

Comparative Evaluation of Three Machine Learning Algorithms on Improving Orbit Prediction Accuracy

Hao Peng¹ and Xiaoli Bai¹ (✉)

Rutgers, The State University of New Jersey, United States

xiaoli.bai@rutgers.edu

Abstract: In this paper, the recently developed machine learning (ML) approach to improve orbit prediction accuracy has been systematically investigated using three ML algorithms, including support vector machine (SVM), artificial neural network (ANN), and Gaussian processes (GP). In a simulation environment consisting of orbit propagation, measurement, estimation, and prediction processes, totally 12 resident space objects (RSOs) in SSO, LEO and MEO are simulated to compare the performance of three ML algorithms. The results in this paper show that ANN usually has the best approximation capability but is easiest to overfit data; SVM is least likely to overfit but the performance usually cannot surpass ANN and GP. Additionally, the ML approach with all the three algorithms is observed to be robust with respect to the measurement noise.

Keywords: Resident Space Objects, Orbit Prediction, Machine Learning, Support Vector Regression, Artificial Neural Network, Gaussian Processes

Nomenclature

RSO(s)	Resident space object(s)
SSO	Solar-Synchronous Orbit
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
SVM	Support Vector Machine
ANN	Artificial Neural Network
GP	Gaussian Processes
RSW	Local orbital frame
e	Orbit prediction errors in RSW frame
e_x, e_y, e_z	Position components of e [km]
e_{vx}, e_{vy}, e_{vz}	Velocity components of e [m/s]
e_ξ	A general reference to one of six components of e above
e_T	Vector of true prediction error
e_{res}	Vector of residual error
\hat{e}_{ML}	Vector of ML-predicted orbit prediction error
$P_{\text{ML}}(e_\xi)$	Performance of the ML model on the component e_ξ
m	Number of learning variables for ML models
n	Number of data points in the training data
k	Number of basis functions of GP

1 Introduction

The number of resident space objects (RSOs) and the number of collision alarms between RSOs are rapidly escalating. One big challenge for Space Situational Awareness (SSA) is to improve orbit prediction accuracy. The 2009 satellite collision of Iridium 33 and Cosmos 2251 [1] reveals the lack of high-accuracy orbit prediction capability. Current approaches for orbit prediction are mostly physics-based. Their success requires a good knowledge of the forces such as Earth gravity, atmospheric drag, and solar radiation pressure. The maneuvering information, geometric information, and other features of the RSO are also necessary for high-accuracy predictions. The analytical model has been well developed, but the RSO's information is limited or hard to access.

The machine learning (ML) methods present a novel approach to extracting knowledge from large amount of data [2], which is similar to human cognition in learning from past experience to predict future events. The ML methods have shown great capability for a wide range of applications, including many in the aerospace field [3–5]. Different kinds of neural network models have drawn great attentions. Barton and McLaughlin [6] have applied the Long Short Term Memory networks to provide more accurate atmospheric density predictions than empirical models. Ertl and Christian [7] used the convolutional neural network to detect space objects that are only partially resolved in images; Parrish and Scheeres [8] applied neural networks to modify low-thrust trajectory on board; Smet et al. [9] carried out a systematic survey of Martian transfer orbit using neural networks to evolve the periapsis Poincaré map; and the neural network has also been combined with advanced controllers [10, 11].

Recently, a framework has been proposed to use the machine learning (ML) approach to improve orbit prediction accuracy for RSOs [12–15]. A supervised learning structure has been developed which learns a function from pairs of input and its output [2]. The supervised ML method is the appropriate approach for improving orbit prediction accuracy based on historical measurements and error information. The improvement can be made without explicitly modeling space objects, spacecraft maneuvers, and space environment, under the condition that the trained ML models are applied to cases similar enough to those during the training. In the proposed framework, for an orbit prediction based on a certain estimate of the RSO, an ML model trained by historical data will generate a compensation for the orbit prediction error, so that the modified orbit prediction is closer to the orbit's true location than the original prediction. Fundamentally, the ML approach aims to capture the relationship between the information practically available to the user and the unknown orbit prediction error. For example, the RSO's altitude and geometry information is usually available to operators but not to others. These relationships are often difficult to model analytically in practice because they depend on the features of the RSO (e.g. the area-to-mass ratio, reflection coefficient, etc.) that can be largely unknown. Additionally, nonconservative forces (e.g. atmosphere drag, etc.) cannot be modeled accurately. With the assumption that the relationships are embedded in the historical data, state-of-art ML algorithms can be used to learn the relationship from the data.

The proposed approach can be implemented with different ML algorithms. In separated studies, we have investigated three effective ML algorithms, including support vector machine (SVM) [12, 13], artificial neural network (ANN) [14], and Gaussian processes (GP) [15]. All the three algorithms have universal approximation capabilities to approximate smooth nonlinear functions. They have all shown good performance to improve the orbit prediction accuracy and each algorithm has its unique characteristics. Therefore, it is beneficial to provide guidelines on how to choose ML algorithms for the proposed ML approach.

In this paper, a comparison study of the three ML algorithms will be presented in a simulation environment using 12 RSOs in SSO, LEO, and MEO. The studied questions arise from the perspective of practical operations, rather than the theoretical aspects of ML studies. Through a series of numerical experiments, the following discussions are presented:

- The general learning capabilities of the three algorithms;
- The overfitting phenomenon;
- The effect of random initializations during the training;
- The capability of handling measurement noise with unknown magnitude.

The remaining part of the paper is organized as follows. In Section II, the background is presented for the following paper, including the ML approach, the design of learning and target variables, and the brief introduction of the three ML algorithms. In Section III, based on a series of numerical experiments, the three ML algorithms are compared and discussed in detail. At the last section, conclusions, suggestions, and future studies are provided.

2 Background

In this subsection, background is briefly reviewed, including the ML approach and the simulation environment; the design of learning and target variables for the proposed supervised learning problem; the basic concept of three ML algorithms; and the specific concerns for the aerospace problems.

2.1 Machine Learning Approach and Simulation Environment

The orbit prediction accuracy is usually improved by using more real dynamic models, using more powerful estimation methods, or building more accurate measurement equipment. In previous publications, we have proposed the ML approach to improve the orbit prediction accuracy through learning from historical data. Since the orbit prediction errors always exist, we assume there is always some information in the historical data that has not been utilized during the practical operations. This information might be hidden and coupled in a complicated way and therefore cannot be characterized by the current analytical modeling techniques. However, if we have large amount of data, many ML algorithms can be used to discover, capture, and model these errors. Afterward, the orbit prediction error can be compensated for based on the learned knowledge.

A simulation environment has been developed to investigate the ML approach. The detailed descriptions and parameters can be found in [12, 13]. To be consistent, in this paper the same simulation environment as in [13] is used. We note this framework is similar but has a few improvements over the simulation environment used in [12]. Additionally, studies using the TLE catalog [15, 16] have achieved similar results as those in the simulation environment, which could partially justify the setup of simulation environment. One advantage of using the simulation environment is that the parameters can be varied to investigate the capability of the ML approach in difference situations, such as the magnitude of measurement errors. As shown in Fig. 1, the truth model (in the gray frame) has the highest accuracy and is used for propagating the true orbit and generate measurements with three simulated ground radar stations; the assumed model (in the dashed blue frame) has lower accuracy and is used for the orbit determination and prediction processes. At last, the true orbit and the prediction are used to active the ML approach to improve orbit prediction accuracy, including data collection, machine learning training, and the performance analysis.

As elaborated in Ref. [13], the true model includes the Newtonian gravity, spherical harmonic gravity with EIGEN-6S model up to 40×40 , third-body perturbations of all the major solar system bodies with DE430 ephemeris data, atmosphere drag force with DTM2000 density model using Marshall solar activity future estimate data, and also the solar radiation pressure. Based on the true model, the assumed model is deliberately modified to have harmonic gravity model up to 10×10 , three-body perturbations of only the sun, moon and Jupiter, and a different atmosphere density model NRLMSISE-00 with average solar activity parameters $F_{10.7}$ and K_p . The simulation environment is implemented using the astrodynamics toolbox Orekit [17] and the Dormand-Prince 8(5,3) numerical propagator is used for the numerical propagations.

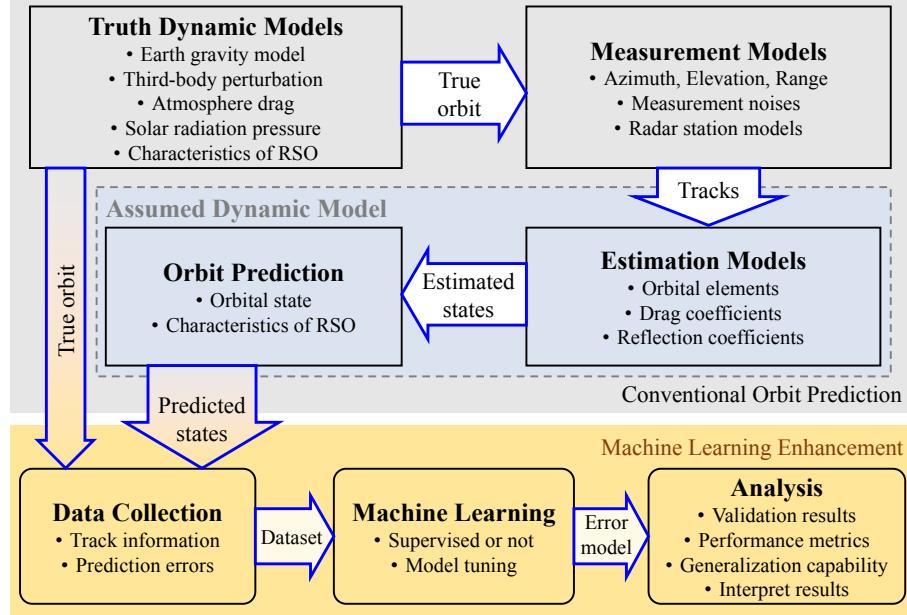


Fig. 1 Flowchart of the simulation environment and machine learning processes [12].

2.2 Learning and Target Variables

The ML approach to improve orbit prediction accuracy is modeled as a supervised learning problem, which is to learn the relationship between the input learning variables and the output target variables. Usually the relationship is nonlinear and cannot be explicitly formulated.

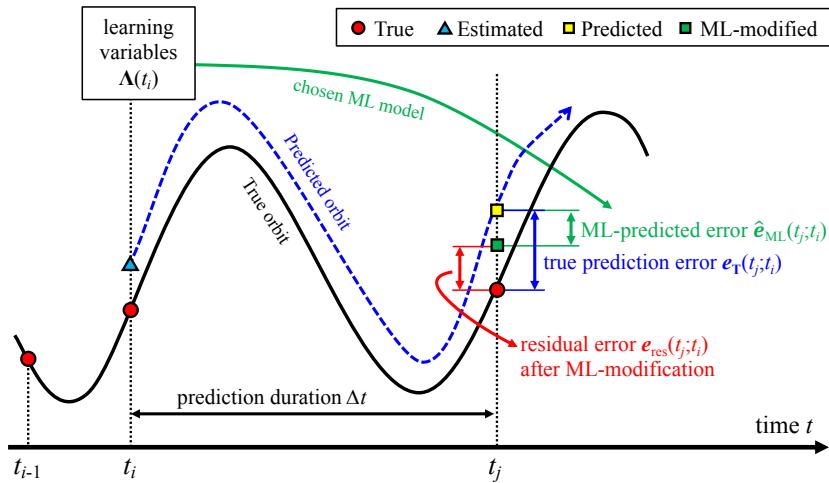


Fig. 2 Illustration of the proposed ML approach [13].

The learning and target variables for the proposed ML approach are designed based on the physics of the problem and the available information. Most of time it needs to be determined through a series of trial-and-error experiments. As shown in Fig. 2, the true prediction error $e_T(t_j; t_i)$ is expected to be reduced if one wants to improve orbit prediction accuracy, so it is chosen as the target variable. The true error is expressed in the RSW frame as $e_T = [e_x, e_y, e_z, e_{vx}, e_{vy}, e_{vz}]^T$, where x -axis is along the radial direction, z -axis is along the angular momentum direction, and y -axis completes the right-handed system. Totally six models will be trained separately for each component of

e_T . Detailed considerations and choices have been presented in the previous publications [13, 14]. Here, they are only briefly summarized for self-completeness:

- Prediction duration Δt to the future epoch t_j ;
- Estimated state $\hat{\mathbf{X}}(t_i)$ at the current epoch t_i expressed both as the orbital elements and the cartesian coordinates in the Earth-centered inertial (ECI) frame;
- Estimated drag coefficient $\hat{C}_d(t_i)$ at current epoch t_i ;
- Maximal measured elevation angle η in the current i -th track, and the corresponding range ρ and azimuth angle α at that epoch;
- Predicted state $\hat{\mathbf{X}}(t_j; t_i)$ at the future epoch t_j based on the current estimate $\hat{\mathbf{X}}(t_i)$, expressed both as the orbital element and in the ECI frame.

We note that the choice of the learning and target variables is consistent with [13, 14] but has been modified from [12].

The dataset of pairs of learning and target variables is collected by propagating all the estimates to the epochs of their subsequent estimates. For example, an estimate $\hat{\mathbf{X}}(t_i)$ will be propagated to the following epoch $t_j > t_i$ if there exists an estimate $\hat{\mathbf{X}}(t_j)$ at t_j ; after the propagation, the learning and target variables are stored in the dataset. The maximum orbit prediction duration Δt_{\max} is set to 7 days. Then, the dataset is used to train the chosen ML models. In the paper, a part of the dataset is chosen as the training data and another part is chosen as the testing data. After the ML model is well trained, it can generate an output $\hat{e}_{\text{ML}}(t_j; t_i)$ for each new set of learning variables. Therefore, we can compensate the true error $e_T(t_j; t_i)$ by subtracting the ML output, where the residual error is $e_{\text{res}} = e - \hat{e}_{\text{ML}}$. In an ideal case where all errors are captured in the ML model, the residual error will be zero.

2.3 Machine Learning Algorithms

Another important choice of the ML approach is the specific ML algorithms (models) to use. In this subsection, we briefly review the three ML algorithms that have been explored in previous publications. They all are supervised learning algorithms with universal approximation capabilities. Although mathematically different, they share some common features in the scope of statistical learning.

2.3.1 Support Vector Machine

The support vector machine (SVM) regression algorithm used in this paper is first proposed by Vapnik (1995) in *The Nature of Statistical Learning Theory**. The SVM algorithm handles nonlinear regression problems through the kernel trick which transforms inputs into a feature space using kernel functions. Then, a linear regression model can be built in the feature space. With the requirement that $\|\mathbf{w}\|$ should be minimized, where \mathbf{w} represents the weights of the regression model, training of the SVM model leads to a convex optimization problem, for which, its dual problem is simpler to solve numerically.

As shown in Figure 3, the SVM allows a soft margin of ε around the real function $f(\mathbf{x})$, so only the “support vectors” falling on the boundary of the margin boundary is used to build the regression model, and the other normal vectors falling inside the ε -margin are not used during the calculation. In this way, the soft boundary allows data with small noise. At the same time, the SVM model has a penalty term for “outliers” in the training data and will leave them outside of the boundary, rather than trying to fit these possibly wrong data. The training complexity of SVM is roughly $O(mn^2)$ where n is the size of training data and m is dimension of learning variables.

Because of the soft ε -margin, the SVM model is more likely to capture the general pattern in the dataset rather than trying to fit all the data. From another point of view, it could miss some detailed information within the ε -margin if ε is too large. The marginal width ε and other hyperparameters are usually determined based on experience or from cross

* A brief history of SVM, <http://www.svms.org/history.html>, accessed on 2018-10-25.

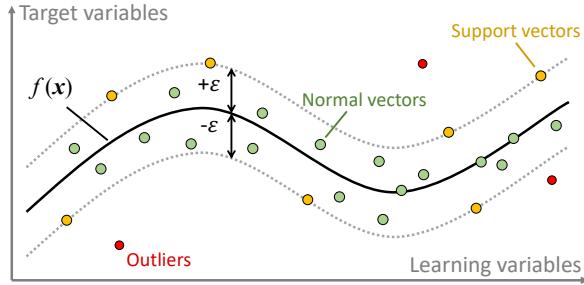


Fig. 3 Illustration of SVM algorithm.

validation and grid searching experiments. More details of the SVM model are referred to the references [18–20].

2.3.2 Artificial Neural Network

The artificial neural network (ANN) can approximate any smooth functions with adequate neurons and layers [21]. One important feature of ANN algorithm is that its layer structure, which makes it easier to use parallel computing techniques. In recent years, many attentions have been attracted to ANN such as the deep convolutional neural networks for image recognition. In this paper, a feedforward network with full connections is used.

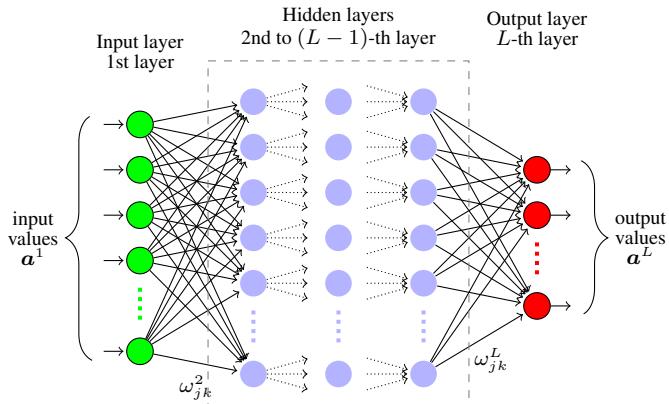


Fig. 4 Illustration of ANN algorithm [14].

Figure 4 shows an ANN with L layers including the input and output layer, so there are $L - 2$ hidden layers. The circle at the input layer corresponds to each learning variable; the circle at the output layer corresponds to each target variable; and the hidden layers contain the artificial neuron which will generate a single value depending on all its connections to the neurons in the last layer and the weights of the connections. The hidden layers are flexible to design, including the number of hidden layers, the number of neurons in each hidden layer, and the activation function that governs how each neuron corresponds to all its inputs. Once the structure is fixed, the ANN is trained by optimizing the cost function defined as the mean square error (MSE) $C_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N \|e_{T,i} - a_i^L\|^2$, where N is the training data size and a_i^L is ANNs i -th output. The time complexity for training is roughly $O(n^3)$ but greatly depends on the structure and implementation. More details of ANN are referred to the references [21–23].

The ANN has great flexibility on the structure, leading to many free parameters to tune either automatically or manually. This makes ANN powerful at extracting information from data. On the other hand, this also makes it possible to purely memorize all the training data if too much freedom is designed or too few data are used for training.

2.3.3 Gaussian Processes

The Gaussian Processes (GP) is a Bayesian inference method which assumes a Gaussian process prior over the underlying function between the learning and target variables. Gaussian process is a special stochastic process of a collection of infinitely number of variables such that any subset of the random variables has a multivariate normal distribution. GP can also be thought of as a Bayesian alternative to the kernel methods such as SVM [24].

An important advantage of GP is that it can provide uncertainty information about its prediction, while classical methods of SVM and ANN do not have this capability. For the orbit prediction problem, the uncertainty information is crucial for operations like conjunction analysis. However, the information comes with a price that the time complexity to train the GP model is $O(n^3)$ due to the inversion of the n -dimensional covariance matrix, where n is the training data size. So GP can be computationally infeasible for large scale problems.

In this paper, one specific sparse GP algorithm capable of learning heteroscedastic noise is used [25, 26], which belongs to a wide range of sparse GP methods [27]. Its time complexity is roughly $O(mn^2)$ where m is the number of its basis functions. It not only handles large amount of data, but also allows the uncertainty varying with respect to the learning variables. Since the scope of this paper is to compare GP with SVM and ANN, the uncertainty output of GP is not discussed. More details about GP are referred to [24, 28, 29].

2.4 Appropriate Design of the ML approach

As introduced in previous subsections, the ML approach is designed as a data-driven methodology to explore and extract potential relationships between the learning and target variables using advanced ML algorithms. It is different from conventional analytical and numerical modeling techniques, so the questions to address when applying the ML approach to practical problems are also different. We provide some insights based on our experience in developing the above three ML algorithms.

The ML algorithm is essentially a statistical modeling algorithm. Both the input and output are assumed having noise with specific distributions. Consequently, it could be less accurate than a precise physical model. However, its advantage is that it can avoid many assumptions made during analytically modeling.

Like how human learns from observation, the ML approach learns from the training data. So, the choice of learning variables and the quality of the data are important to the performance of the trained ML models. Theoretically, if a certain relationship between the inputs and outputs is known to exist, all the three ML algorithms with well designed architectures and hyperparameters can fit the nonlinear function with adequate training data. However, if the existence of relationship between the learning and target variables is unknown, there will be no theoretical guarantee for good performance. In this circumstance, the ML algorithm is used to explore and recover hidden information in the dataset.

In the case of the proposed ML approach for orbit prediction problems, based on the astrodynamics fundamentals that the prediction is obtained through propagation of an imperfect assumed force model, the target variables (orbit prediction errors) will be the results of the unmodeled force, measurements with noise, and estimation processes. By choosing the learning variables to be related to the initial states, the propagation, and the measurements, it is guaranteed that there is some relationship between the learning and target variables. And, the ML algorithm is targeted to recover the relationship from available data. After the ML model is trained, if the generalization performance (to test trained models on a testing data) is good, it is possible that we have found a hidden relationship. However, we cannot conclude that there is no such a relationship if the generalization is not as good as expected. The causes of bad results vary, including:

- The training data is not adequate;
- The hyperparameters of the ML algorithm are not well tuned;

- The numerical implementation needs improvement;
- The target variables are too noisy;
- The relationship does not exist between the designed learning and target variables.

Assuming good generalization performance has been obtained for the designed ML approach, there are additional concerns from the operational point of view, including:

- (1) Although a particular ML algorithm can be more suitable to certain types of problems, the designed ML approach should be capable of achieving similar results using different ML algorithms having the universal approximation feature;
- (2) The ML algorithm should not be prone to overfit data, or there should be a mechanism to prevent overfitting;
- (3) The training results should be insensitive to randomness during initialization or training, or there should be a mechanism to evaluate the effect of randomness.

In the following paper, these three items will be investigated for the comparison of the three ML algorithms. We note that there are other more practical concerns not investigated in this paper, such as the robustness of the performance with respect to outliers in the training data; the computational burden and efficiency to achieve above performance; the optimal design of learning variables; and the capability to incorporate additional information in the future.

3 Comparisons and Discussions

In this section, the results of the ML approach using the three ML algorithms are compared and analyzed in detail. Some conclusions and insightful discussions are presented. Open questions are also pointed out for future study.

3.1 Simulations

The experiments and simulations in this paper are designed to compare characteristics of the three ML algorithms we have used in the previous publications. Before discussing the results, some details and considerations are presented in this subsection.

3.1.1 Studied RSOs

We have simulated 12 RSOs distributed in the SSO, LEO and MEO orbits to evaluate and compare the three ML algorithms. The initial states of the RSOs are summarized in Table 1, where the initial orbital elements are converted from the TLE sets using the SGP4 model at the given epochs and the mass and area information are assumed based on public data[†]. The first four RSOs are in SSO, the next four RSOs are in LEO, and the last four RSOs are in MEO. In previous studies, we have used the same RSOs but those simulations started from different epochs.

3.1.2 Type-II Generalization

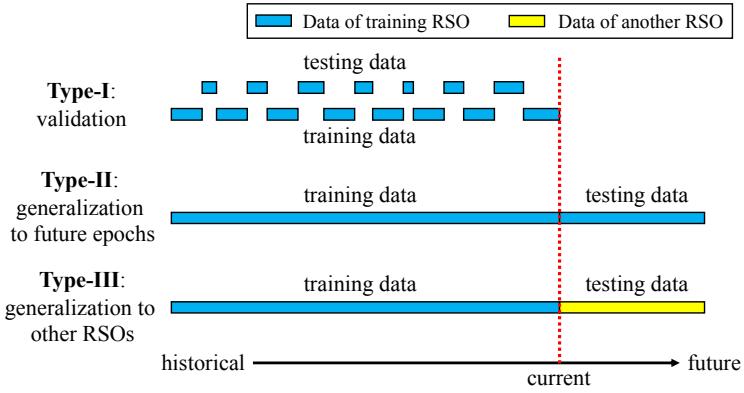
Once the ML model is well trained, a testing data is used to test the knowledge it has learned from the training data. If there is really a relationship and the ML model correctly captures it, the performance on the testing data should be good. This process is referred to as the generalization capability of the trained ML model.

In previous publications, we have proposed three critical types of generalizations for the ML approach to improve orbit prediction accuracy, as shown in Fig. 5. For the type-I generalization, the training and testing data are randomly sampled from the same dataset, which, for the orbit prediction problem, means they occur in the same historical time interval. For the type-II generalization, we add a constraint that the generalization must be from historical to future epoch, which means the epoch of the testing data is after the training data. The type-III generalization is a further extension of type-II while requiring the generalization is among RSOs [12, 14, 15].

[†]List of Supported Missions from International Laser Ranging Service, <https://ilrs.cddis.eosdis.nasa.gov/missions/satellite.missions>, accessed on 2018-10-25.

Table 1 Initial states of the simulated RSOs.

RSO #	NORAD ID	Simulated RSO	Orbit Type	Epoch [UTC]	a [km]	e	i [deg]	ω [deg]	Ω [deg]	M [deg]	Period [min]	Area [m^2]	Mass [kg]
1	27944	LARETS	SSO	2018-09-03T12:05:05.499	7067.3	0.00122	98.1	8.4	15.0	-8.5	98.5	0.05	23.28
2	22824	STELLA	SSO	2018-09-03T04:26:14.000	7184.1	0.00114	98.8	-216.3	-146.8	47.9	101.0	0.05	48.00
3	35871	BLITS	SSO	2018-09-01T05:00:49.660	7203.6	0.00107	98.3	53.7	-113.4	-53.6	101.4	16.00	8211.00
4	37781	HAIYANG-2A	SSO	2018-09-03T13:18:38.326	7352.6	0.00115	99.2	66.3	-104.8	-66.1	104.6	38.94	1500.00
5	39452	SWARM-A	LEO	2018-09-03T04:26:54.389	6821.0	0.00151	87.4	70.0	8.7	-70.0	93.4	1.28	369.00
6	7646	STARLETTE	LEO	2018-09-03T11:18:39.796	7340.0	0.02162	49.9	30.1	168.4	-28.7	104.3	0.05	47.50
7	38077	LARES	LEO	2018-09-03T02:45:37.617	7827.7	0.00049	69.6	150.6	109.9	-150.5	114.9	0.10	400.00
8	16908	AJISAI	LEO	2018-09-03T12:49:57.606	7871.4	0.00211	49.9	137.3	-108.9	41.9	115.8	3.63	685.00
9	22195	LAGEOS-2	MEO	2018-09-02T16:39:02.572	12164.4	0.01383	52.5	-334.4	-95.9	177.4	222.5	0.28	405.38
10	8820	LAGEOS-1	MEO	2018-09-03T05:38:42.907	12272.5	0.00442	109.7	182.4	-75.1	-152.7	225.5	0.28	406.97
11	20026	ETALON-2	MEO	2018-09-01T17:33:32.827	25497.3	0.00150	65.3	-147.0	47.2	56.7	675.3	1.32	1415.00
12	19751	ETALON-1	MEO	2018-09-03T03:40:20.700	25502.2	0.00201	64.2	-157.1	169.6	39.3	675.5	1.32	1415.00

**Fig. 5** Illustration of the three types of generalization [30].

In this paper, the type-II generalization with different ML algorithms is investigated. The RSO has been simulated long enough to collect the full dataset. Then, the training data is selected in the days 1–28 with the testing data in the days 29–35. The sizes of the resulting data for different RSOs are summarized in Table 2. For the LEO and MEO objects, there are 17,000~40,000 data points in the training data. For the MEO objects, as the altitude increases, the number of data points quickly drops from several thousands to just four hundred. For the first 10 RSOs, the data is adequate, but for the last two RSOs, some observed phenomenons may be related to the overfitting effect due to inadequate data, as will be discussed in Sec. 3.4.

Table 2 Number of data points in the training and testing data for each RSO.

RSO	Training (days 1–28)	Testing (days 29–35)
1	48447	14212
2	38931	11693
3	40077	11431
4	34461	10148
5	44681	13279
6	27661	8347
7	20275	5756
8	17067	4887
9	4365	1281
10	6425	1896
11	423	147
12	408	115

Additionally, the training algorithm of the ML model usually has some degrees of randomness and the ML models can be chosen by varying random initializations until achieving satisfying performance. In this paper, we use five fixed random seeds, $\{42, 401, 953, 744, 616\}$, for each case of experiments and then use the best performance for comparison purposes. We note that the choice of five seeds is a compromise between computational feasibility and the best performance.

3.1.3 Implementation of ML algorithms

The statistical and machine learning toolbox of MATLAB is used to implement the SVM models. The neural network toolbox is used to implement the ANN models. A third-party GPz package (<https://github.com/OxfordML/GPz>) is adapted to implement the GP models. MATLAB 2017b is used, on a macOS 10.13.3 platform with 3.7 GHz Quad-Core Intel Xeon E5 CPU. Additional parameters of the ML model and the training parameters are determined through trial-and-error, pursuing the best performance as well as the computational efficiency with the designed dataset.

For the SVM models, the gap tolerance is 10^{-2} for the optimization solver; the soft margin ε is chosen as a quarter of median of the corresponding target variable e_ξ in the training data; the sequential minimal optimization (SMO) algorithm is used to solve dual optimization problem. For ANN models, the network structure is 1 hidden layer with 20 neurons for all the x, y -axis components and 2 hidden layers with 20 neurons for all the z -axis components; the activation function is log-sigmoid function for the hidden layers and pure linear function for the output layer; the Levenberg-Marquardt backpropagation algorithm is used to solve the training problem [14]. For GP models, the number of the basis function is 20 for all components; the precision matrix of basis function is diagonal; the maximum number of iterations and attempts before termination are 500 and 50 respectively; the LBFGS algorithm is used to solve the training problem [30].

We note that a ML model can overfit the training data if it has too much flexibility or the training data is inadequate. Under such situations, the trained ML model will precisely memorize each data point in the training data, but it will have bad performance on new inputs different from the training data. In the machine learning research areas, many approaches have been proposed to prevent overfitting. For the SVM models in this paper, the overfitting is prevented through choosing a proper ε -margin size. In our implementations of ANN and GPR, the last day of the training data is reserved as a validation data during the training, and then the performance on the validation data is monitored iteratively. When the performance on the validation data cannot be further improved, it is assumed that the ML model has started to overfit and therefore the training process is terminated.

3.2 General Learning Capability

The performance of the trained ML models on the testing data is evaluated by a metric $P_{\text{ML}}(e_\xi)$ which we have introduced in the previous studies,

$$P_{\text{ML}}(e_\xi) = 100\% \cdot \frac{\sum_{i=1}^n |e_{\text{ML},\xi,i} - e_{\text{T},\xi,i}|}{\sum_{i=1}^n |e_{\text{T},\xi,i}|} \quad (1)$$

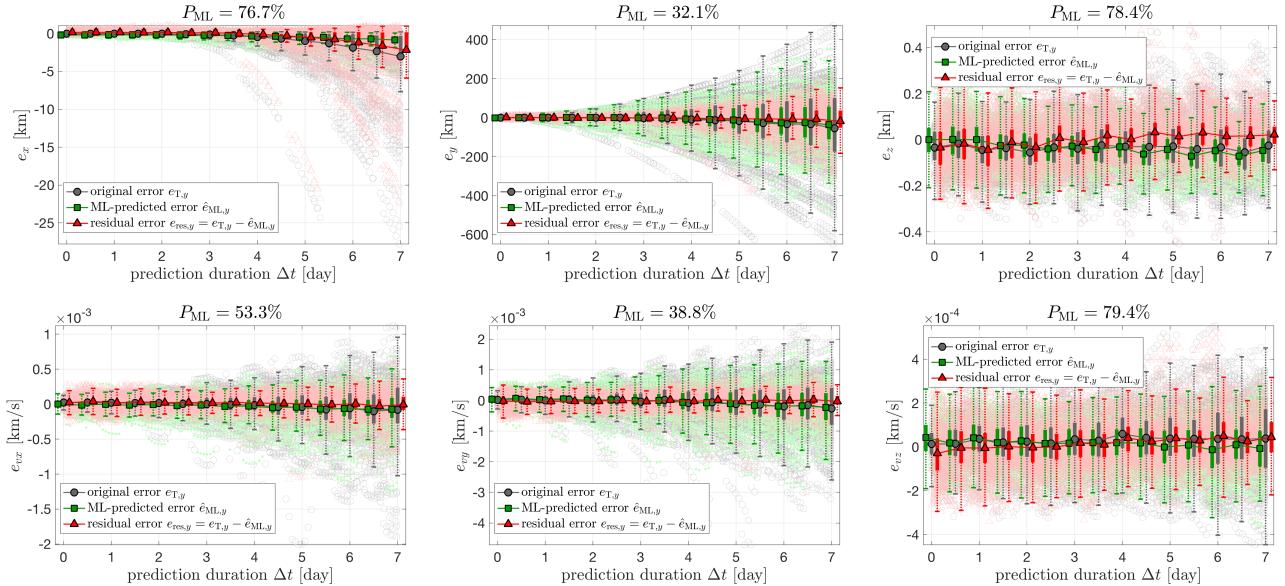
The metric quantifies the percentage of the total absolute residual error with respect to the total absolute true error. If all orbit prediction errors have been compensated for, the metric P_{ML} should be zero. In reality, there will always be residual errors due to either the random noise in the orbit determination process or the ML algorithms limitations.

The best performance of all the six components of all the RSOs are summarized in Table 3. Here, the best performance means that the metrics P_{ML} are the minimum among the experiments using the five random seeds for all the six components of each RSO using the particular ML model. For each component, the value with overline indicates the largest P_{ML} among SVM, ANN, and GP algorithms, while the one with underline indicates the smallest P_{ML} . For the components e_x, e_y, e_{vx}, e_{vy} of most RSOs, SVM generates the largest metrics and ANN generates the smallest. On the contrary, for the components e_z and e_{vz} of many RSOs, ANN generates the largest metrics but

Table 3 Best performance metrics among random initializations of six components of all the RSOs (lower is better).

RSO	$P_{\text{ML}}(e_x)$			$P_{\text{ML}}(e_y)$			$P_{\text{ML}}(e_z)$			$P_{\text{ML}}(e_{vx})$			$P_{\text{ML}}(e_{vy})$			$P_{\text{ML}}(e_{vz})$			
	#	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP
1		76.7	10.9	18.5	32.1	4.3	6.4	78.4	86.6	120.4	53.3	53.9	51.5	38.8	32.2	35.3	79.4	116.5	114.7
2		76.7	19.5	33.8	26.7	4.7	4.9	92.3	81.8	104.9	75.2	74.3	69.5	52.4	42.9	43.7	92.0	97.6	102.2
3		83.9	23.7	27.3	34.0	5.3	6.7	82.1	128.5	91.0	82.4	83.9	87.5	66.8	44.8	57.4	77.9	121.3	92.6
4		89.0	31.6	32.7	43.9	6.8	12.8	68.8	94.6	75.2	71.8	35.5	40.8	48.4	29.4	22.0	70.6	98.2	85.0
5		80.0	11.2	16.1	30.2	5.6	5.7	85.8	105.6	76.4	60.4	36.7	41.6	51.7	29.1	28.2	90.7	107.3	94.8
6		71.3	34.4	18.6	43.7	13.2	14.9	92.9	143.5	87.2	60.1	21.9	33.2	56.1	33.9	19.8	89.8	101.3	79.1
7		95.2	22.8	44.2	59.5	6.6	10.8	92.0	136.8	137.4	78.7	48.2	40.0	57.3	14.8	18.0	88.4	133.5	92.6
8		98.6	76.5	46.3	72.4	16.0	30.9	77.9	108.5	78.6	82.9	147.7	115.1	71.0	76.5	59.5	84.2	105.9	76.3
9		97.3	15.4	92.6	73.8	1.8	17.0	80.1	95.0	70.5	82.8	12.7	37.9	74.8	3.5	26.2	78.9	95.5	81.3
10		79.4	22.7	45.8	50.2	5.9	3.2	106.2	115.8	90.6	68.3	42.7	36.7	47.0	5.2	11.2	111.7	118.9	106.1
11		96.6	151.3	57.7	96.3	26.2	17.3	92.2	38.4	28.7	99.6	70.1	27.6	97.8	19.9	8.2	101.2	69.2	42.3
12		104.3	72.6	63.0	100.7	46.8	10.0	99.8	41.1	50.8	101.2	47.4	30.9	99.6	4.9	4.6	98.5	61.0	37.5

SVM generates the smallest. This phenomenon will be further analyzed in the following subsections.

**Fig. 6** Performance of trained SVM models on RSO-1, corresponding to the first row in Table 3.

Taking RSO-1 as an example, the detailed performance of trained SVM models is shown in Fig. 6. The horizontal axis shows the prediction duration Δt and the vertical axis shows the original/ML-predicted/residual orbit prediction errors respectively. The testing data has been grouped every half day, and the standard boxplot is used to show statistics of each grouped data: the central marker is the median; the solid bar stretches from the first quartile q_1 to the third quartile q_3 ; the dashed line shows the minimum to the maximum after eliminating outliers falling outside of $[q_1 - 1.5\Delta q, q_3 + 1.5\Delta q]$, where $\Delta q = q_3 - q_1$ is the interquartile. The underlying markers in light colors show the actual data. From the figure, it could be observed that

- the medians and interquartiles of e_x , e_y , e_{vx} , and e_{vy} are both increasing with Δt ;
- the medians and interquartiles of e_x , e_y , e_{vx} , and e_{vy} can be significantly reduced, especially for e_y ;
- the medians of e_z and e_{vz} are close to zero and the interquartiles are almost constant with respect to Δt ;
- the medians and interquartiles of e_z and e_{vz} are slightly reduced.

As has been discussed in the previous publication of ANN [14], the improvement of e_z and e_{vz} along z -axis (the cross-track direction) is not as remarkable as those along x - and y -axis. This is because the cross-track motion in the assumed model has been modeled too accurate such that the simulated orbit prediction errors are mainly noise that cannot be removed. If we decrease the harmonic order and degree to only the J_2 perturbation in the assumed model, the orbit prediction error will increase and at the same time the ML approaches performance will be improved [14].

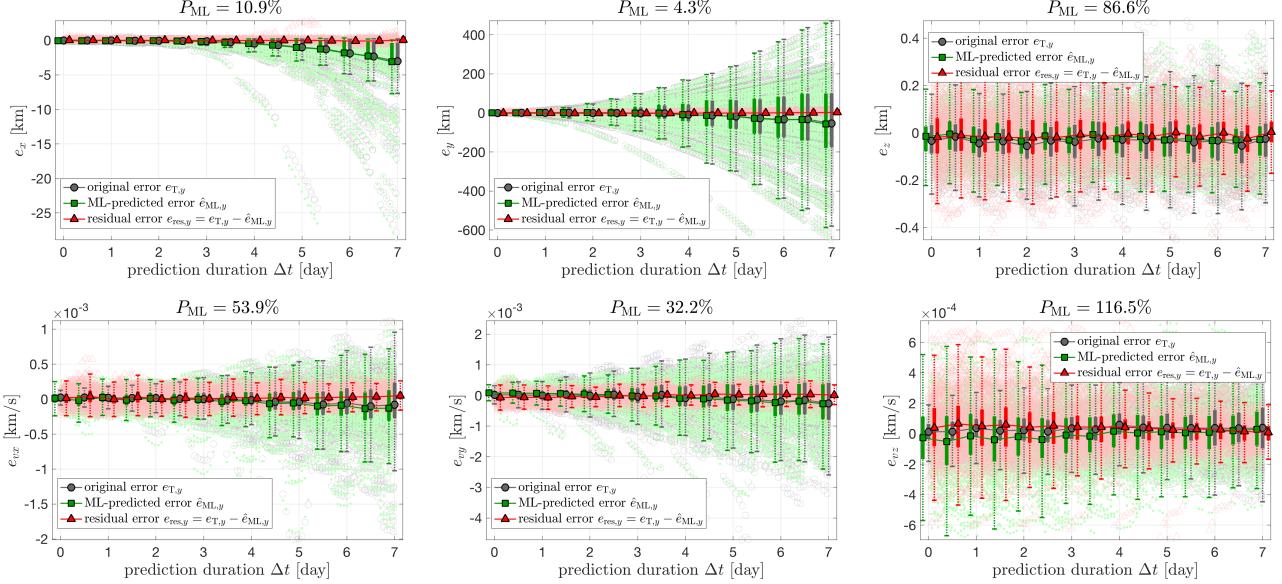


Fig. 7 Performance of trained ANN models on RSO-1, corresponding to the first row in Table 3.

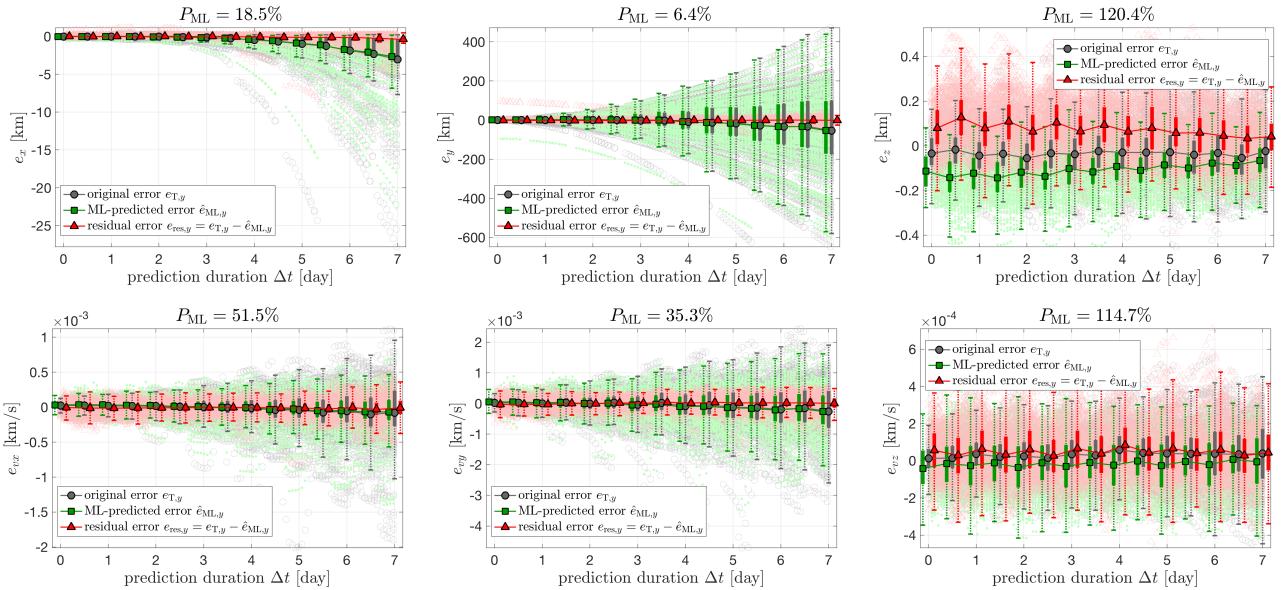


Fig. 8 Performance of trained GP models on RSO-1, corresponding to the first row in Table 3.

The performance using ANN and GP models is shown in Figs 7 and 8 respectively. The patterns are similar to those of SVM models, except that the specific values of P_{ML} are different. It is interesting to notice that the performance of ANN or GP on e_x and e_y has been greatly improved compared with SVM, which is expected because these two algorithms have more flexibility and thus more powerful regression capability.

Comparison on z -axis shows that the metrics of ANN and GP on e_z , e_{vz} are greater than that of SVM. This phenomenon represents the cases when the learning and target variables are mostly noise rather than a true relationship, ANN and GP tend to overfit the training data and therefore lead to bad generalization performance.

At the end, it can be concluded that the learning capability of the proposed ML approach is roughly ANN > GP > SVM, if there is no overfitting.

3.3 Possibility of Practical Overfitting

The supervised ML algorithm is essentially a data-driven modeling technique that captures the relationship between pairs of learning and target variables in the training data. If the assumed relationship really exists, it can be applied onto any new input learning variables following the same relationship. However, it is also possible that the ML algorithm is simply “memorizing” the pairs in training data, rather than revealing a relationship, which is usually referred to as an *overfitting* phenomenon. The causes of overfitting vary. For example, the training data size is too small with respect to the regression capability of the ML algorithm; the target variables do not depend on the designed learning variables; or the random initialization of training process lead to a bad local minimum.

It is difficult to precisely detect or completely prevent overfitting for a general problem. Here, from the application’s viewpoint, we define an alternative concept of “practical overfitting”: if a trained ML model’s performance metric P_{ML} is smaller than 100% on the training data but is much larger than 100% on the testing data, it is considered to be overfitting. For simplicity, all the following discussion of “overfitting” refer to the defined concept of “practical overfitting”.

If the designed learning and target variables have no relationship at all, the learning variables should be irrelevant to the target variable and the trained ML models will generate output of all zeros. The metric is expected to be around 100% (e.g. 90%~110%). However, if the trained ML models fit the pure numerical patterns in the training data and then generalize it onto the testing data, the most possible result is to add wrong modifications to the testing data, leading to metrics much greater than 100% (e.g. >120%).

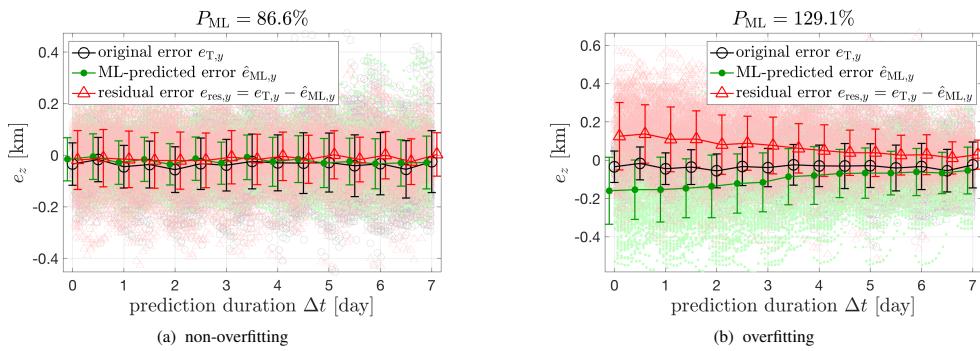


Fig. 9 Two examples of ANNs testing results on e_z of RSO-1 using different random seeds for training.

The results in Table 3 in the last subsection have already shown that ANN and GP have performance metrics greater than 100% for many cases, for example RSO-1’s $P_{\text{ML}}(e_z)$ using GP, RSO-8’s $P_{\text{ML}}(e_{vx})$ using ANN, and RSO-11’s $P_{\text{ML}}(e_x)$ using ANN. Taken e_z of RSO-1 as example, the non-overfitting and overfitting performance of ANN are shown in Fig. 9. The original error (black curves) is centered on zero with an almost constant interquartile, which reflects that e_z has little relationship with Δt . When the ANN model is not overfitting, as shown in Fig. 9(a), the residual error (red curves) is still close to zero and the interquartiles are only slightly varied. However, when the model is overfitting, as shown in Fig. 9(b), the residual error is obviously deviated from zero, which indicates the

ANN model has captured a wrong relationship.

It is helpful to be aware of that overfitting cannot be completely avoided for data-driven models. During the design of the ML approach, it should be understood that sometimes a specific choice of the learning variables or the ML algorithm can lead to overfitting.

3.4 Effect of Random Initialization

As explained before, training the ML model is essentially an optimization process which can introduce some degrees of randomness. In our implementations, for SVM, the randomness comes from the determination of the scale parameter of the kernel function [12]; for ANN, it comes from the algorithm to set initial weights of each neurons [14]; for GP, it comes from the choice of initial center vectors [15].

In this paper, we have tested five random seeds to initialize the training and then to explore the effect of randomness for the designed ML approaches with different algorithms. We note that five random cases are not adequate to discover the overall dependency of the performance on random initializations, but it already can reveal how significant the effect is, which helps design the ML approach.

Firstly, the four RSOs in SSO are analyzed since they are in a relatively more stable environment. In Fig. 10, six components of the orbit prediction error are shown in separated panels. The horizontal axis shows the RSO index and the vertical axis shows the metrics P_{ML} . In each panel, the asterisks show SVM results of five random seeds, the triangles show ANN results, and the squares show GP results. The metrics of SVM concentrate on a very small region, meaning that the effect of random seeds is mild. The metrics of ANN and GP could be lower than those of SVM in most cases, although they are also scattering in broader intervals than those of SVM. It can be observed that for e_x , e_y , e_{vx} , and e_{vy} , ANN and GP can get a better performance than SVM; but for e_z and e_{vz} , ANN and GP can hardly surpass SVM.

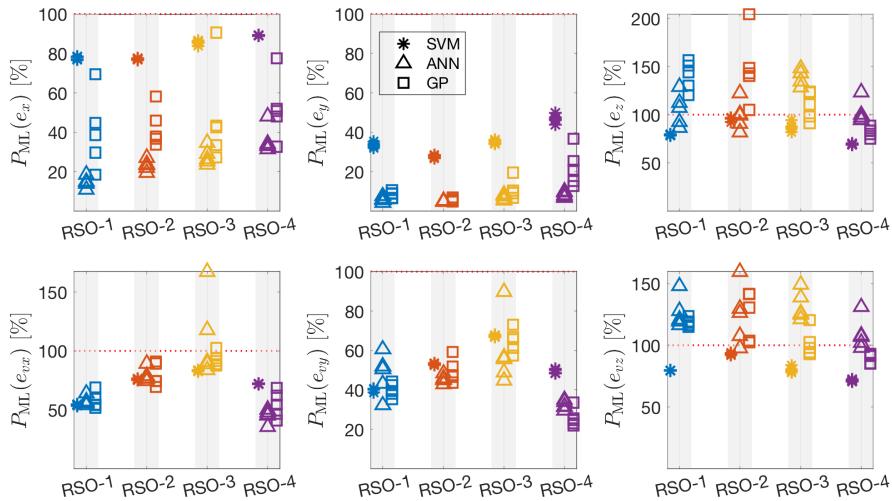


Fig. 10 Metrics distribution of SVM/ANN/GP models using different initializations: RSO-1 to 4 in SSO.

Secondly, the performance on RSO-5 to 8 in LEO is shown in Fig. 11. Generally, the best performance is not as good as those in Fig. 10, as also suggested in Table 3. This is partially because the LEO environment is more complicated than that of SSO. For example, the precession motion is severe; and the atmosphere density is changing with the lightening condition. For RSO-5 to 7, most patterns are similar to RSO-1 to 4: ANN and GP have better performance than SVM but are also more sensitive to the initialization. For RSO-8, ANN and GP both show many overfitting results while SVM appears very stable. Focusing on the overfitting results, above the 100% dotted line, it

can be observed that GP can always achieve less overfitting than ANN in the best cases.

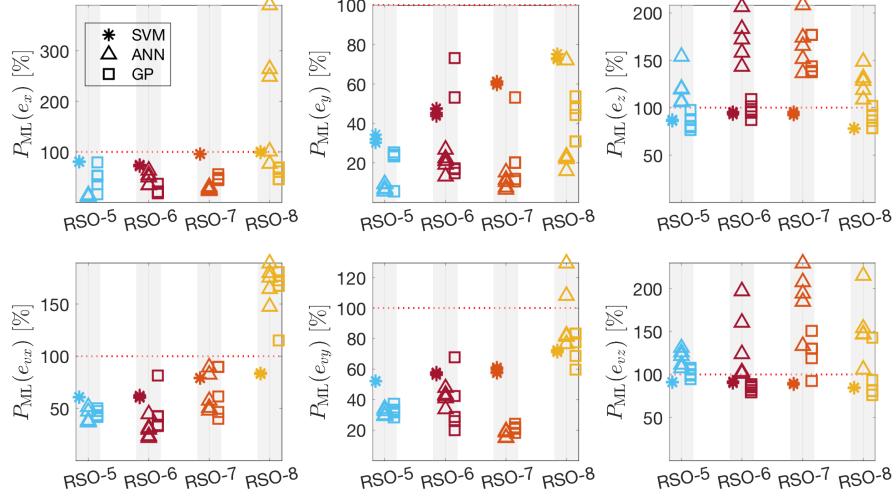


Fig. 11 Metrics distribution of SVM/ANN/GP models using different initializations: RSO-5 to 8 in LEO.

Thirdly, Figure 12 shows the results on RSO-9 to 12 in MEO. In most cases, SVM and GP show good performance and ANN can reach the best performance. It is also observed that the performance on e_z and e_{vz} of RSO-11 and 12 can be bad now, the performance on e_y and e_{vy} also becomes worse than those in Figs 10 and 11. One possible reason is that the assumed model and the orbit determination process in the simulation environment have not considered the variation of solar radiation pressure which depends on the RSO's geometry and attitude. However, it is nice to see that the ML approach can still improve the accuracy in Fig. 12. Further studies will be carried out on MEO or GEO objects using better assumed models.

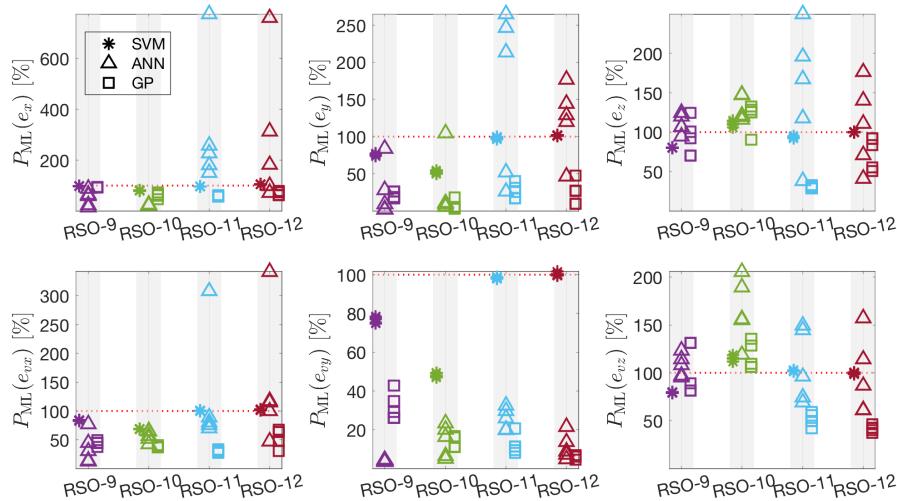


Fig. 12 Metrics distribution of SVM/ANN/GP models using different initializations: RSO-9 to 12 in MEO.

A closer comparison between ANN and GP in all the Figs 10 to 12 shows that GP's best performance is closer or even better than that of ANN, and meanwhile it is less likely than ANN to overfit with a random initialization. As the end, it can be concluded that the robustness of the ML approach with respect to random initializations are SVM > GP > ANN.

3.5 Capability of Handling Various Measurement Noise

In practice, the measurement noise σ_{measure} is usually unknown to the end-user of the RSO catalog. Although the ML approach does not require this information as a priori input, it is helpful to analyze its performance in various situations. The ML approach is expected to be robust so that it will always be effective regardless of the magnitude of measurement noise.

Additional experiments are carried out in the simulation environment by increasing the magnitude of measurement noise σ_{measure} to as large as 5 times of that of the original experiment. For the simulated radar stations in this paper, the noise is added to the measurements of azimuth, elevation, and range. The performance metrics P_{ML} of RSO-1 to 4 in SSO are summarized in Table 4, which are the best one among the five experiments using different random seeds. The second column shows the multiple of the original measurement error σ_{measure} . When the measurement noise increases, the metrics of e_x , e_y , e_{vx} , and e_{vy} are stable and the metrics of e_z and e_{vz} are decreasing. According to the definition of the metric, it quantifies the relative percentage of the residual error with respect to the original true error. The results in Table 4 for each multiple of σ_{measure} could reveal the effectiveness of the ML approach, but comparisons among different P_{ML} are inappropriate here. For example, if the magnitude of e_x is 1 km, a small $P_{\text{ML}}(e_x)$ of 10% indicates the residual magnitude is only reduced by around 0.9 km; nevertheless, if the magnitude of e_y is 10 km, even a big $P_{\text{ML}}(e_y)$ of 60% indicates a reduction about 4 km. In other words, the magnitude of the residual error should also be compared to evaluate the variation of the performance of the ML approach in this circumstance.

Table 4 Best performance metrics P_{ML} among five random seeds on RSO-1 to 4 using all the three ML algorithms, with measurement noise varying from 1 to 5 times of the original magnitude.

RSO	Multiple of σ_{measure}	$P_{\text{ML}}(e_x)$ [%]			$P_{\text{ML}}(e_y)$ [%]			$P_{\text{ML}}(e_z)$ [%]			$P_{\text{ML}}(e_{vx})$ [%]			$P_{\text{ML}}(e_{vy})$ [%]			$P_{\text{ML}}(e_{vz})$ [%]		
		SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP	SVM	ANN	GP
1	1	76.7	10.9	18.5	32.1	4.3	6.4	78.4	86.6	120.4	53.3	53.9	51.5	38.8	32.2	35.3	79.4	116.5	114.7
	2	79.8	14.9	9.8	33.1	4.3	18.3	63.5	96.0	88.4	56.4	50.3	48.7	39.2	30.9	36.0	69.2	81.0	83.7
	3	79.7	12.0	24.4	31.9	5.0	18.2	56.3	78.0	79.3	54.7	47.0	53.8	35.4	21.9	28.4	61.5	75.3	68.4
	4	77.2	8.7	56.9	29.9	4.2	6.9	52.0	84.5	66.1	54.8	31.8	54.0	35.2	20.2	16.4	57.8	82.4	61.6
	5	76.6	8.5	67.4	28.4	4.3	4.4	50.4	61.3	52.1	55.1	19.8	38.6	33.6	14.7	13.6	54.4	59.8	53.3
2	1	76.7	19.5	33.8	26.7	4.7	4.9	92.3	81.8	104.9	75.2	74.3	69.5	52.4	42.9	43.7	92.0	97.6	102.2
	2	75.5	14.9	18.5	24.9	4.6	4.4	83.7	93.6	109.6	72.0	59.4	74.7	45.6	34.5	31.2	84.5	105.4	108.4
	3	77.4	15.6	25.7	25.0	4.8	19.7	75.0	95.3	84.2	71.5	51.1	76.6	42.1	27.5	24.7	77.7	111.6	104.1
	4	76.6	13.8	48.6	24.9	3.9	17.4	69.3	81.4	67.3	65.5	31.7	71.6	36.7	15.2	18.9	73.2	99.3	84.1
	5	74.7	10.6	33.6	23.8	5.9	10.9	68.5	75.8	61.6	62.9	28.6	71.0	33.9	12.3	14.6	71.2	91.8	84.2
3	1	83.9	23.7	27.3	34.0	5.3	6.7	82.1	128.5	91.0	82.4	83.9	87.5	66.8	44.8	57.4	77.9	121.3	92.6
	2	84.7	11.5	19.3	33.7	5.0	8.9	67.5	84.7	73.7	77.7	39.2	51.0	53.3	36.9	29.3	65.9	89.5	79.2
	3	86.7	18.1	51.3	33.1	6.5	17.8	57.2	75.2	60.7	74.6	33.4	47.0	49.2	27.0	26.5	60.8	84.0	68.5
	4	86.2	20.8	70.5	34.9	4.7	18.5	55.1	68.0	53.5	69.1	26.2	45.8	44.6	18.2	18.1	56.5	78.1	63.0
	5	86.9	14.8	52.1	33.7	4.9	7.2	53.4	61.8	53.7	68.2	19.3	31.7	38.9	14.4	14.6	55.9	76.4	60.7
4	1	89.0	31.6	32.7	43.9	6.8	12.8	68.8	94.6	75.2	71.8	35.5	40.8	48.4	29.4	22.0	70.6	98.2	85.0
	2	91.4	35.9	39.1	45.2	7.1	18.0	57.2	81.7	67.9	71.2	22.6	27.5	45.2	13.3	14.4	62.7	90.4	72.0
	3	92.9	19.3	64.5	46.5	4.9	17.9	51.4	68.6	59.6	74.1	14.5	30.6	45.5	10.1	11.3	57.8	67.4	64.1
	4	92.2	34.3	73.9	48.3	2.8	17.0	47.6	58.3	54.4	74.3	13.5	27.7	40.7	8.8	9.8	54.0	74.0	52.4
	5	91.6	30.2	76.9	47.2	5.7	3.8	46.4	54.1	48.2	74.8	16.2	21.2	40.9	7.1	9.1	52.4	69.9	51.2

For each component in the testing data, the true orbit prediction error is varying with respect to the prediction duration Δt , so it is necessary to use a single value to represent the magnitude of the residual error. Take the results of trained SVM models for RSO-1 as an example, the histograms of the original and residual errors of the testing data are shown in Fig. 13. The horizontal axis is the absolute value of the error and the horizontal is the probability of the binned groups. One obvious phenomenon is that the distribution of the residual error has concentrated towards zero, indicating that most of the errors have been compensated for. Another obvious phenomenon is that for e_x , e_y , e_{vx} , e_{vy} , there are a few testing data points with very large values. These data are possibly outliers in the dataset

due to insufficient measurements or wrongly converged estimators during the simulation. They are not eliminated because, differently from in practice, in the simulation environment it is difficult to distinguish from data cleaning to data manipulating, so we have kept all the data. After all, it is nice to observe that the trained ML models are insensitive to these possible outliers.

In Fig. 13, the vertical lines are three statistics to represent the magnitude of the residual errors. The maximum absolute value will be severely affected by the rare but large values. The mean value tends to underestimate the maximum magnitude. The boxplot-max value, corresponding to the maximum bar in a standard boxplot, appears to be an appropriate choice to represent the magnitude of the residual errors.

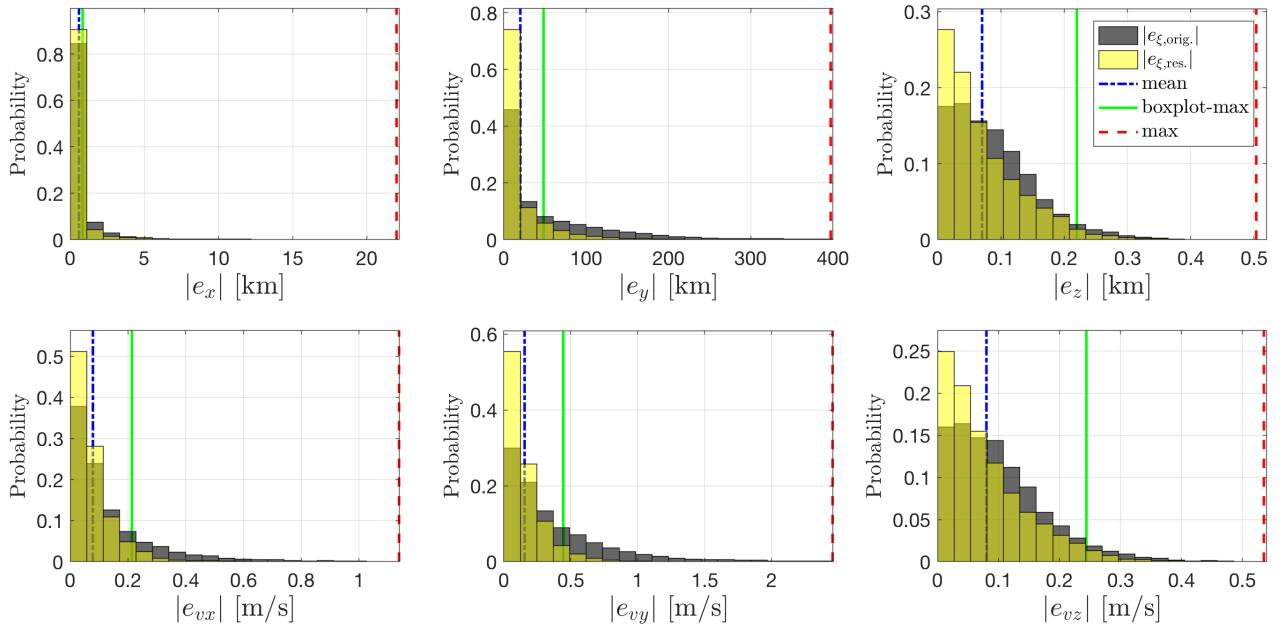


Fig. 13 Histograms of the original and residual errors of the SVM models on the testing data of RSO-1. The verticle lines are statistics to represent the magnitude of the residual errors.

Using the boxplot-max of the absolute residual errors, Table 5 shows the variation of the magnitudes of the residual error of RSO-1 to 4. Each four columns of one component shows the magnitudes of the original training data and the residual errors of SVM, ANN, and GP. It is not surprising that the y -axis components e_y and e_{vy} have the largest magnitudes. For all the RSO, the magnitudes of the residual errors of each components increase with that of the multiple of σ_{measure} for all the four columns, i.e., the original, SVM-residual, ANN-residual, and GP-residual errors. Another observation is that the e_z has the smallest magnitude among position components for all the RSOs, so even if $P_{\text{ML}}(e_z)$ is greater than 100%, the overall performance of the orbit prediction accuracy can still be improved significantly. In order to show the tendency more clearly, the results of RSO-1 in Table 5 are depicted in Fig. 14. It is clear that the residual error is increasing with respect to the multiple of the measurement noise, but the magnitude can always be reduced significantly compared to that of the original error. Similar phenomenon has been observed for RSO-5 to 12 although the details vary.

All the three ML algorithms can reduce the magnitudes of the errors even when the measurement error is increased, so it can be concluded that the performance of the ML approach with these algorithms are practically robust to the measurement noise. This feature can provide some flexibility for potential applications when the measurement noise is unknown.

Table 5 Magnitudes of the original error and the residual errors of RSO-1 to 4 using the three ML algorithms, with measurement noise varying from 1 to 5 times of the original noise magnitude.

RSO Multiple of σ_{measure}	Magnitude of e_x [km]				Magnitude of e_y [km]				Magnitude of e_z [km]				Magnitude of e_{vx} [m/s]				Magnitude of e_{vy} [m/s]				Magnitude of e_{vz} [m/s]				
	Orig.	SVM	ANN	GPR	Orig.	SVM	ANN	GPR	Orig.	SVM	ANN	GPR	Orig.	SVM	ANN	GPR	Orig.	SVM	ANN	GPR	Orig.	SVM	ANN	GPR	
1	1	1.21	0.78	0.22	0.27	199.99	46.11	7.49	9.23	0.26	0.22	0.22	0.32	0.39	0.21	0.23	0.21	1.14	0.43	0.37	0.40	0.29	0.24	0.35	0.34
	2	3.61	2.46	0.64	0.47	350.76	89.34	12.97	55.86	0.45	0.31	0.48	0.42	0.73	0.42	0.40	0.36	2.06	0.81	0.64	0.73	0.51	0.37	0.45	0.44
	3	10.85	7.15	2.30	1.47	591.28	145.41	26.98	96.36	0.73	0.44	0.60	0.58	1.27	0.73	0.52	0.63	3.56	1.29	0.85	1.04	0.84	0.54	0.64	0.58
	4	22.65	13.84	3.43	4.93	856.20	189.51	32.63	42.94	1.04	0.58	0.82	0.67	1.76	1.04	0.48	0.92	5.08	1.82	1.11	0.81	1.24	0.73	1.09	0.77
	5	43.15	26.29	5.78	10.65	1179.12	244.05	45.16	37.51	1.42	0.73	0.85	0.80	2.36	1.44	0.54	0.96	6.90	2.34	0.99	0.95	1.64	0.90	1.06	0.92
2	1	0.86	0.70	0.25	0.22	174.22	32.06	7.44	6.59	0.26	0.27	0.23	0.28	0.21	0.19	0.19	0.17	0.83	0.42	0.36	0.36	0.29	0.28	0.27	0.30
	2	2.71	1.98	0.59	0.49	298.28	51.03	11.13	9.93	0.41	0.39	0.42	0.48	0.36	0.29	0.23	0.29	1.40	0.60	0.44	0.42	0.46	0.42	0.48	0.48
	3	6.78	5.39	1.38	1.01	479.90	76.69	19.34	80.14	0.72	0.61	0.63	0.58	0.59	0.45	0.33	0.49	2.26	0.89	0.58	0.54	0.78	0.65	0.85	0.71
	4	13.71	10.72	2.81	2.93	678.37	110.53	23.60	104.61	1.05	0.80	0.83	0.66	0.85	0.61	0.29	0.63	3.26	1.10	0.49	0.58	1.15	0.88	1.14	0.92
	5	24.38	18.79	3.90	4.37	901.51	140.78	43.62	76.26	1.35	1.00	0.95	0.77	1.17	0.81	0.38	0.84	4.38	1.43	0.53	0.63	1.47	1.11	1.34	1.14
3	1	0.62	0.64	0.27	0.27	146.18	32.77	6.19	6.44	0.30	0.27	0.45	0.29	0.22	0.19	0.19	0.20	0.73	0.53	0.35	0.42	0.33	0.28	0.43	0.31
	2	2.18	1.85	0.58	0.50	270.33	66.34	12.36	15.92	0.56	0.38	0.50	0.43	0.38	0.30	0.15	0.19	1.30	0.74	0.52	0.42	0.61	0.40	0.60	0.47
	3	4.75	3.93	1.03	1.33	402.70	92.58	21.44	54.16	0.86	0.47	0.64	0.47	0.58	0.44	0.20	0.24	1.92	1.00	0.58	0.56	0.92	0.53	0.81	0.60
	4	8.53	7.17	2.04	2.54	541.61	126.42	24.84	83.62	1.18	0.61	0.79	0.57	0.78	0.55	0.20	0.30	2.76	1.21	0.51	0.51	1.26	0.65	0.91	0.75
	5	15.12	13.36	3.34	3.38	722.03	168.49	34.25	30.94	1.49	0.72	0.89	0.70	1.07	0.75	0.22	0.28	3.67	1.42	0.56	0.56	1.56	0.80	1.20	0.91
4	1	1.04	0.89	0.38	0.33	178.91	53.06	11.35	9.72	0.28	0.20	0.29	0.22	0.24	0.19	0.10	0.11	0.86	0.42	0.28	0.21	0.29	0.21	0.30	0.25
	2	2.83	2.44	0.60	0.55	305.36	90.05	12.19	52.82	0.46	0.27	0.41	0.33	0.38	0.30	0.10	0.10	1.38	0.63	0.22	0.24	0.49	0.32	0.47	0.36
	3	5.59	5.58	1.48	1.02	442.11	134.39	21.90	76.77	0.67	0.36	0.49	0.43	0.59	0.44	0.09	0.13	2.05	0.95	0.24	0.28	0.71	0.42	0.50	0.48
	4	13.38	9.63	3.90	3.37	677.15	204.44	20.22	108.58	1.01	0.47	0.59	0.50	0.89	0.60	0.12	0.13	3.12	1.21	0.30	0.33	1.06	0.57	0.81	0.56
	5	23.41	17.67	7.71	5.66	884.95	269.88	40.97	28.25	1.31	0.62	0.73	0.56	1.15	0.79	0.14	0.15	4.17	1.64	0.27	0.35	1.37	0.76	0.96	0.65

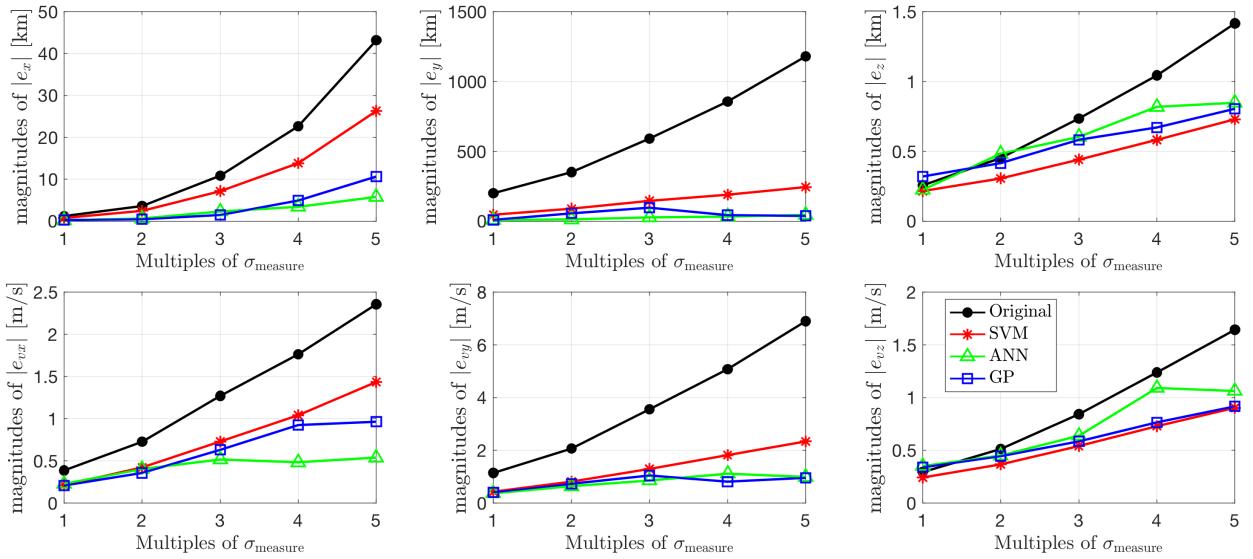


Fig. 14 Variations of the magnitudes of the original and residual errors of the testing data for different components of RSO-1, corresponding to the values in Table 5.

4 Conclusions

In this paper, a comparison study about the previously proposed machine learning (ML) approach to improve orbit prediction accuracy is presented, using 12 RSOs in SSO, LEO, and MEO. Three different ML algorithms used in the previous publications are analyzed, which are SVM, ANN, and GP. The emphasis is to study several aspects related with practical operations. In general, among three algorithms, ANN has the most powerful regression capability due to its more flexible structure; SVM is least likely to overfit the training data; and GP has moderate performance. When dealing with very large measurement noise, they all show stable performance, so the ML approach can be applied without knowing the accurate magnitude of noise in the dataset.

Based on the results in this paper, when it is confident that the designed learning and target variables are related, more powerful and flexible ML algorithms such as ANN or GP shall be adopted. Otherwise, the algorithm more robust to noise and random initializations should be used, such as SVM. Moreover, the ML approach is essentially a data-driven modeling technique, so the good results obtained on specific RSOs should be carefully generalized to other RSOs. For example, with the same design, the performance pattern is different for RSOs in SSO, LEO, and MEO in this paper.

At last, we point out that for the proposed supervised ML approach which are driven by data, it is necessary to build a higher-level outer loop to continuously monitor and adjust the trained ML models in practice. This should be the direction of future studies.

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. Large amount of simulations of RSOs have been supported by the HPC cluster in School of Engineering, Rutgers University.

References

- [1] Kelso T. Iridium 33/Cosmos 2251 Collision. <http://celestak.com/events/collision/>, 2012, accessed on 2017-02-15.
- [2] Abu-Mostafa YS, Magdon-Ismail M, Lin HT. *Learning from Data*, AMLBook2012.
- [3] Hartikainen J, Seppänen M, Särkkä S. 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, Scotland2012.
- [4] Ampatzis C, Izzo D. Machine Learning Techniques for Approximation of Objective Functions in Trajectory Optimisation. In *The IJCAI-09 Workshop on Artificial Intelligence in Space*, volume 2009, Pasadena, California2009, 1–6.
- [5] Sharma S, Cutler JW. Robust Orbit Determination and Classification: A Learning Theoretic Approach. IPN Progress Report 42-203, Jet Propulsion Laboratory, 2015.
- [6] Barton KE, McLaughlin CA. Long Short-Term Memory Neural Networks for the Prediction of Localized Atmospheric Density for Orbit Determination. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–20.
- [7] Ertl CA, Christian JA. Identification of Partially Resolved Objects in Space Imagery with Neural Networks. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–8.
- [8] Parrish NL, Scheeres DJ. Optimal Low-Thrust Trajectory Correction with Neural Networks. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–20.
- [9] Smet SD, Scheeres DJ, Parker JS. Systematic Exploration of Solar Gravity Driven Orbital Transfers in the Martian System Using Artificial Neural Networks. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–20.
- [10] Jia S, Shan J. Neural Network-Based Adaptive Sliding Mode Control for Gyroelastic Body. *IEEE Transactions on Aerospace and Electronic Systems*, 2018: 1–1, .
- [11] Xia X, Jia Y, Xu S, Wang X. An Adaptive Nonsingular Terminal Sliding Mode Tracking Control Using Neural Networks for Space Manipulators Actuated by Cmgs. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–14.
- [12] Peng H, Bai X. Improving Orbit Prediction Accuracy Through Supervised Machine Learning. *Advances in Space Research*, 2018, 61(10): 2628–2646, .
- [13] Peng H, Bai X. Exploring Capability of Support Vector Machine for Improving Satellite Orbit Prediction Accuracy. *Journal of Aerospace Information Systems*, 2018, 15(6): 366–381, .
- [14] Peng H, Bai X. Artificial Neural Network-Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *Journal of Spacecraft and Rockets*, 2018, 55(5): 1248–1260, .
- [15] Peng H, Bai X. Obtain Confidence Interval for the Machine Learning Approach to Improve Orbit Prediction Accuracy. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–17.
- [16] Peng H, Bai X. A Machine Learning Approach to Improve Satellite Orbit Prediction Accuracy: Validation Using Publicly Available Data. In *John L. Junkins Dynamical Systems Symposium*, College Station, TX2018.
- [17] Maisonobe L, Pommier V, Parraud P. Orekit: An Open Source Library for Operational Flight Dynamics Applications. In *4th International Conference on Astrodynamics Tools and Techniques*, 2010.
- [18] Vapnik VN. *The Nature of Statistical Learning Theory*, New York, NY: Springer New York2000, .
- [19] Abu-Mostafa Y, Magdon-Ismail M, Lin H. E-Chapter 8: Support Vector Machines. In *Learning from Data*, 2015.
- [20] Steinwart I, Christmann A. *Support Vector Machines*. Information Science and Statistics, New York, NY: Springer New York2008, .
- [21] Abu-Mostafa Y, Magdon-Ismail M, Lin H. E-Chapter 7: Neural Networks. In *Learning from Data*, 2015.

Astroynamics

<https://mc03.manuscriptcentral.com/astrod>

- [22] Nielsen M. Neural Networks and Deep Learning. <http://neuralnetworksanddeeplearning.com>, 2015.
- [23] Du KL, Swamy MNS. *Neural Networks and Statistical Learning*, London: Springer London2014, .
- [24] Murphy KP. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series, Cambridge, MA: MIT Press2012.
- [25] Almosallam IA, Jarvis MJ, Roberts SJ. GPz: Non-Stationary Sparse Gaussian Processes for Heteroscedastic Uncertainty Estimation in Photometric Redshifts. *Monthly Notices of the Royal Astronomical Society*, 2016, 462(1): 726–739, .
- [26] Almosallam I. Heteroscedastic Gaussian Processes for Uncertain and Incomplete Data. Phd, University of Oxford, 2017.
- [27] Quiñonero-Candela J, Rasmussen CE. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 2005, 6(Dec): 1939–1959.
- [28] Bishop CM. *Pattern Recognition and Machine Learning*. Information Science and Statistics, New York: Springer2006.
- [29] Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning, Cambridge, Mass: MIT Press2006, oCLC: ocm61285753.
- [30] Peng H, Bai X. Generalization Capability of Machine Learning Approach Among Different Satellites: Validated Using TLE Data. In *2018 AAS/AIAA Astrodynamics Specialist Conference*, Snowbird, UT2018, 1–17.

Author biography



Hao Peng has been a Postdoctoral Associate at Rutgers University since August 2016. He received his PhD degree of engineering in 2016 from Beihang University. His current study focus on introducing machine learning, data mining, and artificial intelligence techniques into the aerospace field to solve previously seemingly impossible tasks. His research interests also include dynamical system theory, restricted three-body problem, space trajectory design, orbit control, and optimal control problems. His updated academic activities are available at [Google Scholar](#) as well as [Research Gate](#).



Xiaoli Bai has been an Assistant Professor in the department of Mechanical and Aerospace Engineering at Rutgers since July 2014. She obtained her PhD degree of Aerospace Engineering in 2010 from Texas A&M University. One consequence of her dissertation is a set of methods which significantly enhances and accelerates the fundamental processes underlying the creation and maintenance of space debris catalogs. Her current research interests include astrodynamics and Space Situational Awareness with a focus on the unstable and inactive space debris that are out of control and have uncertain origins; spacecraft guidance, control, and space robotics; and Unmanned Aerial Vehicle navigation and control. Dr. Bai was a recipient of Outstanding Young Aerospace Engineer Award from Texas A&M University in 2018, A. Water Tyson Assistant Professor Award from Rutgers in 2018, The 2016 Air Force Office of Scientific Research Young Investigator Research Program award, the American Institute of Aeronautics and Astronautics Foundation John Leland Atwood Graduate Award, and Amelia Earhart Fellowship.