

Machine Learning Approach to Improve Satellite Orbit Prediction Accuracy Using Publicly Available Data*

Hao Peng · Xiaoli Bai*

Abstract Efficient and high precision orbit prediction is increasingly crucial for improved Space Situational Awareness. Due to the lack of the required information such as space environment conditions and characteristics of Resident Space Objects (RSOs), satellite collisions have happened, partially because that the solely physics-based approaches can fail to achieve the required accuracy for collision avoidance. With the hypothesis that a Machine Learning (ML) approach can learn the underlying pattern of the orbit prediction errors from historical data, in this paper, the Support Vector Machine (SVM) is explored for improving the orbit prediction accuracy. Two publicly available Two-Line Element (TLE) catalog and International Laser Ranging Service (ILRS) catalog are used to validate the proposed ML approach. The position and velocity components of 11 total RSOs maintained at both catalogs are studied. Results of the study demonstrate that the designed dataset structure and SVM model can improve the orbit prediction accuracy with good performance on most cases. The performance on RSOs belonging to different orbit types is analyzed using different sizes of training and testing data. Results of the paper demonstrate the potential of using the proposed ML approach to improve the accuracy of TLE catalog.

Keywords Space Situational Awareness · Orbit Prediction Accuracy · Two-Line Element · Machine Learning · Support Vector Machine

1 Introduction

As of January 4, 2018, there are 18835 Resident Space Objects (RSOs) orbiting the Earth, including 1683 payloads and 14152 rocket bodies and debris [1]. The ability to generate accurate and timely predictions of RSOs' trajectories is the cornerstone of many current and future Space Situational Awareness (SSA) operations, including collision avoidance and observation scheduling. There have been satellite collision events due to inaccurate predictions and false alarms [2]. This is partially because the current orbit prediction practice that solely grounds on physics-based models can fail to achieve the required accuracy for collision avoidance. Fundamentally, the failure is due to the lack of the required information such as the state of the space object at the start of trajectory computation, the environment conditions such as atmospheric drag and solar radiation pressure, RSOs' body characteristics such as the mass, geometry, and material, as well as the intent information for the maneuvering objects.

* Dedicated to Dr. John L. Junkins on the occasion of his seventy-fifth birthday.

H. Peng
Postdoctoral Associate
E-mail: astro.H.Peng@gmail.com

X. Bai (*Corresponding author)
Assistant Professor
Tel.: (848)445-4760
E-mail: xiaoli.bai@rutgers.edu

98 Brett Rd, Piscataway, NJ 08854
Department of Mechanical and Aerospace Engineering
Rutgers, The State University of New Jersey

Machine learning (ML) presents a different modeling and prediction avenue compared with the physics-based approach. The prediction can be made without explicitly modeling shape of RSOs, maneuvers, or space environment. Instead, the models are learned based on large amount of historical data, similar to human cognition in learning from past experience to predict future events. Machine learning methods have shown great capability in a wide range of applications [3, 4] with many in the aerospace area [5–7]. Recently, Peng and Bai have developed a methodology to improve orbit prediction accuracy by learning from historical prediction errors. Using a simulation-based space catalog environment as the test bed, they have explored three types of generalization capability for the proposed ML approach [8, 9], and investigated the limitations when using the support vector machine (SVM) models for improving orbit prediction accuracy [10]. Additionally, they have discovered that the information of the area-to-mass ratio (AMR) can be recovered from the consistency error between two estimated states in the historical data [11].

Considering the varying space environment, it is easy to be critical about these results. On one hand, the results are obtained using a simulation environment. More importantly, RSOs are influenced by both conservative and non-conservative forces and their orbits are dynamically evolving. So the problem of learning orbit prediction errors is different from, and arguably more challenging than many other machine learning applications. Nonetheless, our hypothesis is that it is possible to learn some of the information because the prediction errors are implicitly contained in RSOs' historical data, such as the measurements, the estimation, and the prediction errors.

In this paper, using publicly available resources including the Two-Line Element (TLE) data and the International Laser Ranging Service (ILRS) data, the ML approach, in particular the SVM models, is demonstrated to be able to learn the orbit prediction errors. TLE data is published by the US Strategic Command (USSTRATCOM) at the Space Track website¹ and is the largest publicly available catalog of space objects [12]. On the other hand, the ILRS data provides various services including high-accuracy satellite ephemerides of tracked RSOs and are published regularly as CPF files since 2005 [13]. All tracked RSOs are equipped with retroreflectors, and the stations of ILRS collect the laser ranging data, generate data products, and then distribute them to users. A drawback of the ILRS catalog is that it covers much less RSOs than the TLE catalog.

The goal in this paper is to present a novel and potentially more capable approach to improve TLE accuracy using only publicly available data sources. It is known that the TLE catalog does not have adequate accuracy [14]. Finkleman has discussed the limitations of the current TLE format and proposed solutions to include the variance and covariance information of the generated TLE set [15]. There are many studies on assessing the accuracy of the TLE catalog, characterizing the covariance of the TLE data, or extracting more information about the TLE set [16–24]. Früh and Schildknecht have assessed the accuracy of the TLE sets of RSOs in geostationary and high-eccentricity orbits, using the optical observations as the reference true orbit of the corresponding RSOs [22]. Lemmens and Krag have proposed two approaches to detect maneuvers from the TLE catalog using only the consistency error [24]. Muldoon et al. have developed a data-driven model based on historical TLE sets to generate predictions with accuracy comparable with Simplified General Perturbations-4 (SGP4) model and the methods are tested on 397 RSOs in low Earth orbit (LEO) [25]. Levit and Marshall have proposed a method to use successive TLEs as pseudo-observations which are then fitted by least square estimators with a high-precision model [26]. Their methods are validated against the precision orbital ephemeris from ILRS with significant improvement over the standard SGP4 model. However, as acknowledged by the authors, this method cannot discover the bias without external information. Hejduk et al. have shown that it is possible to improve orbit prediction by comparing the high-accuracy catalog based on higher-order analytical theories to the TLE catalog, which could then incorporate external higher-accuracy information into the TLE catalog [27]. Sang and Li have shown improving the TLE prediction by introducing three error correction functions, which depend on prediction time and are constructed based on observation of the periodicity of prediction errors [28]. Although the approach is tested on four satellites with precise laser ranging data, its generalization capability to other RSOs is not considered in the paper. Rivera and Bai have demonstrated the capability of generalized linear learning model to improve the orbit prediction based on the analysis of the consistency of the TLE data [29]. The bias of the prediction can be removed, as most other researches are capable of, but the standard deviation or the periodicity of the error cannot be removed.

Different from these existing studies, in this paper, the ML approach is explored to improve the orbit prediction accuracy of RSOs under practical situations. Using 11 total RSOs tracked by both the TLE and ILRS catalogs, the generalization capability of the ML approach is investigated systematically. The numerical results in this paper show that: 1) the trained ML models can be successfully generalized to future epochs; 2) good generalization performance can be achieved if adequate training data size is used; 3) the performance is

¹ <http://www.space-track.org/>

stable on a range of situations of the testing data in the future; and 4) the designed ML approach is applicable to all the investigated RSOs in the SSO, LEO, and MEO regions.

With the goal to demonstrate the ML approach for improving orbit prediction accuracy on the TLE and ILRS catalogs, the remaining part of the paper is organized as follows. In Sec. 2, background of the ML approach and the specific ML algorithm are described. In Sec. 3, design of the ML approach for the TLE and ILRS catalogs is presented in detail. In Sec. 4, performance of the ML approach is first analyzed with a particular RSO in detail and then summarized on all RSOs investigated. Discussions and insights on the results are also provided. In Sec. 5, conclusions are summarized and future researches are suggested.

2 Background

In this section, the methodology of introducing the machine learning approach into the orbit prediction problem is presented. First, the model of orbit prediction is presented and the dataset used for the support vector machine is introduced. Second, the support vector machine is briefly reviewed, so that readers with limited background of machine learning can understand the discussions in the paper.

2.1 Machine Learning Approach to Improve Orbit Prediction Accuracy

Orbit prediction is commonly generated by propagating the current state of the RSO using a particular dynamic model to a desired future epoch. Orbit prediction errors are usually inevitable, because the dynamic model is an approximation of the real physics, and the estimated state has errors due to both the measurements and the estimation algorithm. Common ways to improve the accuracy of the orbit prediction include: improving dynamic models to be closer to the truth, improving observation equipment, and developing more capable estimation algorithms. There have been many studies along these topics, but orbit prediction is still challenging because many forces such as atmosphere drag and solar radiation pressure still cannot be modeled with high precision. Additionally, non-conservative forces also depend on the RSO's attitude and specific parameters which often are unavailable.

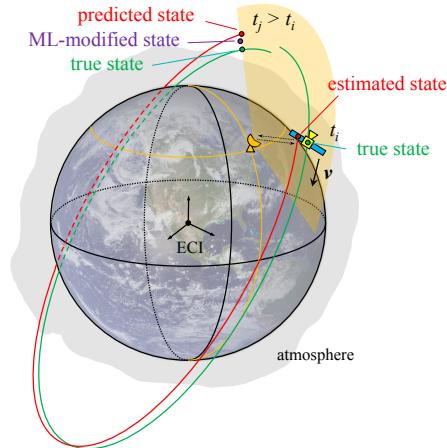


Fig. 1 Using the ML approach to directly modify the orbit prediction at a future epoch.

The proposed ML approach directly modifies the predicted state so that it is closer to the true orbit. This concept is illustrated in Fig. 1. At t_i , the “estimated state” does not coincide with the “true state”. As a consequence, at a future epoch t_j , the “predicted state” based on the estimate and an assumed dynamic model will also deviate from the “true state”. On the other hand, the “ML-modified state” is expected to be closer to the true state. Notice that this approach is different from other approaches which identify the pattern of the prediction errors using accurate dynamic models and then analyze the error dependency, such as in [16]. One benefit of the proposed ML approach is that the learning is only tasked with finding the incremental corrections to the physics-based prediction, which reduces the dimensionality of the learning task. Additionally,

since the ML approach directly works on orbit predictions, it can be established based on the state-of-art dynamic models, and orbit determination and prediction techniques. Furthermore, it can be embedded into existing orbit prediction frameworks without modifying current procedures.

The goal to improve orbit predictions is modeled as a supervised learning problem. The supervised learning refers to the ML methods that learn a function or mapping from labeled data [3]. One “labeled data” consists of a set of learning variables and the corresponding target variables.

At a specific epoch t_i , there are various available information, such as estimates of states, various measurements, maneuvering information at the current and all previous epochs, and also the information that can be generated from them. Theoretically, all relevant information can be chosen as the learning variables, denoted as $\Lambda(t_i)$. The corresponding target variables for the ML approach are chosen as the true orbit prediction error $e_T(t_j; t_i)$, defined as

$$e_T(t_j; t_i) = \mathbf{X}_T(t_j) - \hat{\mathbf{X}}(t_j; t_i), \quad (1)$$

where $\mathbf{X}_T(t_j)$ represents the true state at t_j and $\hat{\mathbf{X}}(t_j; t_i)$ represents the predicted state at t_j based on the estimate $\hat{\mathbf{X}}(t_i)$ at t_i ($t_j > t_i$). In the dataset designed for the ML approach, one data point can be expressed as (Λ, e_T) , where the time variables are omitted for clarity. Depending on the available RSO catalogs and orbital information, different dataset structures can be designed. The design details for the two catalogs used in this paper will be explained in the following sections.

When the ML model is trained, it will produce an approximation $\hat{e}_{ML}(t_j; t_i)$ of the orbit prediction error $e_T(t_j; t_i)$, defined as $\hat{e}_{ML}(t_j; t_i)$. This approximation is then used to generate the “ML-modified state” (see Fig. 1) as

$$\begin{aligned} \hat{\mathbf{X}}_{ML}(t_j; t_i) &= \hat{\mathbf{X}}(t_j; t_i) + \hat{e}_{ML}(t_j; t_i) \\ &= \mathbf{X}_T(t_j) - e_T(t_j; t_i) + \hat{e}_{ML}(t_j; t_i), \end{aligned} \quad (2)$$

where Eq. (1) has been used to substitute $\hat{\mathbf{X}}(t_j; t_i)$ at the second row of Eq. (2). Therefore, if the ML model can approximate $e_T(t_j; t_i)$ successfully, the ML-modified prediction $\hat{\mathbf{X}}_{ML}(t_j; t_i)$ will be closer to the true state $\mathbf{X}_T(t_j)$.

In summary, the proposed ML approach is a methodology to design a dataset based on the RSO catalogs, train ML models to discover the relationship between available information (learning variables) and the orbit prediction error (target variables), and then apply the trained ML models to the conventional orbit prediction to improve the prediction accuracy.

2.2 Support Vector Machine Models

The support vector machine (SVM) method to be used for this study can be used for both classification and regression problems. One strength of the SVM is that they are nonparametric techniques, with no need to specify the basis functions a priori. The SVM regression can handle nonlinear problems since it relies on kernel functions. Moreover, the SVM method has universal approximation capability with various kernels [30], including the Gaussian kernel. The basic concept of the SVM is briefly reviewed below to introduce the proposed ML approach.

Suppose the dataset \mathcal{X} of n labeled data pairs as $\{(\Lambda_1, e_1), (\Lambda_2, e_2), \dots, (\Lambda_n, e_n)\}$, where $\Lambda_i \in \mathbb{R}^m$ represents m learning variables, and $e_i \in \mathbb{R}$ is one component of the orbit prediction error e_T . The ε -SVM regression method aims to find a function $f(\Lambda)$ that has at most a deviation of ε with respect to e for all the training data. In the linear case, the desired function has the form

$$f(\Lambda) = \langle \omega, \Lambda \rangle + b, \quad (3)$$

where $\omega \in \mathbb{R}^m$ is the weight, $b \in \mathbb{R}$ is the bias, and $\langle \cdot, \cdot \rangle$ represents the inner product in \mathbb{R}^m . Then the training task is to find the flattest function in \mathbb{R}^m , which can be represented as a convex optimization problem to minimize a cost function

$$J(\omega) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*), \quad (4)$$

subject to the following constraints for all $i = 1, \dots, n$

$$\begin{aligned} e_i - \langle \omega, \Lambda_i \rangle - b &\leq \varepsilon + \xi_i, \\ \langle \omega, \Lambda_i \rangle + b - e_i &\leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0, \end{aligned} \quad (5)$$

where ξ_i, ξ_i^* are slack variables used to accommodate possible outliers in the training data that have deviations larger than ε , and $C > 0$ is the box constraint representing the trade-off between the flatness of $f(\Lambda)$ and the tolerance of deviations larger than ε . Then, the Lagrangian function is constructed as

$$\begin{aligned} L = & \frac{1}{2} \|\omega\| + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n \eta_i \xi_i - \sum_{i=1}^n \eta_i^* \xi_i^* \\ & - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i - e_i + \langle \omega, \Lambda_i \rangle + b) - \sum_{i=1}^n \alpha_i^* (\varepsilon + \xi_i^* + e_i - \langle \omega, \Lambda_i \rangle - b), \end{aligned} \quad (6)$$

where $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$ are Lagrange multipliers. The KKT conditions are

$$\frac{\partial L}{b} = \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0 \quad (7)$$

$$\frac{\partial L}{\omega} = \omega - \sum_{i=1}^n (\alpha_i^* - \alpha_i) \Lambda_i = 0 \quad (8)$$

$$\frac{\partial L}{\xi_i} = C - \alpha_i - \eta_i = 0 \quad (9)$$

$$\frac{\partial L}{\xi_i^*} = C - \alpha_i^* - \eta_i^* = 0 \quad (10)$$

After solving for ω, η_i, η_i^* and substituting them back into Eq. (6), the dual optimization problem is to maximize

$$\begin{aligned} J'(\alpha_i, \alpha_i^*) = & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \Lambda_i, \Lambda_j \rangle \\ & - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n e_i (\alpha_i - \alpha_i^*), \end{aligned} \quad (11)$$

subject to Eq. (7) and $\alpha_i, \alpha_i^* \in [0, C]$. Solve the dual problem for $\alpha = \{\alpha_i, \alpha_i^*\}$ and Eq. (8) for ω , substitute the results into Eq. (3), then we will have the support vector expansion of the regression function $f(x)$ as

$$f(\Lambda) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \Lambda_i, \Lambda \rangle + b. \quad (12)$$

Notice that in the above equation and the dual problem in Eq. (11), only the inner product of Λ is necessary and the explicit expression of the weight ω is not required. This feature of the SVM makes it possible to deal with nonlinear regressions via kernel functions. Substituting the inner product $\langle \cdot, \cdot \rangle$ by a kernel function $k(\cdot, \cdot)$, the optimization problem is modified to find the flattest function in the feature space indicated by the kernel, and the new regression function is expressed as

$$f(\Lambda) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(\Lambda_i, \Lambda) + b. \quad (13)$$

When given a new learning variable Λ_{test} , according to Eq. (13), the kernel $k(\cdot, \cdot)$ and the solved dual variables α are used to generate $f(\mathbf{x}_{\text{test}})$, which is the output of the trained SVM regression model according to the input data.

More details about the SVM theory are referred to the reference [31, 32] and the references therein. In this paper, the SVM regression function `fitrsvm` in MATLAB is used to implement the proposed ML approach to improve orbit prediction accuracy.

3 The Machine Learning Approach for the TLE Catalog

3.1 TLE and ILRS catalogs

The TLE catalog provides orbit information of RSOs larger than 10 centimeter diameter. The TLE data used in this paper are collected from Space-Track website. The data then are parsed and propagated using the SGP4 model implemented by Orekit, which is a fully tested space dynamics library in Java [33].

There exist other resources that can provide orbital information of RSOs with higher accuracy. For example, operational RSOs may have on-board GPS equipment with the capability of automatic navigation, which should be more accurate than their TLE sets. But these high-accuracy data only belong to the operator and are not publicly available. In this paper, the publicly available data from the International Laser Ranging Service (ILRS) is used. The ILRS provides several kinds of data service, including precise satellite ephemerides for tracked RSOs [13]. The precise ephemeris of a tracked RSO is published in the Consolidated Prediction Format (CPF) file [34] which can be accessed from Crustal Dynamics Data Information System (CDDIS) or EUROLAS Data Center (EDC).

As described in the previous section, the true state of a RSO is required to obtain the true error, which will be used as the target variables for the ML approach. In previous studies carried out in the simulation environment[8–10], the true orbit of a RSO is always available. For the TLE catalog in this paper, the data from ILRS which has higher accuracy will be used as the “true state”. Meanwhile, the orbit prediction using TLE data is chosen to be the estimated state with errors which are expected to be reduced by the ML approach.

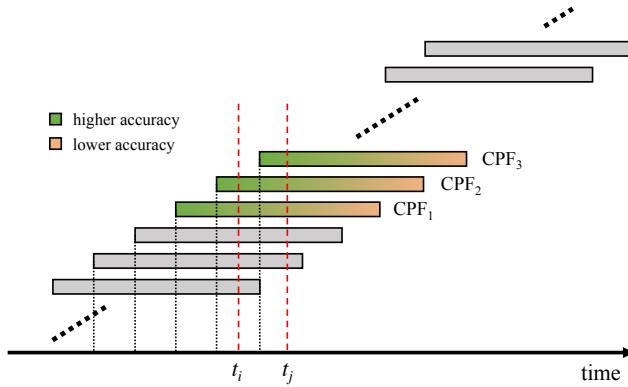


Fig. 2 Illustration of series of CPF files for one RSO. The most updated CPF file will be interpolated for use.

A specific CPF file contains several days of accurate orbit predictions, with a constant step size of several minutes, and the state is represented in the International Terrestrial Reference Frame (ITRF). These data are generated by propagating the RSO using their most accurate force models with a precisely determined initial state. According to the official documents [34], a 10-point baseline Lagrangian interpolator is required to obtain a RSO’s state at an arbitrary epoch covered by the used CPF file, and the interpolation epoch should fall between the middle of the two center points for the best accuracy. The CDDIS website has provided a sample code² that implements such an interpolator. As illustrated in Fig. 2, CPF files are published regularly and each file contains predictions in a certain future time interval, so there are overlapping time intervals among a series of CPF files. With the assumption that the prediction accuracy decreases as the prediction duration increases, the most recently updated CPF file is used when more than one CPF file is available for one epoch. For example, in Fig. 2, CPF₂ is used at t_i while CPF₃ is used at t_j .

² https://ilrs.cddis.eosdis.nasa.gov/docs/2017/cpf_sample_code_v1.01d.tgz, retrieved on 2018/04/17.

3.2 Resident Space Objects to Study

Currently, the CPF catalog tracks only 68 RSOs³, and at past it has tracked other 149 RSOs⁴. In this paper, from all these 217 RSOs, the objects to study are chosen based on three criteria that in the year of interest: 1) they are passive objects without maneuver capabilities; 2) they have consecutive CPF data; and 3) they have adequate TLE and CPF data pairs. The resulting 11 RSOs are summarized in Table 1. The orbital inclination and eccentricity are mean values extracted from all the TLE sets in the chosen year. The beginning RAAN is the RAAN of the first TLE set in the chosen year. The term “Usable TLEs” refers to the remaining TLE sets after removing outliers. When “Varying B*” is “no” it means the B* of the chosen satellite in the TLE catalog is a constant during the year, which could occur for both SSO and LEO. The CPF agent indicates the source of the CPF data, where “HTS” refers to NASA GSFC SLR Misson Contractor⁵. We note that there are many RSOs that we cannot determine their maneuver capabilities and are therefore not included in this study.

Table 1 Parameters and initial states of the RSOs studied to validate the ML approach.

Case #	NORAD ID	Satellite Name	Orbit Type	Year	Usable TLEs	Semi-major Axis [km]	Eccentricity	Incli. [deg]	RAAN [deg]	Period [min]	Varing B*	CPF Agent
1	22824	STELLA	SSO	2008	559	7185.1	0.0013	98.4	-25.1	101.0	yes	HTS
2	27944	LARETS	SSO	2008	493	7068.7	0.00156	98.0	-98.4	98.5	yes	HTS
3	35871	BLITS	SSO	2010	796	7204.4	0.00122	98.7	70.7	101.4	no	HTS
4	16908	AJISAI	LEO	2008	597	7871.5	0.00143	50.0	-46.5	115.8	no	HTS
5	38077	LARES	LEO	2013	983	7827.7	0.00101	69.5	23.7	114.8	no	HTS
6	1328	BEACON-C	LEO	2008	534	7499.8	0.02518	41.2	-17.3	107.7	yes	HTS
7	7646	STARLETTE	LEO	2008	619	7340.0	0.02056	49.8	23.0	104.3	yes	HTS
8	8820	LAGEOS-1	MEO	2008	471	12274.8	0.00393	109.9	34.3	225.4	no	HTS
9	22195	LAGEOS-2	MEO	2008	477	12165.5	0.01346	52.7	-161.1	222.4	no	HTS
10	19751	ETALON-1	MEO	2008	496	25503.8	0.00143	65.1	-56.7	675.2	no	HTS
11	20026	ETALON-2	MEO	2008	506	25499.7	0.00122	64.4	179.8	675.0	no	HTS

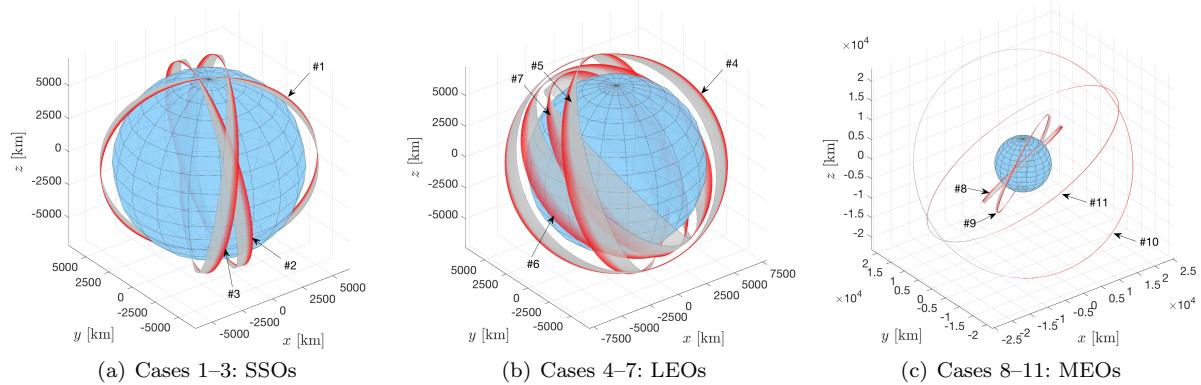


Fig. 3 RSOs’ orbits in the first ten days, extracted from TLE sets and shown in the ECI frame.

Figure 3 shows the orbits of these chosen RSOs in the first ten days of the study, using states extracted from TLE sets and represented in the Earth-centered inertial (ECI) frame. The orbits are precessing along the direction from the gray to the red revolutions in the figure. Although these RSOs are very different from each other, it will be demonstrated next that the same ML approach can be applied to all of them while achieving satisfying performance. This capability is due to the data-driven feature of the proposed ML approach. The hidden relationship between learning and target variables that are intractable to conventional methods can potentially be recovered from historical data by the ML model.

³ https://ilrs.cddis.eosdis.nasa.gov/missions/satellite_missions/current_missions/index.html, retrieved on 2018/04/19.

⁴ https://ilrs.cddis.eosdis.nasa.gov/missions/satellite_missions/past_missions/index.html, retrieved on 2018/04/19.

⁵ https://ilrs.cddis.eosdis.nasa.gov/data_and_products/predictions/prediction_centers.html, retrieved on 2018/04/17.

3.3 Dataset Structure Design

In this section, design of the dataset structure for the ML approach is presented. It is important that there exists a relationship between the designed learning and target variables in the dataset such that the ML models can be trained to capture it. Due to the lack of a rigorous theory about the hidden relationship which is not modeled by the dynamic model, our design procedure is based on a trial-and-error process.

An important concern in machine learning applications is to evaluate the performance of the trained model on a different set of data, which is not included in the training data. In our previous publication [8], it has been discovered that there exist three generalization capabilities for the ML approach to improve orbit prediction accuracy. As shown in Fig. 4, the type I generalization only requires that the testing data has not been used during the training, which is a common practice for machine learning studies. In this paper, the testing data is further required to occur in the future epoch with respect to the training data, corresponding to the type II generalization capability in Fig. 4. This represents the actual orbit prediction operations. The type III generalization capability further requires the test objects are different from the training objects which will be studied in our future studies.

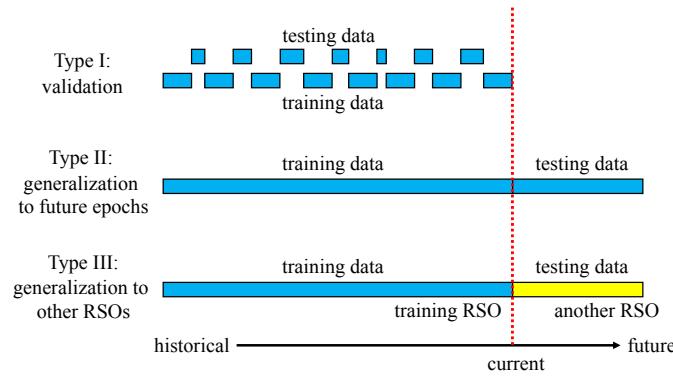


Fig. 4 Illustration of three types of generalization capabilities in [8]. Type II is studied in this paper for the TLE catalog.

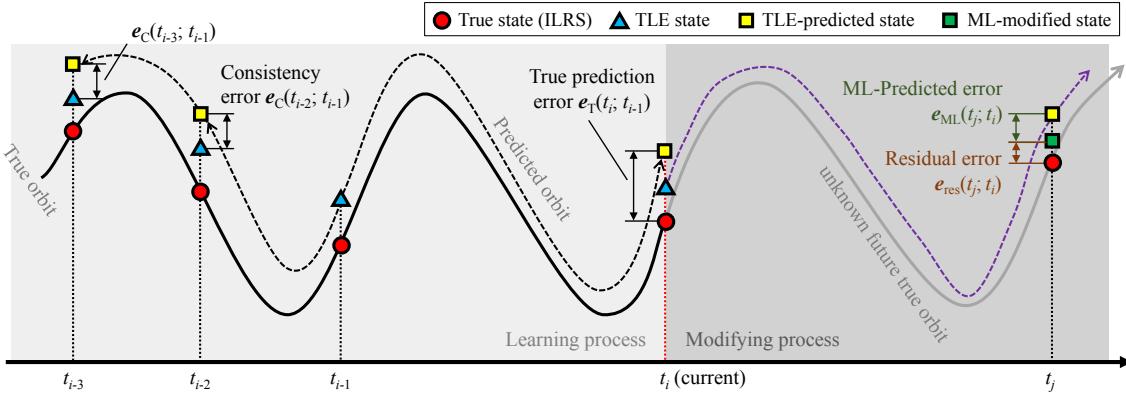


Fig. 5 Illustration of the dataset structure for the ML approach to improve orbit prediction accuracy.

Figure 5 illustrates the key concepts of the dataset structure design for the ML approach. The horizontal axis represents the time t . The dark solid curve represents the true orbit, which is approximated by CPF data as discussed before, and the dashed curves represent the predicted orbits based on the TLE sets. Different markers on the curves represent different states of the RSO, including the true, estimated, predicted and ML-modified states as shown in the figure. The light gray frame on the left side of Fig. 5 illustrates the learning process of the ML approach using historical data. All the TLE sets falling into the learning window are used to generate the training data for the SVM model. The darker gray frame on the right illustrates the modifying process

using the trained SVM model. The dataset structure should be designed to ensure that both learning and target variables are available during the learning process, and all the learning variables are available during the modifying process.

For the learning process in Fig. 5, using historical epochs t_{i-3} , t_{i-2} , t_{i-1} , and t_i as examples, we explore the dataset structure here. At t_{i-1} , the available TLE state $\hat{\mathbf{X}}_{\text{TLE}}(t_{i-1})$ is propagated using SGP4 model to its future epoch t_i , resulting the predicted state $\hat{\mathbf{X}}(t_i; t_{i-1})$. At t_i , the true prediction error between the predicted and the true states is recorded as the target variable, denoted by $\mathbf{e}_T(t_i; t_{i-1}) = \mathbf{X}_T(t_i) - \hat{\mathbf{X}}(t_i; t_{i-1})$. This is the error that is expected to be reduced from the orbit predictions.

The TLE state $\hat{\mathbf{X}}_{\text{TLE}}(t_{i-1})$ is also backwardly propagated to its past epochs t_{i-2} and t_{i-3} to generate the propagated states $\hat{\mathbf{X}}(t_{i-2}; t_{i-1})$ and $\hat{\mathbf{X}}(t_{i-3}; t_{i-1})$. At t_{i-2} , the propagated states are compared with the corresponding TLE states, generating the consistency errors $\mathbf{e}_c(t_{i-2}; t_{i-1}) = \hat{\mathbf{X}}_{\text{TLE}}(t_{i-1}) - \hat{\mathbf{X}}(t_{i-2}; t_{i-1})$. Similarly, we have $\mathbf{e}_c(t_{i-3}, t_{i-1})$ and even earlier consistency errors. These errors are expressed as Cartesian coordinates in the RSW frame as ${}^{\text{RSW}}\mathbf{e}_\xi$, where x -axis is along with the radial direction, the y -axis (along-track direction) is perpendicular to the x -axis in the orbital plane and points to the inertial velocity direction, and the z axis (cross-track direction) is along with the angular momentum direction [35]. The TLE set also provides a coefficient B^* , which is used in the SGP4 model to accommodate the drag effect [14]. This information is also assumed to relate to the orbit prediction error.

In summary, suppose k previous TLE sets are used to calculate consistency errors, the leaning variables $\Lambda(t_{i-1})$ for the ML approach at the epoch t_{i-1} consist of:

- Prediction duration $\Delta t = t_i - t_{i-1}$ to its future epoch t_i ;
- Consistency errors $\{\dots, \delta t_k, \mathbf{e}_c(t_k; t_{i-1}), \dots\}$, where $t_k < t_{i-1}$, $\delta t_k = t_{i-1} - t_k$ is the backward propagation duration;
- Drag information $B^*(t_{i-1})$ and its consistency deviation $\{\dots, \delta B^*(t_k), \dots\}$, where $\delta B^*(t_k) = B^*(t_k) - B^*(t_{i-1})$.

The target variable consists of:

- True orbit prediction error at the future epoch t_i (with respect to t_{i-1}), expressed as $\mathbf{e}_T(t_i; t_{i-1}) = [e_x, e_y, e_z, e_{vx}, e_{vy}, e_{vz}]^\top$ in the RSW frame centered at the predicted state $\hat{\mathbf{X}}_{\text{TLE}}(t_i; t_{i-1})$.

Finally, the whole dataset \mathcal{X} is organized as

$$\begin{aligned} \mathcal{X} = \{ &\dots && \dots && \dots && \dots \\ &(\Lambda(t_{i-1}), \mathbf{e}_T(t_{j-1})), &(\Lambda(t_{i-1}), \mathbf{e}_T(t_j)), &(\Lambda(t_{i-1}), \mathbf{e}_T(t_{j+1})), &\dots \\ &(\Lambda(t_i), \mathbf{e}_T(t_j)), &(\Lambda(t_i), \mathbf{e}_T(t_{j+1})), &\dots && (14) \\ &(\Lambda(t_{i+1}), \mathbf{e}_T(t_{j+1})), &\dots && \dots \\ &&&&&& \dots \}, \end{aligned}$$

where the i -th row contains data from the epoch t_i to all the following epochs t_j of the j -th column. Since the target variable \mathbf{e}_T is a six-dimensional vector, totally six SVM models will be trained for each component e_ξ with $\xi \in \{x, y, z, vx, vy, vz\}$.

In this paper, the maximum of the prediction duration Δt of a TLE set is constrained to 14 days; and eight past TLE sets are used to generate consistency errors based on experimental results. Denote the epoch of the TLE set as t_0 and the epochs of chosen TLE sets as t_{-1}, \dots, t_{-8} . The following constraints are designed to the contiguous time intervals

$$\begin{aligned} t_0 - t_{-1} &> 0 \\ t_{-1} - t_{-2} &> 0.5 \text{ day} \\ t_{-2} - t_{-3} &> 1 \text{ day} \\ t_l - t_{l-1} &> 2 \text{ days} \quad l = -3, \dots, -7. \end{aligned} \tag{15}$$

The constraints in Eq. (15) are chosen such that the resulting δt for eight pairs of consistency errors are roughly $\{0.5, 1.5, 3, 5, 7, 10, 12, 14\}$ days, covering a maximum range of past two weeks.

During the modifying process, as illustrated in Fig. 5 in the darker gray frame, all learning variables $\Lambda(t_i)$ are available as inputs at the current epoch, but the target variables are unknown at the future epoch t_j . The trained SVM models generate an approximation $\hat{\mathbf{e}}_{\text{ML}}(t_j; t_i)$ of the true error $\mathbf{e}_T(t_j; t_i)$. Then, the ML-modified prediction $\hat{\mathbf{X}}_{\text{ML}}(t_j; t_i)$ is obtained by eliminating the ML-predicted true error $\hat{\mathbf{e}}_{\text{ML}}(t_j; t_i)$ from the

TLE-based prediction $\hat{\mathbf{X}}_{\text{TLE}}(t_j; t_i)$, as introduced in Eq. (2). The residual orbit prediction error $e_{\text{res}}(t_j; t_i)$ after the ML-modification is

$$\begin{aligned} e_{\text{res}}(t_j; t_i) &= \mathbf{X}_{\text{T}}(t_j) - \hat{\mathbf{X}}_{\text{ML}}(t_j; t_i) \\ &= \mathbf{e}_{\text{T}}(t_j; t_i) - \hat{\mathbf{e}}_{\text{ML}}(t_j; t_i), \end{aligned} \quad (16)$$

where Eq. (2) has been used. The residual error $\hat{\mathbf{e}}_{\text{res}}(t_j; t_i)$ reaches zero if the SVM model completely captures the underlying errors. Equation (16) indicates that the performance of the ML approach can be directly evaluated using e_{res} (time parameters are omitted for simplicity in the following paper).

At last, we note that the design of dataset structure in this paper is demonstrated to be practically feasible with good performance, but this may not be the best design and certainly is not the unique solution.

3.4 Grid Search for SVM Parameters

For evaluation purpose, a performance metric P_{ML} is used to quantify the trained SVM model, which has been first defined in our previous publication [8]. The metric P_{ML} quantifies the ratio between the total absolute residual errors $|\hat{\mathbf{e}}_{\text{res}}|$ and the total absolute true error $|\mathbf{e}_{\text{T}}|$, mathematically represented as

$$P_{\text{ML}}(\xi) = \frac{\sum_n |\hat{e}_{\text{res},\xi}|}{\sum_n |e_{\text{T},\xi}|} \cdot 100\% = \frac{\sum_n |e_{\text{T},\xi} - \hat{e}_{\text{ML},\xi}|}{\sum_n |e_{\text{T},\xi}|} \cdot 100\%, \quad (17)$$

where n is the size of the testing data, and the subscript $\xi \in \{x, y, z, vx, vy, vz\}$ indicates different components of the error. For clarity, the parameter ξ will be omitted when there is no confusion. The metric reaches the lower boundary zero when the ML-predicted error $\hat{\mathbf{e}}_{\text{ML}}$ is identical to \mathbf{e}_{T} , but has no upper boundary. The smaller P_{ML} is, the better performance the trained SVM model has, because in these cases more errors can be corrected by the trained SVM model. If the training date is used to calculate the metric P_{ML} , the result actually quantifies the learning capability of the SVM model. In this case, the remaining errors represent the information that cannot be modeled with the available learning variables.

Although P_{ML} is shown to be a good average performance metric in previous studies [8–10], there are some limitations in its use, including 1) when some true errors e_{T} are close to zero, P_{ML} can be large and unreliable due to the small denominators; 2) both the reductions of means and standard deviations can lead to small P_{ML} , but cannot be distinguished by P_{ML} ; and 3) P_{ML} does not characterize the variations of performance with respect to Δt .

Using performance metric P_{ML} , parameters of the SVM models are designed through a grid search process, as first implemented in our previous study [10]. Here, the searching results for the TLE catalog using the RSO STELLA are demonstrated. Before training, the learning variables are standardized; the kernel scale parameter is determined automatically by MATLAB; the box constraint is determined by default as $\text{iqr}(e_{\xi})/1.349$, where $\text{iqr}(\cdot)$ is the interquartile range of $\xi \in \{x, y, z, vx, vy, vz\}$; the soft margin ε of the SVM and the gap tolerance between the primary and the dual problem during the training are determined through a grid search process. As will be described later in Sec. 4.1, the data in days 1–300 (starting from the first day of the chosen year) is used as the training data, and the new data in days 301–314 is used as the testing data. At each grid search, the orbit prediction data at the following 14 days (2 weeks) of the training data are used as validation data, and the rest of the training data are used to train the SVM model. In other words, the training data is fixed in days 1–286 and the validation data in days 287–300. Then the metric P_{ML} on the validation data is used to quantify this grid. After a satisfying set of SVM parameters is chosen, all the data in days 0–300 will be used to conduct the actual training of the desired SVM models, in order to make the best of available data.

We remark that this setup of validation data is critical because in the practice the testing data in the future are not available. Therefore, the testing data should not be used to tune SVM parameters even though they are available in the study. Using the testing data to tune will lead to a best-fitting results rather than a reliable evaluation of the performance of the SVM models. Thus, this choice of validation data ensures that: 1) the validation data is in the “future” with respect to the training data; 2) the testing data is not used during the training.

The grid search results of P_{ML} on the testing data are summarized in Tables 2 and 3 for position and velocity components. In Table 2, P_{ML} reaches its largest when ε is 1 km; P_{ML} tends to decrease when ε is reduced; but P_{ML} may slightly increase when ε is further decreased. One reason could be that the random noise in the data cannot be effectively accommodated by a narrow margin. Therefore, the resulting SVM models may overfit the training data and lead to bad generalization capability. In Table 3, similar phenomena are observed.

Usually, smaller ε and gap tolerance will require more computations to train the SVM model, but they are not guaranteed to lead to a better performance. Considering the trade-off between the performance and the computational efficiency, we use the combinations with asterisks in the tables in the following study to train the SVM models. The above chosen ε has a physical interpretation. For example, an $\varepsilon(e_y)$ of 0.1 km indicates that the expected residual orbit prediction error after ML-modification is no larger than 0.1 km.

Table 2 Grid search results of performance on the position components with respect to ε -margin and gap tolerance on the validation data (days 287–300). The combinations with asterisks are chosen.

$P_{\text{ML}} [\%]$	Gap tolerance	Soft margin ε for position			
		0.001 km	0.01 km	0.1 km	1 km
$P_{\text{ML}}(e_x)$	10^{-4}	26.29	26.65	26.93	42.97
	10^{-3}	26.52	26.49*	26.67	42.97
	10^{-2}	26.55	26.48	26.68	42.97
	10^{-1}	26.59	26.51	26.69	42.97
$P_{\text{ML}}(e_y)$	10^{-4}	30.97	31.66	31.26	30.64
	10^{-3}	30.12	29.98	29.80*	30.03
	10^{-2}	30.03	29.97	29.81	29.81
	10^{-1}	30.03	29.99	29.80	29.79
$P_{\text{ML}}(e_z)$	10^{-4}	17.77	17.92	16.87	38.22
	10^{-3}	18.08	18.13	16.75*	38.22
	10^{-2}	18.16	17.99	16.69	38.22
	10^{-1}	18.18	18.01	16.69	38.22

Table 3 Grid search results of performance on the velocity components with respect to ε -margin and gap tolerance on the validation data (days 287–300). The combinations with asterisks are chosen.

$P_{\text{ML}} [\%]$	Gap tolerance	Soft margin ε for velocity			
		0.0001 m/s	0.001 m/s	0.01 m/s	0.1 m/s
$P_{\text{ML}}(e_{vx})$	10^{-4}	14.31	14.48	14.60	14.44
	10^{-3}	14.02	14.16*	14.16	14.36
	10^{-2}	14.20	14.14	14.14	14.35
	10^{-1}	14.18	14.17	14.12	14.35
$P_{\text{ML}}(e_{vy})$	10^{-4}	27.26	27.84	26.59	26.66
	10^{-3}	26.78	26.95	26.94	26.55*
	10^{-2}	26.89	26.86	26.82	26.49
	10^{-1}	26.86	26.85	26.75	26.49
$P_{\text{ML}}(e_{vz})$	10^{-4}	55.67	54.88	50.26	52.81
	10^{-3}	47.58	45.88*	46.63	47.77
	10^{-2}	46.26	46.20	46.04	47.74
	10^{-1}	46.28	46.28	46.09	47.72

We note that an optimal design for the ML approach with SVM is related to practical constraints and requirements, and is not the goal of this paper. In the following paper, it will be demonstrated that satisfying performance can be achieved with this specific design of the dataset structure.

4 Performance and Discussions

In this section, the performance of the ML approach with SVM models on all the chosen RSOs are presented. In the first part, STELLA is taken as an instance to illustrate the performance of the proposed ML approach in details and also the effect of the training or testing data size. Then the results of the ML approach on RSOs in the SSO, LEO, and MEO regions are summarized.

4.1 Performance on STELLA

Since the type II generalization capability to future epochs (see Fig. 4) is investigated, the data in days 1–300 (starting from the first day of the year) is used as the training data, and then the new data in days 301–314 is used as the testing data.

Figure 6 shows the states of STELLA in the ECI (J2000) frame in 2008, where the crosses represent states converted from TLE set and the dots represent states extracted from ILRS data. It is clear that these two catalogs provide close but different states of STELLA. The CPF data is treated as the true state for the ML approach. Each TLE state is propagated to the following TLE states within 14 days, i.e., with prediction duration $\Delta t \leq 14$ days. Then, all the data are collected to build the dataset for the ML approach. Details of these have been described in Sec. 3.

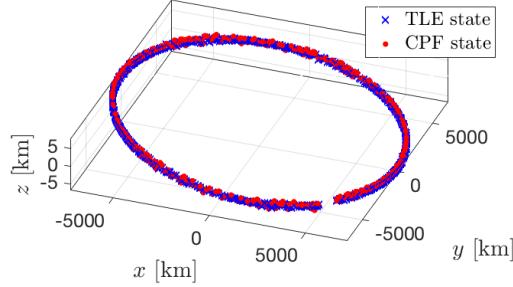


Fig. 6 TLE (blue crosses) and ILRS (red dots) positions of STELLA in the year 2008.

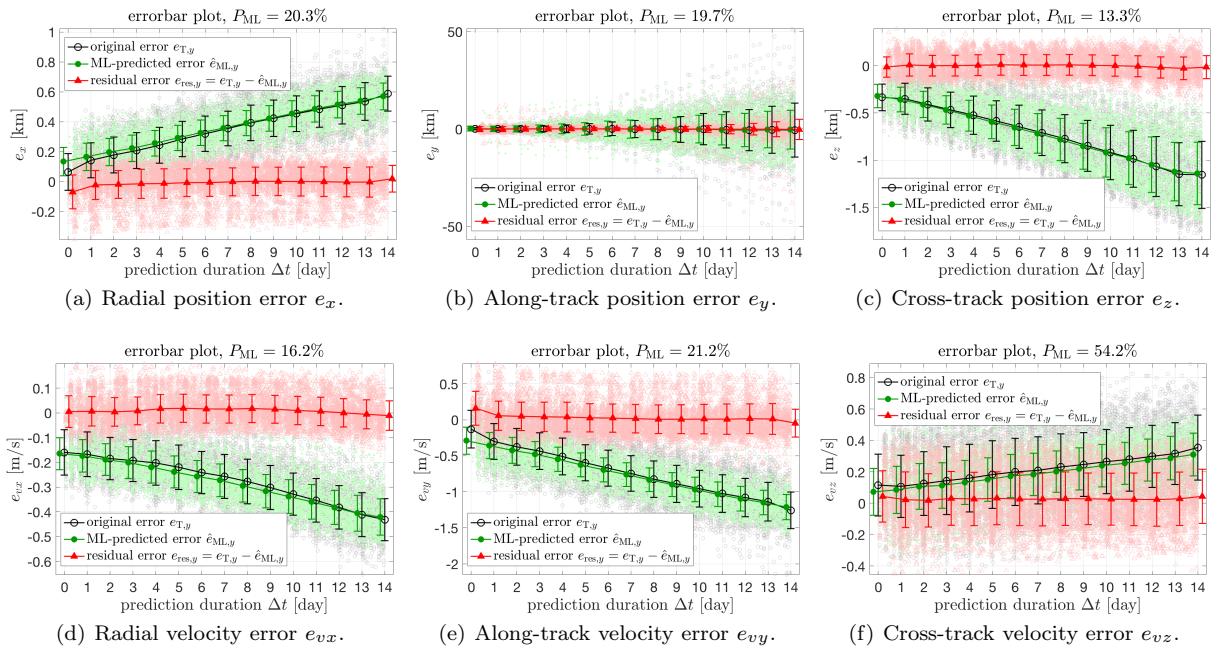


Fig. 7 Performance of the six SVM models on the training data (days 1–300).

Figure 7 demonstrates the performance of the trained SVM model on the training data for all the six components, which reflects the learning capability of the SVM method. In each panel, the horizontal axis represents the prediction duration Δt . The vertical axis shows magnitude of three types of errors, including the true error $e_{T,y}$, the ML-predicted error $\hat{e}_{ML,y}$, and the residual error $e_{res,y} = e_{T,y} - \hat{e}_{ML,y}$. The data points are clustered into 15 groups by Δt for evaluation purpose. The mean value of each group is represented by the center marker, and the standard deviation of the corresponding clustered group is represented by the length from the middle to the top (or bottom) of the bar. For clarity, the three curves are slightly displaced along the horizontal axis to avoid overlapping. The underlying scattering plots with corresponding lighter colors show the raw data of the three kinds of errors. For both the errorbar and the scatter plots, the circles stand

for $e_{T,y}$, the dots stand for $\hat{e}_{ML,y}$, and the triangles stand for $e_{res,y}$. Figures in the remaining part of the paper have used the same annotations if not otherwise described.

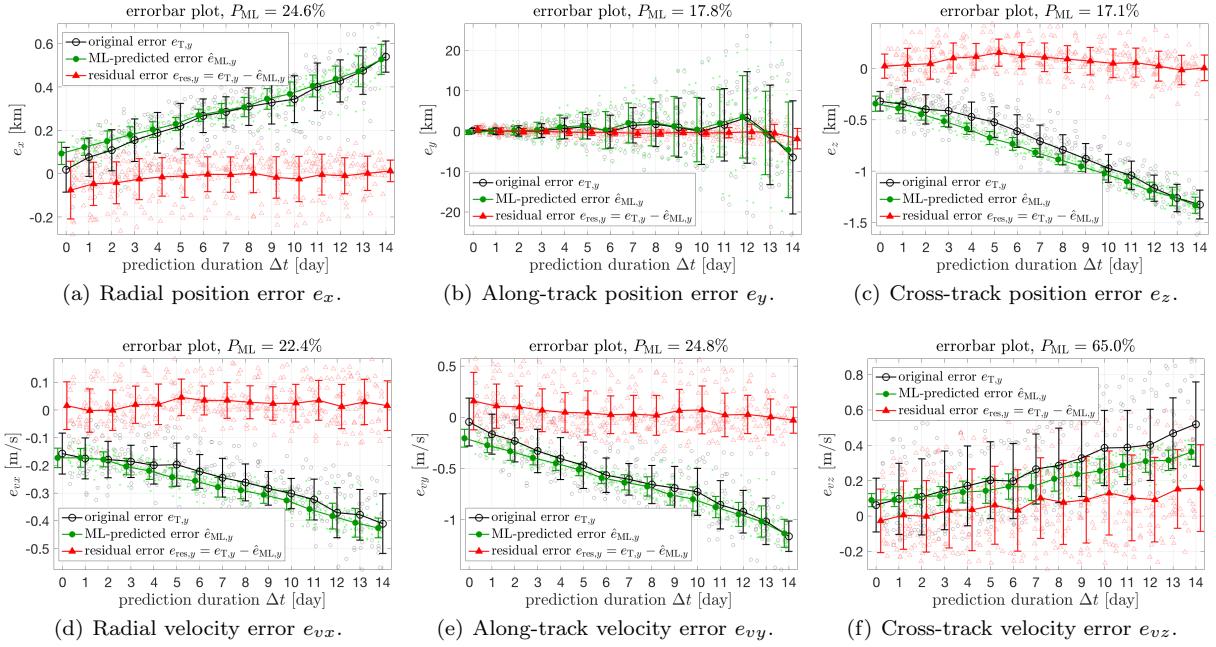


Fig. 8 Performance of the six SVM models on the testing data (days 301–314).

In Fig. 8, the performance of all the six trained SVM models on the testing data is demonstrated. The trained SVM models are effective to improve the orbit prediction accuracy on all the six components. In Figs. 8(a) to 8(f), the means have been reduced to almost zero, while the mean of e_y is originally close to zero and is reserved to be zero after the ML-modification. In Figs. 8(a) to 8(d), the standard deviations have been reduced, while standard deviations for other components have not been increased. The performance metric P_{ML} except that of e_{vz} are all less than 25%, and $P_{ML}(e_{vz})$ is 65.0%. These results reveal that the ML approach with the SVM model can improve the orbit prediction accuracy for STELLA.

Comparing Figs. 7 and 8 shows that the performance is slightly worse on the testing data (except e_y), since the metric P_{ML} are slightly higher. This is expected because the testing data is unknown to the trained SVM models. As shown in Fig. 4, the testing data and the training data of Type II are not overlapped in the time. As a consequence, the testing data can contain information that have not been learned by the SVM models. The minor decrease of $P_{ML}(e_y)$ can be due to less samples in the testing data. However, as shown by the testing performance in Fig. 8, the SVM models have all been successfully generalized to unknown future data. We remark that these results on the TLE catalog agree with our previous research based on a simulated RSO catalog [8].

The results are impressive as all biases have been removed and all standard deviations of position errors have been reduced, which will be helpful for operations such as satellite conjunction analysis. The proposed ML approach to improve orbit prediction of STELLA in future epochs based on its historical data has been demonstrated to be feasible.

4.2 Effect of Training Data Size on Performance

In machine learning studies, using more training data will usually enhance the performance of trained ML models. However, in reality, additional data can also undermine the performance as it may introduce random noise.

To study the effect of the training data size, a series of SVM models are trained with different size of training data, and then they are evaluated on the same testing data. Some example results of e_y are demonstrated in

Fig. 9, using training data size of 100, 150, 200, and 250 days before the testing data fixed in days 300–314. The metric P_{ML} is 28.0% when using only 100 days, decreases to 25.3% when using 150 days, and finally reduces to 17.8% when using 250 days of training data.

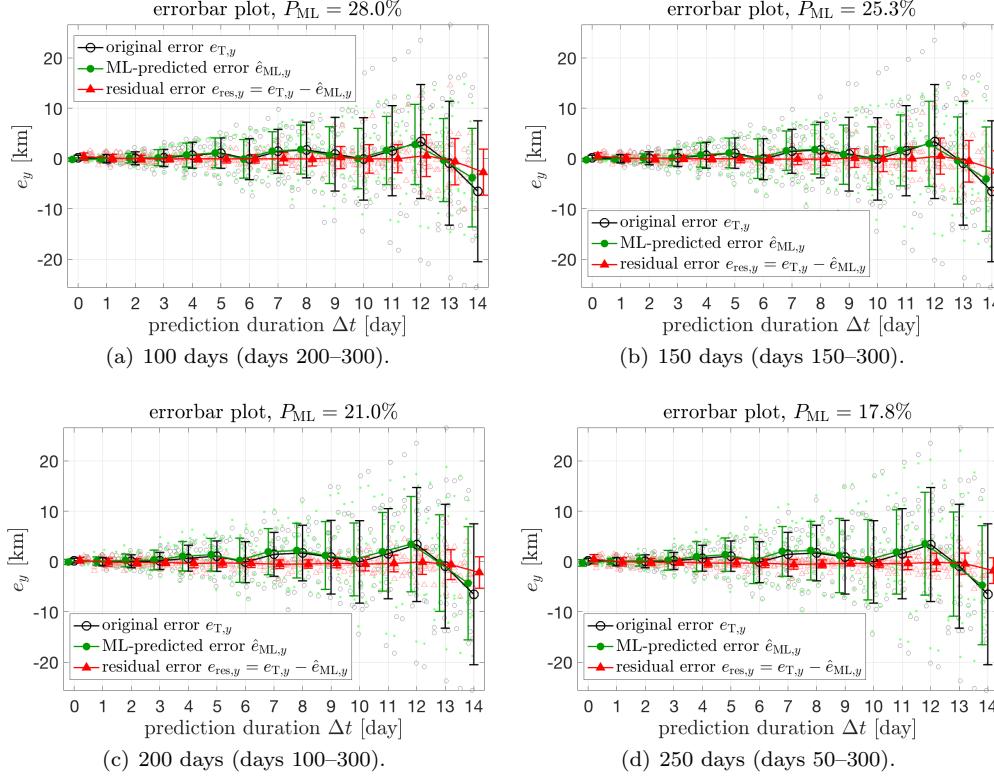


Fig. 9 Result of the SVM models on the same testing data (day 300–314), but trained with different size of data.

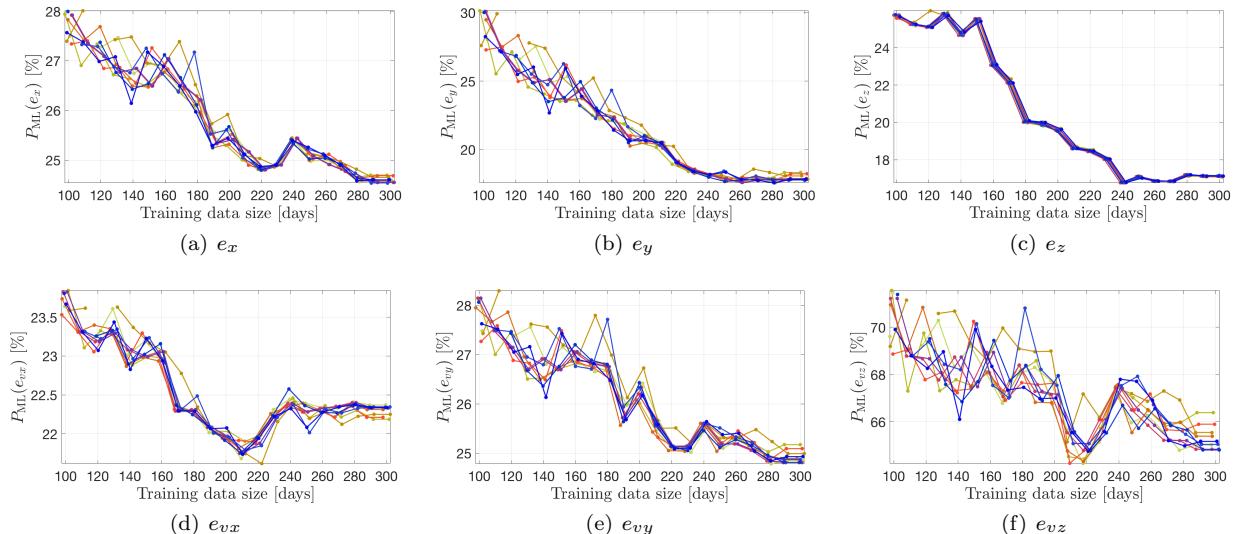


Fig. 10 Variations of P_{ML} with respect to the training data size, illustrated with 10 random seeds.

To observe in more details, the size of the training data is varied from 100 to 300 days with a step size of 10 days, while the testing data is still fixed in days 300–314. In addition to the smaller step size, Monte-Carlo simulations with 10 different random seeds are carried out for each training data, because the scale parameter of the SVM kernel function is heuristically determined by MATLAB through a random sampling strategy at the initialization. All the results are summarized in Fig. 9. In each panel, the horizontal axis shows the training data size of 100 to 300 days, and the curves represent the variation of the performance metric P_{ML} with different random seeds. The random seed for each curve has been kept the same. For all the components, a general trend is that P_{ML} decreases as more data are used for training at the beginning and becomes stable after about training data of 280 days is used. The random seed has some effect on the performance, but the effect diminishes when adequate data is included. Therefore, the performance of trained SVM models is robust with respect to the randomness introduced during the training process. It is also interesting to notice that there seems to be a local optimal solution in Figs. 10(a) and 10(d), but the curve tends to be stable only after more data has been included. These result are expected because more training data usually carries more information about the underlying pattern between the learning and target variables.

Therefore, it can be concluded that the performance of the trained SVM model increases as the size of the training data increases until adequate data has been used. In the following study, the training data size will be fixed to 300 days.

4.3 Performance on Different Testing Data

In previous discussions, the testing data is fixed to be 14 days right after the end of the testing data. For real operations, as shown in Fig. 11, there are at least two situations. The first one is that the testing data has different lengths in the future. One possible application is that the current TLE set is propagated to a certain future time interval for observation scheduling or conjunction analysis, where the length of the time interval can be different due to the requirement. The second one is that the gap between the training and testing data is different. One practical example is that the trained SVM models cannot be updated rightly after the historical data is available, but is delayed with a certain duration, which leads to a gap between the data. It is necessary for the ML approach to be feasible and effective for both situations.

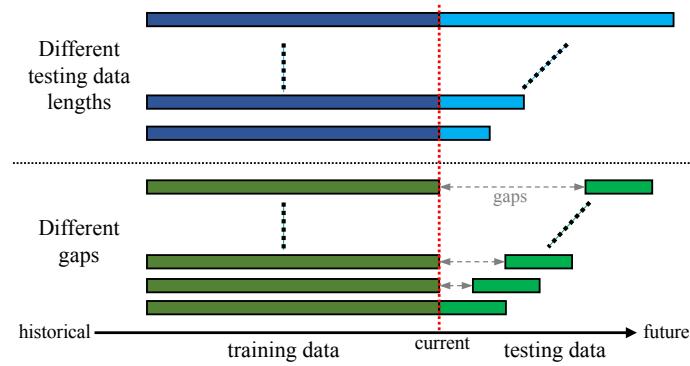


Fig. 11 Two different situations of the testing data in future epochs.

Different Testing Data Length. This situation is investigated by fixing the training data in days 0–300, but varying the testing data from 7 to 40 days (all starting from day 301). For each cases, 10 different random seeds are used to trained SVM models, and then they are applied to the same testing data with the specified length. The results are summarized in Fig. 12. The horizontal axis is the testing data size; the circles show the performance of different SVM models trained with the 10 random seeds. For all the cases, the metrics P_{ML} are much less than 100%, which means the orbit prediction error has been reduced. The metrics P_{ML} of e_x , e_y , and e_{vy} tend to increase as the data size increases; but that of the other three components will increase at first and then start to decrease. However, the total fluctuations for each component are small, which are about 12%, 6%, 4%, 5%, 14%, and 15% respectively.

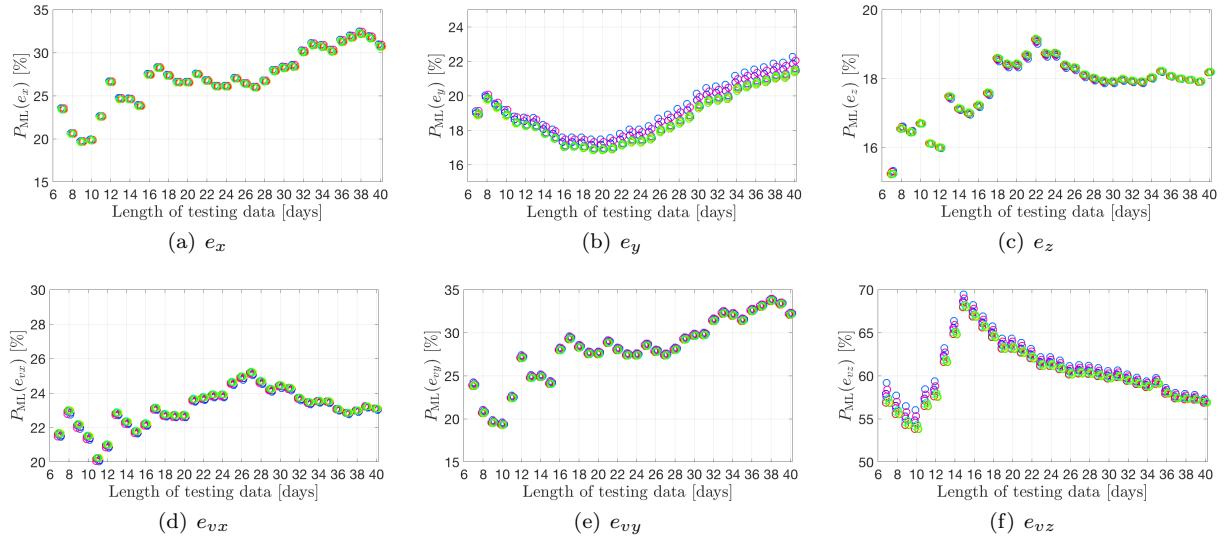


Fig. 12 Variations of P_{ML} with respect to the testing data size, illustrated with 10 random seeds.

Another interesting phenomenon is that, the circles in Fig. 12 corresponding to different seeds show small deviations for all the cases. This reveals that although the random seed has a complicated effect on the results of testing data, as shown in Fig. 10, its effect on the testing data is not significant.

Different Gaps Between Training and Testing Data. The second situation in Fig. 11 is studied by fixing the training data size to be 14 days, and the gap is increased from 1 to 28 days. The results are summarized in Fig. 13. Similarly, the circles correspond to 10 random seeds used for the training and they show uniform effects on the performance. Although there are no clear patterns except for e_y and e_{vz} , the fluctuation of the performance is again small for all the components.

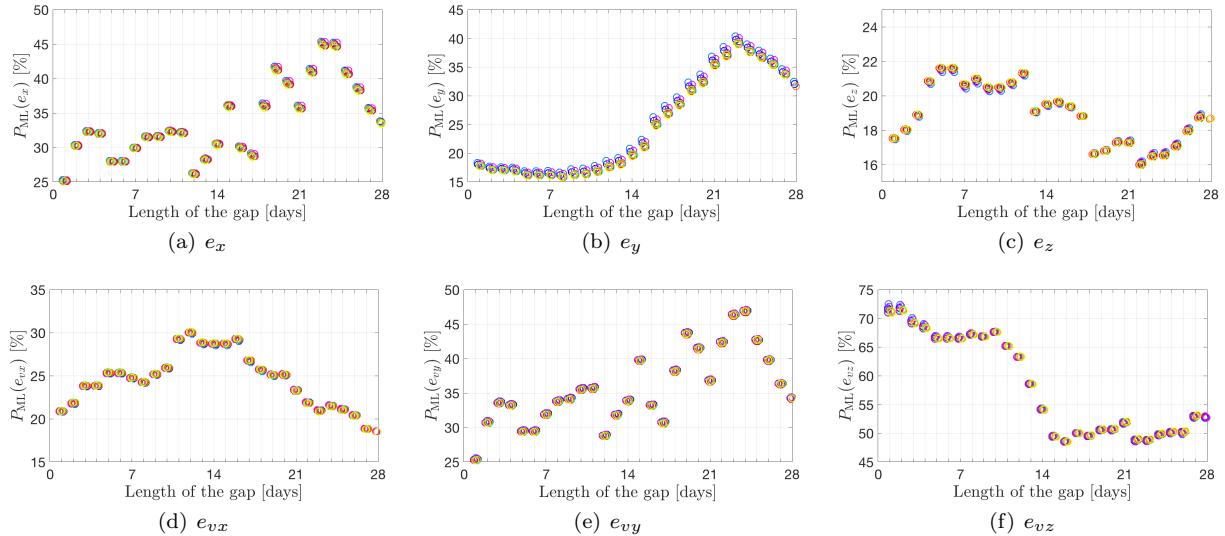


Fig. 13 Variations of P_{ML} with respect to the gap between training and testing data, illustrated with 10 random seeds.

As a conclusion, the trained SVM models have good generalization capabilities to various lengths of future testing data as long as 40 days, and are applicable to testing data with a gap as large as 28 days. In the practice, this means that the trained SVM models can be applied to various lengths of testing data with a satisfying

performance and can be updated with a delay as long as four weeks. This makes the proposed ML approach very flexible.

4.4 Performance on Sun-Synchronous Orbit Objects

In the above discussions, the first RSO (case 1) in Table 1 has been analyzed, which belongs to the SSO orbit type. In Table 4, the performance on all the three SSO RSOs is summarized, with the training data size of 300, 200, and 100 days included. We note that although there are many RSOs belonging to SSO orbit type, only a small number of them are passive objects without maneuvering capability and tracked by both the TLE and ILRS catalogs. In the table, P_{ML} larger than 90% is annotated with a cross mark, which means the performance is bad and not acceptable (similar for the following tables). The results show that the trained SVM models can successfully reduce almost all the residual errors except e_{vz} of LARETS. Moreover, most P_{ML} increases when the training data size decreases from 300 days (0–300) to 100 days (200–300). For example, the three $P_{ML}(e_y)$ of LARETS are 38.5% → 42.8% → 85.5%, which means the performance is becoming worse as smaller training data size is used. However, there are a few exceptions, such as $P_{ML}(e_z)$ of BLITS, which are 63.4% → 62.9% → 49.7%.

Table 4 Performance metrics on SSO RSOs with different training data size. Cross markers indicate metrics greater than 100%.

Case #	NORAD ID	Satellite Name	Orbit Type	Training Data		Testing Data		Performance metric $P_{ML} [\%]$ of					
				Interval	Points	Interval	Points	e_x	e_y	e_z	e_{vx}	e_{vy}	e_{vz}
1	22824	STELLA	SSO	0–300	9268	301–314	361	24.6	17.8	17.1	22.4	24.8	65.0
				100–300	6679	301–314	361	25.6	21.0	19.6	22.0	26.4	67.7
				200–300	3263	301–314	361	27.5	28.0	25.6	23.6	27.5	69.6
2	27944	LARETS	SSO	0–300	6281	301–314	351	36.9	38.5	15.9	14.7	37.0	108.9 \times
				100–300	4581	301–314	351	37.6	42.8	16.8	15.3	36.9	112.0 \times
				200–300	2083	301–314	351	47.3	85.8	27.5	25.3	45.9	130.6 \times
3	35871	BLITS	SSO	0–300	16878	301–314	1395	13.9	20.3	63.4	23.8	15.1	62.9
				100–300	14967	301–314	1395	14.2	21.8	62.9	25.6	15.1	65.1
				200–300	7191	301–314	1395	13.7	39.9	49.7	20.3	14.6	71.7

Figure 14 shows the three bad cases with performance metrics $P_{ML}(e_{vz})$ of LARETS. In the left figures, the trained SVM models have been applied back to the training data, but the metrics are still large. A closer observation on the original and residual errorbars shows that 1) the original errors are small and close to zero, which could invalidate P_{ML} as mentioned in Sec. 3.4; 2) the biases have been learned and compensated; 3) the standard deviations have not been reduced. Possibly these errors are due to randomness and cannot be eliminated. It is also possible that the designed learning variables do not have enough information about e_{vz} for LARETS. As a consequence, the ML-predicted errorbars are not as wide as that of the original errorbars. In the right figures of Figs. 14(a) and 14(b), although P_{ML} is greater than 100%, the ML-predicted biases are close to zero and the residual errorbars (green) are not severely deviated from the original errorbar (black). Differently, in the right figure of Fig. 14(c), the ML-predicted errorbar is very wrong, which indicates that the generalization capability has been almost all lost when only 100 days are used to train the SVM model for LARETS.

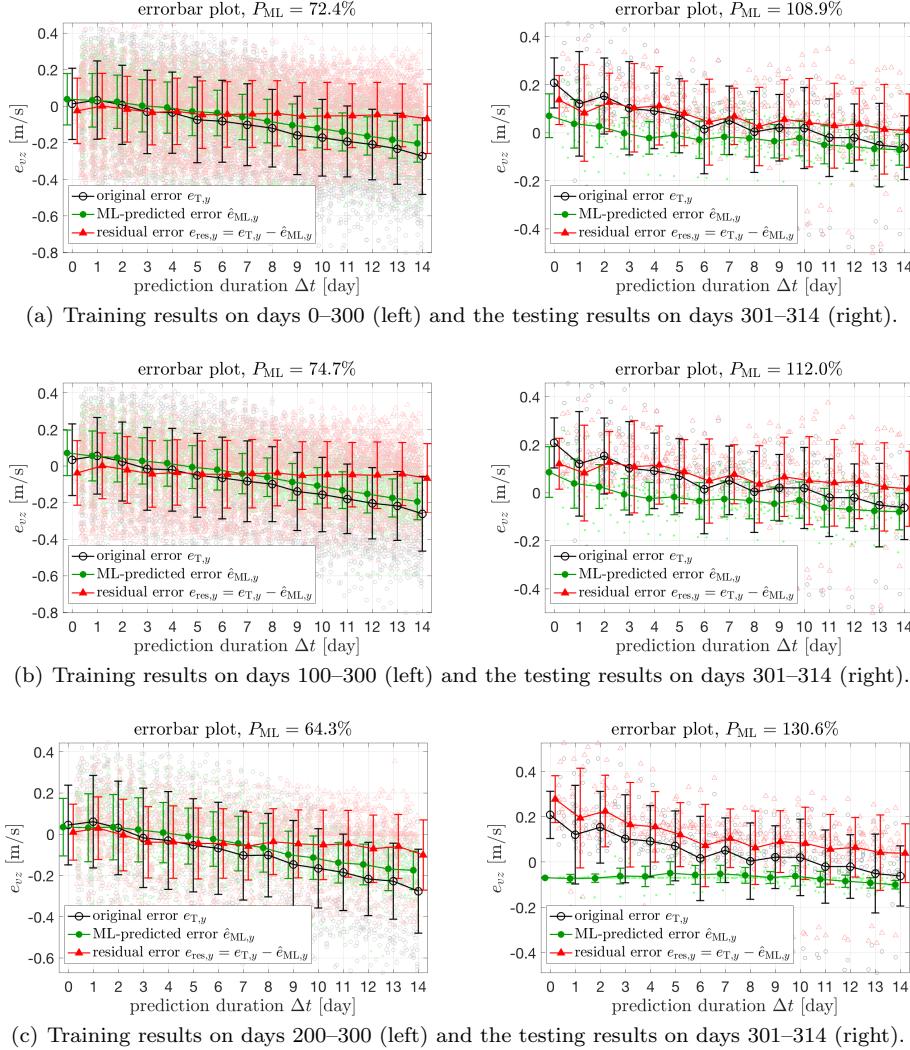


Fig. 14 Training and testing results of the bad performance metrics $P_{ML}(e_{vz})$ of LARETS in Table 4.

4.5 Performance on Low Earth Orbit Objects

RSOs in the LEO region could experience very different space environments. For example, the illumination conditions for every revolution of an SSO RSO are almost the same, but can vary greatly for LEO objects, as can be inferred from Fig. 3. Experiment results are summarized in Table 5. It is interesting to observe that the trained SVM models are still effective on almost all cases. We remark that the ML approach with SVM models is designed only once as described in Sec. 3, and the only difference to study the LEO objects here is the training data.

Some examples with unsatisfying results are demonstrated in Fig. 15. The training performance metrics are good and both biases and standard deviations can be reduced significantly. However, the testing data shows very different patterns compared with the training data, which reveals the ML approach's limitation for those cases. Meanwhile, an interesting phenomenon is that in the three right figures in Fig. 15, all the ML-predicted error bars tend to be closer to the original error when Δt increases. This could mean that the ML approach will have a better performance on e_{vx} and e_{vz} for long predictions of LEO RSOs. There are many possible causes for the failure to generalize for the future epochs, including: 1) the errors of these components are already very small; 2) the input learning variables do not have enough information of the target variables; and 3) the training data contains inconsistent information due to space environment changes. These hypotheses require further studies to confirm.

Table 5 Performance metrics on LEO RSOs with different training data sizes. Cross markers indicate metrics greater than 100%.

Case #	NORAD ID	Satellite Name	Orbit Type	Training Data		Testing Data		Performance metric P_{ML} [%]					
				Interval	Points	Interval	Points	e_x	e_y	e_z	e_{vx}	e_{vy}	e_{vz}
4	16908	Ajisai	LEO	0–300	10262	301–314	489	57.2	47.2	20.0	60.6	53.1	95.3 \times
				100–300	7660	301–314	489	56.5	48.0	20.4	62.9	52.5	96.6 \times
				200–300	3882	301–314	489	49.2	83.7	28.9	67.5	43.7	82.4
5	38077	LARES	LEO	0–300	24544	301–314	419	21.6	43.9	34.0	95.3 \times	24.3	100.0 \times
				100–300	15876	301–314	419	18.6	39.9	32.5	95.4 \times	21.8	100.9 \times
				200–300	8628	301–314	419	21.1	44.4	40.3	85.4	23.5	99.7 \times
6	1328	Beacon-C	LEO	0–300	8306	301–314	295	19.0	33.8	32.6	68.3	18.5	20.6
				100–300	5890	301–314	295	19.0	38.1	34.1	76.5	18.9	22.0
				200–300	2755	301–314	295	29.3	61.3	50.9	83.9	31.6	24.2
7	7646	Starlette	LEO	0–300	10751	301–314	553	86.3	35.7	12.3	74.8	76.9	45.5
				100–300	8013	301–314	553	91.2 \times	37.9	12.5	80.5	80.5	46.2
				200–300	3983	301–314	553	87.5	48.0	10.3	89.4	77.6	41.0

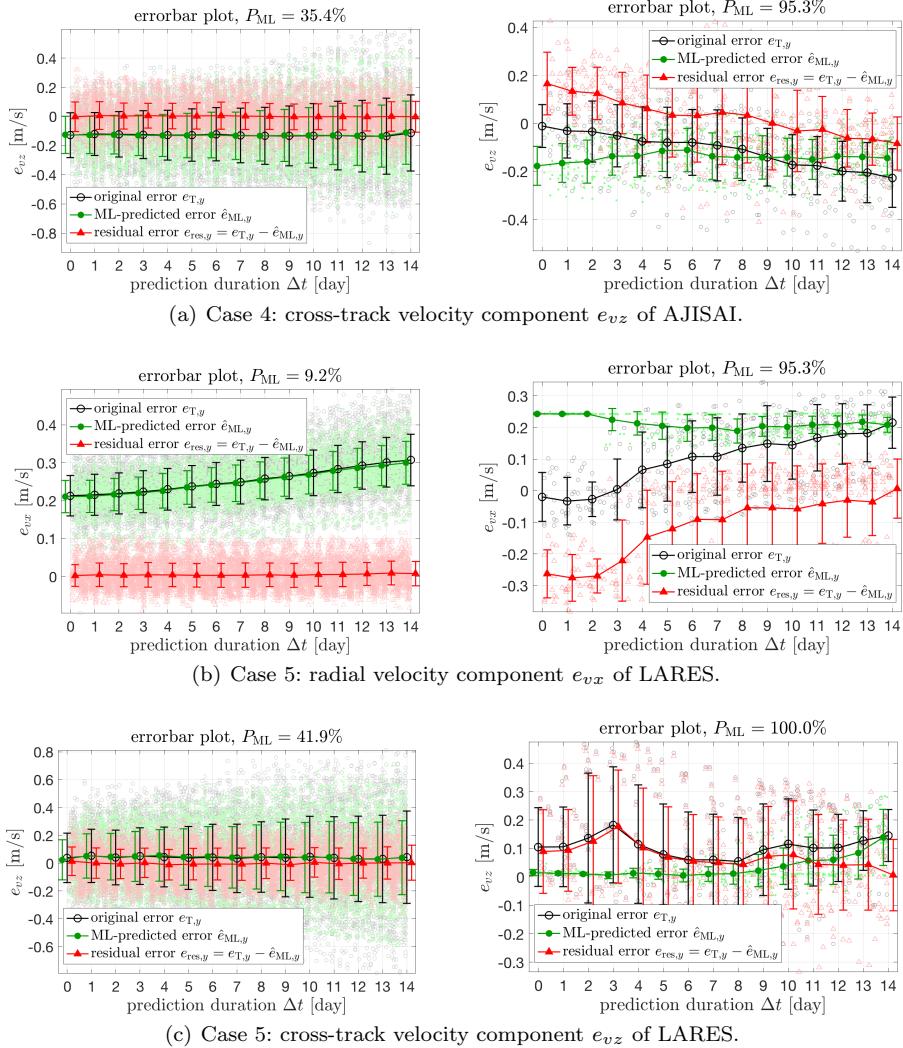


Fig. 15 Example results of the bad performance metrics in Table 5, with training results on days 0–300 (left) and testing results on days 301–314 (right).

4.6 Performance on Medium Earth Orbit Objects

The RSOs in MEO region have much higher altitude, as shown in Fig. 3. Their B^* in TLE sets are constant, as summarized in Table 1. Notice the atmosphere drag force becomes very weak for MEO RSOs, but on the other hand, the effect of solar radiation pressure will become stronger as the RSO moves higher. Unfortunately there is no available information about this effect from TLE catalog. So it could be expected that the ML approach with SVM may not be as effective as on SSO or LEO.

The results are summarized in Table 6. It is nice to see that the trained SVM models have good performance on all the components of case 9 and all components except e_{vz} of case 8, using the maximum training data size of 300 days. For cases 10 and 11, the SVM models are only effective for e_y have satisfying performance. This is possible considering the fact that the dataset structure is designed based on STELLA but ETALON-1 and 2 are much higher.

Table 6 Performance metrics on MEO RSOs with different training data size. Cross markers indicate metrics greater than 100%.

Case #	NORAD ID	Satellite Name	Orbit Type	Training Data		Testing Data		Performance metric $P_{ML} [\%]$					
				Interval	Points	Interval	Points	e_x	e_y	e_z	e_{vx}	e_{vy}	e_{vz}
8	8820	Lageos-1	MEO	0–300	6496	301–314	233	36.2	66.0	77.6	28.2	49.9	98.9 \times
				100–300	4541	301–314	233	39.1	44.9	185.2 \times	32.7	60.0	141.0 \times
				200–300	2141	301–314	233	44.3	45.2	180.0 \times	43.8	58.1	115.8 \times
9	22195	Lageos-2	MEO	0–300	6661	301–314	340	12.7	23.7	31.3	81.6	15.3	38.6
				100–300	4764	301–314	340	14.3	23.1	13.6	75.8	16.5	37.2
				200–300	2329	301–314	340	14.1	40.5	14.3	48.6	21.0	22.9
10	19751	Etalon-1	MEO	0–300	7389	301–314	338	101.3 \times	24.0	108.6 \times	76.9	97.6 \times	90.9 \times
				100–300	5360	301–314	338	102.8 \times	27.0	109.3 \times	78.4	98.2 \times	89.6
				200–300	2646	301–314	338	101.7 \times	47.5	102.1 \times	80.6	99.3 \times	99.0 \times
11	20026	Etalon-2	MEO	0–300	7615	301–314	391	84.1	54.8	113.4 \times	109.6 \times	81.9	146.7 \times
				100–300	5404	301–314	391	83.9	55.1	107.7 \times	108.0 \times	80.7	153.1 \times
				200–300	2554	301–314	391	89.2	80.7	107.4 \times	112.5 \times	80.1	166.9 \times

Three examples with bad performance are demonstrated in Fig. 16. In Fig. 16(a), the training result of e_{vz} of LAGEOS-1 is good and the standard deviation of testing data has been reduced. However, since the original errors are very close to zero, $P_{ML}(e_{vz})$ is large and does not reflect the reduction on the standard deviations. In Figs. 16(b) and 16(c), phenomenons similar to that in Fig. 15 have been observed: the standard deviations in the training data cannot be reduced; the original errors in the testing data are close to zero; and the trained SVM models have bad generalization capabilities.

5 Conclusions

In this paper, a Machine Learning (ML) approach to improve the orbit prediction accuracy is explored using the publicly available Two-Lime Element (TLE) catalog and the International Laser Ranging Service (ILRS) catalog. The dataset structure for the ML approach is designed to use the low-accuracy TLE data as orbit predictions and the high-accuracy ILRS data as the true orbit. The Support Vector Machine (SVM) method is used as the specific ML algorithm to capture the underlying relationship between the learning variables (including pairs of consistency errors of TLE sets) and the target variables (true orbit prediction errors). Totally 11 RSOs having both TLE and ILRS data are used to evaluate the generalization capability of the trained SVM models to future epochs. Using STELLA as an example, the results show that: 1) the ML approach can significantly improve the orbit prediction accuracy for all the position and velocity components; 2) increasing training data size can lead to better performance until adequate data has been used; and 3) the performance metrics are stable on testing data with a wide range of time intervals. Furthermore, the results on all the 11 RSOs reveal that the ML approach is effective for RSOs on SSO and LEO for most position and velocity components.

The ML approach validated on the TLE catalog in this paper is capable of learning information directly from historical data. This data-driven method can be a significant enhancement to the current physical-based prediction frameworks. Future studies include extending the ML approach to a larger range of RSOs in the TLE catalog; exploring limits of the ML approach on the TLE catalog with optimal dataset design; and incorporating more information sources.

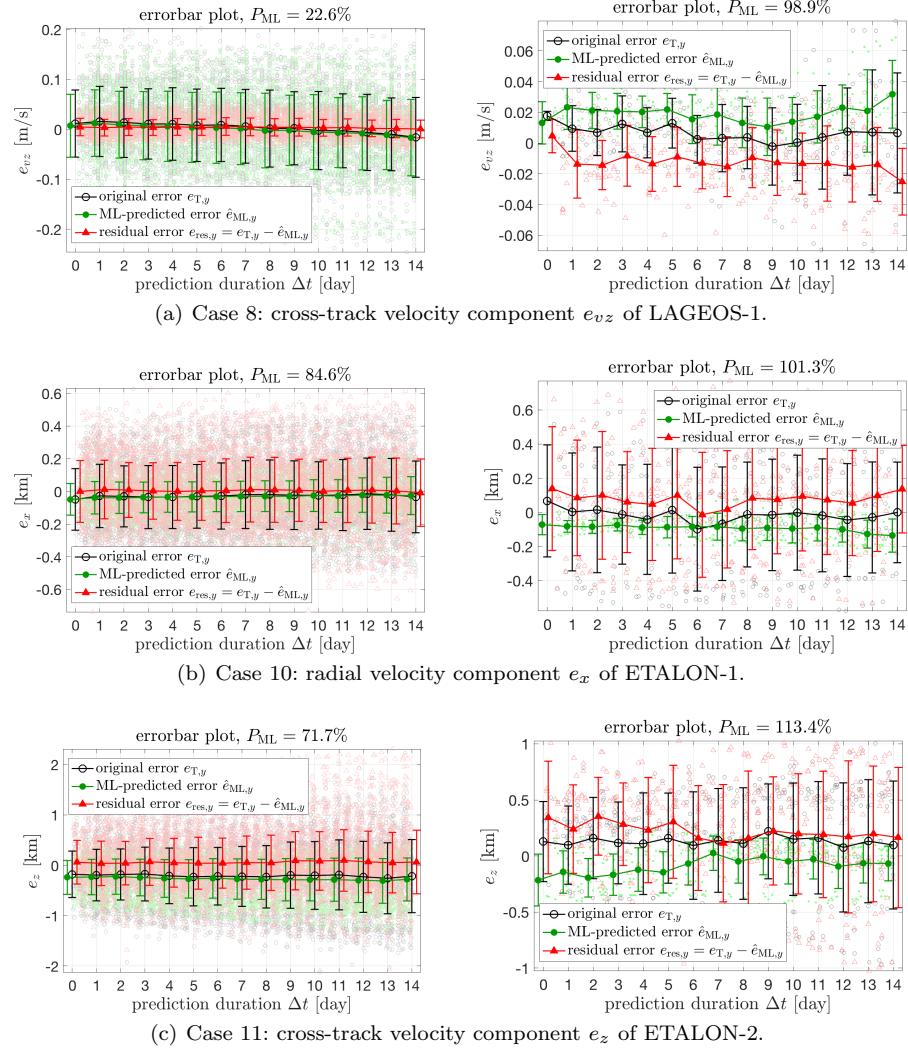


Fig. 16 Example results of the bad performance metrics in Table 6, with training results on days 0–300 (left) and testing results on days 301–314 (right).

Acknowledgements The authors would acknowledge the research support from the Air Force Office of Scientific Research (AFOSR) FA9550-16-1-0184 and the Office of Naval Research (ONR) N00014-16-1-2729.

Conflict of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Phillip Anz-Meador and Debi Shoots. Orbital Debris Quarterly News (2018 Feb). NASA Orbital Debris Program Office, 22(1):1–12, February 2018. URL <https://www orbitaldebris jsc nasa gov/quarterly-news/pdfs/odqnv22i1.pdf>.
- T.S. Kelso. Iridium 33/Cosmos 2251 Collision, July 2009. URL <http://celestak.com/events/collision.asp>.
- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. Learning from Data. AMLBook, 2012. ISBN 1-60049-006-9. URL <http://amlbook.com/>.

4. Tom M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill Education, New York, NY, 1 edition edition, 1997. ISBN 0-07-042807-7. URL <http://www.cs.cmu.edu/%0020tom/mlbook.html>.
5. Christos Ampatzis and Dario Izzo. Machine learning techniques for approximation of objective functions in trajectory optimisation. In *The IJCAI-09 Workshop on Artificial Intelligence in Space*, volume 2009, Pasadena, California, 2009. URL <https://www.esa.int/gsp/ACT/doc/AI/pub/ACT-RPR-AI-2009-09-aanapproximation.pdf>.
6. Jouni Hartikainen, Mari Seppänen, and Simo Särkkä. State-Space Inference for Non-Linear Latent Force Models with Application to Satellite Orbit Prediction. In *29th International Conference on Machine Learning (ICML-12)*, Edinburgh, Scotland, June 2012.
7. Srinagesh Sharma and James W. Cutler. Robust Orbit Determination and Classification: A Learning Theoretic Approach. IPN Progress Report 42-203, Jet Propulsion Laboratory, November 2015.
8. Hao Peng and Xiaoli Bai. Improving Orbit Prediction Accuracy through Supervised Machine Learning. *Advances in Space Research*, March 2018. ISSN 02731177. doi: 10.1016/j.asr.2018.03.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S0273117718302035>.
9. Hao Peng and Xiaoli Bai. Artificial Neural Network-Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *Journal of Spacecraft and Rockets*, pages 1–13, June 2018. ISSN 0022-4650, 1533-6794. doi: 10.2514/1.A34171. URL <https://arc.aiaa.org/doi/10.2514/1.A34171>.
10. Hao Peng and Xiaoli Bai. Exploring Capability of Support Vector Machine for Improving Satellite Orbit Prediction Accuracy. *Journal of Aerospace Information Systems*, pages 1–16, 2018. doi: 10.2514/1.I010616. URL <https://doi.org/10.2514/1.I010616>.
11. Hao Peng and Xiaoli Bai. Recovering Area-to-Mass Ratio of Resident Space Objects through Data Mining. *Acta Astronautica*, 142:75–86, January 2018. ISSN 0094-5765. doi: 10.1016/j.actaastro.2017.09.030. URL <http://www.sciencedirect.com/science/article/pii/S0094576517304058>.
12. David Vallado, Benjamin Virgili, and Tim Flohrer. Improved SSA Through Orbit Determination of Two-Line Element Sets. In *6th European Conference on Space Debris*, Darmstadt, Germany, April 2013. doi: 10.13140/2.1.4644.2241. URL <http://adsabs.harvard.edu/abs/2013ESASP.723E..48V>.
13. M. R. Pearlman, J. J. Degnan, and J. M. Bosworth. The International Laser Ranging Service. *Advances in Space Research*, 30(2):135–143, July 2002. ISSN 0273-1177. doi: 10.1016/S0273-1177(02)00277-6. URL <http://www.sciencedirect.com/science/article/pii/S0273117702002776>.
14. David Vallado, Paul Crawford, Ricahrd Hujasak, and T.S. Kelso. Revisiting Spacetrack Report #3. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 1–88. American Institute of Aeronautics and Astronautics, August 2006. ISBN 978-1-62410-048-2. doi: 10.2514/6.2006-6753. URL <http://arc.aiaa.org/doi/10.2514/6.2006-6753>.
15. David Finkleman. “TLE or not TLE?” That is the Question. In *AAS/AIAA Astrodynamics Specialist Conference 2007*, Mackinac Island, Michigan, August 2007.
16. Xian-Zong Bai, Lei Chen, and Guo-Jin Tang. Periodicity characterization of orbital prediction error and Poisson series fitting. *Advances in Space Research*, 50(5):560–575, September 2012. ISSN 02731177. doi: 10.1016/j.asr.2012.05.018. URL <http://linkinghub.elsevier.com/retrieve/pii/S0273117712003456>.
17. Paul Legendre, Béatrice Deguine, Romain Garmier, and Bruno Revelin. Two Line Element Accuracy Assessment Based On A Mixture of Gaussian Laws. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 1–13, Reston, Virginia, August 2006. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-048-2. doi: 10.2514/6.2006-6518. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2006-6518>.
18. R. Wang, J. Liu, and Q. M. Zhang. Propagation errors analysis of TLE data. *Advances in Space Research*, 43 (7):1065–1069, April 2009. ISSN 0273-1177. doi: 10.1016/j.asr.2008.11.017. URL <http://www.sciencedirect.com/science/article/pii/S0273117708006121>.
19. Jacco Geul, Erwin Mooij, and Ron Noomen. TLE uncertainty estimation using robust weighted differencing. *Advances in Space Research*, 2017. ISSN 0273-1177. doi: 10.1016/j.asr.2017.02.038. URL <http://www.sciencedirect.com/science/article/pii/S0273117717301515>.
20. Jizhang Sang, James C. Bennett, and Craig Smith. Experimental results of debris orbit predictions using sparse tracking data from Mt. Stromlo. *Acta Astronautica*, 102:258–268, September 2014. ISSN 00945765. doi: 10.1016/j.actaastro.2014.06.012. URL <http://linkinghub.elsevier.com/retrieve/pii/S0094576514002112>.
21. Lei Chen, Xian-Zong Bai, Yan-Gang Liang, and Ke-Bo Li. Orbital Error Analysis Based on Historical Data. In *Orbital Data Applications for Space Objects*, pages 77–105. Springer, Singapore, 2017. ISBN 978-981-10-2962-2 978-981-10-2963-9. doi: 10.1007/978-981-10-2963-9_3. URL <https://link.springer.com/>

- chapter/10.1007/978-981-10-2963-9_3.
- 22. Carolin Früh and Thomas Schildknecht. Accuracy of Two-Line-Element Data for Geostationary and High-Eccentricity Orbits. *Journal of Guidance, Control, and Dynamics*, 35(5):1483–1491, 2012. ISSN 0731-5090. doi: 10.2514/1.55843. URL <https://doi.org/10.2514/1.55843>.
 - 23. F. Kenneth Chan. *Spacecraft Collision Probability*. American Institute of Aeronautics and Astronautics, Inc., Washington, DC, March 2008. ISBN 978-1-884989-18-6. doi: 10.2514/4.989186. URL <http://arc.aiaa.org/doi/book/10.2514/4.989186>.
 - 24. Stijn Lemmens and Holger Krag. Two-Line-Elements-Based Maneuver Detection Methods for Satellites in Low Earth Orbit. *Journal of Guidance, Control, and Dynamics*, 37(3):860–868, 2014. ISSN 0731-5090. doi: 10.2514/1.61300. URL <http://dx.doi.org/10.2514/1.61300>.
 - 25. Alana R. Muldoon, Gabriel H. Elkaim, Ian F. Rickard, and Brian Weeden. Improved orbital debris trajectory estimation based on sequential TLE Processing. In *Paper IAC-09. A6. 2.9 Presented at the 60th International Astronautical Congress*. Daejeon, South Korea. Citeseer, 2009. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.294.5898&rep=rep1&type=pdf>.
 - 26. Creon Levit and William Marshall. Improved orbit predictions using two-line elements. *Advances in Space Research*, 47(7):1107–1115, 2011. ISSN 02731177. doi: 10.1016/j.asr.2010.10.017. URL <http://dx.doi.org/10.1016/j.asr.2010.10.017>.
 - 27. M. D. Hejduk, S. J. Casali, D. A. Cappellucci, N. L. Ericson, and D. E. Snow. A catalogue-wide implementation of general Perturbations orbit determination extrapolated from higher order orbital theory solutions. *Advances in the Astronautical Sciences*, 148:619–632, 2013. ISSN 00653438.
 - 28. Jizhang Sang and Bin Li. A Tle-Based Representation of Precise Orbit Prediction Results. In *The 6th International Conference on Astrodynamics Tools and Techniques*, 2016.
 - 29. Jullian Rivera and Xiaoli Bai. Improving the Orbit Propagation Accuracy of Two-Line-Element Satellite. In *67th International Astronautical Congress*, Guadalajara, Mexico, September 2016.
 - 30. Barbara Hammer and Kai Gersmann. A Note on the Universal Approximation Capability of Support Vector Machines. *Neural Processing Letters*, 17(1):43–53, February 2003. ISSN 1370-4621, 1573-773X. doi: 10.1023/A:1022936519097. URL <http://link.springer.com/article/10.1023/A:1022936519097>.
 - 31. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, New York, NY, 2000. ISBN 978-1-4419-3160-3 978-1-4757-3264-1. doi: 10.1007/978-1-4757-3264-1. URL <http://link.springer.com/10.1007/978-1-4757-3264-1>.
 - 32. Alexander J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004. ISSN 0960-3174. doi: 10.1023/B:STCO.0000035301.49549.88. URL <http://link.springer.com/10.1023/B:STCO.0000035301.49549.88>.
 - 33. Luc Maisonobe, Véronique Pommier, and Pascal Parraud. Orekit: An Open Source Library for Operational Flight Dynamics Applications. In *4th International Conference on Astrodynamics Tools and Techniques*, May 2010. URL https://www.researchgate.net/publication/310250345_OREKIT_AN_OPEN-SOURCE_LIBRARY_FOR_OPERATIONAL_FLIGHT_DYNAMICS_APPLICATIONS.
 - 34. Randall L. Ricklefs. Consolidated Laser Ranging Prediction Format v1.01. *ILRS Data Format and Procedures Working Group*, 2006. URL https://ilrs.cddis.eosdis.nasa.gov/docs/cpf_1.00.pdf.
 - 35. David A. Vallado. *Fundamentals of Astrodynamics and Applications*. The McGraw-Hill Companies, Inc., New York, NY, 1st edition, 1997. ISBN 978-0-07-066834-8. URL <http://www.springer.com/us/book/9780070668348>.