AUTOENCODER-BASED THERMOSPHERIC DENSITY MODEL FOR UNCERTAINTY QUANTIFICATION AND REAL-TIME CALIBRATION

M. Manzi and M. Vasile

Aerospace Centre of Excellence,
University of Strathclyde, Glasgow, UK,
Email: {matteo.manzi, massimiliano.vasile}@strath.ac.uk

ABSTRACT

The dynamics of space objects in Low Earth Orbit (LEO) is strongly determined by the effects of atmospheric drag: this complex interaction, dependent on the physical properties of the atmosphere, comprises several of the primary sources of orbit uncertainty in LEO. However, most atmospheric density models, both empirical and semi-empirical, do not explicitly provide an estimate of the uncertainty.

In this work, we propose a Deep Learning technique based on the use of autoencoders, a family of artificial Neural Networks (NN), to construct a grey-box model of the dynamics of the thermosphere density field. This gray-box model allows for a nearly real-time quantification of the uncertainty in the dynamics using satellite tracking data.

Building on the U.S. Air Force High Accuracy Satellite Drag Model (HASDM), we develop a reduced-order dynamical model of the thermosphere density: autoencoders can be used to construct a nonlinear embedding of the high-dimensional, chaotic dynamics associated with the physical space, mapping it onto a low-dimensional manifold.

The Neural Network can be structured following the encode-propagate-decode model so that the dynamics associated with the reduced-order model can be reconstructed in tandem with the model reduction and decomposition. The dynamics can be reconstructed via Deep Symbolic Regression, leading to a nonlinear differential equation governing the system and can include space weather effects (related to the $F10.7$ index) and geomagnetic activities (related to the a_p index).

Making use of such a grey-box model, the estimation of the density field can be coupled with the dynamics of tracked objects, leading to real-time calibration to deal with both measurement and model uncertainties. The obtained model, because of its explicit differential formulation, enables one to take into account parametric uncertainties, leading to a more informed uncertainty quantification for a system of interest.

In the paper, we start by applying the proposed methodology to a simple, low-dimensional dynamical system whose underlying model is explicitly known. After that, the technique will be applied to model the dynamics of the thermosphere density field.

Keywords: Machine Learning; Data-driven models; Uncertainty Quantification; Space Weather; Space Situational Awareness; Thermosphere density; Satellite drag.

1. INTRODUCTION

In the context of the so-called New Space [27], Low-Earth Orbit (LEO) is characterized by an unprecedented increase in space traffic. The clearer example of this is the launch of mega constellations [33, 28]. At the same time, the dynamics of LEO satellites is strongly influenced by atmospheric drag, which is also the largest source of uncertainty in orbital motion: in fact, at LEO altitudes, atmospheric drag is one of the main forces acting on spacecraft and debris. Being particularly hard to model and predict, the drag acceleration $\mathbf{a}_{drag}$ is given in Equation (1):

$$\mathbf{a}_{drag} = -\frac{1}{2}\frac{C_D A}{m}\rho v_{rel}^2 \frac{\mathbf{v}_{rel}}{v_{rel}} \tag{1}$$

where C_D, A, m are the drag coefficient, the effective area and mass of the object; $\mathbf{v}_{rel}$ is its velocity with respect to the moving atmosphere, characterized by a density ρ.

Once the physical properties of a satellite or space debris are given, the drag acceleration is therefore only a function of the wind vector field and the density scalar field[1]. Introducing spherical coordinates associated to the Earth-centered inertial reference frame, such density can be obtained from the evaluation of the underlying field (Equation (2)):

[1] While a number of missions has been recently dealing with the collection of wind data, this work focuses on the modelling of the density field only.

$$\rho = \rho(r, \theta, \phi, t) \tag{2}$$

The density of the thermopshere is particularly difficult to model because its behaviour is strongly coupled to the space environment, particularly to the dynamics of the ionosphere, magnetosphere and of the Sun, which are by themselves difficult to predict [6]. Because of such a context, it is interesting to restate the following quote, given in [17]: "the central difficulty in climate change science is that the dynamical equations for the actual climate are unknown. All that is available from the true climate in nature are some coarse-grained observations [..]. Thus, climate change science must cope with predicting the coarse-grained dynamic changes of an extremely complex system only partially observed from a suite of imperfect models for the climate [20]." It should be clear how such considerations associated to climate science are equally applicable to the study of the thermospheric density field, and that the forecasting aspects are particularly relevant in space engineering.

In fact, accurate orbit determination and uncertainty quantification are required for the automation of space operations, in particular in order to perform conjunction assessment and collision avoidance [6]; in more general terms, the increase in space traffic motivates the ability to efficiently track and catalogue objects.

In order to do this, a number of existing density models could be used. On the one hand, physics-based models, associated to the solution of the continuity, momentum and energy equations are particularly accurate but computationally expensive, because of the need to solve Navier-Stokes equations and because of the associated high-dimensionality [34, 30]. On the other hand, empirical models are used to describe the average behaviour of the atmosphere, via mathematical formulations arising from historical observations. While their evaluation is really fast, making them suitable for the automation of space operations, their accuracy and forecasting capability are limited, particularly in the context of highly nonlinear phenomena, such as space weather events. Some examples of empirical models are:

- the Drag Temperature Model (DTM-2013) model [2]; it is a semi-empirical model describing the temperature, density and composition of the Earth's thermosphere.
- the NRLMSISE-00 model [29]; it is an empirical atmospheric model extending from the ground to the exobase and is an upgrade of the MSISE-90 model.
- the Jacchia-Bowman 2008 (JB2008) model [16, 1]; based on Jacchia's diffusion equations, it is an empirical atmospheric density model making use of solar and geomagnetic indices.
- the U.S. Air Force High Accuracy Satellite Drag Model (HASDM) [40, 41], an empirical model calibrated using observations of calibration satellites.

A number of recent works have proposed a trade-off between the model accuracy and computational efficiency, making use of Reduced Order Modelling (ROM) techniques [24, 25, 26]. In line with such works, and following the aforementioned motivations, it becomes necessary to build on existing atmospheric density models to construct a grey-box model of the dynamics and efficiently perform real-time calibration [11, 12] and uncertainty quantification [10].

While there is a plethora of Reduced Order Modelling techniques to be used, going from linear techniques to approaches aiming at leveraging coherent structures associated to the flow [43, 32], the use of Machine Learning techniques, leveraging the power of Artificial Neural Networks (and Deep Neural Networks, in particular) has emerged as a promising tool to perform Reduced Order Modelling [13, 14], whose application is particularly compelling in the field of fluid mechanics [3]. In particular autoencoders, a specific kind of Artificial Neural Network, have been applied in order to deal with fluid mechanics problems and perform flow field prediction [9, 5, 15, 8, 23, 4].

The goal of this work is therefore the following:

we aim at using autoencoders to obtain a reduced-order representation of the dynamical system governing the evolution of the thermospheric density field.

This is done, among other things, to model the uncertainty in the dynamics of the thermosphere density field, which has direct implications on the uncertainty of the dynamics of LEO satellites, and therefore on their position at the time of closest approach with other objects [37].

This work is structured as follows:

- reduced-order modelling techniques both based on linearity assumptions and on autoencoders (both fully connected and convolutional) will be presented;
- results associated to a simple case study, the Chaotic Lorenz System, used to demonstrate the capabilities of the proposed methodology, will be given;
- results associated to the modelling of the thermospheric density field will be given, after a brief discussion about the data used for this;
- the methodology will be expanded in order to connect the given results with data-driven reconstruction of governing equations;
- conclusions and recommendations for future works will be given.

2. METHODOLOGY

In order to overcome the high-dimensionality of physics-based density models, Reduced-Order Modelling (ROM) techniques can be used; in this way, the original high-dimensional system is represented using a smaller number of parameters. In recent works, e.g. [24, 11], the order reduction is performed making use of Proper Orthogonal Decomposition [35] (also known as Principal Component Analysis). This technique makes use of Singular Value Decomposition (SVD) in order to compute orthogonal basis functions (also called spatial modes), capturing the dominant characteristics of the system. The main limitation with this approach is its linearity: SVD is a factorization technique of matrices, and therefore leads to a linear mapping between the an high-dimensional and a low-dimensional space. The thermospheric dynamics is however highly nonlinear, and such Reduced Order Modelling technique has limited accuracy, as outlined in [42].

2.1. Autoencoders

This is why in this work we propose to use autoencoders to represent thermospheric density data in a reduced space, making use of non-linear dimensionality reduction. In fact, autoencoders lead to a nonlinear embedding of the high-dimensional field associated with the physical space, mapping it onto a low-dimensional manifold. Manifold learning relies on the manifold assumption, stating that most real-world high-dimensional datasets lie close to a much lower-dimensional manifold [14]. We will assume this to hold for thermospheric density data as well.

An autoencoder is a kind of Neural Networks that, without any supervision (i.e., the training set is unlabeled), takes an input $\mathbf{x}$, encodes it as a vector $\mathbf{z}$ and then decodes it to $\hat{\mathbf{x}}$, which is a reconstruction of the input itself [13]. It is made of two distinguishable parts (Figure 1), an encoder Φ and a decoder Ψ, such that:

$$\Phi : \mathbf{x} \rightarrow \mathbf{z} \tag{3}$$

$$\Psi : \mathbf{z} \rightarrow \hat{\mathbf{x}} \tag{4}$$

Because of their most common structure, in which inner layers of the network are characterized by a smaller number of neurons, compared to the input and the output (i.e., undercomplete autoencoders), the vector $\mathbf{z}$, associated to the hidden layer of the network (also known as its bottleneck), is a latent, compressed representation of the input vector $\mathbf{x}$. In fact, learning an undercomplete representation forces the autoencoder to capture the most salient features of the training data.

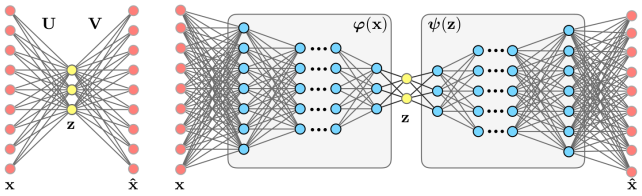


Figure 1. Schematics of shallow and deep autoencoder [3]. The input $\boldsymbol{x}$ is reduced to $\boldsymbol{z}$ via the decoder; an approximation of the input $\hat{\mathbf{x}}$ is reconstructed from $\boldsymbol{z}$ via the decoder.

The procedure can be thought as a generalization of dimensionality reduction techniques, such as Proper Orthogonal Decomposition. In fact, when using only one layer for the encoder and the decoder, and when using linear activation functions to construct them (see left graph in Figure 1), the procedure is equivalent to Single Value Decomposition from which the matrices U and V arise.

2.2. Convolutional Autoencoders

A subcategory of autoencoders can be introduced: convolutional autoencoders perform the same task of fully connected autoencoders, but making use of a slightly different technique. Their relevance for the problem at hand can be understood because of their ability to deal with spatially coherent information. In Convolutional Autoencoders, the layers of the network perform the sampling and convolution operations.

While a fully connected architecture will be used to deal with the Lorenz Systems (the reasons for this choice are outlined in Section 3.1), for the compression of the thermospheric density field convolutional autoencoders will be used. The main reason behind this design choice is that the convolution operation takes into account the relative position of the measured field: this is ideal for the modelling of observed phenomena governed by partial differential equations, in which local interactions drive the evolution of the field [18, 19].

Convolutional Autoencoders (whose schematic network is represented in Figure 2) are associated to quicker training and reduced probability of overfitting, compared to fully connected networks, because of the reduced number of trainable weights in the model.

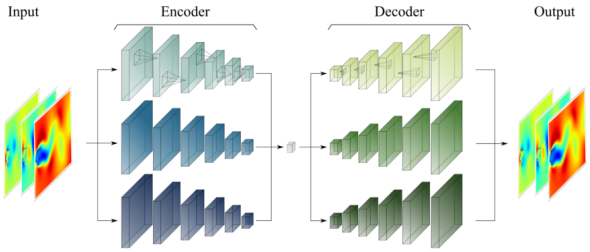


Figure 2. Schematic of a Convolutional Neural Network-Autoencoder [15].

Being autoencoders, and in particular convolutional autoencoders, a special case of feedforward neural networks, they can be trained using the same techniques: in this work, we make use of backpropagation and mini-batch gradient descent. Moreover, in order to perform the training, available data are divided into training and validations data; this is usually done in the context of neural networks. The idea is that the network is trained against the training data and at the same time its performance is assessed against validation data [13]. The goal of this is to avoid overfitting, i. e. to avoid the network to adapt too much to the behaviour of training data and to maintain generalizability.

In order to perform the training of the autoencoders, throughout this work the mean squared error is used as a loss function (Equation (5)):

$$L(\mathbf{x}, \Psi(\Phi(\mathbf{x}))) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{5}$$

3. RESULTS

3.1. Chaotic Lorenz System

We start by applying the proposed methodology to a dynamical system whose differential equations are explicitly known, and which is indirectly observed through an high-dimensional embedding. We do this because we are interested in reconstructing the dynamics of the low-dimensional system which, being explicitly known, can be propagated and compared to the results of the compression associated to the use of autoencoders.

Following [7], we construct a high-dimensional problem whose dynamics follows from the chaotic Lorenz system. The highly nonlinear dynamics of the Lorenz System is governed by the following set of equations:

$$\begin{aligned} \dot{z}_1 &= \sigma(z_2 - z_1) \\ \dot{z}_2 &= z_1(\rho - z_3) - z_2 \\ \dot{z}_3 &= z_1 z_2 - \beta z_3 \end{aligned} \tag{6}$$

This system has been chosen because, displaying chaos, its dynamics has relations with the one of the atmospheric density field[2].

From it, we can construct a mapping between $\mathbb{R}^3$ and $\mathbb{R}^{128}$, using six fixed spatial modes associated to Legendre polynomials; in particular, such polynomials are evaluated along a uniform grid defined in [-1, 1]. We do this since we are interested in observing high-dimensional fields, to be compressed with autoencoders: we therefore assume that the true three-dimensional dynamics is not directly observed.

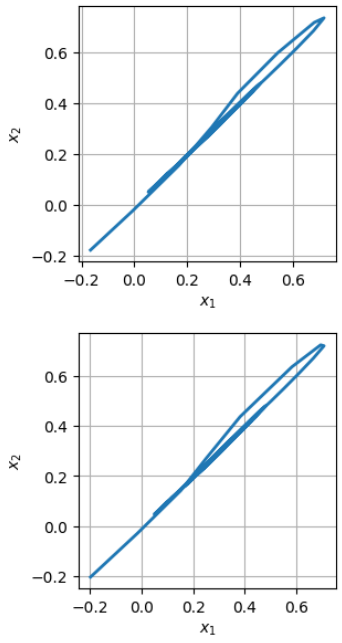
We will consider two mappings, one linear and one non-linear, in which cubic terms are added to the linear terms:

$$\mathbf{x}_1(t) = \mathbf{u}_1 z_1(t) + \mathbf{u}_2 z_2(t) + \mathbf{u}_3 z_3(t) \tag{7}$$

$$\mathbf{x}_2(t) = \mathbf{x}_1(t) + \mathbf{u}_4 z_1(t)^3 + \mathbf{u}_5 z_2(t)^3 + \mathbf{u}_6 z_3(t)^3 \tag{8}$$

In equations (7) and (8), the components of the $\mathbf{u}_n$ vectors are the 128 evaluations of the n^{th} Legendre polynomial.

It is now possible to consider different initial conditions of the Lorenz System, in order to obtain a number of 128-dimensional observed trajectories to be divided, as usual, into training and validation trajectories. In particular, we make use of 1024 training and 10 validation trajectories, associated to different initial conditions: this is done for data augmentation purposes.



[2] In fact, the Lorenz system has been formulated as a simplified mathematical model for atmospheric convection.

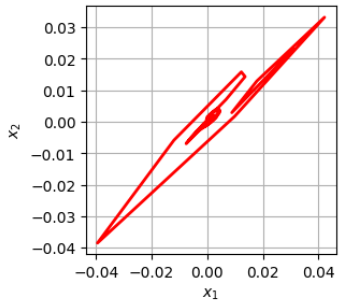


Figure 3. Reference trajectory and reconstructed one; reconstruction error.

Since the input/output size is 128, and the latent space dimension of the network is 3, we make use of fully connected layers to tackle this problem; in fact, since the mapping used to go from $\mathbb{R}^3$ and $\mathbb{R}^{128}$ is analytical, we expect fully connected layers to reconstruct such relation.

Associated to the linear mapping given in Equation (7), we make use of a linear activation function to construct our autoencoder. As previously stated, such linear autoencoder mimics the behaviour of Proper Orthogonal Decomposition (i.e., the encoder behaves like a matrix). Once the autoencoder is trained, we can compare one of its inputs, taken from the validation dataset, and compare it to the decoded trajectory (Figure 3).

The two blue trajectories are the input and the decoded one, while the red trajectory is their mismatch.

Input
Encoder
Flattening Layer
Densely-connected NN layer, 128 units, hyperbolic tangent activation function
Densely-connected NN layer, 64 units, hyperbolic tangent activation function
Densely-connected NN layer, 3 units (latent dimension), hyperbolic tangent activation function
Decoder
Densely-connected NN layer, 64 units, hyperbolic tangent activation function
Densely-connected NN layer, 128 units, hyperbolic tangent activation function

Table 1. Structure of the fully connected autoencoder associated to the nonlinear mapping of the Lorenz system.

It is now possible to apply the same methodology for the case in which the mapping between $\mathbb{R}^3$ and $\mathbb{R}^{128}$ is nonlinear (Equation (8)). In order to deal with this problem, the structure of the autoencoder needs to be generalized: for building it, we make use of five layers; moreover, we substitute the activation function with the hyperbolic tangent (Table 1). Before training, data are normalized, in order for the network to be able to reconstruct their behaviour. Again, it is possible to compare validation inputs with its decoded trajectory (Figure 4).

For this case the reconstruction error, smaller than before, is in the order of 10^{-5}.

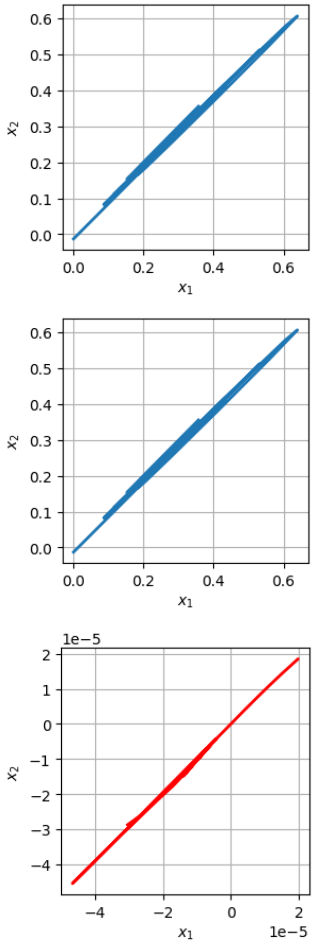


Figure 4. Reference trajectory and reconstructed one; reconstruction error.

We now focus on the latent space representation of data; in fact, when using autoencoders, and in general when dealing with dimensionality reduction, the interesting result, together with the ones used to assess the ability to reconstruct the original data, are compressed data themselves, which can be thought as a byproduct of the autoencoder's attempt to learn the identity function under some constraints. In order to do this, once the network is trained, it is possible to follow the conceptual steps associated to the reconstruction of the reduced-order trajectory as outlined in Figure 5:

- first, associated to a fixed three-dimensional initial

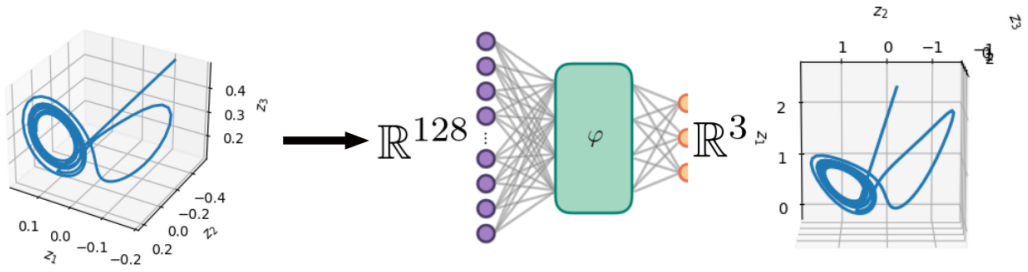


Figure 5. Conceptual steps for the reconstruction of the reduced-order trajectory.

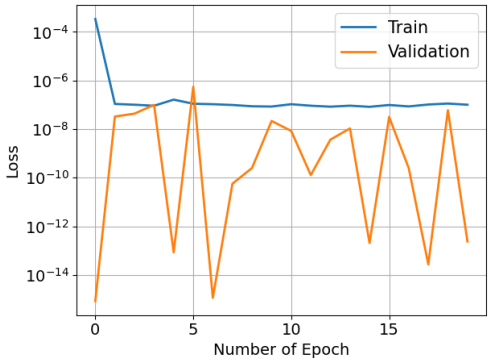
condition, the dynamics of the Chaotic Lorenz system is given;

- via the analytical mapping the dynamics is lifted from $\mathbb{R}^3$ to $\mathbb{R}^{128}$, leading to the observed trajectory, used as an input to the autoencoder;

- finally, making use of the trained encoder, it is possible to represent the trajectory associated to the reduced-order representation given in the latent space.

The decoder would match the analytical map from $\mathbf{z}$ to $\mathbf{x}$ only in the case in which the bottleneck is $\mathbf{z}$: this is not the case in the given results. Nevertheless, the observed trajectories are correctly reconstructed. The compressed trajectory exhibit an attractor with a 2-lobe structure, like the one of the original Lorenz attractor. Moreover, we are not really interested in reconstructing the "true" trajectory, as long as the one we are reconstructing contains equivalent information.

The behaviour associated to the convergence, during training, is represented in Figure 6.



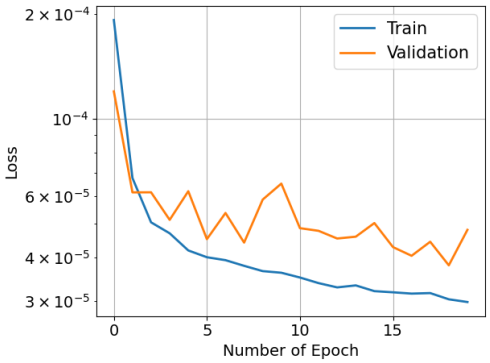


Figure 6. Loss history associated to linear and nonlinear mappings.

The first plot does not show the usual behaviour of converging networks, because the validation loss is really small, already from epoch one; this is because a linear activation function has been used for the associated autoencoder. The second plot, associated to an autoencoder with a nonlinear activation function, displays a clear convergence.

Finally, we focus again on the first and last plots in Figure 5: while the exact trajectory of the reduced-order state, associated to the Lorenz system's set of equations, has not been exactly reconstructed, the autoencoder is successful in embedding the high-dimensional, observed field into a low-dimensional space, whose decoding leads to an approximation of the recorded data. The fact that there is a similarity between the Lorenz trajectory and the one associated to the latent space is an additional factor supporting the successful behaviour of the autoencoder. Because of this, it is reasonable to now apply the same methodology to a real-world problem, such as the thermospheric density field.

3.2. Thermospheric Density

Dealing with a problem with more practical implications, let us now apply the proposed methodology for the modelling of the thermospheric density field. To do this, in this section we make use of the U.S. Air Force High Ac-

curacy Satellite Drag Model (HASDM) data [41], derived real time from accurate radar observations of multiple calibration satellites[3].

An alternative choice would have been to make use of the DTM-2013 model data [2], available on the Space Situational Awareness Space Weather Service Network[4]. However, DTM-2013 available data cover only the time interval October 2017 - April 2021, which are associated to less than one solar cycle (Figure 7).

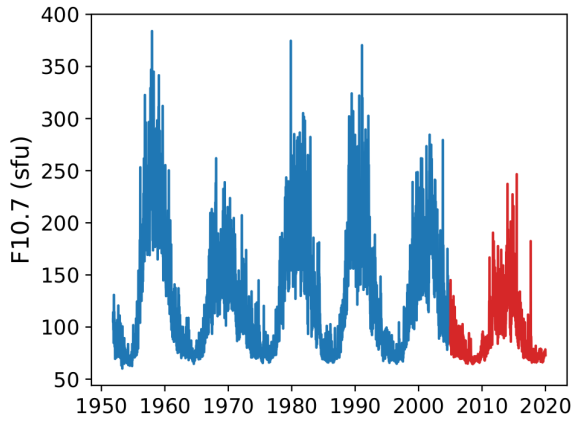


Figure 7. Solar activity history expressed via the F10.7 index [39]. The F10.7 time series is split into training (blue) and validation (red) datasets.

On the other hand, the HASDM available data cover two full solar cycles, going from January 1^{st} 2000 to December 31^{st} 2019, and are thus representative of the overall behaviour of the density field. This is particularly relevant for extrapolation.

For each day, the temporal resolution is 3 hours; the database has a grid size of 10° latitude, 15° longitude, 25 km altitude (175 - 825 km) (Table 2).

	Domain:	Resolution:
Local solar time [hr]	[0, 24[	3
Latitude [deg]	[-90, 90]	10
Longitude [deg]	[0, 360[	15
Altitude [km]	[175, 825]	25

Table 2. HASDM data charateristics in the time interval from January 1^{st} 2000 to December 31^{st} 2019.

[3]Last visited on 6^{th} April 2021: `https://spacewx.com/hasdm/`

[4]Last visited on 6^{th} April 2021: `https://swe.ssa.esa.int/current-space-weather`

Input
Encoder
3D convolution layer 8 filters, kernel size (3 3 3), ReLu activation function
Max pooling operation for 3D data Pool size (2 2 2)
3D convolution layer 16 filters, kernel size (3 3 3), ReLu activation function
Max pooling operation for 3D data Pool size (2 2 2)
Flattening
Densely-connected NN layer ReLu activation function
Decoder
Densely-connected NN layer ReLu activation function
Reshaping
3D upsampling, size (2 2 2)
3D deconvolution layer 16 filters, kernel size (3 3 3), ReLu activation function
3D upsampling, size (2 2 2)
3D deconvolution layer 8 filters, kernel size (3 3 3), ReLu activation function
3D convolution layer 1 filter, kernel size (1 1 1), ReLu activation function

Table 3. Structure of the convolutional autoencoder making use of HASDM data for the modelling of thermospheric density field.

As opposed to the Lorenz System, the dynamics of the observed field is associated to a partial differential equation, in which local interactions are relevant for the evolution of the density field. Because of this, convolutional autoencoders, and in particular 3-dimensional convolutional autoencoders are used. This means that the input is a 3D tensor associated to polar coordinates.

The network is made of 4 convolutional and 2 fully connected layers (Table 3). Dealing with hyperparameters, after tuning, the latent state size has been set to 32. The rectifier (ReLu) activation function has been used in all the layers of the network.

Again, data have been normalized, making use of the maximum density value in space and time: in this way all the data used for the training are defined in $[0, 1]$, and the use of the ReLu activation function is justified.

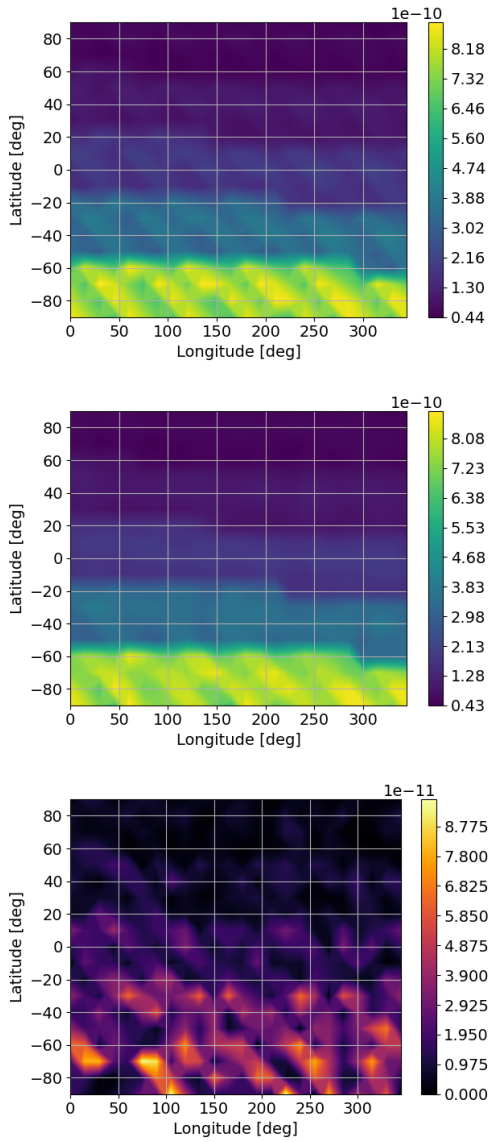


Figure 8. Reference and reconstructed density field; reconstruction error.

Once the training is performed, it is possible to test the performance of such autoencoder on a given dataset, for example on the density field at 175 km altitude, on the 1^{st} January 2000, at midnight[5]. Figure 8 represents the comparison between the input field and the decoded one, also via the error field, which in this case is the absolute value of the difference between the two.

Such results represent the reconstruction behaviour of the autoencoder on a subset of the three-dimensional tensor given as an output by the autoencoder. As previously described, these are obtained using polar coordinates to define the data used for the autoencoder training. In fact, the convolution operation is associated to orthogonal directions, and the information about the physical geometry of the data is shared with the input data tensors. While spherical convolution neural networks exist, more classical convolution techniques are still able to deal with spherical coordinates, as these are associated to an orthonormal basis. The only limitation in the given results is that the information about the periodicity of the field in the longitude axis is lost; nevertheless, the represented results do not display an increasing error in the regions with longitude 0^o and 360^o.

As already done for the Lorenz system, it is also possible to investigate the behaviour of the reduced-order representation of the state of the system, once training is performed. Figure 9 represents the evolution of the first two components (out of 32) of the $\mathbf{z}$ vector, embedding the density field.

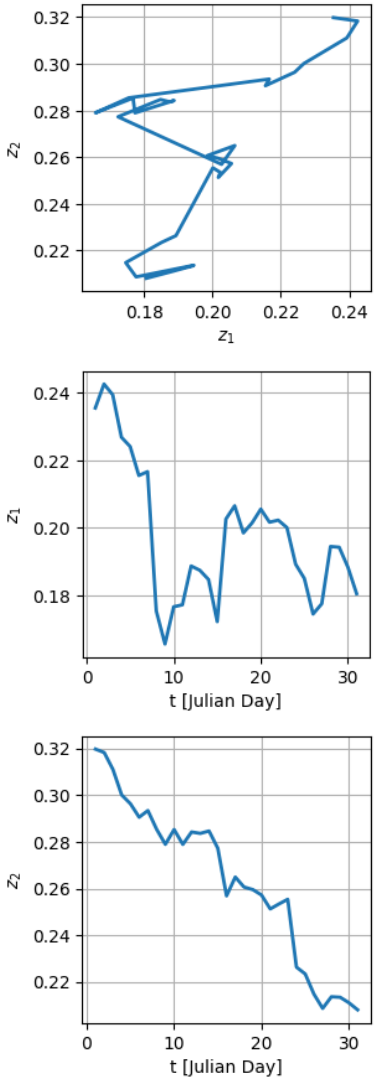


Figure 9. Evolution of the first two components of the reduced-order state $\mathbf{z}$, associated to January 2000.

[5]We restrict the analysis to a fixed altitude for representability; the output of the autoencoder is nevertheless a 3D tensor reconstructing the density field on the altitude interval.

A visual similarity between the first plot in Figure 9 and a two-dimensional Lévy flight can be detected. This may be an additional confirmation of the underlying chaotic nature of the dynamics of the density field: such random process is in fact strongly related to the motion associated to turbulent flows [38].

It should be clear how the trajectories given in Figure 9 define a set of data to be used in the reconstruction of a governing differential equation. Moreover, making use of state-of-the-art space weather forecasting techniques [39], it is possible to extrapolate results in the future, combining the grey-box dynamical model associated to the reduced-order state compressed by the autoencoder with solar and geomagnetic activity inputs associated to future times.

During the training process, different batch sizes have been considered to investigate the convergence behaviour; in Figure 10 it is shown how a bigger batch size leads to smoother convergence.

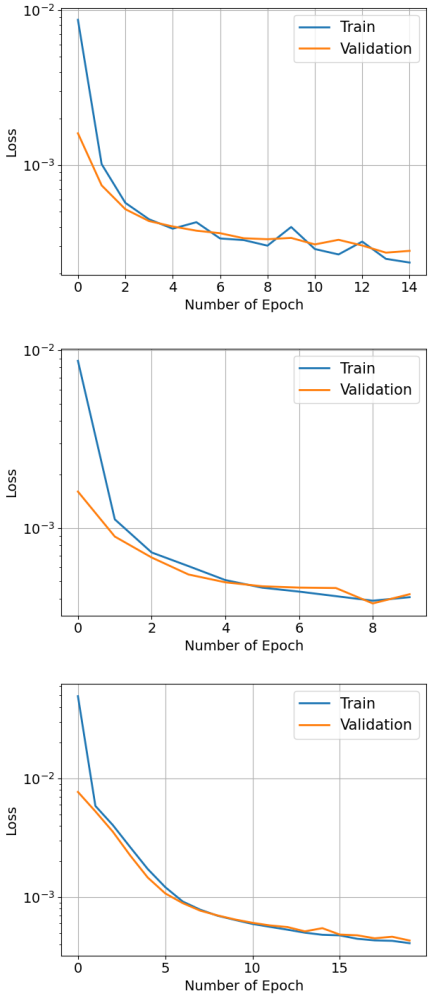


Figure 10. Loss history associated to training. Batch size 8, 16, 50.

4. DATA-DRIVEN RECONSTRUCTION OF GOVERNING EQUATIONS

In analogy with what has been shown with the Lorenz system, the assumption here is that the density field is governed by an ordinary differential equation associated to the reduced-order representation of the system. The prediction of the density field is therefore performed using its compressed representation, not the original state.

In other words, with the proposed methodology, once the reduced order representation of the density field is obtained, it is possible to construct an explicit differential equation governing the evolution of the reduced-order state, which is associated to the evolution of the thermospheric density field: reduced-order model data of the autoencoder can be used to generate expressions and obtain best-fitting differential equations associated to the autoencoder's bottleneck governing the spatial and temporal evolution of the density field.

Previous works aiming at modelling the evolution of the thermospheric density field made use of Dynamic Mode Decomposition ([12], [11]). While linearity may be a desirable features in some contexts, it is a big limitation in order to estimate the "true" dynamics (i.e., in order to best fit the available state observations). In fact, we aim at reconstructing a more general formulation of the dynamics, in which the right hand side is characterized by nonlinear functions, and in which the interplay between the state of the system and control inputs is not neglected; the ability to capture non-linearities in the dynamics is particularly relevant during storms.

Symbolic Regression leads to nonlinear expressions: with such technique, data are fitted using compact, closed-form analytical expressions [21]. Making use of Genetic Programming, expressions are generated to obtain a differential equation fitting (reduced-order model) data.

Symbolic Regression can be coupled with Sparse Regression to reduce computational cost and avoid overfitting. Parsimonious models are constructed using sparsity-promoting techniques [22]. In Figure 11 the relation between the use of autoencoders and the regression part of the algorithm is shown: the latent representation of the given data are used to construct best-fitting differential equations, which can be expressed as a linear combination of nonlinear functions associated to Symbolic Regression (Equation (9)).

It is worth underlying that, while it is possible to make use of Deep Learning and Neural Networks for the reconstruction of differential equations as well [42, 36, 31], this approach is more prone to overfitting, compared with the use of Symbolic Regression.

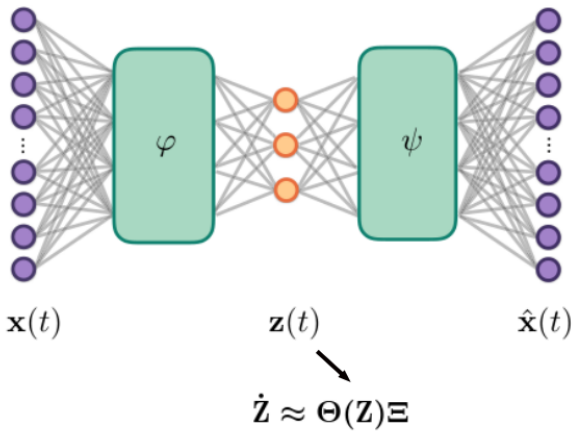


Figure 11. Relation between the latent state representation of the autoencoder [7] and the observable matrix $\Theta(\mathbf{Z})$, defined by Symbolic Regression, used to perform nonlinear parameter optimization and Sparse Regression.

$$\Theta(\mathbf{Z}) = \left[\quad \right] \tag{9}$$

5. CONCLUSIONS AND RECOMMENDATIONS

Convolutional autoencoders constitute a promising non-linear reduced-order modelling technique, to be used in the context of atmospheric density modelling. Moreover, Deep Symbolic Regression can be used to obtain an explicit non-linear differential formulation of the governing equations.

Once the explicit, interpretable differential formulation of the system is available, i.e. both the one of the spacecraft and the one associated to the reduced order model of the density field, it is possible to:

- make use of available tracking data to perform real-time calibration of the density field, e. g. via a Kalman Filter.
- implement Uncertainty propagation techniques, e. g. Polynomial Chaos Expansion, to take into account the non-Gaussianity of the probability density functions, leading to an accurate computation of the probability of impact. In this context, the use of Symbolic Regression enables one to also take into account the uncertainty associated to the reduced-order model dynamics parameters, leading to the state uncertainty of a spacecraft of interest. This additional source of uncertainty can be used to refine the computation of the probability of collision.

ACKNOWLEDGEMENT

This work was supported through the MSCA ETN Stardust-R grant agreement number 813644.

The SET HASDM density data are provided for scientific use courtesy of Space Environment Technologies.

Matteo Manzi would like to thank Davide Bagheri and Victor Rodriguez-Fernandez.

DATA AND SOURCE CODE

The Python Implementation developed in the context of this work will be made available at `https://github.com/strath-ace/smart-ml` soon after the submission of this paper.

REFERENCES

1. Bowman B. R., and Tobiska W. K., and Marcos F., and Huang C., and Lin C., and Burke W. (2008). A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices. *Preceedings of the AIAA/AAS Astrodynmics Specialist Conference*. 10.2514/6.2008-6438
2. Bruinsma S., (2015). The DTM-2013 thermosphere model, *J. Space Weather Space Clim.*, 10.1051/swsc/2015001
3. Brunton S. L., and Noack B. R., and Koumoutsakos P., (2020). Machine Learning for Fluid Mechanics, *Annual Review of Fluid Mechanics*, **52**(1), 477–508. 10.1146/annurev-fluid-010719-060214
4. Bukka S. R., and Gupta R., and Magee A. R., and Jaiman R. K. (2021). Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models, *Physics of Fluids*, **33**(1), 13601. 10.1063/5.0030137
5. Bukka S. R., and Magee A. R., and Jaiman R. K. (2020). Deep Convolutional Recurrent Autoencoders for Flow Field Prediction, *arXiv*, 2003.12147
6. Bussy-Virat C. D., and Ridley A. J., and Getchius J. W. (2018). Effects of Uncertainties in the Atmospheric Density on the Probability of Collision Between Space Objects, *Space Weather*, **16**(5), 519–537. 10.1029/2017SW001705
7. Champion K., and Lusch B., and Kutz N., and Brunton S. (2019). Data-driven discovery of coordinates and governing equations, *Proceedings of the National Academy of Sciences*, 10.1073/pnas.1906995116
8. Erichson N. B., and Muehlebach M., and Mahoney M. W. (2019). Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction, *arXiv*, 1905.10866

9. Fukami K., and Nakamura T., and Fukagata K., (2020). Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, *Physics of Fluids*, **32**(9), 95110. 10.1063/5.0020721

10. Gondelach D., and Linares R., (2019). Atmospheric Density Uncertainty Quantification for Satellite Conjunction Assessment, *arXiv*, 1912.01069

11. Gondelach D., and Linares R. (2020). Real-Time Thermospheric Density Estimation via Two-Line Element Data Assimilation, *Space Weather*, 10.1029/2019SW002356

12. Gondelach D., and Linares R. (2021). Real-Time Thermospheric Density Estimation Via Radar And GPS Tracking Data Assimilation, *Space Weather*, 10.1029/2020SW002620

13. Goodfellow I., and Bengio Y., and Courville A. (2016). Deep Learning, *MIT Press*

14. Gron A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, *O'Reilly Media, Inc.*, 10.5555/3153997

15. Hasegawa K., and Fukami K., and Murata T. et al. (2020). Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes, *Theoretical and Computational Fluid Dynamics*, **34**, 367–383. 10.1007/s00162-020-00528-w

16. Jacchia L. G. (1970). New static models of the thermosphere and exosphere with empirical temperature profiles. *SAO Special Report*, 313.

17. Kutz J. N., and Brunton S. L., and Brunton B. W., and Proctor J. L. (2016). Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. 10.1137/1.9781611974508

18. Licata R., and Mehta P. (2020). Physics-informed Machine Learning with Autoencoders and LSTM for Probabilistic Space Weather Modeling and Forecasting, *17th Conference on Space Weather - 100th AMS Annual Meeting*, 10.13140/RG.2.2.17039.74401

19. Licata R., and Mehta P., and Tobiska K. (2020). Data-Driven HASDM Density Model using Machine Learning, *Earth and Space Science Open Archive*, 10.1002/essoar.10505213.1

20. Majida A. (2012). Challenges in climate science and contemporary applied mathematics, *Communications on Pure and Applied Mathematics*, **65**, 920–948

21. Manzi M., and Vasile M., (2020). Discovering Unmodeled Components in Astrodynamics with Symbolic Regression, *IEEE Congress on Evolutionary Computation*, 10.1109/CEC48606.2020.9185534

22. Manzi M., and Vasile M., (2020). Orbital Anomaly Reconstruction Using Deep Symbolic Regression, *71st International Astronautical Congress*

23. Maulik R., and Lusch B. and Balaprakash P. (2021). Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, *Physics of Fluids*, **33**(3), 37106. 10.1063/5.0039986

24. Mehta P., and Linares R. (2017), A methodology for reduced order modeling and calibration of the upper atmosphere, *Space Weather*, 10.1002/2017SW001642

25. Mehta P., and Linares R., and Sutton E. (2018), A Quasi-Physical Dynamic Reduced Order Model for Thermospheric Mass Density via Hermitian Space-Dynamic Mode Decomposition, *Space Weather*, 10.1029/2018SW001840

26. Mehta P., and Linares R. (2020), Real-Time Thermospheric Density Estimation from Satellite Position Measurements, *Journal of Guidance, Control, and Dynamics*, **43**(9), 1656–1670, 10.2514/1.G004793

27. Muelhaupt T. J., and Sorge M. E., and Morin J., and Wilson R. S. (2019), Space traffic management in the new space era, *Journal of Space Safety Engineering*, **6**(2), 80–87. 10.1016/j.jsse.2019.05.007

28. Murakami D. D., and Nag S., and Lifson M., and Kopardekar P. H. (2019). Space Traffic Management with a NASA UAS Traffic Management (UTM) Inspired Architecture, *AIAA Scitech 2019 Forum*, 10.2514/6.2019-2004

29. Picone J. M., and Hedin A. E., and Drob D. P., and Aikin A. C. (2002). NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues, *Journal of Geophysical Research: Space Physics*, **107**(A12), 1468. 10.1029/2002JA009430

30. Qian L., and Burns A., and Emery B., and Foster B., and Lu G., and Maute A., and Richmond A., and Roble R. G., and Solomon S., and Wang W. (2013). The NCAR TIE-GCM: A community model of the coupled thermosphere/ionosphere system, *Geophysical Monograph Series*, **201**, 73–83. 10.1029/2012GM001297

31. Rackauckas C., and Ma Y., and Martensen J., and Warner C., and Zubov K., and Supekar R., and Skinner D., and Ramadhan A., Edelman A. (2020), Universal Differential Equations for Scientific Machine Learning, *Research Square*. 10.21203/rs.3.rs-55125/v1

32. Ramezanian D., and Nouri A. G., and Babaee H. (2021). On-the-fly Reduced Order Modeling of Passive and Reactive Species via Time-Dependent Manifolds, *arXiv*, 2101.03847

33. Reiland N., and Rosengren A. J., and Malhotra R., and Bombardelli C. (2021), Assessing and minimizing collisions in satellite mega-constellations, *Advances in Space Research*, 10.1016/j.asr.2021.01.010

34. Ridley A. J., and Deng Y., and Tóth G. (2006). The global ionosphere–thermosphere model, *Journal of Atmospheric and Solar-Terrestrial Physics*, **68**(8) 839–864. 10.1016/j.jastp.2006.01.008

35. Rowley C. W., and Colonius T., and Murray R. M. (2004). Model reduction for compressible flows using POD and Galerkin projection, *Physica D: Nonlinear Phenomena*, **189**, 115–129. 10.1016/j.physd.2003.03.001

36. Shankar V., and Portwood G., and Mohan A., and Mitra P., and Rackauckas C., and Wilson L., and Schmidt D., and Viswanathan V. (2020). Learning nonlinear spatio-temporal dynamics with convolutional

Neural ODEs, *Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020)*

37. Shelton C. T., and Junkins J. L. (2019), Probability of collision between space objects including model uncertainty, *Acta Astronautica*, **155**, 462–471. 10.1016/j.actaastro.2018.11.051

38. Shlesinger, M. and Klafter, J., West, B. (1986), Levy walks with applications to turbulence and chaos, *Physica A: Statistical Mechanics and its Applications*, **140**, 212–218. 10.1016/0378-4371(86)90224-4

39. Stevenson E., and Rodriguez-Fernandez V., and Minisci E., and Camacho D. (2020), A Deep Learning Approach to Space Weather Proxy Forecasting for Orbital Prediction, *71st International Astronautical Congress*

40. Storz M. F., and Bowman B. R., and Branson J. I., and Casali S. J., and Tobiska W. K. (2005). High Accuracy Satellite Drag Model (HASDM), *Advances in Space Research*, **36**(12), 2497–2505

41. Tobiska K., and Bowman B., and Bouwer D. et al. (2021). The SET HASDM density database, *Space Weather*, **19** 10.1029/2020SW002682

42. Turner H., and Zhang M., and Gondelach D., and Linares R. (2020). Machine Learning Algorithms for Improved Thermospheric Density Modeling, *Dynamic Data Driven Applications Systems*, 10.1007/978-3-030-61725-7_18

43. Xie X., and Nolan P., and Ross S., and Mou C. and Iliescu T. (2020). Lagrangian Reduced Order Modeling Using Finite Time Lyapunov Exponents, *Fluids*, 10.3390/fluids5040189