EasyChair Preprint

№ 15676

Improved Orbit Prediction Method Based on Two-Line Elements with Dynamic Loss Function

Wenxin Li, Yanfang Tao and Hao Deng

January 6, 2025

Improved Orbit Prediction Method based on Two-Line Elements with Dynamic Loss Function

Wenxin Li[1][0009−0009−0457−7636], Yanfang Tao[2][0000−0001−7763−2653], and Hao Deng*,[1][0000−0003−3841−958X]

[1] College of Informatics, Huazhong Agricultural University, Wuhan 430070, China
lwxx@webmail.hzau.edu.cn; dengh@mail.hzau.edu.cn
[2] School of Information Engineering, Wuhan Business University, Wuhan 430056, China
taoyf@wbu.edu.cn

Abstract. Accurate orbit prediction is crucial for space situational awareness. However, Physics-based approaches can fail to achieve the required accuracy for collision avoidance of Resident Space Objects (RSOs). This paper presents a Machine Learning-based approach for RSOs orbit prediction leveraging Two-Line Element (TLE). Taking the dynamic nature of orbital deviations into consideration, we integrate a dynamic loss function into the orbit prediction framework, allowing for a more adaptive and accurate prediction model. The experiments demonstrate the superior performance of our proposed method in predicting RSOs orbits over extended periods.

Keywords: Orbit Prediction · Space Objects · Dynamic Loss Function · Convolutional Neural Networks · Long Short Term Memory.

1 Introduction

The increasing number of Resident Space Objects (RSOs), particularly in low-Earth-orbit (LEO) [1], has made accurate prediction of RSOs orbits crucial. Traditional prediction methods take into account various influences, such as gravity, atmospheric drag, and solar radiation pressure from the Earth, Moon, and Sun, enabling high-precision RSOs trajectory predictions through simulation and calculation. However, these physics-based approaches often lack crucial information including initial state, environmental conditions, and target characteristics [4], limiting their accuracy and requiring computationally demanding processes [2, 3].

In order to address this issue, machine learning technology has emerged as a novel approach that has garnered significant research attention. Machine learning presents an alternative methodology for modeling and prediction compared to traditional methods. Through the analysis of historical orbital data and the application of relevant algorithms, machine learning techniques can identify patterns and enhance the predictive capability of RSOs orbital models. Currently,

RSOs orbit prediction based on TLE(Two-Line Elements) datasets is a well-established practice in the aerospace industry, with TLE datasets providing detailed information on RSOs orbit elements for rapid and efficient orbit prediction [5]. Previous studies have demonstrated that the utilization of machine learning techniques can substantially enhance the accuracy of orbit prediction. Hao et al. conducted extensive research using artificial neural networks to significantly enhance orbit prediction accuracy[6] and explored the application of SVM[7] in satellite orbit prediction, demonstrating the broad potential of machine learning methods for improving orbit prediction precision. Their study utilized a simulation-based spatial catalog environment to showcase three generalization abilities of the proposed machine learning methods[8], systematically investigating the use of three algorithms (SVM, ANN, and Gaussian processes) for orbit prediction accuracy[9]. Additionally, they proposed an innovative fusion strategy that integrates machine learning outputs into a traditional framework using the Extended Kalman Filter (EKF) for orbital estimation and prediction, showing improved accuracy and precision in orbit prediction through simulations[10]. Li et al. introduced a novel orbit prediction method using TLE, employing GBDT and CNN to model and analyze error datasets, achieving over a 75% increase in prediction accuracy along the orbit direction[11]. Giridhar improved the accuracy of traditional TLE prediction by using curve fitting and LSTM to predict historical track data[12]. Rohit proposed a computer vision approach based on real-time Earth images to predict satellite orbits when ground communications are lost. This method, which uses neural networks trained on image datasets, outperforms traditional Kalman filters[13]. Estel et al. trained supervised learning models based on Kepler and SGP4 orbital models to predict inter-satellite links. Their study demonstrates how data from these orbital models can enhance ISL prediction accuracy, thereby optimizing link management and space resource utilization in satellite networks[14].

Although TLE-based orbit prediction has made some progress, there is still a research gap in developing methods to dynamically adjust prediction models based on real-time orbit data. Current methods often rely on predefined parameters and assumptions that may not accurately reflect the complex dynamics of RSOs orbits. Inspired by L2T-DLF, the model dynamically adjusts the loss function during training [15]. Different from the traditional TLE-ML orbit prediction method, this paper presents a new and effective method using TLE for accurate orbit prediction by training the orbit error model. In our study, building on the existing TLE-ML orbit prediction framework and learning from changing orbit prediction errors by introducing a dynamic loss function, our proposed approach aims to bridge this gap and improve the prediction accuracy of TLE-based orbit prediction models. As a feedback mechanism, the dynamic loss function continuously evaluates the accuracy of the prediction and updates the model parameters accordingly. By combining orbital prediction error with feedback signal from the loss function, our approach aims to enhance long-term predictive capability of tle-based orbital predictions.

Currently, traditional orbital prediction methods are mainly based on physical models, and there are three general ideas, namely numerical, analytical, and semi-analytical methods [17], which are essentially all based on physical models. TLE published by the North American Aerospace Defense Command (NORAD) is the most commonly used orbital data by the majority of aerospace worker, and is published on the space-track website, which is the largest publicly available catalog of space objects [16]. The Simplified General Perturbation 4 (SGP4) propagator that accompanies the TLE data, a mathematical model for calculating and predicting the orbits of space objects, was developed by NASA and is a common analytical method.The SGP4 propagator allows for the equations of motion to be solved quickly because of some simplification of the ingress model, yielding a lower approximation of accuracy. As a result the computation is extremely fast and the computational and time costs are dramatically reduced. The orbital dynamic motion equation of a space object in the geocentric inertial coordinate system(ECI) is as follows [17, 18]

$$f(x, c, t) = \ddot{\boldsymbol{r}} = -\frac{\mu_{\oplus}}{r^3}\boldsymbol{r} + a_{pert}(\boldsymbol{r}, \dot{\boldsymbol{r}}, t) + \boldsymbol{\Gamma}, \tag{1}$$

Where x is the state vector of the space object, consisting of six dimensions:$x = [r, \dot{r}]$, with r, $\dot{r}$ and $\ddot{r}$ representing the position vector, velocity vector, and acceleration vector of the space object, respectively. The parameter c refers to the force model parameters, including the drag coefficient C_d, which is used to calculate atmospheric drag effects. The variable t represents time. $r = ||r||$ denotes the Euclidean norm of the vector $\boldsymbol{r}$.$\mu_{\oplus} = GM_{\oplus}$, where G is the gravitational constant and $M_{\oplus}$ is the mass of the Earth. a_{pert} is the perturbation acceleration vector caused by non-spherical gravity, third-body gravity attraction, atmospheric drag, solar radiation pressure, etc. $\boldsymbol{\Gamma}$ is the thrust acceleration vector.

To predict the motion of a space object using the SGP4 propagator, the initial state $x_0 = [r, \dot{r}]_{t=t_0}$ and the resistance coefficient C_d of the future epoch t must be provided. With these inputs, the SGP4 propagator can be used to determine the predicted state at time t.

In this study, the determination and prediction of space object orbits are carried out using only TLE data and the SGP4 propagator. In practice, the true orbit of a space object is unknown and can only be estimated through orbit determination methods. Therefore, the TLE data at a specific moment is considered as the corresponding estimated orbit. The orbit prediction error is defined as the difference between the real orbit value and the predicted orbit value. For a specific space object's TLE, an important characteristic of TLE-based Orbit Prediction (OP) error is that the orbital error around the TLE reference epoch is relatively small [19]. Based on this observation, in the existing technique, the state vector derived from each TLE at its epoch is treated as an approximation of the true vector. Since the forecast error of TLE within the orbital period centered on its epoch is relatively small, the forecast state during that period is used as the reference state in this study. The previously predicted state of the TLE at that interval is used as a prediction [20]. Therefore, the prediction

error can be obtained by subtracting the orbital reference state quantity from the predicted orbital state quantity at a specific epoch time, denoted as:

$$e = \hat{X}_{pred} - X_{true}, \tag{2}$$

At the jth TLE moment, take the long orbital period interval centered on the epoch moment, divide this time interval into n parts, so that $n+1$ time points can be obtained.

$$t_{j,k} = t_j - \frac{T}{2} + k\frac{T}{n}, \quad k = 0, 1, 2, \ldots n, \tag{3}$$

In this time period, the reference state vector of $t_{j,k}$ is estimated according to the t_j TLE, and the predicted state vector of $t_{j,k}$ is estimated by the t_i TLE, so equation (2) is rewritten as:

$$e = \hat{X}_{(j;i)} - X_{(j)} = \hat{X}_{(j,k;i)} - \hat{X}_{(j,k;j)}, \tag{4}$$

Where, t_i and t_j are the epoch corresponding to the ith and jth TLE respectively, $i<j$, indicating that the TLE at epoch i is used to obtain the predicted state vector, and the TLE at epoch j is used to obtain the predicted vector, which is used as an approximation of the real state vector. Then, the ith TLE is extrapolated to all epoch within the forecast time threshold T one by one, and compared with the subsequent TLE derived state vector to obtain the prediction error. Fig.1 is the schematic diagram of the process.

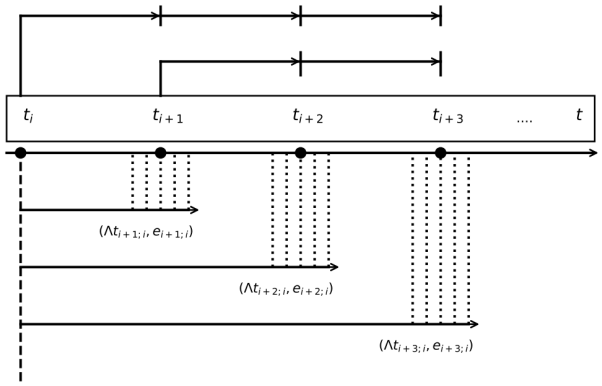


Fig. 1: Generation of TLE orbit prediction error

Our research stands out among previous studies and makes two significant contributions to enhance the accuracy and reliability of orbit predictions. These contributions include:

(1)Introduction of a new strategy: We propose a novel strategy to improve the accuracy of orbit prediction by incorporating a dynamic loss function into the

TLE-ML framework. Unlike traditional TLE-ML methods that employ a static loss function with a fixed weight during training, our approach dynamically adjusts the weight of the loss function based on the characteristics of the input data. This dynamic adjustment enables the model to prioritize critical moments and important features, thereby enhancing the prediction accuracy.

(2)Experimental validation: We extensively validate the effectiveness of our proposed strategy through empirical experiments in various orbit prediction tasks. By comparing our strategy with traditional TLE-ML methods, we observe significant improvements across multiple datasets and assessment metrics, confirming the practical benefits of our approach.

In the following, section 2 provides an overview of data and models, encompassing dataset descriptions and deep learning algorithms with dynamic loss functions. This section highlights the use of convolutional neural networks (CNN) and Long Short-Term Memory (LSTM) models as the primary approaches for error pattern learning. Section 3 presents experimental results aimed at enhancing the accuracy of orbit prediction. Finally, Section 4 offers conclusions and outlines future directions for research.

2 Data and Pineline

2.1 Dataset Description

Machine learning algorithms rely on input space $\mathcal{X}$ and output space $\mathcal{Y}$, and can learn an underlying mapping relationship $\mathcal{Y} = \psi(\mathcal{X})$. The success of machine learning algorithms largely depends on the selection of features, so in the modeling of orbit prediction errors, it is necessary to carefully select the features that have a decisive influence on the target variable y. In general, too many variables will lead to overfitting of the model, while too few variables may lead to underfitting. Features should be reasonably selected to adequately describe the evolution characteristics of orbit prediction errors.

The OP error is converted from an Earth-centered inertial coordinate system (ECI) to a RSOs-based UNW (in-orbit, normal, and cross-orbit) coordinate system $e = [\Delta U, \Delta N, \Delta W]$. Take e as the output variable at epoch j, then the corresponding input variable is: $[\Delta t, \mathbf{x}(t_i), \mathbf{x}(t_j; t_i), B^*]$.

1. $\Delta t = t_j - t_i$, $j > i$ indicates the interval between the start epoch t_i and the end epoch t_j.
2. $\mathbf{x}(t_i)$ represents the initial state vector in the ECI frame, consisting of position and velocity vector, expressed as $\mathbf{x}(t_i) = [r_x^i, r_y^i, r_z^i, \nu_x^i, \nu_y^i, \nu_z^i]$.
3. $\mathbf{x}(t_j; t_i)$ represents the orbital prediction state vector in the ECI frame, and the predicted vector about the epoch j is obtained based on the epoch i.It can also be expressed as $\mathbf{x}(t_j; t_i) = [r_x^{j \leftarrow i}, r_y^{j \leftarrow i}, r_z^{j \leftarrow i}, v_x^{j \leftarrow i}, v_y^{j \leftarrow i}, v_z^{j \leftarrow i}]$.
4. In solving the equation of state of a space object, the atmospheric drag effect is an important parameter and therefore the value of B^* is taken into consideration.

According to the above definition of input variables and output variables, a series of data pairs (x_i, y_i) can be obtained, thus constituting the error data set $D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. x_i is a 14-dimensional vector, and y_i is a 3-dimensional vector. So the goal of machine learning approach is to learn the mapping between x and y. If the machine learning approach is good at capturing potential error patterns, it can be used to correct future time orbit predictions.

Due to observation errors and uncontrollable external factors, there will be some outliers in TLE data, outliers in the generated forecast track, and outliers in the corresponding error data set. Taking into account the nature of TLE's prediction errors that grow over time, a data-cleaning strategy based on box plot, which displays a large amount of information in a simplified format, has been adopted to find outliers. It can show the concentration of the data set, dispersion trends, and highlight outliers. The box plot of the integer error dataset is drawn, and five representative values are obtained, namely the minimum $Q1$, first quartile $Q2$, median $Q3$, third quartile $Q4$, and maximum $Q5$. Define the difference $IOR = Q3 - Q1$, called interquartile range. Data points outside the range $[Q1 - 3 * IQR, Q3 + 3 * IQR]$ are defined as outliers, which need to be eliminated to obtain a new error dataset.

2.2 Deep Learning Methods with Dynamic Loss Function

In this paper, our goal is to use a machine learning model to learn a dataset of historical OP errors and apply it to the performance evaluation of prediction errors at future moments, which is a regression problem. For this purpose, we designed two different models, a CNN model based on one-dimensional convolution and an LSTM model containing eight LSTM layers and one linear layer. By using the one-dimensional convolutional CNN model, we can extract features from historical OP error data and make predictions. The one-dimensional convolution operation can effectively capture local patterns and features in time series data, providing useful information for subsequent prediction tasks. On the other hand, our LSTM model consists of eight LSTM layers and one linear layer. This architecture allows the model to gradually build up an understanding of the sequence data and learn long-term dependencies.The memory units and gating mechanisms of the LSTM enable it to efficiently process the sequence data and capture the contextual information in the sequence. The linear layer is used for final output prediction. These two models allow us to learn and predict historical OP error data and evaluate their performance in terms of OP error at future moments. This helps to understand the accuracy and reliability of the models and their applicability in OP tasks.

Generally speaking, the loss function of a machine learning model needs to be predetermined, for example, the loss functions commonly used in regression problems are mean square error(MSE), root mean square error($RMSE$) and mean absolute error(MAE), etc. Inspired by the teaching relationship in real life, Wu et al. introduced this relationship into machine learning, proposed the concept of learning to teach, and proposed L2T-DLF on this basis. In this framework, the loss function of the machine learning model (called the student model)

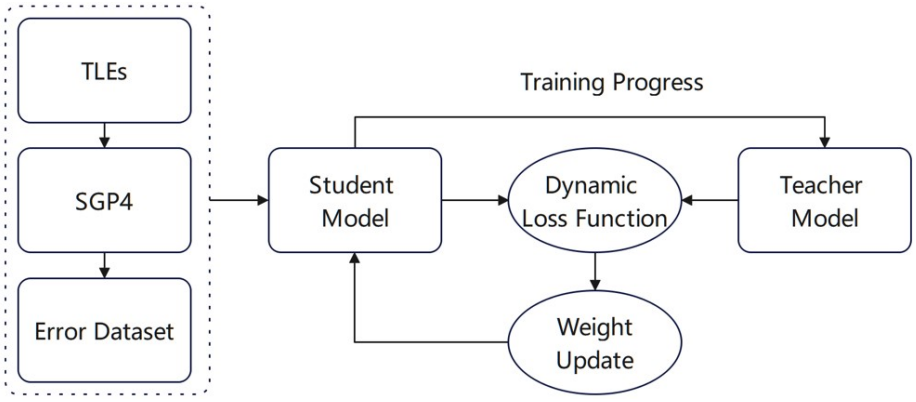


Fig. 2: Dynamic feedback loop between student model and teacher model in training process

is defined by another machine learning model (called the teacher model). As a parametric model, the teacher model dynamically outputs different loss functions that will guide the use and optimization of the student model in different training stages [13]. Inspired by L2T-DLF, we introduce the dynamic loss function into the orbit prediction problem. Based on the traditional machine learning model (called the student model), a parametric model (called the teacher model) is introduced that is responsible for setting the appropriate loss function l for the student model by outputting the appropriate loss function coefficients Φ. At each stage of the iteration of the student model, a state vector s_t is generated, which contains the current iteration number t, the precision p of the current training. The input of the teacher model is a state variable s_t, and the output is a loss function coefficient Φ. The loss function is dynamically updated by the loss function coefficient.

$$\Phi_t = \mu_\theta(s_t), \tag{5}$$

$$l_{\Phi_t}(\hat{y}, y) = \Phi_t MSE, \tag{6}$$

Equation (5) calculates the coefficient of the loss function in the teacher model, where the parameter is θ, and it calculates the loss coefficient Φ_t for the tth iteration. On the other hand, Equation (6) represents the loss function that is calculated using the mean square error (MSE), which is defined as: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Here, y_i represents the actual value and $\hat{y}_i$ represents the predicted value. The mean square error is a commonly used metric to measure the average squared difference between the predicted and actual values. The implementation flow of deep learning methods with dynamic loss function in the error pattern modeling is given in Fig.2.

3 Experimental Results

In this experiment, a total of five RSOs were chosen as the experimental objects. Their relevant information is presented in Table 1. The table includes the average values of orbital inclination, eccentricity, and orbital period, which were obtained from all sets of TLE data for the specific year. TLE data from the first to the sixtieth day(1st to 60th day) were utilized to create error datasets for training purposes. The TLE data from the sixty-first day were then employed to generate updated orbit predictions for the subsequent fourteen days (61st to 74th day). These updated predictions were used as new inputs to evaluate the performance of the trained machine-learning model in terms of its ability to make improved orbit predictions for future periods.

Table 1: Information of RSOs

NORAD ID	Object type	Year	Eccentricity	Incli.[deg]	Period[min]	Perige[km]
17732	DEBRIS	1977-12-21	0.16723	65.8	103.39	778
24843	DEBRIS	1997-06-06	1.02854	63.3	129.74	1222
24871	PAYLOAD	1997-07-09	0.01226	86.4	99.99	749
24907	PAYLOAD	1997-08-21	0.00221	86.4	100.37	773
45859	PAYLOAD	2020-07-04	0.13273	98.2	98.69	600

3.1 Metric

In order to evaluate the performance of error model trained by ML method, an metric of model performance is proposed. Let the predicted error value $\hat{e}$ obtained by ML method be, which can be calculated as $\hat{y} = \psi(\mathcal{X})$. Where $\hat{y}$ is the output value obtained by the machine learning model at the given $\mathcal{X}$. So the residual error can be expressed as Δe. In order to evaluate the performance of error model trained by ML method, an metric of model performance is proposed. Let the predicted error value $\hat{e}$ obtained by ML method be, which can be calculated as $\hat{y} = \psi(\mathcal{X})$. Where $\hat{y}$ is the output value obtained by the machine learning model at the given $\mathcal{X}$. So this residual error Δe can be expressed as

$$\Delta e = e - \hat{e}, \tag{7}$$

The metric PM is used to measure the performance of ML approach in improving OP, which is defined as

$$PM = (1 - \frac{RMS_{\Delta e}}{RMS_e}) \times 100\%, \tag{8}$$

Where $RMS_e = \sqrt{\frac{1}{n}\sum_{k=1}^{n} e_k^2}$, $RMS_{\Delta e} = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \Delta e_k^2}$.In this context, a higher metric value indicates better model training. Specifically, a smaller value

of $RMS_{\Delta e}$ and a closer value of PM to 100% are desirable. However, if the predictive performance of the model is not optimal, the PM value may result in a negative value. This suggests that an incorrect model was established during the training process, leading to poor predictions. Compared to traditional error metrics, PM more directly reflects the accuracy of the model's predictions. It offers a clear and effective method for assessing the model's performance in orbital prediction tasks, aiding in the identification and optimization of the model's predictive capabilities.

3.2 Performance Comparison

Our methodology aims to improve the accuracy and robustness of orbit predictions by combining the efficiency of TLE data with the adaptability of a dynamic loss function. To validate the effectiveness of our proposed methodology, we conducted experiments using real-world RSOs orbit data. In these experiments, we compared the performance of our dynamic TLE-based prediction method with traditional static prediction models. The results of these experiments demonstrated that our method significantly enhances the predictive accuracy of orbits over extended time periods. By dynamically adjusting loss function of prediction models based on evolving orbital conditions, our method outperforms static prediction models. This adaptive approach enables us to achieve enhanced precision in orbit determination, leading to more accurate and reliable RSOs orbit predictions.

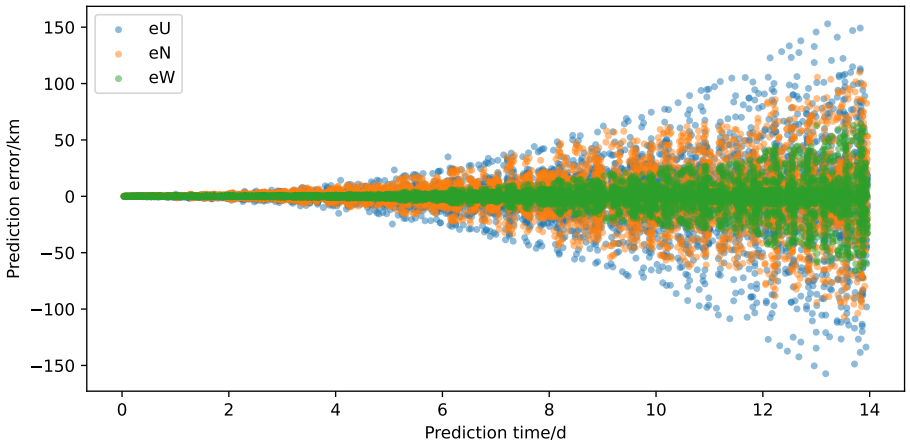


Fig. 3: Orbit prediction error variation of the object 17732 in U, N, and W directions.

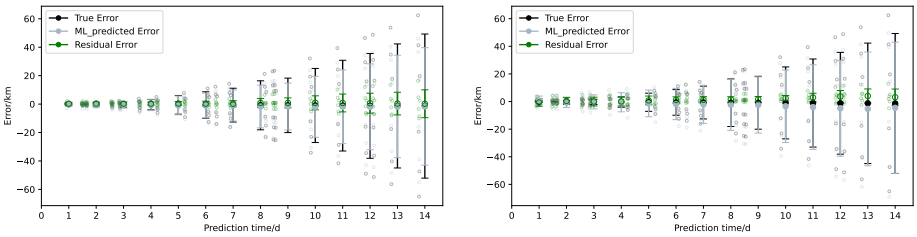
Table 2 presents a summary of the performance of the trained CNN and LSTM models, as well as their combinations with the dynamic loss function

Table 2: Performance metric on RSOs with different ML approaches

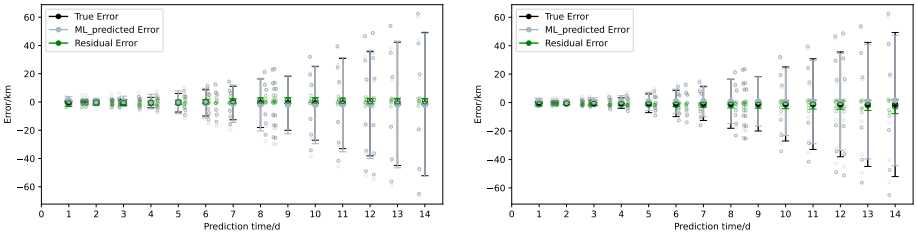
NORAD ID	PM_U		PM_N		PM_W	
	CNN	LSTM	CNN	LSTM	CNN	LSTM
	CNN+DLF	LSTM+DLF	CNN+DLF	LSTM+DLF	CNN+DLF	LSTM+DLF
	CNN+HE	LSTM+HE	CNN+HE	LSTM+HE	CNN+HE	LSTM+HE
17732	84.55	84.50	80.24	88.43	84.98	93.96
	94.99	**94.42**	**96.51**	**94.71**	**95.70**	**94.22**
	82.07	89.91	82.45	84.09	85.78	87.92
24843	85.35	78.39	84.19	77.31	90.16	88.21
	92.94	**90.83**	**93.30**	**91.04**	**93.86**	**90.76**
	75.95	65.25	73.07	77.76	90.73	80.64
24871	94.34	96.27	93.24	94.30	93.43	93.65
	96.70	**96.35**	97.14	**95.70**	**96.64**	**96.16**
	93.31	94.26	**98.14**	95.41	91.38	95.84
24907	95.41	87.03	92.34	83.80	**96.76**	93.55
	96.45	**96.22**	**96.78**	**95.74**	96.52	**94.99**
	96.61	90.24	95.61	95.25	96.35	84.59
45859	91.09	95.92	95.39	93.81	91.63	95.61
	96.54	**97.14**	**96.94**	**97.06**	**96.68**	96.42
	84.49	96.58	85.63	94.47	92.19	**96.94**

(CNN+DLF and LSTM+DLF), in terms of their ability to generalize orbit prediction errors for the next 14 days across five different RSOs. The performance of trained ML model is measured using the metric PM.The evaluation of the models considers three error directions obtained from the decomposition: e_U, e_N, and e_W. From Table 2, it can be observed that the accuracy of error prediction significantly improves with the trained CNN+DLF and LSTM+DLF models compared to their respective single models.

The fluctuation of the error components with prediction time in each direction for object 17732 is shown in Fig. 3, where it is evident that the error increases quickly with increasing prediction time. Because of the considerable influence of atmospheric drag on the orbit, the orbit prediction error typically diverges with forecast time. The results presented in Fig. 4 indicate that the forecast error in the N direction is approximately 100 km at a prediction period of 14 days, the prediction error in the W direction is around 50 km for 14 days, and the prediction error in the U direction is nearly 150 km. Plotting the error prediction time scatter plot for each of the other RSOs is found that U is the main error component for all of them, and the errors in the N and W directions are closer together or are all relatively small, so the analysis that follows focuses on that direction. The performance of CNN, LSTM, and their combined models with the dynamic loss function on five space objects is illustrated in Fig. 4-8. The horizontal axis of each graph represents the prediction time, while the vertical axis represents the error. The black dots represent the true error, the gray dots represent the

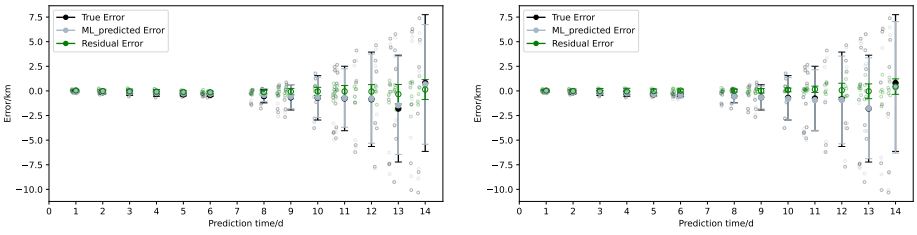


(a) CNN, $PM = 84.55\%$ (b) CNN+DLF, $PM = 94.99\%$

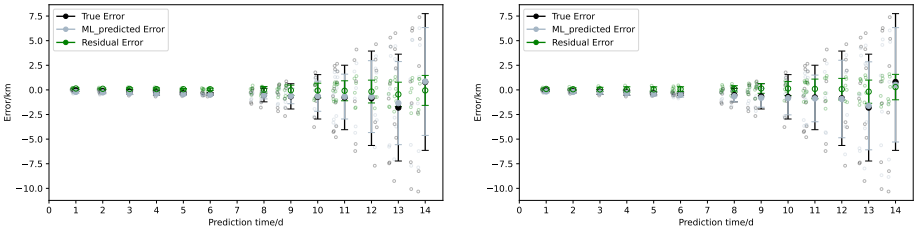


(c) LSTM, $PM = 84.50\%$ (d) LSTM+DLF, $PM = 94.42\%$

Fig. 4: Performance of different ML models for object 17732

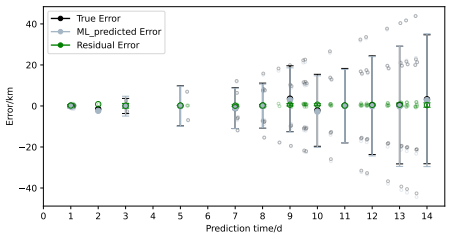


(a) CNN, $PM = 85.35\%$ (b) CNN+DLF, PM $= 92.94\%$

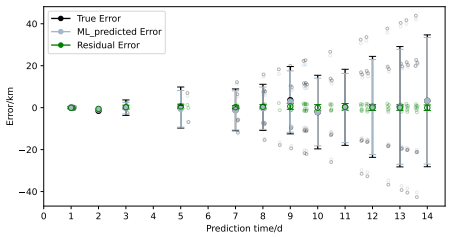


(c) LSTM, $PM = 78.39\%$ (d) LSTM+DLF, $PM = 90.83\%$

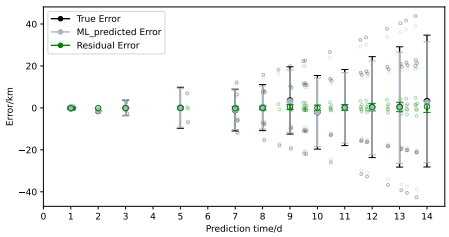
Fig. 5: Performance of different ML models for object 24843



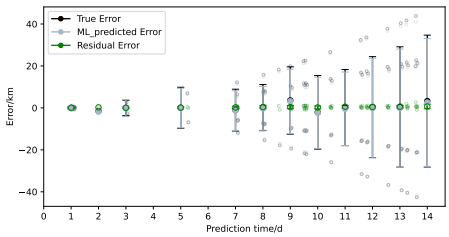
(a) CNN, $PM = 94.34\%$



(b) CNN+DLF, $PM = 96.70\%$

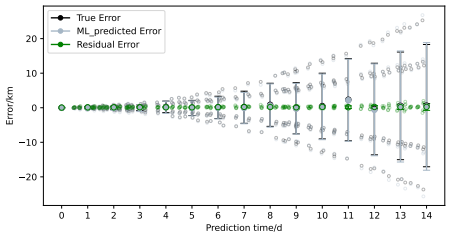


(c) LSTM, $PM = 96.27\%$

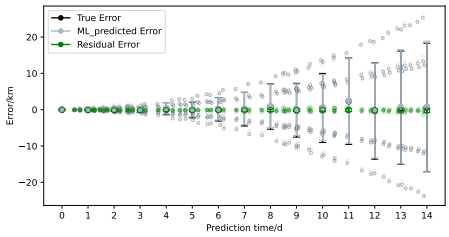


(d) LSTM+DLF, $PM = 96.35\%$

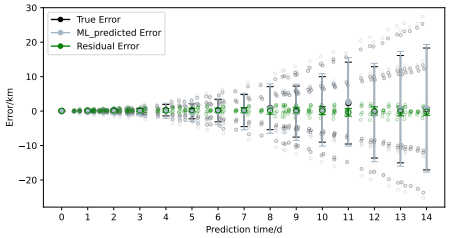
Fig. 6: Performance of different ML models for object 24871



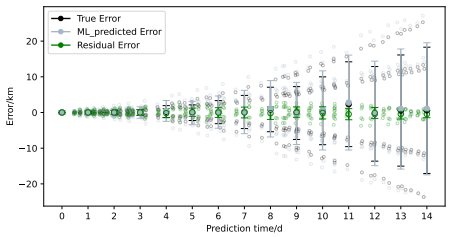
(a) CNN, $PM = 95.41\%$



(b) CNN+DLF, $PM = 96.45\%$

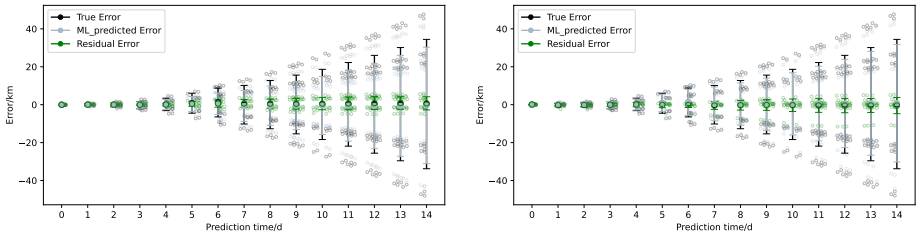


(c) LSTM, $PM = 87.03\%$

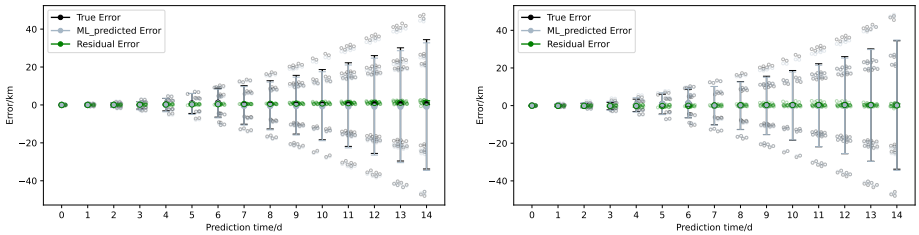


(d) LSTM+DLF, $PM = 96.22\%$

Fig. 7: Performance of different ML models for object 24907



(a) CNN, $PM = 91.09\%$ (b) CNN+DLF, $PM = 96.54\%$



(c) LSTM, $PM = 95.92\%$ (d) LSTM+DLF, $PM = 97.14\%$

Fig. 8: Performance of different ML models for object 45859

prediction error of the machine learning model, and the green dots represent the difference between them. The data points were grouped into 14 categories based on prediction time t. Each group's mean is denoted by a center marker, and its standard deviation is represented by the length from middle to top (or bottom) of the bar chart. When a model effectively captures prediction error patterns, it results in close proximity between prediction error and true error, indicating residual closeness to zero. In Fig. 4, it can be observed that trained ML models excel at extracting potential TLE error patterns and demonstrate strong performance on an error dataset over a period of 14 days with PM exceeding 84%. Specifically, CNN and LSTM models achieve performances of 84.55% and 84.50%, respectively; whereas CNN+DLF and LSTM+DLF show improvements by 10.44% and 9.92%, respectively compared to individual models.The prediction error curves of CNN+DLF and LSTM+DLF in Fig. 5 demonstrate a high level of consistency with the actual prediction error curves, with PM values of 96.45% and 96.22% respectively, representing increases of 1.04% and 9.19% compared to the single model. Fig. 6-8 illustrate that the CNN+DLF and LSTM+DLF models exhibit superior generalization ability in the U direction compared to their single counterparts, resulting in significantly improved PM values.

To demonstrate the efficacy of the dynamic loss function strategy in capturing the prediction error mode of RSOs, this paper discusses a novel approach called the HawkEye loss function. This loss function is designed to proficiently handle outliers and noise in Support Vector Regression (SVR)[21]. As shown in Table

2, the model's performance with the HawkEye loss function improves over the traditional square loss function. For instance, with NORAD ID 24907, the model using the HawkEye loss function achieves PM of 96.61%, 90.24%, 95.61%, and 95.25% in the U and N directions, respectively—surpassing the results obtained using the square loss function. Although the HawkEye loss function enhances the model's performance to some degree, it still falls short compared to the dynamic loss function across the majority of satellites. In general, the HawkEye loss function significantly improves the model's performance when compared to the baseline model.

4 Conclusion

In this paper, we have introduced a novel approach to RSOs orbit prediction by integrating a dynamic loss function into the conventional TLE-based prediction framework. Our proposed method utilizes real-time orbital data to adapt the prediction models, resulting in enhanced accuracy and adaptability of RSOs orbit predictions over extended time periods. The experimental findings demonstrate that our methodology outperforms traditional static prediction models and offers a more reliable solution for precise orbit determination. By utilizing the trained model for prediction error correction in future instances, we can achieve an orbit prediction accuracy of over 90% in the along-track direction for the next 14 days. This significant improvement in accuracy is particularly noteworthy and provides a more precise representation of real-world variations. These findings have important implications for various applications that rely on accurate RSOs orbit predictions, such as space debris monitoring and RSOs collision avoidance. In terms of future research directions, there are several avenues to explore. One promising direction is the integration of advanced machine learning techniques to further enhance the predictive capabilities of TLE-based orbit prediction models. This could involve leveraging deep learning algorithms, recurrent neural networks, or other state-of-the-art methodologies to capture complex temporal dependencies and improve prediction accuracy. Additionally, investigating the impact of external factors such as solar activity, atmospheric conditions, and gravitational perturbations on orbit prediction could lead to more robust and comprehensive models. Incorporating these factors into the prediction framework may allow for better long-term predictions and enable accounting for unexpected changes in RSOs orbits.Overall, our research contributes to advancing RSOs orbit prediction methodologies and opens up possibilities for enhanced accuracy and reliability in future predictions.

Acknowledgements. This work was supported by Education Science Planning Project of Hubei Province (2020GB198) and Natural Science Foundation of Hubei Province (2023AFB523).

References

1. McDowell, J.C., 2020. The low earth orbit RSOs population and impacts of the spacex starlink constellation. Astrophys. J. Lett. 892 (2), L36
2. MONTENBRUCK O. Numerical integration methods for orbital motion[J]. Celestial Mechanics and Dynamical Astronomy 1992, 53(1):59-69
3. Kim H K, Han C Y. Analytical and numerical approaches of a solar array thermal analysis in a low-earth orbit RSOs[J]. Advances in Space Research, 2010, 46(11): 1427-1439
4. Peng H, Bai X. Machine learning approach to improve RSOs orbit prediction accuracy using publicly available data[J]. The Journal of the astronautical sciences, 2020, 67(2): 762-793
5. Hoots F R, Roehrich R L. Models for propagation of NORAD element sets[M]. Office of Astrodynamics, 1980, 22(1), 33-50
6. Peng H, Bai X. Artificial neural network–based machine learning approach to improve orbit prediction accuracy[J]. Journal of Spacecraft and Rockets, 2018, 55(5): 1248-1260.
7. Peng H, Bai X. Exploring capability of support vector machine for improving satellite orbit prediction accuracy[J]. Journal of Aerospace Information Systems, 2018, 15(6): 366-381.
8. Peng H, Bai X. Improving orbit prediction accuracy through supervised machine learning[J]. Advances in Space Research, 2018, 61(10): 2628-2646
9. Peng H, Bai X. Gaussian processes for improving orbit prediction accuracy[J]. Acta astronautica, 2019, 161: 44-56
10. Peng H, Bai X. Fusion of a machine learning approach and classical orbit predictions[J]. Acta astronautica, 2021, 184: 222-240
11. Li B, Zhang Y, Huang J, et al. Improved orbit predictions using two-line elements through error pattern mining and transferring[J]. Acta Astronautica, 2021, 188: 405-415
12. Jadala G, Meedinti G N, Delhibabu R. RSOs Orbit Prediction Using a Machine Learning Approach[C]//ICAI Workshops. 2022: 28-46
13. Khorana R. Low-Earth Satellite Orbit Determination Using Deep Convolutional Networks with Satellite Imagery[J]. arXiv preprint arXiv:2305.12286, 2023
14. Ferrer E, Ruiz-De-Azua J A, Betorz F, et al. Inter-Satellite Link Prediction with Supervised Learning Based on Kepler and SGP4 Orbits[J]. International Journal of Computational Intelligence Systems, 2024, 17(1): 217
15. Wu L, Tian F, Xia Y, et al. Learning to teach with dynamic loss functions[J]. Advances in neural information processing systems, 2018, 31
16. Vallado D A, Virgili B B, Flohrer T. Improved SSA through orbit determination of two-line element sets[C]//ESA Space Debris Conference. 2013: 1-7
17. D. A. Vallado Fundamentals of Astrodynamics and Applications, vol. 12. Berlin, Germany: Springer, 2001
18. J.T. Horwood, N.D. Aragon, A.B. Poore, Gaussian sum filters for space surveillance: Theory and simulations, J. Guid. Control Dyn. 34 (6) (2011) 1839–1851
19. Levit C, Marshall W. Improved orbit predictions using two-line elements[J]. Advances in Space Research, 2011, 47(7): 1107-1115
20. Bai X Z, Chen L, Tang G J. Periodicity characterization of orbital prediction error and Poisson series fitting[J]. Advances in space research, 2012, 50(5): 560-575
21. Akhtar M, Tanveer M, Arshad M. HawkEye: Advancing robust regression with bounded, smooth, and insensitive loss function[J]. arXiv preprint arXiv:2401.16785, 2024.