DOI: 10.13009/EUCASS2022-6137

Machine Learning techiques applied to space objects uncontrolled re-entry predictions

Cpt. Alessandro Panico[⋆#†] *and Pierluigi Di Lizia*[⋆]
[⋆]*Department of Aerospace Science and Technology, Politecnico di Milano*
Via La Masa 34, 20156 Milano, Italy
[#]*Italian Air Force, Flight Test Wing*
Via di Pratica di Mare, 45, 00071, Pomezia (RM), Italy
alessandro.panico@polimi.it† · pierluigi.dilizia@polimi.it
[†]Corresponding author

Abstract

Space debris overpopulation and the increased number of uncontrolled atmospheric re-entry is fostering the agencies and the operational centres to investigate modern techniques to deal with uncertainties and providing more reliable predictions and services. This paper presents a Machine Learning approach to characterize the re-entry of space objects, identifying similar conditions in the historical data and provide re-entry windows up to 45 days in advance. First, a statistically relevant database is collected throughout publicly available sources. Data are pre-processed and then provided to several architectures and configurations for validation and test. The final results are finally compared with current scenarios that are handled with conventional techniques by the Italian SST Operations Centre (ISOC).

1. Introduction

Over the decades, Space Situational Awareness (SSA) has been growing in importance and public attention as thousands of civil and military assets populate the orbits around the Earth. Their survivability depends primarily on the capacity of detecting, identifying and handling potential risks or threats.[1] The prediction of the re-entry window is still an open issue since current algorithms have limited accuracy for long period estimations, especially for the objects poorly characterized and/or moving on highly elliptical orbits. The main approaches to deal with this problem are based on numerical propagation and conventional statistical regression. The numerical propagation includes all the relevant perturbations: zonal and tesseral harmonics of the geo-potential, third body attraction of the moon and the sun, solar radiation pressure, aerodynamic drag and solar activity.[2–4] The ideal conditions to effectively propagate the object and estimate the re-entry window is to have a deep understanding of its physical properties and to collect several observations to determine the evolution of the orbital parameters over time. Within these conditions, the final window can be found with high accuracy, having slight variations of the epoch only during the last days and a progressive reduction of the uncertainty.[5] Space debris and objects orbiting in highly elliptical orbits may not meet these requirements. Indeed, the characterization of the former depends on the specific genesis, but some features like size, shape, mass and the value of the exposed area of debris are usually unknown and might be estimated with regression techniques through a dedicated observation campaign.[6,7] However, limited measurements related to any highly elliptical orbiting object are typically acquired. This is especially true when their orbits show significant variations, which make medium-long term predictions inaccurate and consequently the re-entry window usually converges a few orbits before the actual re-entry epoch.

The description of the problem addressed by this paper is presented and discussed in Section 2, showing the current state of the art and the potential applications. Since the solution proposed is based on a data driven approach, an analysis of the modern architectures and solution is presented in Section 3.

Section 4 presents the activities carried out to collect, filter and pre-process the datasets, the derivation of the input-output couples, the analysis of the network architectures and the configuration parameters.

All the data used for this work were downloaded from publicly available sources and a conventional training-validation dataset split was adopted to verify the goodness of the model obtained. The results obtained with the trained networks are summarized in Section 5 and specific case scenarios are compared with official estimations carried out by the United States Space Command (USSPACECOM) in the framework of the publicly available *spacetrack.com* service, showing

DOI: 10.13009/EUCASS2022-6137

a tendency to reduce the error in medium-long term prediction (i.e., from 15 to 45 days before the real re-entry epoch) with a significant reduction in the computational time.

2. Problem Statement

As described in Section 1, the prediction of the re-entry was conventionally addressed as the time needed by the space object to reach an altitude equal to 80 km, an experimental limit in which no space object can orbit around the Earth. Dynamic Models can predict this event providing atmospheric and orbital parameters, such as the mean motion, the ballistic coefficient, the air density, the space activity and propagate the orbit until the threshold is reached.[2–4] Typical statistical approach consists in the determination of the ballistic coefficient by processing a series of Two Line Elements (TLE) as observations. Then, using the latest TLE available and the optimized parameters, orbit propagation is performed until an altitude of 80 km is reached.[6,7] The position vector **r** is computed by integrating the following equation of motion:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a_E} + \mathbf{a_D} + \mathbf{a_S} + \mathbf{a_M} + \mathbf{a_{SRP}} \tag{1}$$

where μ is the gravitational parameter, $\mathbf{a_E}$ is the disturbance due to the Earth gravitational model, $\mathbf{a_D}$ is the atmospheric drag, $\mathbf{a_S}$ is the third body gravitational force (luni-solar attraction), $\mathbf{a_M}$ is the magnetic force and $\mathbf{a_{SRP}}$ is the solar pressure.

Despite the model complexity, the final orbits can be significantly affected by the satellite rotational dynamics, inaccurate orbital parameters, variability in the atmospheric parameters, variability in the atmospheric density and in the magnetic field, especially in high solar activity conditions.

The main idea behind a data driven approach is that similar behaviours can be found in several objects, especially when they have similar features, such as rocket bodies or high elliptical objects. Fig. 1 shows the comparison between a classical and a data driven approach to this problem.

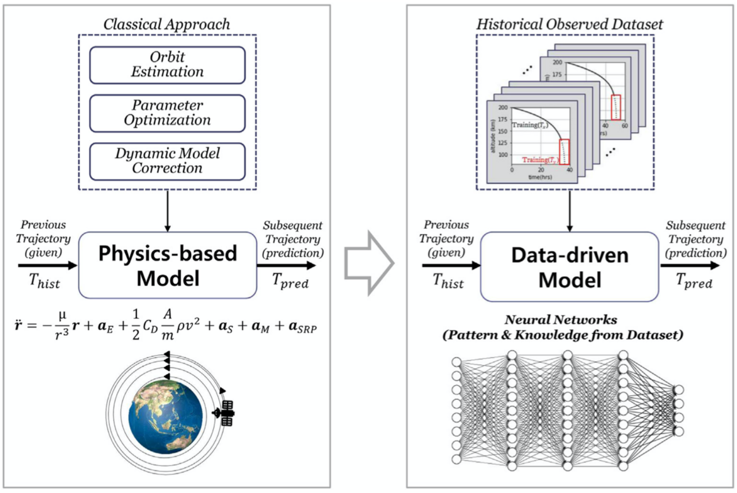

Classical Approach
Orbit Estimation
Parameter Optimization
Dynamic Model Correction
Previous Trajectory (given)
T_{hist}
Physics-based Model
Subsequent Trajectory (prediction)
T_{pred}
$\ddot{r} = -\frac{\mu}{r^3}r + a_E + \frac{1}{2}C_D\frac{A}{m}\rho v^2 + a_S + a_M + a_{SRP}$
Historical Observed Dataset
200
175
150
125
100
altitude (km)
Training(T_x)
Training(T_y)
0
20
40
60
time(hrs)
Previous Trajectory (given)
T_{hist}
Data-driven Model
Subsequent Trajectory (prediction)
T_{pred}
Neural Networks
(Pattern & Knowledge from Dataset)

Figure 1: Classical vs Data Driven Approach[8]

A possible strategy to handle the uncertainty of re-entries is given by the Recurrent Neural Network (RNN).[8] The idea behind this approach is to consider the history of the TLE of a specific object and train the neural network consequently, with an approach called Sequential to Sequential (Seq2Seq), in which an input time-series is transformed into an output sequence. The results shown in this study are remarkable, with high confidence in the re-entry window assessment. However, the study is limited to nearly circular orbits and it is mainly focused on the final re-entry predictions, with a mean altitude ranging from 130 to 160 km and a residual lifetime that is typically below 24 hours.

The approach investigated in this study in mainly based on medium-term predictions and takes advantage of a larger and more general dataset. The main goal is to span the entire catalogue and to consider the last available TLE in order to rapidly identify the objects that should be included in the 30 days list, requiring specific sensor tasking campaign since the re-entry epoch is rapidly approaching. So that, instead of using a RNN, a conventional Artificial Neural Network (ANN) is considered, providing a [n x 1] array with all the relevant pieces of information available at the TLE epoch and obtaining the estimated residual lifetime as network output.

DOI: 10.13009/EUCASS2022-6137

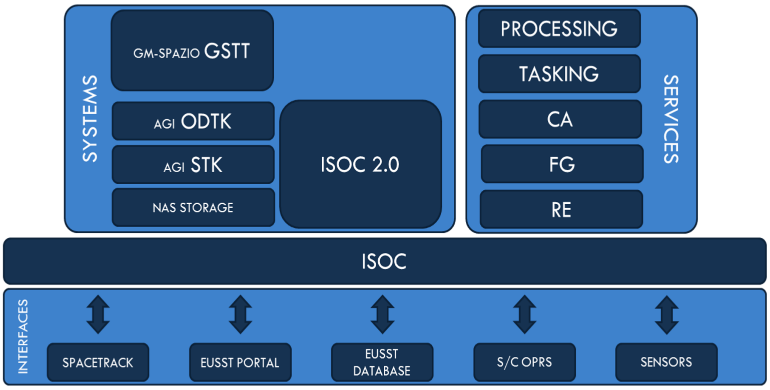


Figure 2: ISOC architecture

2.1 Italian SST Operations Centre

Since 2014, the Italian Air Force (ItAF) has been involved in the Space Surveillance & Tracking (SST) domain, providing the relative services in the framework of the European Union - Space Surveillance & Tracking (EU-SST), in partnership with the Italian Space Agency (ASI) and the National Astrophysical Institute (INAF). It consisted in the creation of a Command and Control (C2) system, the Italian SST Operations Centre (ISOC), that was made to plan and operate a variety of national sensors (optical, radar, laser), to process incoming data (internal and external) and providing the services required by the European Community:

- Re-entry: analysis of large uncontrolled objects atmospheric re-entry, in order to identify possible area of impact and alert the population;
- Conjunction: identification of possible collisions, analysis of the probability and identification of the mitigation strategies by proposing feasible manoeuvers to satellite operators.
- Fragmentation: parent identifications and data correlations, fragment clouds propagation for medium-long term impact analysis.

Over the decades, the consciousness of the system has increased, as long as the collaborations with national industries and academics entities. Originally, most of the tools were Commercial Off The Shelf (COTS) or based on commercially available routines. Nowadays, ISOC is evolving towards a complete suite (whose high level architecture is depicted in Fig. 2), a sophisticated environment in which COTS and proprietary algorithms live together due to the modern micro-services architecture that was realized in collaboration the national partners involved. in the project. The next objectives rely to the extension of the conventional SST applications to the entire space domain, to protect and support the assets, improving the terrestrial applications, automatizing the operational and information distribution flows and the international co-operations as well.

3. Overview of Machine Learning

The Machine Learning (ML) approach is considered a disruptive technology for several engineering branches. It is essentially based on the key idea that computer algorithms can emulate human learning, by means of a process that improves the performance of a very specific task with the experience.[9] However, the theoretical fundamentals were originally defined in the late '40s[10] with the investigation of the possible similitude between human intelligence and electronic computers[11] and the evolution of more and more complex uses of the language by the computers.[12] These pioneer projects were too premature because there were not enough computational power to support them. Only the tremendous technological innovation that came starting from the last decade of XX century gave the necessary conditions for the development of ML applications for real life problems (virus-malware detection, fraud operations, prediction of trends, autonomous navigation, decision making systems...)[13] Most of the applications rely to the general category of the regression/classification problems, resulting in a linear combination of non-linear basis functions $\phi_j(x)$:

$$y(\mathbf{x}, \mathbf{w}) = h\Big(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x})\Big) \tag{2}$$

DOI: 10.13009/EUCASS2022-6137

in which $h(\cdot)$ is either the identity function (regressions) or a non-linear function (classifications), e.g. logistic sigmoid, hyperbolic tangent, softsign, rectified tanh, rectified linear, etc. The parameter w_j represents the normalized weight allocated to each of the M functions $\phi_j(\mathbf{x})$.[14]

ANN can be interpreted as a combination of several nested functions, that represent subsequent layers of the neural network and have the general structure of Eq. 2 and may be represented visually as in Fig. 3.

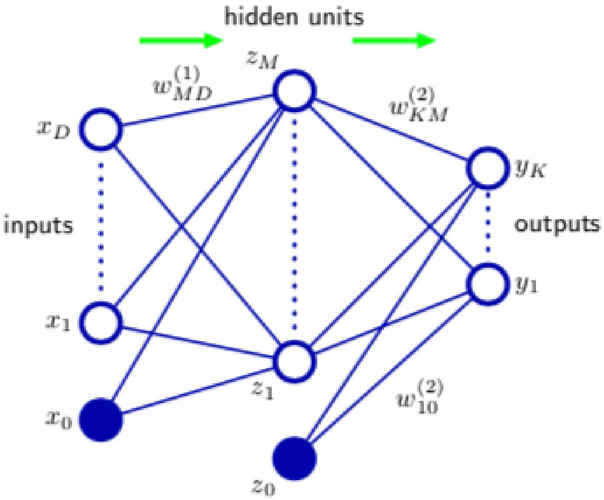


Figure 3: Feedforward Neural Network Diagram[14]

From a semantic point of view, each layer extracts specific features from data, with higher order information the deeper the network. Different results can be obtained with alternative architectures, like changing the number of hidden layers, the number of neurons ant their connections, the chosen activation functions. However, once the network hyper-parameters are defined, the weights are computed by means of a training process, that consists of computing the residual error with known input-output couples (supervised training). The typical cost function is the L^2 norm, defined as follows:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} |\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n|^2 \tag{3}$$

with $E(\mathbf{w})$ that is the resulting error in the regression/classification task, as obtained with the weight tentative combination ($\mathbf{w}$). Intuitively, $\mathbf{t}$, $\mathbf{y}$ and $\mathbf{s}$ are the vectors representing the truth, the actual output and the network input respectively. Back-propagation is used to compute the error gradient by reversing the network,[15] so that the weights can be tuned with an optimization algorithm to minimize the cost function in Eq. 3 (gradient descend, stochastic gradient descend, AdaGrad, RMSProp or ADAM).[16,17]

Among the numerous ML architectures, Convolutional Neural Network (CNN) are specifically designed to manage data with a grid-topology, in which input data have an intrinsic connection and are not independent, like time series, images or multidimensional maps. These networks can handle huge amounts of data because they can simplify the problem, by squeezing data, extracting the main features and therefore reducing the problem complexity.[18] This approach reduces the computational power required to train the network due to the introduction of the "sparce" connectivity, meaning that only adjacent pixels (or subsequent samples or frames for time series) have a statistical and physical relationship. These kind of networks have shown tremendous results, especially in image classification problems.[19–21] Another solution for topological related problems is provided by RNN, in which feedback connections throughout the net are provided.[22] Moreover, this architecture can even be used for self-feeding applications, like propagation dynamics,[23] music generation and encoding,[24,25] sentiment classification.[26]

Various ML algorithms and tools have been developed and used in a wide variety of space applications, such as automatic space objects characterization,[27,28] planning and optimization of spacecraft trajectories,[29,30] orbit determination[31,32] and spacecraft propagation.[33–37]

DOI: 10.13009/EUCASS2022-6137

4. MACHINE LEARNING ARCHITECTURES TESTED

4.1 Database

The first step of the problem was gathering data and generating a suitable and representative database to train the neural network. Data were collected in the framework 2010-2020, associating a unique ID, based on the NORAD identification number, for each decayed object. A dedicated "Decayed Object" class has been programmed in order to import and organize all the relevant data coming from different public sources. The process is summarized in Fig.4.

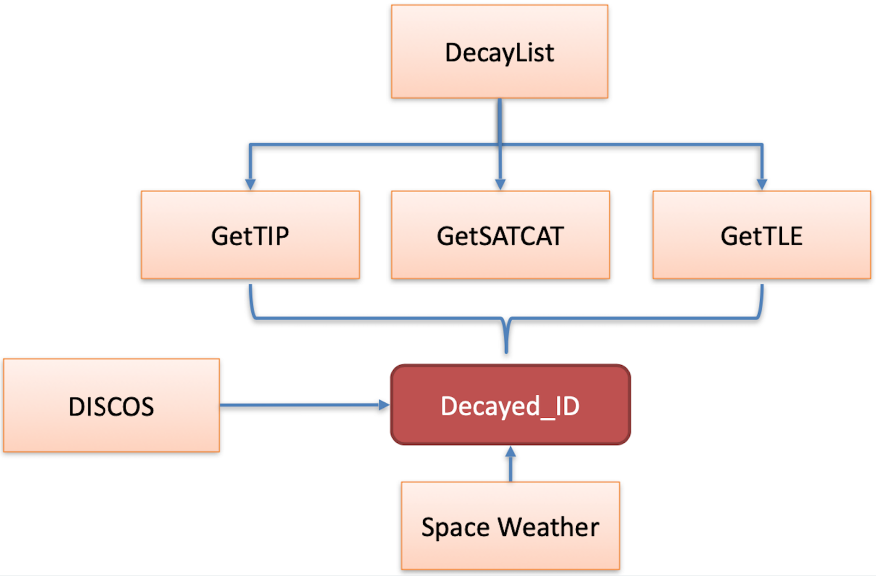


Figure 4: Dataset Collection

The class handles each space object separately, populating its tags with orbital data, Space Situational Awareness (SSA) messages and object related features (e.g. shape, nationality, launching year..), as shown in the list below:

- the decay epochs
- the list of the TLE and the Tracking and Impact Prediction (TIP) messages published by the 18^{th} Space Defence Squadron (SDS) of the USSPACECOM on *spacetrack.org* up to 45 days before the declared re-entry epoch.
- the physical data that are disseminated on *spacetrack.org* and Database and Information System Characterising Objects in Space (DISCOS) catalogues.
- the space weather data that are published on *celestrak.com*.

For completeness, Tab. 1 shows the extrapolated pieces of information from each data source. Most of them have a number representation, but a few ones are categorical (e.g. nationality, shape...). So that, the correlation between the spacecraft propagation and the launching state can be assessed and taken into account in the model, as well as the effect of complex shapes or satellite typology on uncontrolled re-entry.

Table 1: Class Data

Data Source	**Extrapolated Data**
spacetrack.org TLE	Epoch, Mean Motion, Eccentricity, Inclination, Radius of the Ascending Node, Argument of the Perigee, Mean Anomaly, B*,First Derivative of the Mean Motion, Second Derivative of the Mean Motion, Semi-Major Axis, Orbital Period, Apogee, Perigee

DOI: 10.13009/EUCASS2022-6137

Data Source	**Extrapolated Data**
spacetrack.org Catalogue	Decay Epoch, Object Nationality, Object Type, Launching State, Launching year, Launching Piece, RADAR Cross Section (RCS), Categorical RCS size
Space Weather	Geomagnetic Data (Planetary Range Index Sum, Average of the Planetary Equivalent Amplitude, Planetary Daily Character Figure), International Sun-Spots Number, Observed 10.7 cm Solar Radio Flux, Centered and Shifted Solar Radio Flux Prediction
ESA DISCOS	Minumum/Maximum/Average Cross Section, Object Mass, Object Dimensions (Length, Height, Depth), Categorical Object Type and Shape

A specific built-in function was then created to initialize the neural network training datasets. It consists of generating the input-output couples that will populate the training and validation database. Essentially, the operational process that is assumed for this application consists of verifying the re-entry epoch of a space object once a new TLE is received. So that, the built-in function explores the class of the object ID, collecting all the relevant data available at the TLE epoch. Three kinds of scenarios are considered, as shown in Tab. 2.

Table 2: Training Scenarios

Data	**Model A**	**Model B**	**Model C**
spacetrack.org TLE	X	X	X
spacetrack.org Catalogue	X	X	X
Space Weather		X	X
ESA DISCOS			X

In all cases, the output vector is a decimal number that represents the number of days up to the declared re-entry epoch. This number is obtained by the difference between the declared re-entry and the TLE epoch. This computation gives generality to the process that becomes independent from the time variable.
The dataset composition is described in Tab.3

Table 3: Dataset Composition

	TRAINING SET		VALIDATION SET		TOTAL DATASET	
Dataset	**Objects**	**Epochs**	**Objects**	**Epochs**	**Objects**	**Epochs**
Rocket Bodies	279	19.203	110	7.374	389	26.577
Payload	366	18.524	178	8.972	544	27.496
Debris	599	18.575	244	8.121	843	26.696
High Eccentricity	90	5.113	33	1.567	123	6.680
Total	1.244	56.302	532	24.467	1.776	80.769

4.2 Architectures

The architecture network adopted consists of a feed-forward neural network able to generate the regression model that identifies the re-entry epoch of a generic space object.
The number of inner layers as long as their depth are meta-parameters of the neural network. Usually, the deeper the network, the higher the accuracy, but the longer the training time. However, a proper balancing is required in any case, mainly to avoid local minima in the solution optimization process. Indeed, the back-propagation algorithm, that consists in the derivation of the model weights variations to minimize the problem cost function, is carried out by means of an optimization process and may stop prematurely because of a local minimum.
Since the scenarios indicated in Tab. 2 have different input size, i.e. the input vector has different number of elements depending on the kind of data used for the regression, the architecture is defined in terms of multiple of the input vector.

DOI: 10.13009/EUCASS2022-6137

The "standard" model assumed as benchmark for this analysis has the structure shown in Tab. 4. Throughout the fully connected network, each node of the layer *i* has a specific connection to each node of the layer *i + 1* that is activated by means of a Rectified Linear unit (ReLu) or Leaky ReLu function. This choice has been driven by the computational efficiency during the network training, that can be even six time faster with respect to conventionally used non-linear functions (tanh, softmax...)[19] The Leaky ReLu have been tested to avoid the occurrence of the dead neurons. It consists on introducing a slight inclination in the negative side of the curve to reduce the possibility of having 0 value as output. In terms of computational efficiency, this approach takes advantage of the combination of linear curves, being faster than conventional non-linear functions.[38]

Table 4: Standard Architecture

Layer	Model A	Model B	Model C
Input	20	26	35
Layer 1	20	26	35
Layer 2	20	26	35
Layer 3	40	52	70
Layer 4	40	52	70
Layer 5	80	104	140
Layer 6	80	104	140
Layer 7	40	52	70
Layer 8	40	52	70
Layer 9	20	26	35
Layer 10	20	26	35
Output	1	1	1

Further configurations have been tested, including an architecture with two separate networks to compute both the prediction and the associated variance and additional preliminary tests in which the effect of depth and width variations were verified.

The final activation function, that is basically aimed at providing the regression outcome, i.e. the re-entry prediction, has been tested with either ReLu or a linear function, to consider positive only or positive and negative results respectively. The training approach consisted of randomly dividing the dataset, so that 70% of data can be used to train the neural network, whereas the other 30% can be used for the model validation. However, the division was carried out in terms of space objects instead of epochs/TLE in order to make the datasets statistically independent. Otherwise, the same object could have been used both for training and validation at different instants of time, making the neural network over-fitted for the categorical data.

In terms of pre-processing, the database was filtered to remove some errors in data. Specifically:

- some objects had TLE epoch after the declared time of re-entry, causing negative predictions in the algorithm. Sometimes the declared decay was clearly wrong (one year before/after the last available TLE) .
- some objects were declared re-entered even with very high orbits. All the objects with a perigee above an arbitrary limit of 250 km in the 45 days before the declared decay were not taken into account in this study.

Moreover, input vectors passed through a Gaussian Transformation (Quantile Transformer) to get rid of outliers and consider the 90^{th} percentile only. On the contrary, the output vector was normalized with a Min-Max Scaler between 0 and 45. This solution was preferred with respect to the conventional $0 - 1$ range to immediately visualize the error in terms of days.

The proposed approach can be improved with the quantification of the prediction uncertainty, such as modelling a mean-variance estimator.[39] To achieve this goal, the training dataset shall be divided in two equal parts, so that the former is used to train a neural network to estimate the decay epoch (mean estimator), whereas the latter can be devoted to forecast the solution uncertainty (variance estimator). The training dataset split was necessary to have stochastically independent trainings of the networks, keeping the validation dataset intact for the final testing phase. Once the mean network is fully trained, it is applied on the second training dataset to predict the decay epoch of the input objects. This operation in functional for the identification of the second training truth: the squared deviation becomes the target for the variance estimator. The specific cost function is hence modified as follows:

$$E(\mathbf{w}_2) = \frac{1}{2}\sum_{n=1}^{N} \left|\mathbf{y}_2(\mathbf{x}_n, \mathbf{w}_2) - \hat{\mathbf{e}}_n^2\right|^2 = \frac{1}{2}\sum_{n=1}^{N} \left|\mathbf{y}_2(\mathbf{x}_n, \mathbf{w}_2) - |\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n|^2\right|^2 \tag{4}$$

being equal to the application of the L2 norm to the predicted squared error $\hat{\mathbf{e}}^2$.

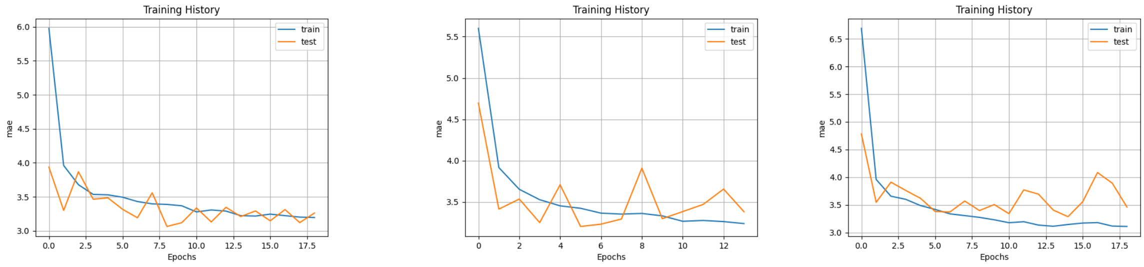
DOI: 10.13009/EUCASS2022-6137

5. Training Results

Training operations were carried out on a dual GPU workstation (NVIDIA RTX A5000) using the open source Deep Learning backend Tensorflow (version 2.4.0) and its high-level Python interface Keras (version 2.4.3).
The configuration adopted to initialize the networks, propagate and compute the errors, and finally derive the inner weights is showed in Tab.5. The loss function used for this study is the Mean Squared Error presented in Eq. 3. However, for validation purposes and to better visualize results, Mean Absolute Error was considered as well. Despite the maximum number of epochs was assumed equal to 100, trainings used to stop between 20 and 50 epochs because no relevant improvements were found during the network tests with the validation dataset (Early Stopping Strategy). The number of input elements randomly selected within the dataset to test the network and update the weight, i.e. the batch size, was set to 100, but it depends on the available memory and the system computational power. The higher the value the faster the training convergence in terms of number of epochs required to find the best solution. Finally, the optimization function was the Adaptive Moment Estimation (ADAM) with standard configuration values[16] as shown in Tab.5.
Figures 5-6 show the results of the Mean Absolute Error computed at each training epoch and the 5-95th percentile range of the propagation error with respect to the TLE epoch obtained by the analysis of the statistics computed with the global validation dataset. In Figures 7-10 the same neural network architecture was trained with specific clusters of data (debris, rocket bodies, payload, HEO), showing no relevant improvements.
In terms of input data, it seems that the prediction accuracy does not improve significantly by considering space weather (Model B) and DISCOS categorical data (Model C) and the range 5-95th percentile becomes slightly noisier as can be observed by the graphs. All the analysis show that the mean error is usually in the order of 2-4 days for 20-30 days propagations. On the contrary, longer propagations tend to underestimate the decay epoch. The noisy 5-95th percentile range for High Elliptical Orbits (HEO) objects is justified by the poor dataset available for training and especially for validation (only 33 objects).

Table 5: Training Configuration

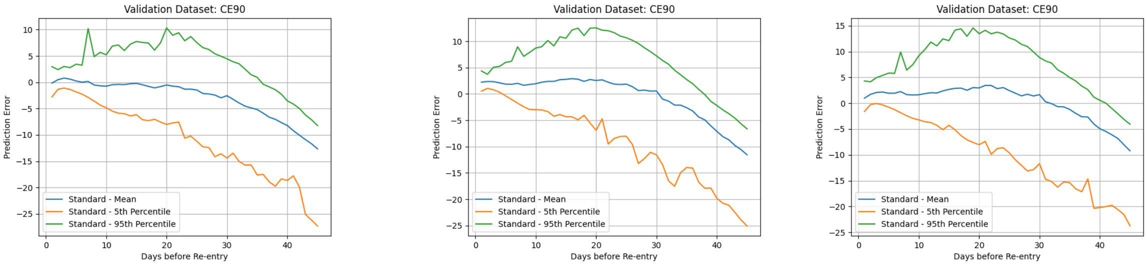
Training Configuration	Setting
Loss Function	Mean Squared Error
Validation Function	Mean Absolute Error
Max Training Epochs	100
Early Stopping Strategy	10
Batch Size	150
Dropout	20%
Optimizer	ADAM
Learning Rate	0.001
β (ADAM)	0.8
β_1 (ADAM)	0.9



Model A Model B Model C

Figure 5: Global Dataset: Mean Absolute Error vs Epochs

Besides, considering the topic from an operational point of view, *spacetrack.com* TIP messages alert the space operators community about imminent re-entry. These messages are usually disseminated about 4-5 days before the effective decay epoch. However, despite the re-entry windows declared on *spacetrack.com* are very accurate, there are some objects with premature alarms and large errors. Fig.11-12 show the results obtained by the application of the "Model

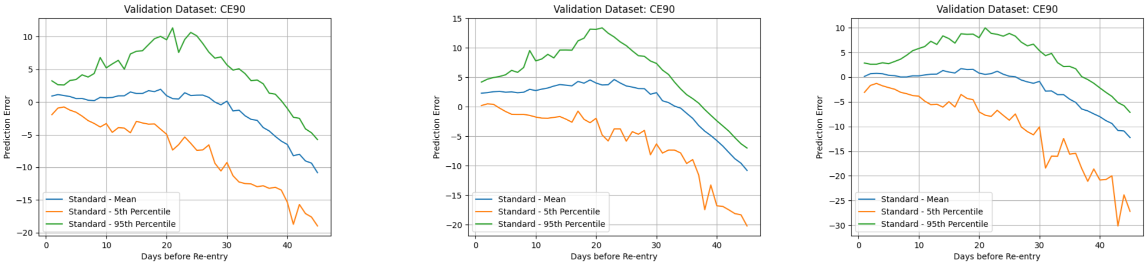


Model A

Model B

Model C

Figure 6: Global Dataset: CE90 Estimation

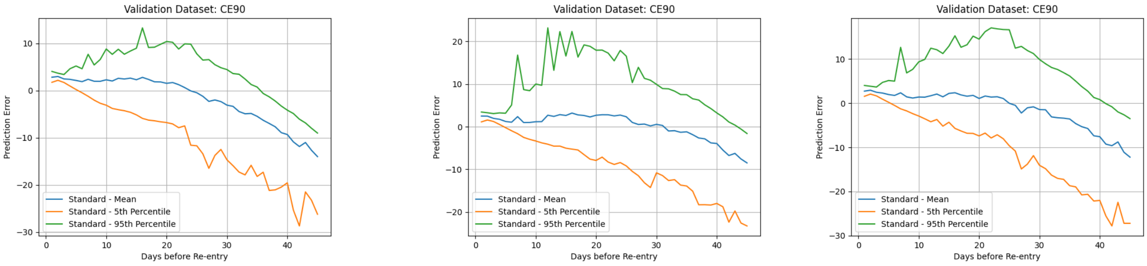


Model A

Model B

Model C

Figure 7: Debris: CE90 Estimation

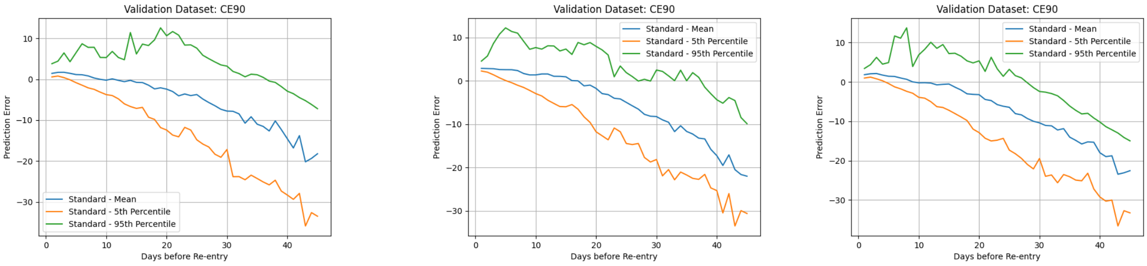


Model A

Model B

Model C

Figure 8: Rocket Bodies: CE90 Estimation



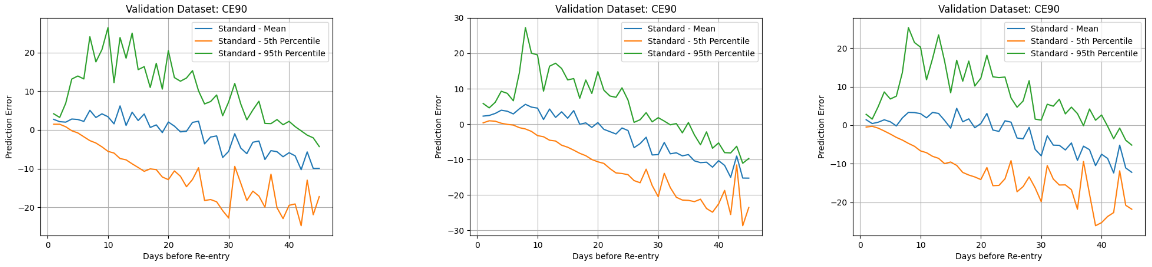
Model A

Model B

Model C

Figure 9: Payloads: CE90 Estimation

C" architecture. The tracked objects are two rocket bodies, that usually receive particular attention before decay as some components may pass through the atmosphere and impact the Earth surface. The x axis shows the real time difference between the TLE epoch and the decay epoch, whereas on y axis the model prediction is reported with the blue dots. Ideally, the dots should be aligned with the theoretical plain line. The further they are the higher the error.



Model A

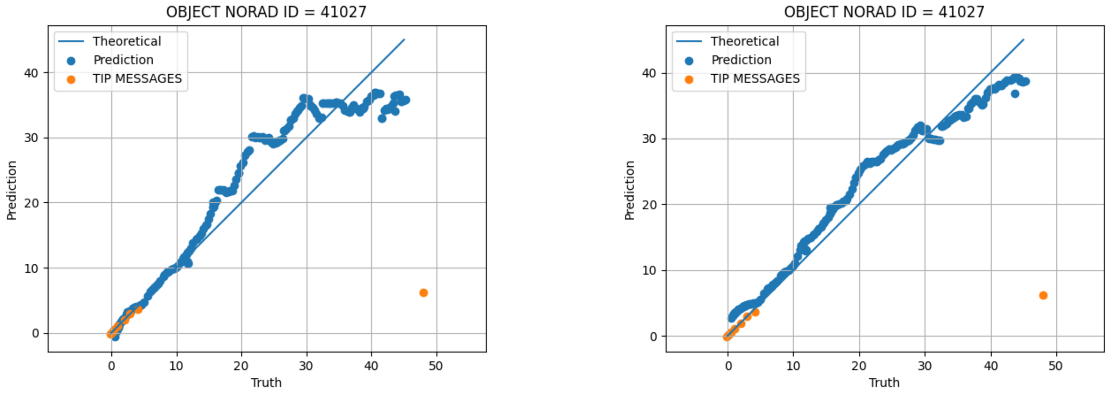
Model B

Model C

Figure 10: HEO: CE90 Estimation

The figures show also the TIP messages (orange dots), with very precise final predictions and some false alarms even 40 days before the actual object decay epoch. For these scenarios the models presented in this paper are more robust. In particular, the images presented side by side show the results based on different training datasets, specifically the global dataset and the rocket bodies cluster for Fig. 11 and also HEO for Fig. 12. Despite being very similar in terms of performance, a model trained on a very specific dataset gives better results with the respective inputs. Indeed, since the object 27412 have a starting eccentricity that is higher than 0.5 (value assumed as cut-off), the HEO model was tested as well (see Fig.12), being even more accurate then the other models for both short and medium-long term propagations, confirming that specific datasets may lead to better results for confined data clusters. Nevertheless, the application of the rocket body model on debris or payload input data might result in poor accuracy, as the model is too specific. Same inaccuracies are expected for HEO model with circular and stable orbits.

Figure 13 shows the orbital trend of the object 27412 and in particular the behaviour of the perigee history, that is quite common for HEO objects. As the TLE is an orbit approximation that depends on the initialization point and HEO orbit encounter very different disturbances from apogee to perigee, the real lowest altitude reached by object may be different. This is the situation that reasonably happened for the object 27412 with the perigee that seems to decrease fast at about 80 km about 35 days before decay and then peaking again above 100 km.



Global Dataset: Model C

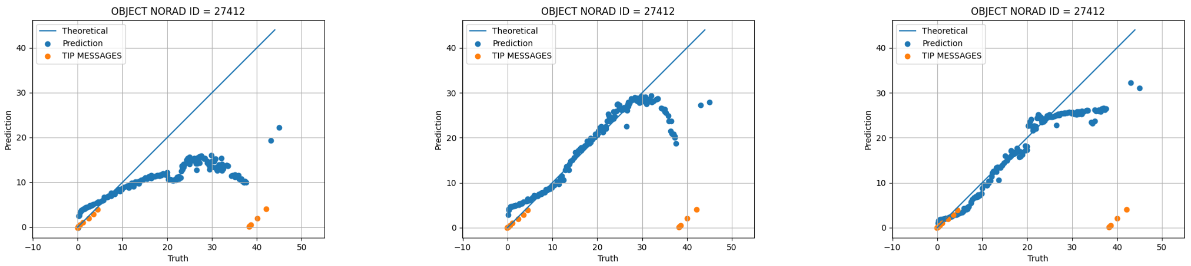
Rocket Body: Model C

Figure 11: NORAD ID: 41027 - CZ-4B Rocket Body

Besides, the application of the mean-variance architecture slightly decreases the performance for the pure estimation of the decay epoch, since the training dataset is smaller and less general. However, this is largely compensated by the possibility to provide both the prediction and the relevant confidence level, as depicted in Fig.14.

Finally, in terms of computational resources for operational use, these models offer significant improvements with respect to conventional deterministic algorithms. The SW routine AGI STK High Precision Orbit Propagator (HPOP), that can be considered a reference for re-entry assessments, takes 1-3 minutes for 20-40 days propagations on the same dual GPU workstation used for training the networks. It is worth to point out that performance can change significantly depending on the integration step and the propagation duration, but the order of magnitude is in terms of minutes. On the contrary, the proposed ML approach reduces the computational time of 3-4 orders of magnitude as shown in Tab.6.

DOI: 10.13009/EUCASS2022-6137

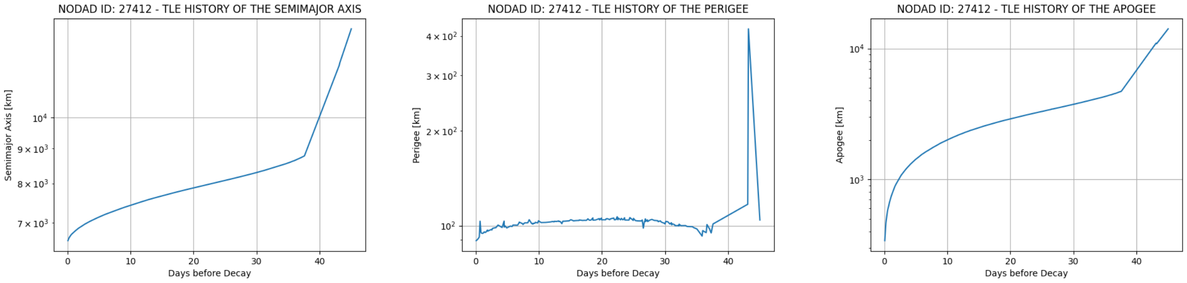


Global Dataset: Model C

Rocket Body: Model C

HEO Dataset: Model C

Figure 12: NORAD ID: 27412 - SL-6 Rocket Body (HEO)

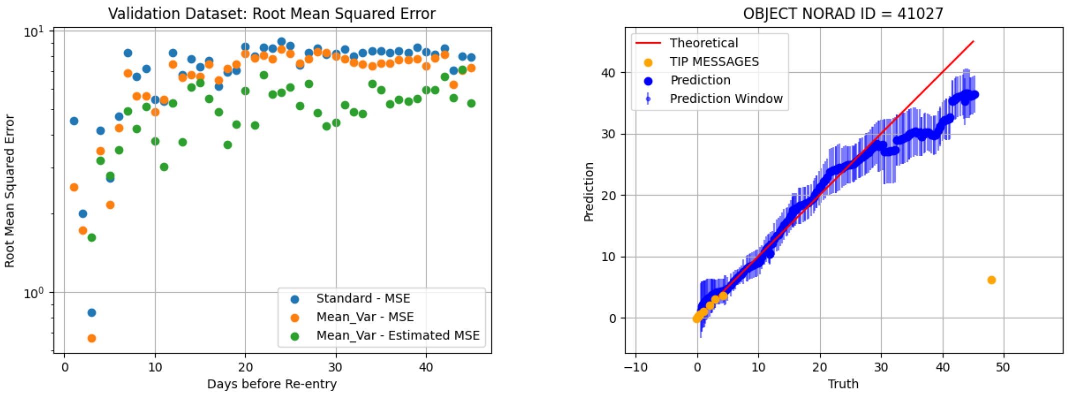


Semimajor axis

Perigee

Apogee

Figure 13: NORAD ID: 27412 - SL-6 Rocket Body (HEO) Orbital Trend



RMSE - Computed vs Estimated

Rocket Body: Mean-Variance Model C

Figure 14: Mean-Variance Model

Table 6: Model Computational Performance

Algorithm Step	**Performance**
Initializing Functions	11.96 s
Downloading & Loading Data	7.86 s
Inference	0.065 s

DOI: 10.13009/EUCASS2022-6137

6. Conclusions

The architectures tested in this research show that ML techniques can be applied to infer the decay epoch of re-entering objects. The ANN is a powerful tool because it requires a limited number of data inputs (just one TLE along with available information regarding space weather and general features of the space object) and can perform accurate medium-long term predictions, whereas reducing the computational time of 3-4 orders of magnitude.

The mean-variance estimator can also provide the confidence window associated to the re-entry epoch, with the intrinsic assumption of Gaussian error distribution. Further research may include a quantile estimator as well as other architectures that combine the confidence level directly in the outcome. Moreover, the ANN might be combined with multi-temporal approaches, e.g. including CNN or RNN to provide more robust estimations, especially for the last propagation hours.

The results obtained in this work suggest that the next-generation database scanning algorithms will likely take advantage of these ML approaches to provide a first guess of the re-entry window and to trigger a data acquisition process in order to refine the re-entry epoch and location predictions with the available deterministic and stochastic techniques.

References

[1] US DoD. Defence space strategy summary. Technical report, US Department of Defence, 2020.

[2] Carmen Pardini and Luciano Anselmo. Reentry predictions of three massive uncontrolled spacecraft. In *23rd International Symposium on Space Flight Dynamics*, 2012.

[3] Carmen Pardini and Luciano Anselmo. Re-entry predictions for uncontrolled satellites: results and challenge. In *6th IAASS Conference*, 2013.

[4] Carmen Pardini and Luciano Anselmo. Re-entry predictions of potentially dangerous uncontrolled satellite: Challenges and civil protection applications. In *Stardust Final Conference on Atsteroids and Space Debris*, 2016.

[5] Carmen Pardini and Luciano Anselmo. The uncontrolled reentry of progress-m 27m. *Journal of Space Safety Engineering*, 3(3):117–126, 2016.

[6] Alessandro Gasparetto, David J. Gondelach, Roberto Armellin, and Aleksander A. Lidtke. Ballistic coefficient estimation for reentry prediction of rocket bodies in eccentric orbits based on tle data. *Mathematical Problems in Engineering*, 2017.

[7] V. S. Yurasov, A. I. Nazarenko, K. T. Alfriend, and P. J. Cefola. Reentry Time Prediction Using Atmospheric Density Corrections. In D. Danesy, editor, *4th European Conference on Space Debris*, volume 587 of *ESA Special Publication*, page 325, August 2005.

[8] Okchul Jung, Jaedong Seong, Youyeun Jung, and Hyochoong Bang. Recurrent neural network model to predict re-entry trajectories of uncontrolled space objects. *Advances in Space Research*, 68(6):2515–2529, September 2021.

[9] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1 edition, 1997.

[10] B.G. Buchanan. A (very) brief history of artificial intelligence. *AI Magazine*, 26(4):53–60, 2005.

[11] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–60, 1950.

[12] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon. Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, 1955. Available online: `http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html`, Accessed on 2019/06/25.

[13] Stefan Strauß. From Big Data to Deep Learning: A Leap Towards Strong AI or "Intelligentia Obscura"? *Big Data and Cognitive Computing*, 2(3):16, 2018.

[14] C. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[15] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

DOI: 10.13009/EUCASS2022-6137

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] S. Ruder. An overview of gradient descent optimization algorithms, 2016. `https://arxiv.org/abs/1609.04747`, Accessed on 2019/07/20.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] K.-L Du and M.N.s Swamy. *Recurrent Neural Networks*, pages 337–353. 12 2014.

[23] A. Aussem, F. Murtagh, and M. Sarazin. Dynamical recurrent neural networks - towards environmental time series prediction. *International Journal of Neural Systems*, 6(2):145–170, June 1995.

[24] Lam Hoang. Applying and comparing 1d-cnn and bi-rnn architecture in music genre classification. 2021.

[25] Nicolas Mauthes. Vgm-rnn: Recurrent neural networks for video game music generation. 2021.

[26] Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, E. Cambria, and U. Rajendra Acharrya. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Gener. Comput. Syst.*, 115:279–294, 2021.

[27] Walter D. Bennette, Kayla Zeliff, and Joseph Raquepas. Classification of objects in geosynchronous earth orbit via light curve analysis. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, 2017.

[28] Roberto Furfaro, Richard Linares, and Vishnu Reddy. Shape Identification of Space Objects via Light Curve Inversion Using Deep Learning Models. In S. Ryan, editor, *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 17, September 2019.

[29] D. Izzo, D. Hennes, L. F. Simões, and M. Märtens. Designing complex interplanetary trajectories for the global trajectory optimization competitions. *Space Engineering*, pages 151–176, 2016.

[30] Alessio Mereta, Dario Izzo, and Alexander Wittig. Machine learning of optimal low-thrust transfers between near-earth objects. pages 543–553, 06 2017.

[31] S. Sharma and J. W. Cutler. Robust Orbit Determination and Classification: A Learning Theoretic Approach. *Interplanetary Network Progress Report*, 42-203:1–20, November 2015.

[32] Byoung-Sun Lee, Wonsul Kim, Junho Lee, and Yoola Hwang. Machine learning approach to initial orbit determination of unknown leo satellites. In *2018 SpaceOps Conference*, number 2566, 2018.

[33] Hao Peng and Xiaoli Bai. Improving orbit prediction accuracy through supervised machine learning. *Advances in Space Research*, 61(10):2628–2646, May 2018.

[34] Hao Peng and Xiaoli Bai. Artificial Neural Network-Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *Journal of Spacecraft and Rockets*, 55(5):1248–1260, September 2018.

[35] Bin Li, Jian Huang, Yanming Feng, Fuhong Wang, and Jizhang Sang. A Machine Learning-Based Approach for Improved Orbit Predictions of LEO Space Debris With Sparse Tracking Data From a Single Station. *IEEE Transactions on Aerospace Electronic Systems*, 56(6):4253–4268, December 2020.

[36] Davide Amato, Aaron Jay Rosengren, and Roberto Furfaro. Solving the main problem in satellite theory through recurrent neural networks. In *42nd COSPAR Scientific Assembly*, volume 42, pages PSD.1–22–18, July 2018.

DOI: 10.13009/EUCASS2022-6137

[37] Alaa Osama, Mourad Raafat, Ashraf A. Darwish, Sara Abdelghafar, and Aboul Ella Hassanien. Satellite orbit prediction based on recurrent neural network using two line elements. *2022 5th International Conference on Computing and Informatics (ICCI)*, pages 298–302, 2022.

[38] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[39] Gideon Mendels. Estimating uncertainty in machine learning models, 2019. *https://towardsdatascience.com/estimating-uncertainty-in-machine-learning-models-part-3-22b8c58b07b*, Accessed on 2021/10/20.