Machine Learning Approach for Accurate and Robust Satellite Tracking in Optical Space-to-Ground Communication using Time-Series Prediction for LEO Satellites

Maurice Uteg
German Aerospace Center (DLR), Responsive Space Cluster Competence Center
Eugen-Sänger-Str. 50, 29328 Faßberg, Germany; maurice.uteg@dlr.de

Helmut Ribel
German Aerospace Center (DLR), Responsive Space Cluster Competence Center
Eugen-Sänger-Str. 50, 29328 Faßberg, Germany; helmut.ribel@dlr.de

Sacha Tholl
German Aerospace Center (DLR), Responsive Space Cluster Competence Center
Eugen-Sänger-Str. 50, 29328 Faßberg, Germany; sacha.tholl@dlr.de

Marcus T. Knopp
German Aerospace Center (DLR), Responsive Space Cluster Competence Center
Muenchener Str. 20, 82234 Wessling, Germany; marcus.knopp@dlr.de

Abstract

As space becomes increasingly congested with Resident Space Objects (RSOs) in Low Earth Orbit (LEO), improving the accuracy of orbit prediction is crucial for ensuring operational reliability, particularly for satellite tracking in optical communication and Telemetry, Tracking, and Command (TTC) operations. This work focuses on refining orbit prediction by leveraging machine learning techniques to enhance tracking capabilities. Traditional orbit approximation relies on the Simplified Perturbations model (SGP4), which calculates a satellite's position and velocity by considering various perturbations, such as Earth's gravitational irregularities and atmospheric drag, using an empirical model for efficient orbit determination. However, this approach is prone to errors, as it simplifies complex orbital dynamics. To address this limitation, this paper explores the potential of machine learning algorithms to analyze time-dependent data, with a particular focus on systematic deviations from SGP4 predictions that are inherently captured in historical orbit information. To achieve this, we create a set of data consisting of time-series satellite position data sets of LEO Objects from past Two-Line Elements (TLEs) as well as orbital messages derived form Global Navigation Satellite System (GNSS) observations and Laser Ranging Data. These data sets are used to train various machine learning models specialized in time series data, such as Long-Short-Term Memory (LSTM) networks to evaluate their potential for improving the robustness and accuracy of orbit forecasting. Finally, the performance of the machine learning model is evaluated by comparing its predictions with those from the traditional SGP4 model. In the future, we will be assessing prediction accuracy and analyze Radial, In-Track, and Cross-Track (RIC) errors to ensure the new model's effectiveness using measurements from Optical Ground Stations at DLR.

Introduction

The rapid growth of Resident Space Objects (RSOs) in Low Earth Orbit (LEO) has transformed orbital safety into a critical challenge for space operations. With over 42,000 trackable objects currently cataloged and millions of smaller debris fragments, the risk of catastrophic collisions has increased significantly [1, 2]. Accurate orbit prediction is a key factor towards ensuring safe satellite operations. Even minor errors in predicting a satellite's trajectory can lead to mission failures due to collisions or the mission ending early due to costly avoidance maneuvers. Following up on this, satellite collisions have occurred due to inaccuracies in prediction data and false alarms [3]. This paper addresses this challenge by proposing a hybrid approach that combines traditional orbital perturbation models with a machine learning (ML) approach to refine orbit prediction accuracy, focusing on LEO being the orbital regime characterized by high object density and collision risk.

Current orbit prediction relies on the Simplified General Perturbations Model 4 (SGP4), especially for smaller Satellites like CubeSats. It is a framework that approximates satellite motion by accounting for perturbations such as Earth's gravitational field and atmospheric drag [4]. While SGP4 remains the industry standard for processing Two-Line Element (TLE) data, its propagation accuracy degrades significantly over time. Studies show that TLE-derived predictions become unreliable after less than 7 days, rendering them unsuitable for long-term operational planning [5]. This limitation stems from SGP4's reliance on simplified physical models and its inability to capture systematic errors arising from un-modeled perturbations, such as variations in atmospheric density or spacecraft-specific characteristics (e.g. mass, geometry, and reflectivity).

Compounding these challenges, traditional methods like the Extended Kalman Filter (EKF) struggle to combine real-time observational data with a dynamic model. While EKF excels in short-term accuracy, its performance relies on the quality of the initial state estimates and its sensitivity to uncertainties in environmental conditions (e.g. solar activity affecting atmospheric drag) [6]. For LEO objects, where atmospheric drag dominates perturbations, these uncertainties are increased by the empirical nature of drag models, which often fail to generalize across varying spacecraft geometries and orientations [7].

Recent advances in ML, particularly recurrent neural networks (RNNs) and their Long Short-Term Memory (LSTM) variants, offer a promising alternative to address these limitations. Unlike traditional methods, ML models are able to learn complex and non-linear relationships directly from historical data, capturing systematic deviations in SGP4 predictions that arise from un-modeled dynamics. By training on time-series data sets derived from openly accessible TLEs, CPFs created by the International Laser Ranging Service Catalogue (ILRS) using Laser Ranging Data and orbital messages in the Consolidated Prediction Format (CPFs) provided by the German Space Operation Centre (GSOC), LSTMs can identify patterns without modeling potentially unknown parameters such as the shape of RSOs, maneuver epochs/duration and orientation of the spacecraft. [8, 9]

Conservative forces (e.g. Earth's gravitation) and Non-Conservative forces alike (e.g. solar radiation) influence the RSOs and their respective orbits. This work explores the possibility that ML-enhanced orbit prediction can learn to model the orbit propagation because most of these dependencies are inherently contained in the RSOs' historical data. Specifically, we investigate whether training LSTMs or similar structures on historical orbital data from the past 230 to 900 TLEs can yield more accurate 14-day predictions than traditional approaches. This time frame aligns with operational needs for collision avoidance and mission planning, where even small improvements in accuracy can reduce false alarms and unnecessary maneuvers. Results show the potential of the proposed ML approach to improve the publicly available TLE catalogue accuracy. Improvements are achieved in all position directions with the In-Track-direction being the hardest to model. The quality of the results depend on the number of training points used as well as the area to mass ratio of the satellites tested.

Proposed Framework

The following section reviews past and on-going machine learning applications in orbit determination and propagation, positioning the proposed hybrid SGP4-ML framework within the state-of-the-art. Then, the implemented machine learning approach is explained and the model structure specifically designed for LEO propagation is presented.

Background

Previous work has integrated machine learning into satellite orbit determination, each addressing limitations of the traditional approach. Peng and Bai (2019) implemented a support vector machine (SVM) framework that learned residuals between SGP4 predictions and actual trajectories using TLE data from the North American Aerospace Defense Command (NORAD) and laser ranging observations from the International Laser Ranging Service (ILRS). Their study covered eleven satellites across LEO, MEO, and sun-synchronous orbits (SSO), demonstrating consistent reduction in position and velocity errors. Improvements are especially apparent after one day of the TLE generation. However, the validation methodology exhibited limitations when TLE-derived state errors were close to zero, as the performance metric became unstable in these regimes. [10]

Nor'asnilawati et al. (2021) explored LSTM networks in combination with SGP4 for orbit propagation using NORAD TLE catalog data combined with real-time optical observations from the Malaysia Space Centre. Their implementation showed measurable improvement in three-to-seven-days predictions compared to standalone SGP4. The study notably highlighted the challenge of atmospheric drag modeling in LEO as well as the frequent change in illumination conditions for every revolution. [5]

Mortlock et al. investigated the Simultaneous Tracking and Navigation (STAN) framework. Their approach employs an EKF that dynamically switches between two operational modes. The first being standard tracking using Global Navigation Satellite System signals while it is available. When GNSS signals become unavailable, inter-satellite links from other LEO satellites are utilized as pseudo-observables to estimate the state vector. This system is enhanced by incorporating a neural network to correct propagation errors during periods with no GNSS signal. However, the implementation remained constrained by the quality of initial state estimates from the EKF. [11]

Machine Learning Enhanced Orbit Propagation

Orbit prediction is challenging due to the complexity of non-conservative forces acting on the RSOs. Forces such as atmospheric drag and solar radiation pressure undergo high variability which can be caused by spacecraft-specific properties (e.g. area-to-mass ratio and orientation of the spacecraft) and rapidly changing environmental conditions (e.g. atmospheric density fluctuations, solar activity). Traditional propagators like SGP4 simplify these dynamics through empirical approximations, introducing systematic errors that accumulate with longer propagation spans [4]. The approach of this work addresses this by embedding machine learning as an error-correction layer within existing operational workflows, avoiding disruptive changes to established prediction pipelines.

With the hypothesis that a low-dimensional problem can be handled easier by the model, our algorithm trains ML models to forecast residual errors between SGP4/TLE-propagated states and true trajectories rather than predicting absolute orbital states. The hypothesis is that systematic error patterns (e.g. drag-induced along-track decay) are easier to learn by an algorithm than raw orbital dynamics. The model uses errors as input to then predict the error of the next time step. The machine learning error is given by the model, and the state calculated from it ideally lies between the true state and the TLE state. The following equations are used to calculate the inputs for the proposed ML-model:

$$\mathbf{e}_{\text{true}}(t;t-1) = \mathbf{x}_{\text{true}}(t) - \mathbf{x}_{\text{TLE}}(t;t-1) \tag{1}$$

The true error $\mathbf{e}_{\text{true}}(t)$ marks the target for the ML-model. It is computed by subtracting $\mathbf{x}_{\text{TLE}}(t;t-1)$ – the TLE propagated state at time t propagated from time $t-1$ – from $\mathbf{x}_{\text{true}}(t)$ – the true state at time t being the orbital data provided by ILRS or GSOC.

$$\mathbf{e}_{\text{ML}}(t;t-1) = \mathbf{x}_{\text{ML}}(t;t-1) - \mathbf{x}_{\text{TLE}}(t;t-1) \tag{2}$$

$$\mathbf{e}_{\text{ML}}(t;t-1) = \mathbf{x}_{\text{ML}}(t;t-1) - \mathbf{x}_{\text{true}}(t) + \mathbf{e}_{\text{true}}(t;t-1) \tag{3}$$

In the equations above, $\mathbf{x}$ marks the 6-dimensional state vector containing x-,y-,z- positions and velocities of the spacecraft in those directions. Equation 2 defines the machine learning error as the difference between the corrected state by the model $\mathbf{x}_{\text{ML}}$ and the TLE-propagated state $\mathbf{x}_{\text{TLE}}$. Substituting $\mathbf{x}_{\text{TLE}}$ using the true error definition yields Equation 3, expressing the ML error in terms of the states $\mathbf{x}_{\text{true}}$ and $\mathbf{x}_{\text{ML}}$ and error $\mathbf{e}_{\text{true}}$.

$$|\mathbf{x}_{\text{ML}}(t) - \mathbf{x}_{\text{true}}(t)| < |\mathbf{x}_{\text{TLE}}(t) - \mathbf{x}_{\text{true}}(t)| \tag{4}$$

Equation 4 shows the fundamental principle of the framework. When the ML model successfully approximates the true error, the corrected state lies closer to the actual trajectory than the original TLE prediction.

Model Structure

The model architecture employs a version of recurrent neural networks (RNNs), with LSTM and Gated-Recurrent-Unit (GRU) variants selected for their ability to capture time-dependencies in orbital propagation. LSTMs mitigate vanishing gradients through three adaptive gates regulating the internal information flow. The Forget Gate determines which prior error states (e.g. historical drag signatures) to discard. For orbital data, this gate learns to suppress outdated information like atmospheric density estimates when orbital conditions change. The Input Gate controls the incorporation of new error observations. The Output Gate filters the cell state to produce the hidden state, which encodes persistent error patterns like drag cycles. GRUs merge LSTM functionality into two gates for computational efficiency. GRUs make use of the Update Gate which blends prior states (history of the error pattern) and candidate states (proposed state from new input), where the reset gate controls relevance of historical errors. [12]

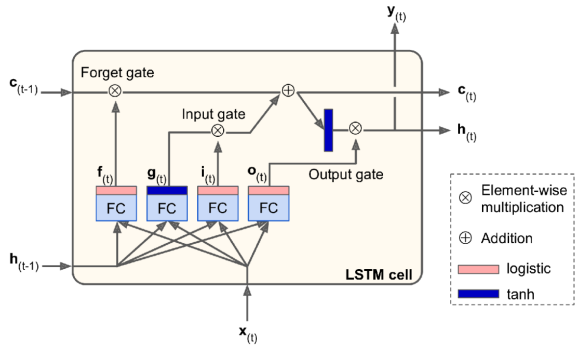


Figure 1: Scheme of an LSTM Cell [13]

Four configurations are implemented in Python using Tensorflows high-level API "Keras". The first two configurations involve an LSTM/GRU input layer followed by an LSTM/GRU hidden layer with a Dense output layer with the input vector being a 14-Day error sequence.

The other two configurations include a TimeDistributed layer as opposed to the previous Dense output layer which applies an identical Dense layer independently to each time step (e.g. predicting errors at *t+1, t+2, ..., t+5* simultaneously). This preserves the temporal structure and adds the model's ability to preserve time-dependent information for each time point separately. All models are using ELU or tanh activation in hidden layers with minor performance differences between the two. The loss function is considered a hyperparameter with either the Normalized Root Mean Squared Error (NRMSE), Mean Absolute Error, or Huber loss as hybrid alternative. NRMSE punishes outliers severely and equally across the different magnitudes of input values. MAE is more robust towards outliers and gives each error the same significance depending on its value. Huber-Loss tries to combine both ideas, by being quadratic for low values near the chase value and linear for large values.

Machine Learning Approach

Signal Processing

Input sequences comprise consistency error vectors from the previous 14 data points, derived by comparing TLE-propagated states with measured orbital data messages applicable to Equation 2. Each error vector includes position and velocity deviations in the Radial-, In-Track-, Cross-Track-coordinate system (RIC). Additionally, the input includes B* drag coefficient deviations present in the TLE-Data representing atmospheric drag changes and the time gaps between the data points, as they vary between TLE epochs. All input variables are standardized to zero mean and unit variance to ensure numerical stability and equal learning importance during training.

For validation, we analyze four LEO satellites tracked by the ILRS and 10 satellites provided by GSOC. Ground truth references (ILRS laser ranging, GSOC CPFs) are sampled at 60–180 second intervals; when comparing to TLEs, the nearest ground truth epoch is selected to minimize interpolation errors. Training and test sets were constructed chronologically to simulate real-world operational conditions where future orbits represent unknown variables. Peng and Bai experimented with alternating training and testing data on the time scale with the conclusion that a linear approach performs just as well [10].

Early tests showed that different model architectures increase the prediction accuracy of different parameters. For that reason, six independent models are trained with

	Training Data Points	Validation Data Points	Test Data Points	Available TLEs
Ajisai	2674	434	434	624
Lares	4228	700	700	982
BeaconC	1974	322	322	552
Starlette	2688	448	448	634
#1	8400	1400	1386	868
#2	5068	840	840	576
#3	2352	392	378	307
#4	3346	546	546	402
#5	9380	1554	1554	982
#6	2030	336	336	266
#7	1862	308	308	231
#8	2268	378	364	298
#9	2310	378	378	302
#10	3948	658	658	439

Table 1: Table of the used satellites and available TLEs and the corresponding data points; numbered satellites are anonymous

one for each output dimension (*x, y, z* position and velocity residuals). Each model learns the temporal gap between epochs to predict forward propagation states, while improving consistency errors using the 14 most recent backward-propagated error vectors.

Table 1 shows the available data of each satellite that is investigated. The number of data points result from the number of TLEs and true states available with the ratios between training, validation and testing respectively. GSOC provided RSOs #8 and #9 emitted erroneous signals during coordinate conversion that could not be resolved during the frame of this work. For that reason, those data are omitted for performance evaluation.

Parameter Study

Following Peng and Bai [10], we adopted the position mean loss (PML) as prediction improvement metric:

$$P_{\mathrm{ML}}(\mathrm{var}) = \frac{\sum_n |e_{\mathrm{true,var}} - e_{\mathrm{ML,var}}|}{\sum_n |e_{\mathrm{true,var}}|} * 100\% \qquad (5)$$

where $\mathbf{e}_{\mathrm{true}}$ is the true error and $\mathbf{e}_{\mathrm{ML}}$ is the ML-predicted error. While PML quantifies relative improvement over TLE propagation, it fails to capture prediction uncertainty as it reports only mean improvement without confidence intervals. For the hyperparameter optimization, a complete search rather than traditional grid search is implemented to ensure comprehensive exploration of the parameter space. This approach evaluates all possible combinations within defined ranges, eliminating the risk of missing optimal configurations that might fall between grid points.

All input features undergo column-wise standardization to a normal distribution ($\mu = 0$, $\sigma = 1$), calculated independently for each feature column. This approach preserves the physical dimension of error components, ensuring that inherently larger-magnitude errors maintain proportionally greater influence in the loss calculation compared to smaller errors.

Three loss functions are evaluated as explained in subsection "Model Structure". This configuration enables direct comparison of loss functions and their individual impacts on prediction accuracy for radial, in-track, and cross-track error components.

Performance

This section presents a comprehensive assessment of the proposed machine learning orbit propagation framework across all 12 RSOs. The evaluation methodology employs CPF data as the true state reference, with each TLE propagated forward to predict the subsequent 14 data points. This approach enables quantification of error reduction across radial, in-track, and cross-track

components, as well as their corresponding velocity errors.

The GRU architecture demonstrated better prediction accuracy compared to LSTM with less computational time. Among loss functions, Huber and NRMSE outperformed MAE, indicating that controlled penalization of outliers better suits the non-Gaussian error distributions present in orbital information.

Figures *2 (a) to 2 (f)* show the performance of Beacon-C, representative of a successful implementation of the proposed ML algorithm. The position and velocity error evolution demonstrates major improvements across all components.

Each graph presents prediction duration in time points along the horizontal axis against the error magnitude expressed in meters (*m*) for position components and meters per second (*meter/second*) for velocity components on the vertical axis. Time points have non-uniform gaps as a time point is the epoch of an available TLE that also has a true state in close time proximity. The data representation depicts the mean error as a central dot with standard deviation visualized through error bars surrounding it, with a slight offset between error types for better visibility. Error bars in red represent the true error representing baseline performance, blue indicates the residual error remaining after ML correction has been applied, and green represents the ML-predicted error.

The results demonstrate consistent error reduction across all components albeit with varying magnitudes. The in-track position error proves most challenging to model. This may be because it is dominant in LEO due to atmospheric drag effects or because it has a high correlation with the error in the radial direction. Nevertheless, the residual absolute error after 14 prediction steps remains several kilometers lower than the original TLE-error. Systematic error patterns are effectively captured not only in position but also in all three velocity components, reducing propagated velocity errors in the RIC-frame.

Table 2 shows the performance analysis with a significant variability across satellites, with area-to-mass ratio emerging as a potential correlating factor. Satellites with higher area-to-mass ratios demonstrate reduced model capacity in capturing atmospheric drag effects, particularly evident in in-track error prediction where the correlation between PML improvement and area-to-mass ratio (σ = 0,652) suggests insufficient drag modeling. This indicates potential value in incorporating area-to-mass ratio as an input parameter during training. The models struggle to correct the velocity error in the radial and cross-track direction which indicates that input features lack sufficient information in that regard.

Figure 3 depicts a case where the model demonstrates limited predictive capability, achieving an error reduction with a PML value of 87. This sub-optimal performance may be an indicator for inadequate representation of the satellite's area-to-mass ratio in the feature set or insufficient temporal resolution between observational data points. These limitations will be the focus of future works.

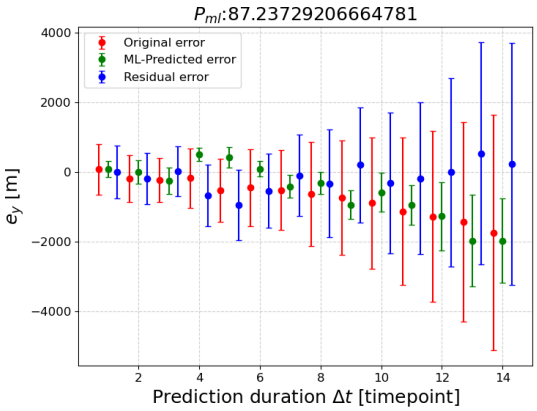


Figure 3: In-Track error e_y of RSO #10 showing insufficient improvements through the PML metric

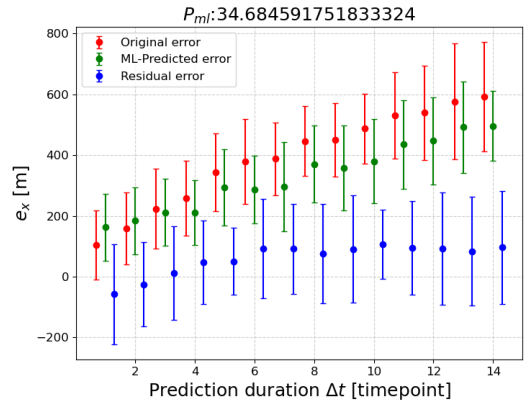
Smaller absolute values of original velocity errors may further limit PML improvement potential as the baseline magnitude is already low. Additionally, satellite maneuvers likely degrade prediction accuracy during and following maneuver periods. Future implementations could benefit from excluding maneuver-affected TLEs.

Preliminary analysis indicates that a training data set size beyond 200 TLEs yields diminishing returns, although expanded testing is needed to validate this finding.

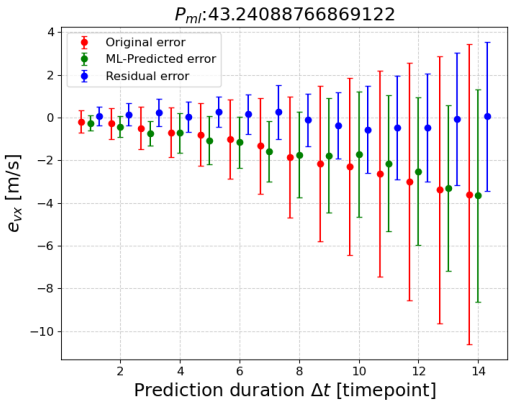
Conclusion

This study demonstrates that machine learning techniques can enhance LEO satellite orbit prediction by modeling TLE propagation errors across twelve Resident Space Objects. The results reveal consistent error

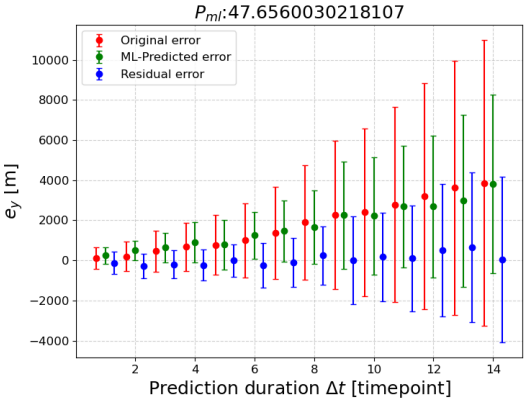
Figure 2: Beacon-C error components in RIC frame



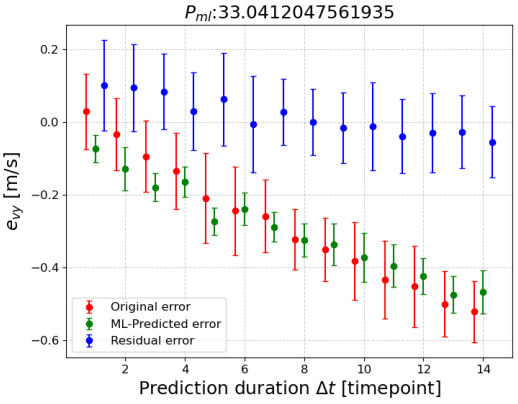
(a) Radial position error



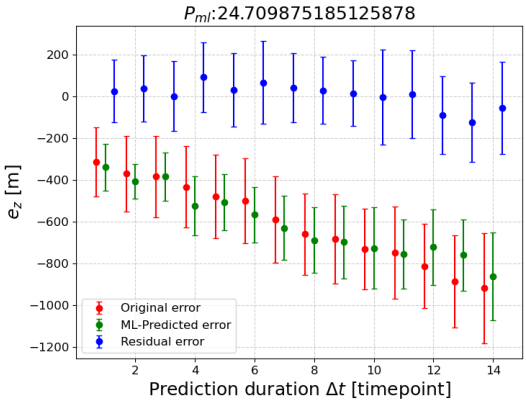
(d) Radial velocity error



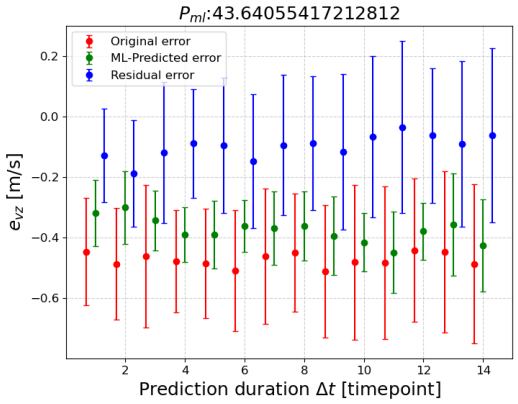
(b) In-track position error



(e) In-track velocity error



(c) Cross-track position error



(f) Cross-track velocity error

reduction across all six RIC-frame parameters for satellites with a small area-to-mass ratio. However, performance varies considerably across the investigated parameters, with cross-track position errors proving especially challenging for satellites with higher area-to-mass ratios, while radial and cross-track velocity errors remain difficult to predict across all RSOs.

A critical limitation of the current TLE framework is its implicit assumption of zero initial error, treating propagated states as ground truth when in reality they contain systematic biases that accumulate over time. Incorporating satellite-specific area-to-mass ratio as an explicit input parameter could improve ML-enhanced atmospheric drag modeling.

The proposed ML approach offers a practical pathway to enhance the operational TLE catalog by extending prediction validity periods and reducing position uncertainties. For objects without continuous orbit determination processes, a more reliable conjunction assessment and maneuver planning can be implemented with minimal infrastructure changes.

Our approach enables more accurate orbit predictions for satellites that lack frequent tracking or GNSS-based orbit determination. This enhanced prediction capability is particularly valuable for ground-based tracking systems, which can leverage the improved orbital knowledge to maintain continuous observation of space objects between sparse catalog updates. Future work will focus on validating the improvement of the detection accuracy through real telescope observations and developing adaptive models that account for significant variations in illumination conditions and atmospheric drag profiles across different orbital regimes. Additionally, investigating the impact of area-to-mass ratio and time gaps between data points may yield further improvements in prediction quality.

Acknowledgements

The authors would like to express their sincere gratitude to the GSOC Flight Dynamics Services at the German Aerospace Center (DLR) for their provision of precise orbital information, which was essential for the validation and development of the proposed methodology. Their data, both TLEs and accurate CPFs, and expertise in orbit determination greatly enhanced the accuracy and reliability of our analysis.

References

[1] European Space Agency. *Space debris by the numbers*. ESA. 2024. URL: https://www.esa.int/Space_Safety/Space_Debris/Space_debris_by_the_numbers (visited on 07/23/2025).

[2] Carmen Pardini and Luciano Anselmo. "Evaluating the impact of space activities in low earth orbit". In: *Acta Astronautica* 184 (2021), pp. 11–22.

[3] TS Kelso. "Iridium 33/cosmos 2251 collision". In: *Celestrak. com* 5 (2009).

[4] Ilia V Baranov. "SGP4 propagation program design and validation". In: *University of Waterloo: Faculty of Engineering Department of Electrical and Computer Engineering* (2009).

[5] Nor'asnilawati Salleh, Nurulhuda Firdaus Mohd Azmi, and Siti Sophiayati Yuhaniz. "An adaptation of deep learning technique in orbit propagation model using long short-term memory". In: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE. 2021, pp. 1–6.

[6] Oliver Montenbruck and Pere Ramos-Bosch. "Precision real-time navigation of LEO satellites using global positioning system measurements". In: *GPS solutions* 12.3 (2008), pp. 187–198.

[7] Yang Zhang et al. "Numerical analysis of the influence of typical perturbation forces on LEO satellites orbit prediction at different altitudes". In: (2024).

[8] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[9] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[10] Hao Peng and Xiaoli Bai. "Machine learning approach to improve satellite orbit prediction accuracy using publicly available data". In: *The*

Journal of the astronautical sciences 67.2 (2020), pp. 762–793.

[11] Trier R Mortlock and Zaher M Kassas. “Performance analysis of simultaneous tracking and navigation with LEO satellites”. In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*. 2020, pp. 2416–2429.

[12] Rahul Dey and Fathi M Salem. “Gate-variants of gated recurrent unit (GRU) neural networks”. In: *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE. 2017, pp. 1597–1600.

[13] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O’Reilly Media, Inc.", 2022.

Satellite Denomination	PML Radial	PML In-Track	PML Cross-Track	PML Radial Vel.	PML In-Track Vel.	PML Cross-Track Vel.	AMR
Ajisai	65,89	96,74	28,5	98,6	61,54	168,66*	0,0058
Lares	25,95	69,07	32,8	59,08	24,6	92,88	0,00027
Beacon-C	34,68	47,65	24,70	43,24	33,04	43,64	-
Starlette	112,43*	61,46	26,47	62,67	93,03	59,37	0,00096
#1	52,65	79,12	97,25	83,19	70,93	100,07*	0,0068
#2	35,87	93,35	89,63	88,37	51,96	86,84	0,001837
#3	54,10	99,23	75,98	172,18*	55,83	90,79	0,005573
#4	51,97	102,24*	79,9	99	57,36	94,79	0,017699
#5	77,24	101,18*	102,72*	90,36	100,66*	101,3*	0,009926
#6	108,70*	74,49	107,05*	90,36	94,6	99,06	0,002175
#7	34,70	91,28	96,98	86,98	43,59	102,48*	0,002191
#10	47,26	87,23	85,82	89,8	50,1	104,37*	0,004966

Table 2: Performance of all investigated Satellites, GSOC provided satellites are anonymized. Columns represent the respective error of each variable with the last column being the area-to-mass ratio of the RSO