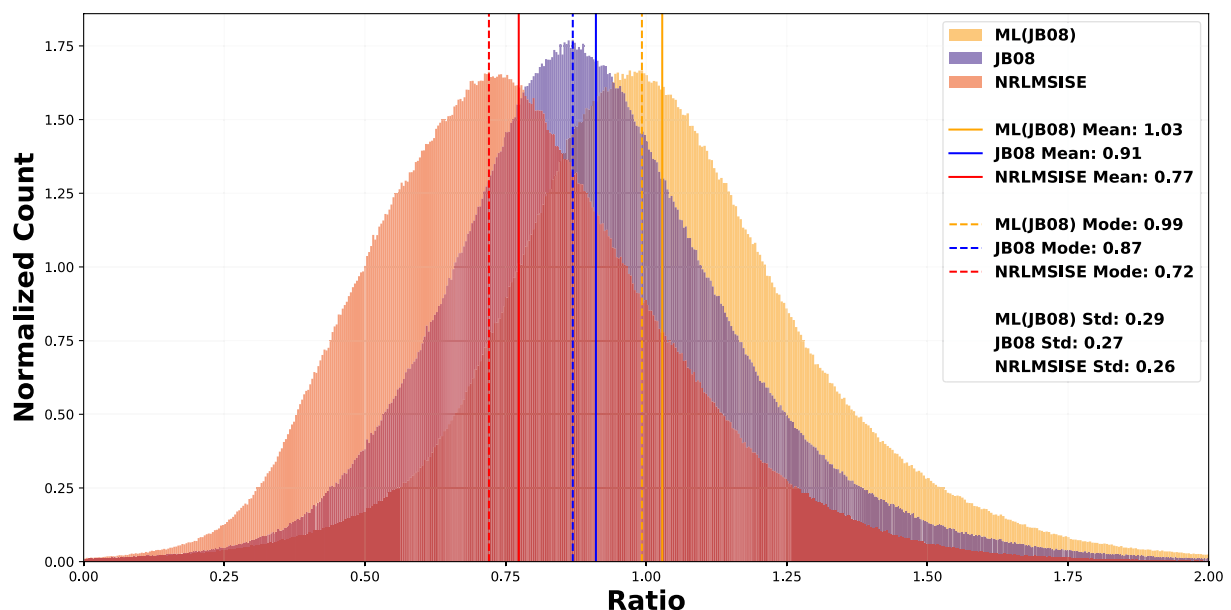


Figure A1. Ratio of Observed Density to Model Density for an machine learning (ML) model trained on JB08 inputs, NRLMSISE-00, and JB08 thermospheric neutral density models.

Table A2
Mean and Standard Deviation of Ratio Between Observed and Model Densities

Category	NRLMSISE-00		JB08		KML	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Overall	0.77	0.26	0.91	0.27	1.03	0.29
400–450 km	0.64	0.21	0.82	0.27	0.99	0.30
450–500 km	0.86	0.26	0.97	0.25	1.05	0.28
500–550 km	0.85	0.23	0.95	0.18	1.01	0.20
Quiet (Ap 0–15)	0.76	0.26	0.91	0.27	1.01	0.28
Mild (Ap 15–30)	0.82	0.24	0.94	0.25	1.12	0.29
Minor (Ap 30–50)	0.89	0.28	0.96	0.27	1.17	0.32
Major (Ap 50+)	0.92	0.31	0.97	0.34	1.19	0.46

Data Availability Statement

The space weather indices used as inputs for NRLMSISE-00 empirical model are available through Celestrack (Kelso & Sean, 2010), while those used as inputs for JB-08 empirical model are available through SET (Kent, 2023). The authors have used CCMC software and ATMOS to retrieve the empirical model's densities (Drob et al., 2023; C. Li, 2023). The thermospheric density satellite data is available online through TU Delft thermosphere group (Siemes et al., 2023). The OMNIWeb high-resolution data is available online through NASA Goddard Space Flight Center (NASA, 2023). The code used to process all the inputs, train ML models, and benchmark their performances was developed during this study, and it is available through our open-source repository Karman (Acciarini, Brown, & Baydin, 2023).

Acknowledgments

The material is based upon work supported by NASA under award No NNX13AT27A. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics Space Administration.

References

- Acciarini, G., Brown, E., & Baydin, A. G. (2023). Karman [Software]. Karman Software. Retrieved from <https://karman-project.readthedocs.io/en/latest/>
- Acciarini, G., Brown, E., Bridges, C., Baydin, A. G., Berger, T. E., & Guhathakurta, M. (2023). Karman-a machine learning software package for benchmarking thermospheric density models. In *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*.
- Berger, T., Holzinger, M., Sutton, E., & Thayer, J. (2020). Flying through uncertainty. *Space Weather*, 18(1), e2019SW002373. <https://doi.org/10.1029/2019sw002373>
- Bezděk, A. (2007). Lognormal distribution of the observed and modelled neutral thermospheric densities. *Studia Geophysica et Geodaetica*, 51(3), 461–468. <https://doi.org/10.1007/s11200-007-0027-2>
- Bonasera, S., Acciarini, G., Pérez-Hernández, J., Benson, B., Brown, E., Sutton, E., et al. (2021). Dropout and ensemble networks for thermospheric density uncertainty estimation. In *Bayesian deep learning workshop, neurIPS*.
- Bowman, B., Tobiska, W. K., Marcos, F., Huang, C., Lin, C., & Burke, W. (2008). A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices. In *AIAA/AAS astrodynamics specialist conference and exhibit* (p. 6438).
- Bruinsma, S. (2015). The DTM-2013 thermosphere model. *Journal of Space Weather and Space Climate*, 5, A1. <https://doi.org/10.1051/swsc/2015001>
- Bruinsma, S., Boniface, C., Sutton, E. K., & Fedrizzi, M. (2021). Thermosphere modeling capabilities assessment: Geomagnetic storms. *Journal of Space Weather and Space Climate*, 11, 12. <https://doi.org/10.1051/swsc/2021002>
- Bruinsma, S. L., Sánchez-Ortiz, N., Olmedo, E., & Guijarro, N. (2012). Evaluation of the DTM-2009 thermosphere model for benchmarking purposes. *Journal of Space Weather and Space Climate*, 2, A04. <https://doi.org/10.1051/swsc/2012005>
- Chamberlin, P. C., Eparvier, F. G., Knoer, V., Leise, H., Pankratz, A., Snow, M., et al. (2020). The flare irradiance spectral model-version 2 (FISM2). *Space Weather*, 18(12), e2020SW002588. <https://doi.org/10.1029/2020SW002588>
- Chen, H., Liu, H., & Hanada, T. (2014). Storm-time atmospheric density modeling using neural networks and its application in orbit propagation. *Advances in Space Research*, 53(3), 558–567. <https://doi.org/10.1016/j.asr.2013.11.052>
- Dang, T., Li, X., Luo, B., Li, R., Zhang, B., Pham, K., et al. (2022). Unveiling the space weather during the Starlink satellites destruction event on 4 February 2022. *Space Weather*, 20(8), e2022SW003152. <https://doi.org/10.1029/2022sw003152>
- Doornbos, E. (2012). *Thermospheric density and wind determination from satellite dynamics*. Springer Science & Business Media.
- Drob, D., Emmert, J., & Yue, J. (2023). CCMC [Software]. NRLMSISE-00 model. Retrieved from <https://ccmc.gsfc.nasa.gov/models/NRLMSIS~00/>
- Emmert, J. T. (2015). Thermospheric mass density: A review. *Advances in Space Research*, 56(5), 773–824. <https://doi.org/10.1016/j.asr.2015.05.038>
- He, C., Yang, Y., Carter, B., Kerr, E., Wu, S., Deleflie, F., et al. (2018). Review and comparison of empirical thermospheric mass density models. *Progress in Aerospace Sciences*, 103, 31–51. <https://doi.org/10.1016/j.paerosci.2018.10.003>
- Jackson, D. R., Bruinsma, S., Negrin, S., Stolle, C., Budd, C. J., Gonzalez, R. D., et al. (2020). The space weather atmosphere models and indices (SWAMI) project: Overview and first results. *Journal of Space Weather and Space Climate*, 10, 18. <https://doi.org/10.1051/swsc/2020019>
- Kelso, T., & Sean, T. (2010). Celestrak [Dataset]. Public domain satellite tracking data. Retrieved from <https://www.celestrak.com/SpaceData/>

- Kent, T. (2023). Jb-08 solar indices [Dataset]. Solar indices used in JB-08 model. Data is Retrieved from <https://sol.spacenvironment.net/JB2008/indices/>
- King, J., & Papitashvili, N. (2005). Solar wind spatial scales in and comparisons of hourly wind and ACE plasma and magnetic field data. *Journal of Geophysical Research*, 110(A2), A02104. <https://doi.org/10.1029/2004ja010649>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- LASP. (2023). Solar irradiance data [Dataset]. Solar irradiance data from the Laboratory for Atmospheric and Space Physics, at CU Boulder. Data is Retrieved from <https://lasp.colorado.edu/our-expertise/data-systems/data-products/>
- Li, C. (2023). Atmos [Software]. JB-08 model. Retrieved from <https://github.com/lcx366/ATMOS>
- Li, W., Liu, L., Chen, Y., Xiao, Z., Le, H., & Zhang, R. (2023). Improving the extraction ability of thermospheric mass density variations from observational data by deep learning. *Space Weather*, 21(7), e2022SW003376. <https://doi.org/10.1029/2022sw003376>
- Licata, R. J., & Mehta, P. M. (2022). Uncertainty quantification techniques for data-driven space weather modeling: Thermospheric density application. *Scientific Reports*, 12(1), 7256. <https://doi.org/10.1038/s41598-022-11049-3>
- Licata, R. J., Mehta, P. M., Tobiska, W. K., Bowman, B. R., & Pilinski, M. D. (2021). Qualitative and quantitative assessment of the SET HASDM database. *Space Weather*, 19(8), e2021SW002798. <https://doi.org/10.1029/2021sw002798>
- Licata, R. J., Mehta, P. M., Tobiska, W. K., & Huzurbazar, S. (2022). Machine-learned HASDM thermospheric mass density model with uncertainty quantification. *Space Weather*, 20(4), e2021SW002915. <https://doi.org/10.1029/2021sw002915>
- Licata, R. J., Mehta, P. M., Weimer, D. R., Drob, D. P., Tobiska, W. K., & Yoshii, J. (2022). Science through machine learning: Quantification of post-storm thermospheric cooling. *Space Weather*, 20(9), e2022SW003189. <https://doi.org/10.1029/2022sw003189>
- Marcos, F., Bowman, B., & Sheehan, R. (2006). Accuracy of Earth's thermospheric neutral density models. In *AIAA/AAS astrodynamics specialist conference and exhibit* (p. 6167).
- Mehta, P. M., & Linares, R. (2017). A methodology for reduced order modeling and calibration of the upper atmosphere. *Space Weather*, 15(10), 1270–1287. <https://doi.org/10.1002/2017sw001642>
- Mehta, P. M., Linares, R., & Sutton, E. K. (2018). A quasi-physical dynamic reduced order model for thermospheric mass density via hermitian space-dynamic mode decomposition. *Space Weather*, 16(5), 569–588. <https://doi.org/10.1029/2018sw001840>
- NASA. (2023). Omniweb data [Dataset]. OMNIWeb high-resolution data. Retrieved from https://omniweb.gsfc.nasa.gov/form/omni_min.html
- Oliveira, D. M., Zesta, E., Hayakawa, H., & Bhaskar, A. (2020). Estimating satellite orbital drag during historical magnetic superstorms. *Space Weather*, 18(11), e2020SW002472. <https://doi.org/10.1029/2020sw002472>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (Vol. 32).
- Pérez, D., & Bevilacqua, R. (2015). Neural network based calibration of atmospheric density models. *Acta Astronautica*, 110, 58–76. <https://doi.org/10.1016/j.actaastro.2014.12.018>
- Pérez, D., Wohlberg, B., Lovell, T. A., Shoemaker, M., & Bevilacqua, R. (2014). Orbit-centered atmospheric density prediction using artificial neural networks. *Acta Astronautica*, 98, 9–23. <https://doi.org/10.1016/j.actaastro.2014.01.007>
- Picone, J., Hedin, A., Drob, D. P., & Aikin, A. (2002). NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research*, 107(A12), SIA–15. <https://doi.org/10.1029/2002ja009430>
- Richmond, A., Ridley, E., & Roble, R. (1992). A thermosphere/ionosphere general circulation model with coupled electrodynamics. *Geophysical Research Letters*, 19(6), 601–604. <https://doi.org/10.1029/92gl00401>
- Siemes, C., de Teixeira da Encarnação, J., Doornbos, E., Van Den Ijssel, J., Kraus, J., Perešty, R., et al. (2016). Swarm accelerometer data processing from raw accelerations to thermospheric neutral densities. *Earth, Planets and Space*, 68(1), 1–16.
- Siemes, C., Doornbos, E., & Iorfida, E. (2023). Thermosphere data [Dataset]. TU Delft thermospheric density data extracted from GOCE, CHAMP, GRACE, Swarm Missions. Retrieved from <http://thermosphere.tudelft.nl/>
- Solomon, S. C., & Qian, L. (2005). Solar extreme-ultraviolet irradiance for general circulation models. *Journal of Geophysical Research*, 110(A10), A10306. <https://doi.org/10.1029/2005SW002588>
- Storz, M. F., Bowman, B. R., Branson, M. J. I., Casali, S. J., & Tobiska, W. K. (2005). High accuracy satellite drag model (HASDM). *Advances in Space Research*, 36(12), 2497–2505. <https://doi.org/10.1016/j.asr.2004.02.020>
- Tobiska, W. K., Bouwer, S. D., & Bowman, B. R. (2008). The development of new solar indices for use in thermospheric density modeling. *Journal of Atmospheric and Solar-Terrestrial Physics*, 70(5), 803–819. <https://doi.org/10.1016/j.jastp.2007.11.001>
- Tobiska, W. K., Bowman, B. R., Bouwer, S. D., Cruz, A., Wahl, K., Pilinski, M. D., et al. (2021). The SET HASDM density database. *Space Weather*, 19(4), e2020SW002682. <https://doi.org/10.1029/2020sw002682>
- Turner, H., Zhang, M., Gondelach, D., & Linares, R. (2020). Machine learning algorithms for improved thermospheric density modeling. In F. Darema, E. Blasch, S. Ravela, & A. Aved (Eds.), *Dynamic data driven applications systems* (pp. 143–151). Springer International Publishing.
- Vallado, D. A. (2001). *Fundamentals of astrodynamics and applications* (Vol. 12). Springer Science & Business Media.
- Vourlidas, A., & Bruinsma, S. (2018). Euv irradiance inputs to thermospheric density models: Open issues and path forward. *Space Weather*, 16(1), 5–15. <https://doi.org/10.1002/2017sw001725>
- Wang, P., Chen, Z., Deng, X., Wang, J., Tang, R., Li, H., et al. (2022). The prediction of storm-time thermospheric mass density by LSTM-based ensemble learning. *Space Weather*, 20(3), e2021SW002950. <https://doi.org/10.1029/2021sw002950>
- Wang, P., Chen, Z., Deng, X., Wang, J.-S., Tang, R., Li, H., et al. (2023). The short-time prediction of thermospheric mass density based on ensemble-transfer learning. *Space Weather*, 21(10), e2023SW003576. <https://doi.org/10.1029/2023sw003576>
- Wang, Y., & Bai, X. (2023). Thermospheric density predictions during quiet time and geomagnetic storm using a deep evidential model-based framework. *Acta Astronautica*, 211, 316–325. <https://doi.org/10.1016/j.actaastro.2023.06.023>
- Weng, L., Lei, J., Zhong, J., Dou, X., & Fang, H. (2020). A machine-learning approach to derive long-term trends of thermospheric density. *Geophysical Research Letters*, 47(6), e2020GL087140. <https://doi.org/10.1029/2020gl087140>
- Zhang, Y., Yu, J., Chen, J., & Sang, J. (2021). An empirical atmospheric density calibration model based on long short-term memory neural network. *Atmosphere*, 12(7), 925. <https://doi.org/10.3390/atmos12070925>