Research in Astronomy and Astrophysics, 23:085014 (18pp), 2023 August
https://doi.org/10.1088/1674-4527/acd7be

A Data-driven Method for Realistic Covariance Prediction of Space Object with Sparse Tracking Data

Hong-Kang Liu[1], Bin Li[1,2], Yan Zhang[3], and Ji-Zhang Sang[1,2]
[1] School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; bli@sgg.whu.edu.cn
[2] Hubei Luojia Laboratory, Wuhan 430079, China
[3] Xi'an Satellite Control Center, Xi'an 710043, China

Received 2022 December 19; revised 2023 April 28; accepted 2023 May 19; published 2023 July 11

Abstract

Covariance of the orbital state of a resident space object (RSO) is a necessary requirement for various space situational awareness tasks, like the space collision warning. It describes an accuracy envelope of the RSO's location. However, in current space surveillance, the tracking data of an individual RSO is often found insufficiently accurate and sparsely distributed, making the predicted covariance (PC) derived from the tracking data and classical orbit dynamic system usually unrealistic in describing the error characterization of orbit predictions. Given the fact that the tracking data of an RSO from a single station or a fixed network share a similar temporal and spatial distribution, the evolution of PC could share a hidden relationship with that data distribution. This study proposes a novel method to generate accurate PC by combining the classical covariance propagation method and the data-driven approach. Two popular machine learning algorithms are applied to model the inconsistency between the orbit prediction error and the PC from historical observations, and then this inconsistency model is used for the future PC. Experimental results with the Swarm constellation satellites demonstrate that the trained Random Forest models can capture more than 95% of the underlying inconsistency in a tracking scenario of sparse observations. More importantly, the trained models show great generalization capability in correcting the PC of future epochs and other RSOs with similar orbit characteristics and observation conditions. Besides, a deep analysis of generalization performance is carried out to describe the temporal and spatial similarities of two data sets, in which the Jaccard similarity is used. It demonstrates that the higher the Jaccard similarity is, the better the generalization performance will be, which may be used as a guide to whether to apply the trained models of a satellite to other satellites. Further, the generalization performance is also evaluated by the classical Cramer von Misses test, which also shows that trained models have encouraging generalization performance.

Key words: celestial mechanics – methods: analytical – miscellaneous

1. Introduction

Space Situational Awareness (SSA) encompasses intelligence and surveillance of all space objects, prediction of space events, possible collisions, threats, and activities (Bobrinsky & Del 2010). SSA-related tasks include collision probability computation (Xu & Xiong 2014), sensor tasking and scheduling (Jiang et al. 2017), uncorrelated track (UCT) association (Liu et al. 2021), catalog maintenance, and anomaly detection, etc. Common among these applications is that a proper uncertainty characterization of the orbit state of an individual resident space object (RSO), mainly referring to the orbital covariance (Poore et al. 2016), is required. Unrealistic covariance could lead to failure in predicting collisions, false correlation between UTCs and RSOs, inefficient use of sensor resources, and undetected maneuvers. The notion of covariance realism is not without precedent, but a clear definition has been often out of reach (Vallado & Seago 2009). In other tracking domains, the term of covariance consistency is used in place of the term covariance realism (Drummond et al. 2007), which describes the proper characterization of the covariance of the state of a system. Covariance realism requires the estimate of the mean to be the true mean (namely, the estimate is unbiased) and the covariance to possess the right size, shape, and orientation (namely, consistency). Most often, covariance realism is synonymous with the covariance accuracy, gauged by comparing the predicted covariance (PC) with the truth orbit prediction error (PE) (Vallado & Seago 2009).

In the orbit dynamics field, an estimate of the orbit state and its covariance matrix is obtained from an orbit determination (OD) process with available measurements, and then the state and covariance are propagated forward, namely orbit prediction (OP), in which the PC describes a certain accuracy envelope of the predicted orbit state. In this process, the orbital dynamics is usually assumed to be deterministic, and the measurement errors are of Gaussian distribution and regarded as the only uncertainty source. Thus, the estimated covariance matrix only accounts for the measurement noises, known as the noise-only covariance (Montenbruck & Gill 2000), also called formal state covariance (Lopez et al. 2021). In this way, other sources of uncertainty, such as force model errors, are typically not accounted for during the OD and subsequent OP. This usually leads to an overly-optimistic or conservative covariance matrix, eventually, causing the accuracy loss of covariance realism.

Techniques of achieving better covariance realism continue to be increasingly needed by the Space Surveillance and Tracking (SST) and Space Traffic Management (STM) operators (Montenbruck & Gill 2000). A number of statistical techniques have been developed by researchers with various degrees of success to produce relatively realistic covariance. The mechanisms used by space communities differ in their realizations, and can be sorted into two main types: (1) scaling methods, which inflate the covariance by certain factors with the purpose of increasing the volume of a Gaussian ellipsoid based on statistical metrics; some researchers propose to compute such scale by increasing the initial position uncertainty to match the velocity error, whereas another option is to employ the Mahalanobis distance of the orbital differences to find the scale factor; (2) State Noise Compensation (SNC) method, which uses a process noise matrix to consider the uncertainties in the dynamics system or measurements. A well-known example is the realistic covariance predictions of the Earth Science Constellation satellites: Aqua, Aura, and Terra (Junkins et al. 1996; Duncan & Long 2006; Pawloski et al. 2018). It is necessary to clarify that the Consider Analysis method is another example of the SNC technique which accounts for modeling uncertainties in the covariance estimation and prediction. Besides, there exist other techniques conceived to improve covariance realism instead of modeling the sources of uncertainty. For instance, state and covariance representation in mean orbital elements (Junkins et al. 1996) and in the nonlinear reference frames (Aristoff et al. 2021) are widely studied to prolong the realism maintenance upon propagation. These methods have been widely used and achieved great improvements in covariance realism. However, they have limitations of various degrees from a practical point of view. The scaling method lacks physical insight and only relies on a statistical analysis of empirical data to provide covariance corrections. It is difficult to be generalized or extrapolated since the scaling factors are not linked to any physical process. The SNC method requires a significant amount of tuning, and this tuning is prone to failure. Moreover, the type of dynamics in the SNC method appears limited by the pre-assumed Gauss-Markov functional form. Appending dynamics parameters to the state can be an effective method for recovering dynamics, but it requires a known noise pattern for those unmodeled dynamics.

With the employment of more accurate sensors and the growing population of RSOs (Zhang et al. 2022), the potential number of newly discovered objects is likely to increase by an order of magnitude within the next decade, thereby placing an ever-increasing burden on current operational systems. The above limitations have made existing techniques ill-suited for future needs. Moving forward, the implementation of new, innovative, robust, and intelligent methods for covariance realism is required to enable the development and maintenance of the present and future space catalog and to support the overall SSA mission.

Machine learning (ML) techniques have been developed significantly in recent years and promise viable approaches within the space domain's solution space to augment, supplement, and potentially replace traditional methods with encouraging performance. Different algorithms and tools have been developed and utilized in the SSA domain. Liu and Schreiber reported the application of a deep neural network for an automatic classification scheme for space objects based on a single track of a photometric light curve (Liu & Schreiber 2021). Mereta et al. showed that in the case of multi-revolution transfers between near-earth objects (NEOs) the ML approach is vastly superior to the commonly used impulsive Lambert estimate (Mereta et al. 2017). Shen et al. developed both manifold and deep learning methods to determine the space object behavior pattern classification (Shen et al. 2019). Curzi et al. proposed a novel approach to estimate TLE prediction errors using recurrent neural network (Curzi et al. 2022). Moreover, using ML modeling techniques to reduce OP errors shows promising results. Peng and Bai studied the support vector machine, Gaussian Processes, and artificial neural network to improve the OP accuracy (Peng & Bai 2018, 2019, 2021). Li et al. achieved better than 70% OP accuracy improvement by applying the ML-predicted OP errors to correct the physics-based OPs, and the 7 days OP errors were reduced from hundreds or even thousands of meters to only tens of meters through the error correction obtained from the learned error models (Li et al. 2020, 2021). These applications clearly show the excellent capability of ML in SSA, which prompts authors to apply ML methods to achieve better covariance realism.

In order to achieve better covariance realism, various approaches have been attempted, including new propagation methods, reference frames representing covariance, and calibration methods. In this paper, we aim to improve the realism of the propagated covariance by the calibration approach, a novel method is proposed to produce more realistic covariance through the combination of the classical error propagation model and the data-driven model. The evolution pattern of the inconsistency between the error of the propagated orbit (PE) and the PC is learned by the ML method from historical data and is then used to produce realistic covariance for future epochs via the error compensation. A complete ML framework for the improved covariance realism is implemented in three steps, including constructing the data set of the inconsistency between PC and measured PE, training error models with ML methods, and correcting the future PC using

ML-predicted inconsistency. In the developed ML framework, the trained inconsistency model can be used as a corrector to the physics-based orbit error prorogation system. Thus, accurate covariance can be obtained over a long duration if the trained ML model captures the majority of the inconsistency.

In what follows, the generation of state and its associated initial covariance, as well as their predictions, are described in Section 2. The implementation of the ML method to improve the covariance is given in Section 3. In Section 4, experiments with sparse satellite laser ranging (SLR) data of the low-Earth-orbit (LEO) Swarm satellites are conducted to show the performance of the proposed method. In the final section, some conclusions are drawn.

2. Statistical Characterization of Orbit Prediction Errors

In the real orbital data process, the initial orbit covariance (IOC) is generated as a part of an OD solution and then propagated forward with the linear or nonlinear models. Ideally, the PC provides appropriate information of the PE over the prediction time span. In this section, the fundamental computations of the PC and PE are presented.

2.1. PC Generation

An accurate orbit state is determined from an orbit estimation process, and the batch least-square (BLS) is typically used to provide a best-fit orbit matching the available observations (Montenbruck & Gill 2000). IOC is also obtained in the BLS OD system, providing a measure of the variability and correlation of the estimated orbit state parameters. A mathematical description of the IOC matrix can be found in Ref. (Vallado & Seago 2009). The IOC at the initial epoch t_0, denoted as $\boldsymbol{P}(t_0)$, is expressed as

$$\boldsymbol{P}(t_0) = \hat{\boldsymbol{\sigma}}_0^2(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})^{-1} \tag{1}$$

where $\boldsymbol{A}$ is the partial-derivative matrix (partial derivatives of the measurements with respect to the estimated parameters); $\boldsymbol{W}$ is the weight matrix of the measurements, and $\hat{\boldsymbol{\sigma}}_0^2$ is the posterior unit variance.

However, Equation (1) disregards the inherent uncertainty of dynamic and measurement models. In this sense, $\boldsymbol{P}(t_0)$ is commonly found to underestimate the true characterization of the estimated orbit state uncertainty. Evidence of this phenomenon is abundant in literature (Junkins et al. 1996; Duncan & Long 2006; Lopez et al. 2021). For the sake of simplicity and efficiency, linear dynamics and linear propagation of covariance are assumed. Linear covariance propagation is an analytical method that is relatively straightforward, in which the IOC is propagated from t_0 to the future epoch t using the state transition matrix $\boldsymbol{\Phi}(t, t_0)$ as follows

$$\boldsymbol{P}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{P}(t_0)\boldsymbol{\Phi}(t, t_0)^T, \tag{2}$$

where $\boldsymbol{P}(t)$ is the propagated covariance matrix at t.

The diagonal elements of $\boldsymbol{P}(t)$ yield the variances of the position and velocity components while the off-diagonal terms are a measure of the correlation between errors of individual components. The final size of $\boldsymbol{P}(t_0)$ is a 6×6 matrix, which means that the uncertainties in the dynamical parameters such as drag or solar radiation pressure coefficients are not included, such that the unmodelled dynamical parameter uncertainty will contribute to the inconsistency of the PC. To better understand the error characteristics, the covariance in the Cartesian coordinate system is usually transformed into the RSW coordinate system, where *R, S*, and *W* represent the radial, in-track (in-plane normal to radial), and cross-track directions, respectively (Vallado 2007). In the following, $\boldsymbol{\sigma}^P$ is used to denote the square root of the variance of a position or velocity component in the RSW coordinate system. Considering that a principal application of the covariance is in the conjunction assessment (CA) of space objects, where only the position portion of the covariance is used, it is reasonable to focus only on the position portion of the covariance. So, in our study, only the position covariance, denoted as (σ_R^P, σ_S^P, σ_W^P), is concerned, and the velocity covariance will not be discussed in the following.

2.2. PE Generation

PE, the difference between the OPs and their "truth" or precise ephemerides, is a measure to assess the accuracy of state predictions. The position differences are calculated and then expressed in the RSW coordinate frame for the orbit error analysis. The root mean square (RMS) value of PEs for each position component is computed in a preset interval (e.g., an orbital period), and used as a measure of the PE magnitude, representing the 1σ error. This process is basically the same as that in Ref. (Duncan & Long 2006). In the following, the computed positional sigma errors, denoted as (σ_R^C, σ_S^C, σ_W^C), are regarded as the realistic magnitude of the state uncertainty in the OD and OP process.

In theory, if the PC is accurate, it should be consistent with the computed $\boldsymbol{\sigma}^C$, that is, the magnitudes of $\boldsymbol{\sigma}^C$ and $\boldsymbol{\sigma}^P$ should be close to each other to some extent. Figure 1 shows an example of inconsistency between $\boldsymbol{\sigma}^C$ and $\boldsymbol{\sigma}^P$. So, the problem turns out to bridge the gap between $\boldsymbol{\sigma}^P$ and $\boldsymbol{\sigma}^C$. Hence, the inconsistency or the gap is computed as

$$e_{\text{dir}}^{\text{gap}} = \sigma_{\text{dir}}^C - \sigma_{\text{dir}}^P \tag{3}$$

where $e_{\text{dir}}^{\text{gap}}$ is the inconsistency between σ_{dir}^P and σ_{dir}^C, dir $\in R$, *S*, *W*.

Because $\boldsymbol{\sigma}^C$ is computed from the differences between the propagated orbit and the "truth" orbit, it is regarded as reliable. In this sense, the inconsistency is mostly due to $\boldsymbol{\sigma}^P$. The key idea of this study is to model the inconsistency and then apply the modeled inconsistency pattern to correct PC, such that the

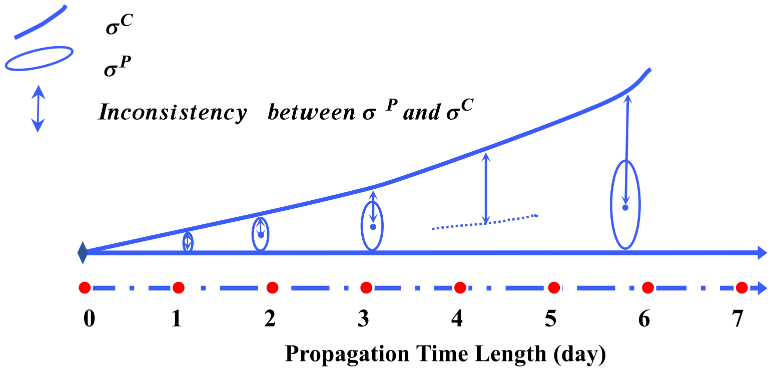

σ^C
σ^P
Inconsistency between σ^P and σ^C
0
1
2
3
4
5
6
7
Propagation Time Length (day)

Figure 1. Schematic illustration of the inconsistency between PC and PE.

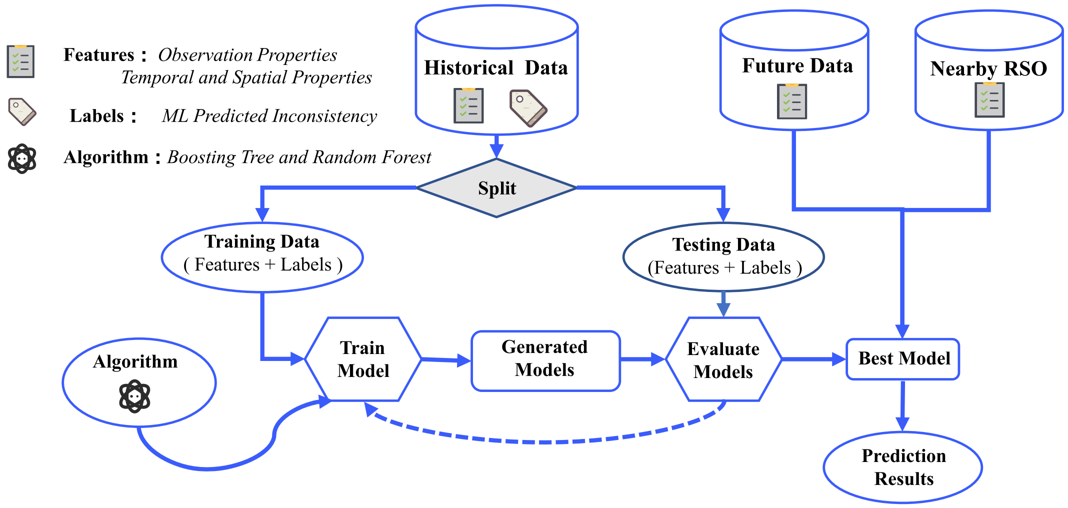


Figure 2. ML framework for improved covariance realism.

inconsistency between the corrected σ^P and σ^C is significantly reduced.

3. Development of ML Framework

In our previous studies (Li et al. 2020, 2021), an ML framework to improve OP accuracy has been developed. This study expands the ML framework to include the capability of modeling the inconsistency, defined in Equation (3). The modified ML framework is illustrated in Figure 2.

3.1. Data Source

Given that optical and radar measurement data is difficult to obtain, and emulated measurements may not fully simulate real observation scenes, so, in this study, we consider the single-station SLR tracking scenario. The tracking station is Changchun SLR station (CHAL), China, with its geodetic latitude of 43°79, the longitude of 125°44, and altitude of 274.90 m. In the single-station SLR tracking scenario, the collected consolidated ranging data (CRD) tracklets are sparse, mostly less than three passes daily for an SLR satellite. The CRD is available at the International Laser Ranging Service (ILRS) data center (ftp://cddis.gsfc.nasa.gov/). Also, the precise Consolidated Prediction Format (CPF) orbits of the SLR satellites are downloaded and used as the reference to compute the PE.

Training data samples are generated based on a consistency analysis of orbital differences between the reference orbits and predicted orbits, which are from independent ODs using the SLR data. The graphical representation of the orbital differences is depicted in Figure 3. Furthermore, the orbital differences are time-tagged, forming the population of the PEs and associated PCs with respect to the prediction time length.

Sufficient statistical samples for the ML would be acquired by considering a sufficient number of independent orbits. To compute the orbital differences between the reference and predicted orbits, two additional steps are performed:

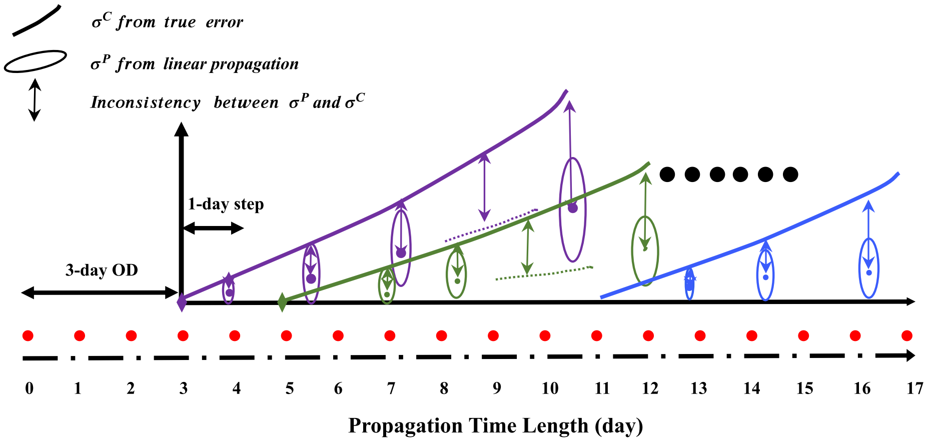

σ^C from true error
σ^P from linear propagation
Inconsistency between σ^P and σ^C
1-day step
3-day OD
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Propagation Time Length (day)

Figure 3. OD sliding windows for generating PE and PC.

1. Requirement on the number of data points per track, for a track to be used in the OD, it must have at least three data points.
2. Requirement on the number of tracks within 3 days, for an OD process to be successfully completed, there are at least 2 tracks available over 3 days (the OD span used in this study).

The sliding-window manner favored by NASA (Pawloski et al. 2018; Duncan & Long 2006) has been implemented and is also demonstrated effective in the work (Li et al. 2020; Lopez et al. 2021). In this study, the sliding step is set as 1 day.

For clarity, we briefly introduce the OD strategy and settings used in this study. The equations of motion of an Earth-orbiting satellite in the Earth-centered inertial (ECI) coordinate system is given by a system of three second-order differential equations of

$$\ddot{\boldsymbol{r}} = -\mu\frac{\boldsymbol{r}}{r^3} + a_{\text{per}} \tag{4}$$

where $\boldsymbol{r}$ and $\ddot{\boldsymbol{r}}$, are the satellite's position and acceleration vectors in the ECI frame, respectively; $\mu = GM$ is the Earth gravitational constant; $\boldsymbol{a}_{\text{per}}$ is the perturbation acceleration vector.

In our experiments, the main perturbations considered are listed in Table 1. An orbit determination and analysis software for processing the ranging observations is developed by Li et al. (Li et al. 2020). It applies the Cowell integrator to propagate satellite position and velocity vectors in Equation (4), and the BLS algorithm for the optimal estimations. In the BLS OD process, three days' laser tracking arcs of a satellite are used to compute the optimal estimation of the state. Commonly, model parameters, such as C_d and C_r, may be estimated. However, given the sparsity of tracking arcs, only the position and velocity of the satellite are estimated. The OD fit span is set starting at the epoch time of the first tracking arc. The initial orbit state is computed from the latest TLE before the OD fit span. Further details about the orbit estimation strategy can also be found in Ref. (Li et al. 2020).

Table 1
OD Setting in Force Modeling

Parameters	Modeling
Earth Geopotential Model	30 × 30 Joint Gravity Model (JGM) -3
Non-Central Bodies	Sun, Moon (JPL DE200 Ephemeris File)
Atmospheric Density Model	MSIS86
Ocean Tide Model	CSR
drag coefficient C_d	2.2
solar radiation pressure coefficient C_r	1.1
Numerical Integrator	Cowell Integrator with 60 s step size

3.2. Learning Features

ML method works on sample data (training and testing data) to learn the hidden pattern. For a given problem, the success of the learning relies strongly on data preparation. Generally, the sample data consists of the input $\mathbf{Z}$, usually a vector, and the output Y. The objective of the learning is to capture the mapping $Y = f(\mathbf{Z})$. Therefore, it is imperative that the sample data should be well-prepared. Determining the target variable Y is always straightforward. In this study, the target variable Y is the inconsistency defined in Equation (3). But the choice of feature variables is subjective, as there are many latent features affecting Y. Fortunately, most ML algorithms are not sensitive to irrelevant feature variables. Theoretically, the more feature variables are included in $\mathbf{Z}$, the more accurate but complex the trained model would be. Therefore, it is usually favorable to take into account the main feature variables that affect Y in ML learning.

According to the orbit dynamics theory, the factors that affect $e_{\mathrm{dir}}^{\mathrm{gap}}$ at a specific epoch t_i should be considered. At some time t_i, the predicted state $\hat{\boldsymbol{x}}_{t_i}$, the PC $\boldsymbol{P}_{t_i}$, the initial orbit state $\hat{\boldsymbol{x}}_{t_0}$, and the IOC $\boldsymbol{P}(t_0)$ are all obtained from the OD and OP process together with the observation information. The reference CPF state $\boldsymbol{x}_{t_i}$ is also available. Based on these key elements, the feature variables included in $\boldsymbol{Z}$ can be determined as:

1. The prediction duration $\Delta t_j = t_j - t_i$. This variable is important, since the longer the prediction time, the larger the inconstancy would be.
2. Initial orbit state $\hat{\boldsymbol{x}}_{t_0}$ for state propagation. $\hat{\boldsymbol{x}}_{t_0}$ in both the form of classical orbital element (COE), $\hat{\boldsymbol{x}}_{t_0}^{\mathrm{COE}}$, and the form of Cartesian coordinates of the ECI position and velocity (six-dimensional), $\hat{\boldsymbol{x}}_{t_0}^{\mathrm{ECI}}$, are used. They contain necessary orbit characteristics of a RSO, such as the orbit altitude, period, shape, and state.
3. The predicted state $\hat{\boldsymbol{x}}_{t_i}$, expressed in the COE form as $\hat{\boldsymbol{x}}_{t_i}^{\mathrm{COE}}$, and the ECI position and velocity form $\hat{\boldsymbol{x}}_{t_i}^{\mathrm{ECI}}$. They are used to compute the sigma errors $\boldsymbol{\sigma}^C$.
4. The properties of measurements used in the ODs denoted as $\mathrm{Obs}_{\mathrm{properties}}$, such as the number of tracks N_{pass}, the number of data points of a track N_{points}, observation time duration of a track N_{duration}, the starting time t_s of each track within OD span, maximum elevation angle el over a track, and the corresponding azimuth angle az. The starting time t_s and the (el, az) pair reflect the temporal and spatial distribution of measurements. The above factors have a direct influence on the accuracy of IOC, thus affecting the growth rate of the inconsistency.
5. The OD accuracy, RMS. This variable reflects the accuracy of OD results.

The target variable Y refers to the inconsistency as:

1. $\boldsymbol{e}^{\mathrm{gap}}$ at time t_i, expressed in the RSW coordinate system as $\boldsymbol{e}^{\mathrm{gap}} = [\mathrm{e}_R^{\mathrm{gap}}, \mathrm{e}_S^{\mathrm{gap}}, \mathrm{e}_W^{\mathrm{gap}}]$, where $\mathrm{e}_R^{\mathrm{gap}}$, $\mathrm{e}_S^{\mathrm{gap}}$, and $\mathrm{e}_W^{\mathrm{gap}}$ denote the radial, in-track, and cross-track position inconstancy, respectively.

Given the feature and target variables defined above, a sample of the training/testing data is formed as ($\boldsymbol{Z}$, Y), where $\boldsymbol{Z} = [\Delta t_j, \hat{\boldsymbol{x}}_{t_0}^{\mathrm{COE}}, \hat{\boldsymbol{x}}_{t_0}^{\mathrm{ECI}}, \hat{\boldsymbol{x}}_{t_i}^{\mathrm{COE}}, \hat{\boldsymbol{x}}_{t_i}^{\mathrm{ECI}}, Obs_{\mathrm{properties}}, \mathrm{RMS}]$, and $Y \in \boldsymbol{e}^{\mathrm{gap}}$.

Based on the above configuration of ($\boldsymbol{Z}$, Y), there exists an implicit mathematical relation that maps $\boldsymbol{Z}$ to Y. With the well-prepared samples generated in a sliding-window manner and the well-designed feature variables, the mapping function, defined as $Y = \hat{f}(\boldsymbol{Z})$, is hopefully found through the ML training, if this mapping exists. Then, the trained ML model is expected to find the relationship between the PC and PE. In particular, if data beyond the learning period, or at a future epoch, are input into the trained model, the model is expected to make good predictions about the relationship between the PC and PE at the epoch. The correction to PC can be performed as below.

The ML-predicted inconsistency, denoted as $\boldsymbol{e}_{\mathrm{ML}}^{\mathrm{gap}}$ is computed by

$$\boldsymbol{e}_{\mathrm{ML}}^{gap} = \hat{f}(\boldsymbol{Z}) \tag{5}$$

where $\boldsymbol{e}_{\mathrm{ML}}^{\mathrm{gap}}$ is the predicted gap corresponding to given data $\boldsymbol{Z}$. Then, the predicted gap is applied to correct the propagated sigma, $\boldsymbol{\sigma}^P$, such that a modified sigma, $\boldsymbol{\sigma}^{\mathrm{ML}}$, is obtained by

$$\boldsymbol{\sigma}^{\mathrm{ML}} = \boldsymbol{\sigma}^P + \boldsymbol{e}_{\mathrm{ML}}^{\mathrm{gap}} \tag{6}$$

where $\boldsymbol{\sigma}^{\mathrm{ML}}$ is the modified sigma that will replace the original $\boldsymbol{\sigma}^{\boldsymbol{P}}$ in applications.

The residual error $\Delta\boldsymbol{e}$ corresponding to $\boldsymbol{\sigma}^{\mathrm{ML}}$ is

$$\Delta\boldsymbol{e} = \boldsymbol{e}^{\mathrm{gap}} - \boldsymbol{e}_{\mathrm{ML}}^{\mathrm{gap}} \tag{7}$$

It is expected that $\Delta\boldsymbol{e}$, the inconsistency between $\boldsymbol{\sigma}^C$ and $\boldsymbol{\sigma}^{\mathrm{ML}}$, should be significantly less than $\boldsymbol{\sigma}^{\mathrm{gap}}$ if the ML models capture the underlying pattern.

The performance of the learned ML model can now be assessed using the following metric

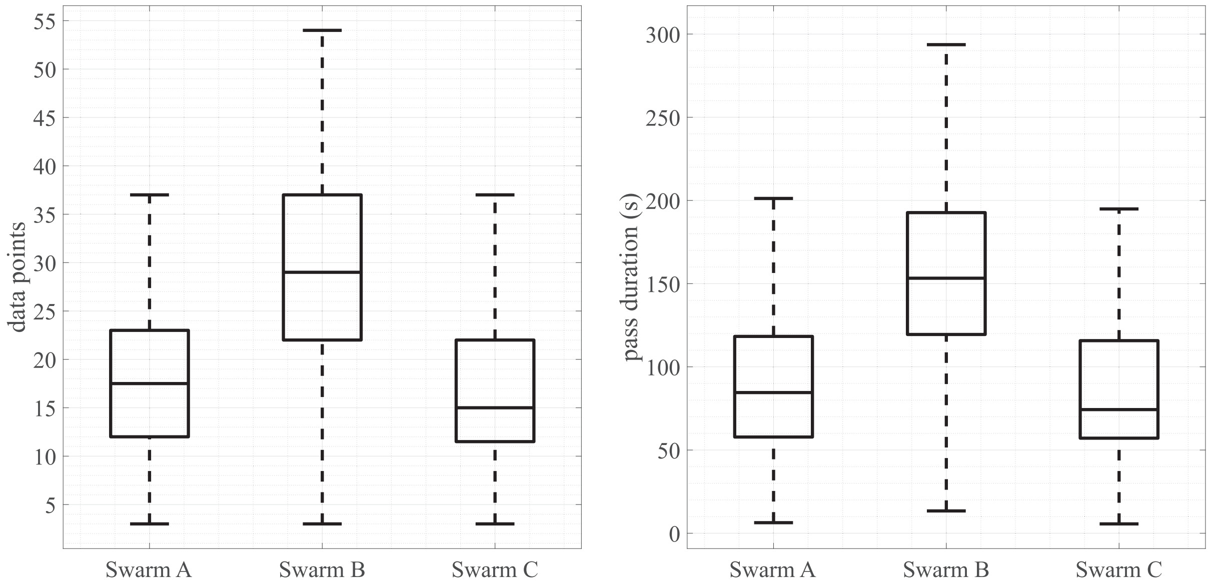
$$P_{\mathrm{ML}} = \frac{\sum_{i=1}^{n} |\Delta\boldsymbol{e}|}{\sum_{i=1}^{n} |\boldsymbol{e}^{\mathrm{gap}}|} \times 100\% \tag{8}$$

where n is the number of training or testing samples.

In the mathematical sense, P_{ML} measures the percentage of the residual error $\Delta\boldsymbol{e}$ with respect to the true inconsistency of the training or testing data. The lower the P_{ML} value is, the more errors are corrected, thus the better the performance of the trained ML models would be. The metric reaches its lower boundary of zero when $\boldsymbol{e}_{\mathrm{ML}}^{\mathrm{gap}}$ is identical to $\boldsymbol{e}^{\mathrm{gap}}$, but has no upper boundary. When the P_{ML} is larger than 100%, it means the models are invalid in compensating the covariance inconsistency.

4. Experimental Results

The developed ML framework is applied to train the orbit covariance inconsistency models of three Swarm satellites. To implement the ML framework, the boosting tree (BT) and random forest (RF) (Friedman et al. 2000) are employed. These two models are both based on the decision tree, have high accuracy with moderate computation complexity, are versatile in case of large data sets having high dimensionality, and have a high degree of interpretability (Molnar 2020). Sharing well reputation in the ML community, these two models of different mechanisms are chosen to validate each other and to demonstrate the effectiveness of the ML framework. More details about BT and RF models are referred to in Ref. (Russell & Norvig 2003). In the experiments, the propagated orbits and PC are obtained from the OD/OP system first. Then, the CPF orbits are used as references to compute the PE. Lastly, the PC is calibrated by the learned ML model such that the



(a) . Data Points distribution of Swarm in CHAL **(b) . Pass duration distribution of Swarm in CHAL**

Figure 4. SLR data distributions of Swarm satellites in January 2021 at CHAL station.

inconsistency between the corrected (σ_R^P, σ_S^P, σ_W^P) and (σ_R^C, σ_S^C, σ_W^C) is reduced.

To demonstrate the learning and generalization capabilities of the ML models, the three satellites in the Swarm constellation are chosen as the test objects. The three identical satellites, named Alpha, Bravo, and Charlie (or A, B, and C), were launched into three near-polar orbits on 2013 November 22. Swarm A and Swarm C form the lower pair of satellites flying side-by-side (1°4 separation in longitude at the equator) at an altitude of 460 km and at the inclination of 87°4, whereas Swarm B is cruising at a higher orbit of 520 km and at 88° inclination angle. Orbit parameters of the Swarm satellites are summarized in Table 2.

4.1. Statistical Characterization of Tracking Data in OD

The pattern of the inconsistency between the PC and PE is to be learned from historical data. The SLR tracking data of the Swarm satellites at CHAL in 2021 January have the distribution properties shown in Figure 4. The left of Figure 4 represents the distribution of the data points of each arc, and the right represents the distribution of the pass durations. The bottom of each box is the first quartile, the top of the box is the third quartile, and the line inside of the box is the median.

From Figure 4, it can be found that Swarm A and Swarm C have similar distributions of data points and track durations owing to their pair-flying attribute, while Swarm B has the best distributions among the three satellites. There are 35, 27, and 26 tracks in 2021 January for Swarm A, Swarm B, and Swarm C, respectively. The average track duration of each satellite is 88.4 s, 157.7 s, and 83.7 s, with their corresponding average number of data points per track being 17, 29, and 16.

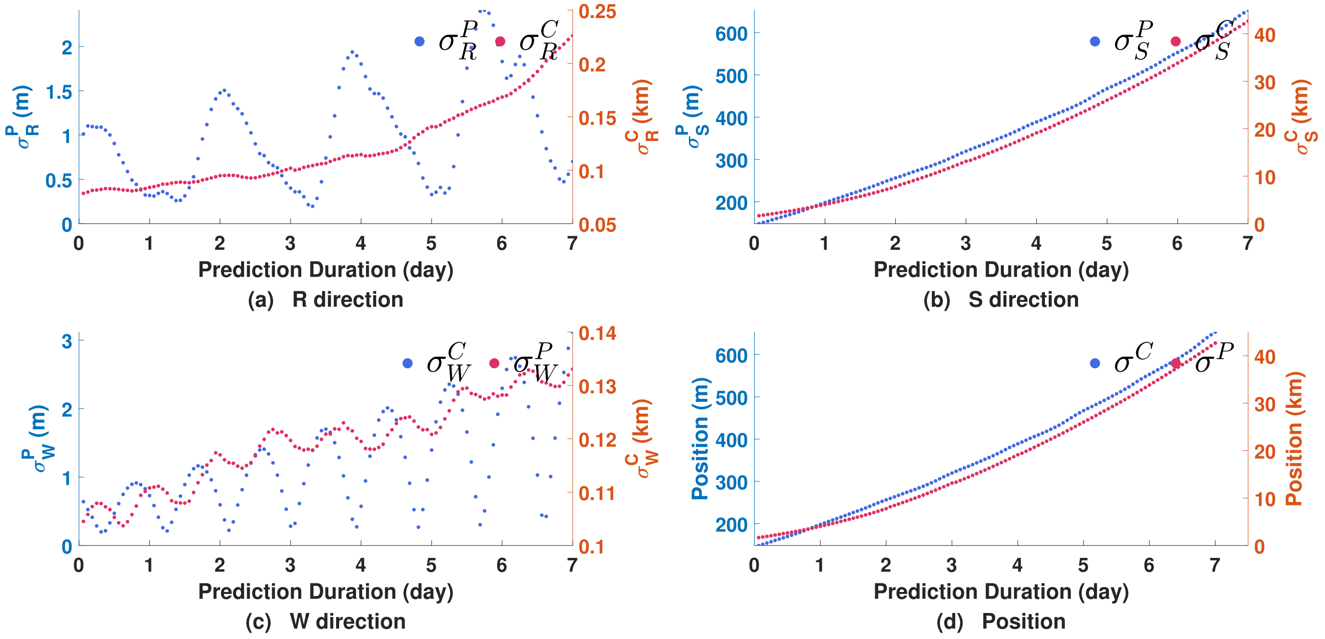
Table 2
Orbit Parameters of Swarm Satellites

Parameters	Swarm A (Alpha)	Swarm C (Charlie)	Swarm B (Bravo)
Altitude	460 km	460 km	520 km
Inclination	87°4	87°4	88°
Period	5618 s	5618 s	5618 s
Eccentricity	0.00139	0.00139	0.00139

With the available data gathered by the CHAL station within 90 days from January to April in 2021, the BLS OD with a 3 days fit span is carried out, and the OD span is sliding forward with a 1 day step. 35/56/43 OD runs are successful for Swarm A, Swarm B, and Swarm C, respectively, and each of these successful OD runs produces the initial orbit state and IOC, which are then propagated forward for 7 days from the end of the OD span, generating the required PC and PE. Here, the 7 days OP is believed sufficient for a timely and reliable SSA application.

4.2. Inconsistency between PC and PE

Examination of the statistical characterization of PC and PE reveals that PC is a very poor representation of PE dispersion. Figure 5 displays an example growing trend of the position components of PC and PE of Swarm A over a 7 days OP time span. It is noted that each PC/PE point is computed once for 1.5 hr.



(a) R direction

(b) S direction

(c) W direction

(d) Position

Figure 5. PC compared with PE in the RSW coordinate system, Swarm A.

Figure 5 clearly shows the inconsistency between the PC (blue dots) and the PE (red dots) when only sparse data is available. That is, (σ_R^P, σ_S^P, σ_W^P) greatly underestimate (σ_R^C, σ_S^C, σ_W^C) during the OP time span. (σ_R^P, σ_W^P) are virtually not growing. While the in-track position error σ_S^P is significantly larger than the other two components with its magnitude reaching nearly 40 km at the end of 7 days, similar results can be noticed in Duncan & Long (2006). Because the in-track position error contributes the most to the position error, great attention is paid to the model training of inconsistency in the in-track position component. It is observed that at the end of the 7 days propagation, the ratio of PC to PE in the in-track direction is about 0.015, the magnitude of this kind of ratio is also found in references (Zaidi & Hejduk 2016), and the level of PC is within the range of what is considered normal for this type of analysis.

It is very intuitive to see that the gaps between (σ_R^P, σ_S^P, σ_W^P) and (σ_R^C, σ_S^C, σ_W^C) increase as the prediction time increases. It is also clear that PC does not provide the proper characterization of the true orbit prediction errors. In fact, they are too conservative or "optimistic" to describe the errors of the predicted orbits.

Figure 6 shows the inconsistency values in the S direction over 7 days OP span of Swarm A from the whole data set in 3 months. Examination of Figure 5 has shown that the in-track position gap is significantly larger than the other two components. In this figure, different growth trends can be observed, and this can be explained by the fact that the dynamic model errors and observation conditions produce different OP accuracy. So, technically speaking, the more diverse observation properties contain, the more inconsistency patterns can be displayed.

4.3. ML Model on the Training and Testing Data

The 3 months time span is divided into a 1 month training period and a 2 months validation period. That is, the data in January are used to train the models. To avoid overfitting, the Cross-Validation (CV) that splits the data set in the training period into training and testing data is applied. Randomly chosen 70% of the whole data set is used to train the model for each of the R/S/W directions while the remaining 30% is used to evaluate the performance of the learned models. As discussed earlier, the feature variables of the testing data are the input to the model, and the output is the ML-predicted covariance inconsistency in each direction.

Figure 7 presents the results of the trained BT (left) and RF (right) models on Swarm A training data (3137 samples). The horizontal axis represents the prediction duration. In the figure, the green dots are the "true" inconsistency $\sigma^{\text{gap\_true}}$ and the red dots are the ML-predicted inconsistency $\sigma^{\text{gap\_pred}}$. It is intuitive to see the RF models perform better. It can be seen that the red dots with the BT model, on the left of the figure, are more dispersed, while the red dots with the RF model, on the right of the figure, are much closer to the "true" green dots, or even overlapped. The performance metric P_{ML} of the BT model is 35.7%, 34.9%, and 20.1% for the radial, in-track, and cross-track inconsistency, respectively. Great improvement can be noticed when the RF model is used, with the P_{ML} being 1.2%, 1.5%, and 0.4% for the radial, in-track, and cross-track

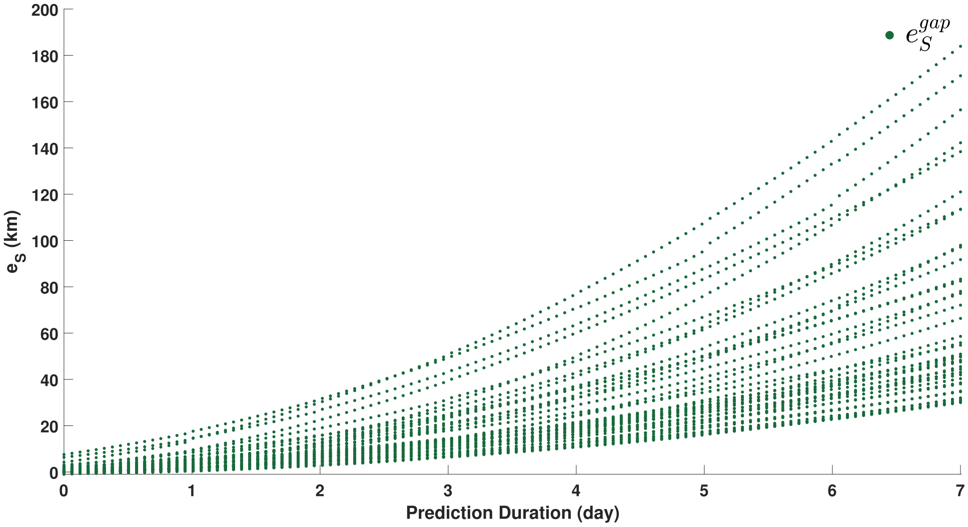


Figure 6. The inconsistency over the 7 days OP span, Swarm A.

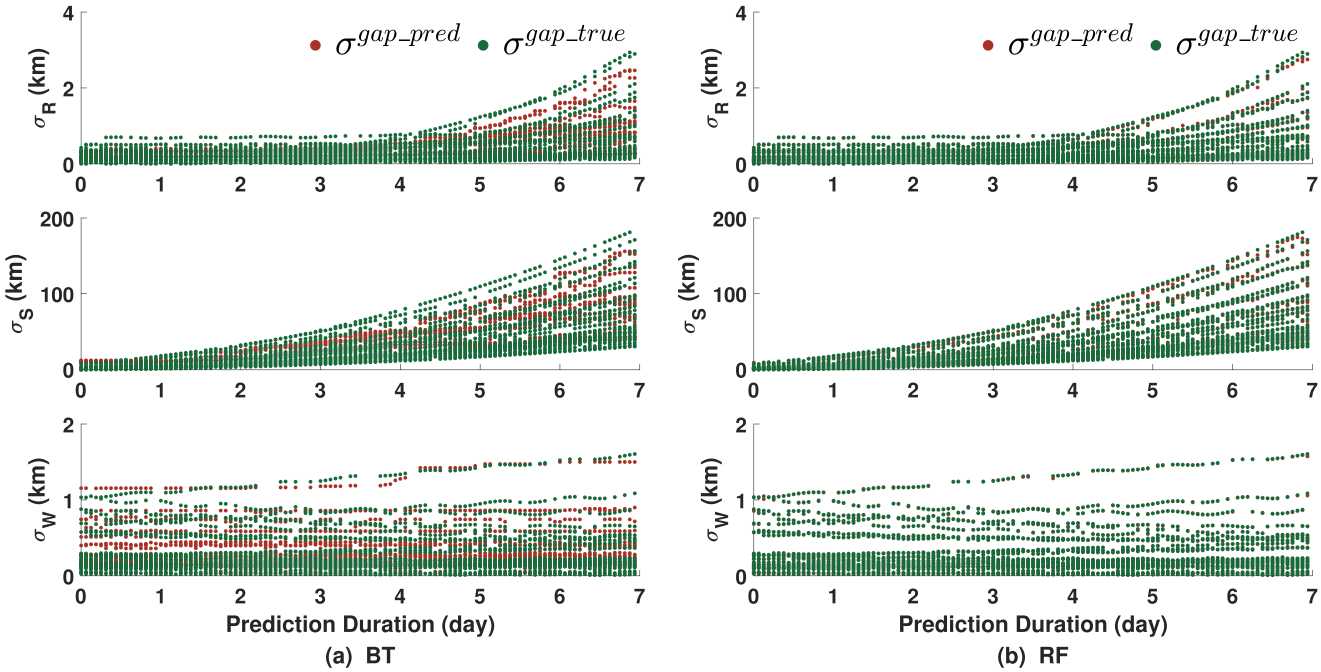
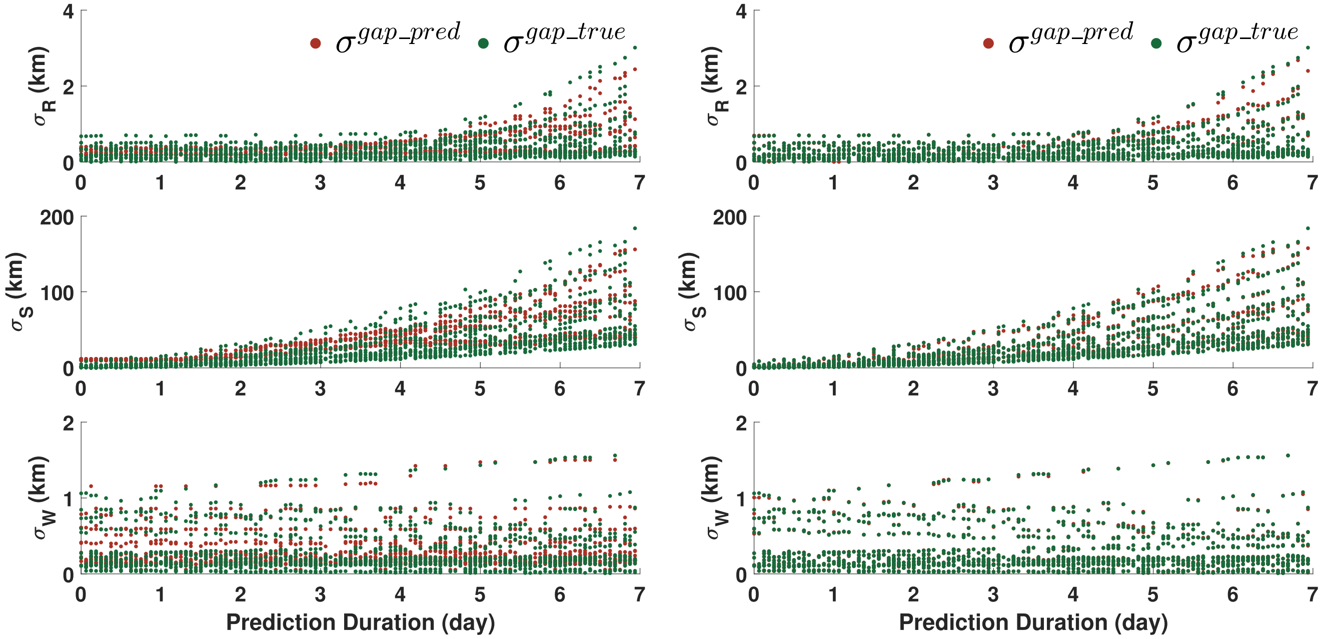


Figure 7. Results of the trained BT (a) and RF (b) models on the training data (3,137 samples), Swarm A.

inconsistency, respectively, indicating that the trained models have captured about 99% of the underlying pattern. Therefore, the conclusion can be made that the trained models capture the inconsistency in the training data set very well. Similar results are obtained for Swarm B and Swarm C.

Figure 8 shows the results of the performance of the trained BT (left) and RF (right) models on the testing data (1344 samples) for Swarm A. These two models show a slight drop when they are used on the testing data. Meanwhile, the RF model outperforms the BT model. It is intuitive to explain that if the trained model performs poorly on the training data, it is difficult to obtain satisfactory predictions on the testing data.

Table 3 gives the P_{ML} values of the trained models on the training data and testing data for all Swarm satellites.



(a) BT

(b) RF

Figure 8. Results of the trained BT (a) and RF (b) models on the testing data (1,344 samples), Swarm A.

Table 3
Performance P_{ML} of the BT and RF Models in Fitting the Training and Testing Data

Objects	BT			RF		
	σ_R	σ_S	σ_W	σ_R	σ_S	σ_W
	train/test	train/test	train/test	train/test	train/test	train/test
Swarm A	35.7%/35.9%	34.9%/36.9%	20.1%/21.3%	1.2%/2.8%	1.5%/3.8%	0.4%/1.0%
Swarm B	39.0%/40.8%	49.2%/49.3%	36.7%/37.4%	0.6%/1.7%	1.2%/3.3%	0.5%/1.4%
Swarm C	45.2%/50.8%	34.0%/36.0%	23.6%/24.7%	0.7%/1.9%	1.6%/4.2%	0.2%/0.6%

In Table 3, it can be seen that all P_{ML} values are below 50% for both the BT and RF models, indicating more than half of the inconsistency errors can be corrected by the trained models. Moreover, the trained RF models are much more powerful than the trained BT models in capturing the inconsistency patterns on the same training and testing data, with the P_{ML} all less than 5% in all three directions for all the satellites. Overall, the developed ML framework is capable of capturing the majority of the underlying inconsistency pattern with well-designed samples. If the learned patterns agree well with the realistic patterns, the trained models can be employed to predict the inconsistency in future epochs.

4.4. Generalization of ML Model to Future Epochs

For practical applications, one is more concerned with the performance of the trained ML models on new data in the future. So, a desirable ML model should have good generalization capability, that is, making good predictions with new input beyond the training period. In this study, after the trained models on the inconsistency are obtained from the use of the historical covariance inconsistency data set, it can be generalized to future epochs to predict new inconsistency at these epochs. Thus, the improved covariances are obtained via the model compensation.

In order to demonstrate the generalization capability of the trained ML models on the data beyond the training period, the data in 2021 January are used to train the ML models, and the data in February and March are used as the validation data for the evaluation of capability in the generalization into future of the trained ML models. The performance metric P_{ML} of the ML models on the validation data is shown in Figures 9–11. Since the in-track error contributes the most to the position error, in this section, only the in-track component is displayed. In each of these figures, the improvement in terms of the percentage of

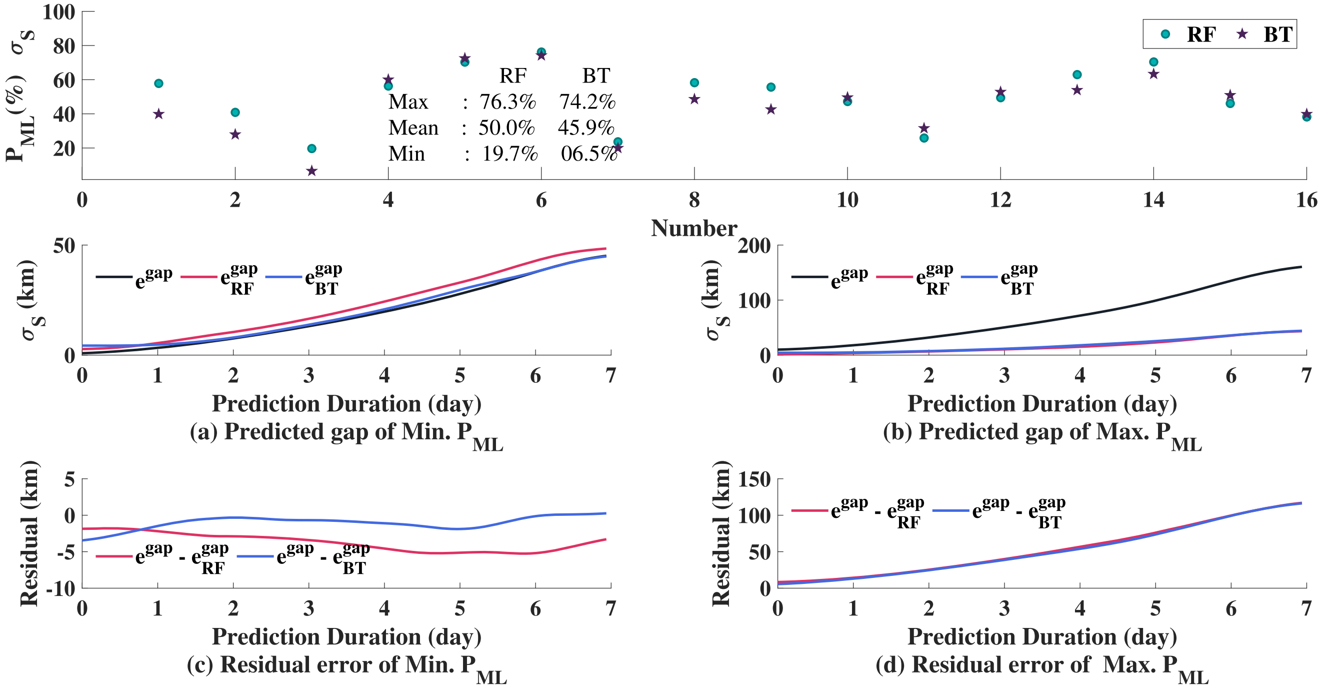


Figure 9. Improved covariance realism beyond the training period, Swarm A.

every single validation case is presented. Besides, panels (a) and (b) in each figure show the predicted gap when the model performs best (lowest in P_{ML} metric value) and performs worst (highest in P_{ML} metric value), with panels (c) and (d) showing the residual errors after the corrections are made to PC.

Observing Figures 9–11, there are 16/30/28 validation cases for Swarm A, Swarm B, and Swarm C, respectively. It reveals that the RF and BT models behave similarly and both show good generalization ability. The means of P_{ML} of the trained BT and RF models on the validation data are all below 50%. The trained models perform best on the swarm B data set, with the mean P_{ML} metrics of RF and BT models at 25.4% and 26.7%, respectively. This means that the ML-predicted inconsistency over the 7 days OP time span is significantly reduced compared with the original one, confirming the models capability of the generalization into the 2 months after the training time. The trained models of Swarm A are poorer in generalization ability, with the mean P_{ML} values at nearly 50%, while the trained models of Swarm C perform moderately. It is also seen that the P_{ML} on the validation data is generally larger than that on the training data. This is owing to the variations of the orbital environment which cause the inconsistency pattern in the validation time to deviate from the pattern in the training time. In each of these figures, the best-performed predictions and the worst-performed predictions have also been shown in panels (a) and (b), respectively, with the corresponding residual errors shown in (c) and (d).

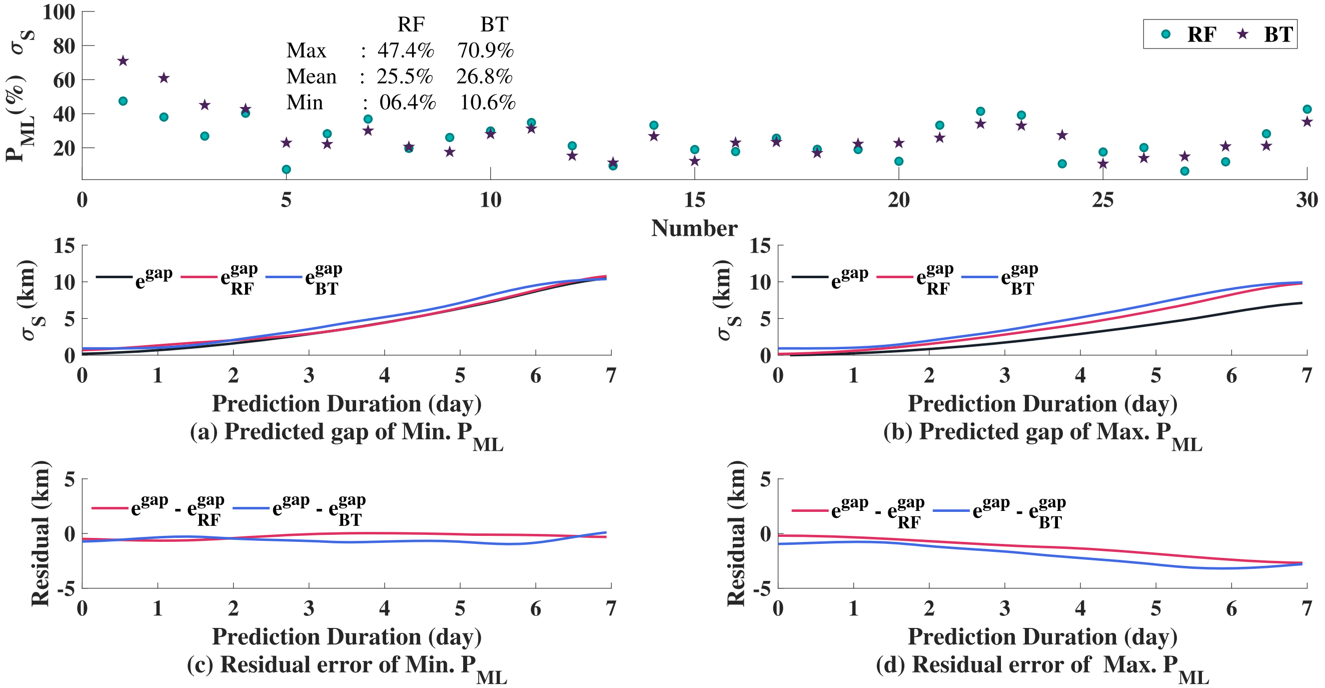
Taking Swarm A in Figure 9 as an example, panel (a) shows the predicted gap when the P_{ML} metric value is 19.7% and 6.5% for RF and BT models, respectively. It can be seen that the $e^{\mathrm{gap}}_{\mathrm{BT}}$ are closely aligned with the e^{gap}, while the $e^{\mathrm{gap}}_{\mathrm{RF}}$ are generally higher than the e^{gap}, but within a well-accepted margin. In panel (b), the predicted gaps, when the P_{ML} metric value is 76.2% and 74.1% for RF and BT models, respectively, are shown. It clearly shows that despite the predictions $e^{\mathrm{gap}}_{\mathrm{BT}}$ and $e^{\mathrm{gap}}_{\mathrm{RF}}$ making the corrected PCs closer to PEs, the improvement is far less impressive. Similar results can be observed in Figures 10 and 11.

In summary, the trained models for each satellite have the remarkable capability of generalization into the future, and a reduction of the inconsistency is achieved for all satellites, achieving a better covariance realism.

4.5. Generalization of ML Model to Other Satellites

In this section, the generalization capability of trained models of a satellite into other satellites is examined, which is highly important for the SSA applications, since there may be no sufficient samples for training a desirable model of other RSOs. So, the question is whether the learned models of a specific RSO can be applied to other RSOs that result in improved covariance realism.

Here, the trained models of Swarm A are applied to Swarm B and Swarm C. The Swarm A models are trained using all data over the 3 months. The validation data is also over the same 3



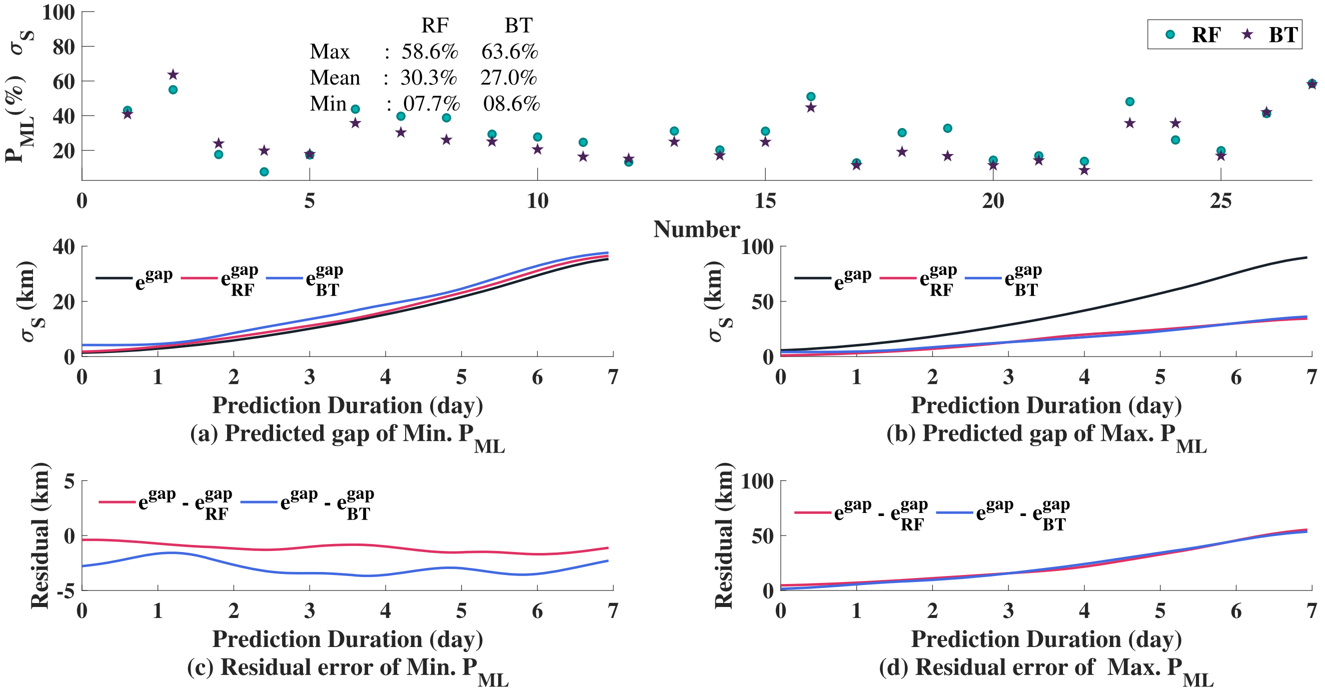
(a) Predicted gap of Min. P_{ML}

(b) Predicted gap of Max. P_{ML}

(c) Residual error of Min. P_{ML}

(d) Residual error of Max. P_{ML}

Figure 10. Improved covariance realism beyond the training period, Swarm B.



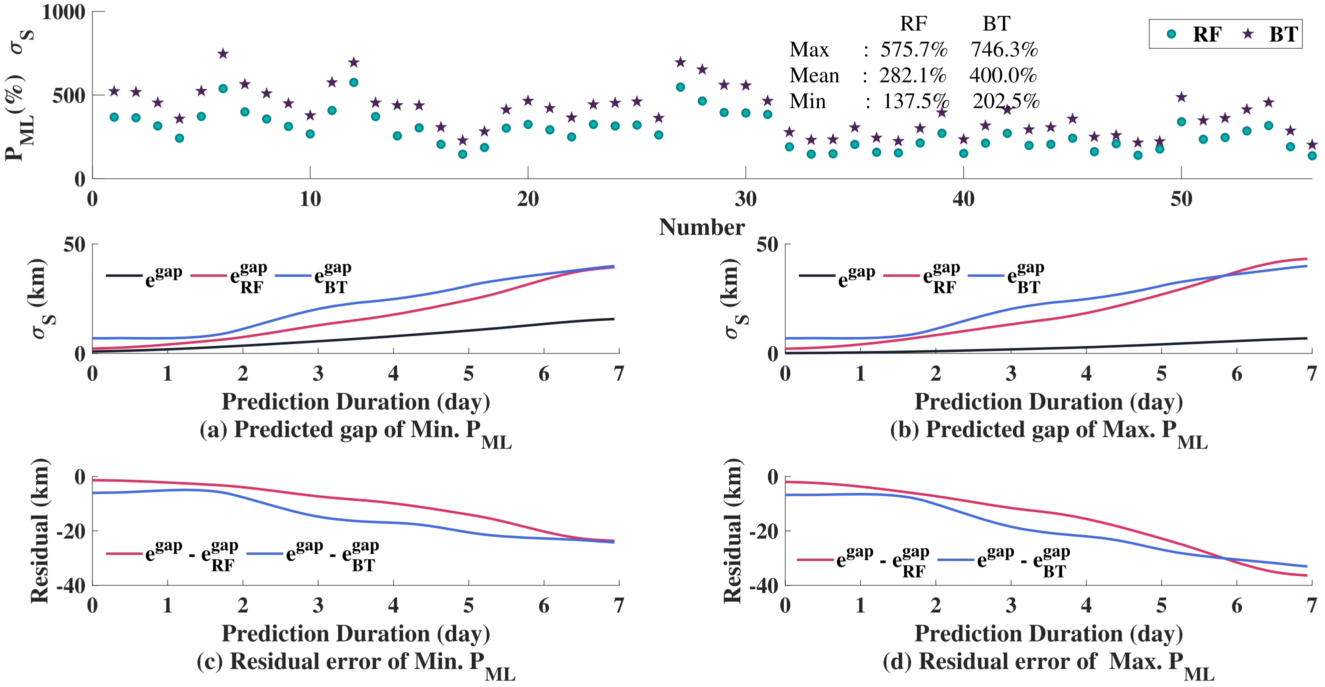
(a) Predicted gap of Min. P_{ML}

(b) Predicted gap of Max. P_{ML}

(c) Residual error of Min. P_{ML}

(d) Residual error of Max. P_{ML}

Figure 11. Improved covariance realism beyond the training period, Swarm C.



(a) Predicted gap of Min. P_{ML}

(b) Predicted gap of Max. P_{ML}

(c) Residual error of Min. P_{ML}

(d) Residual error of Max. P_{ML}

Figure 12. Performance of Swarm A models generalization to Swarm B.

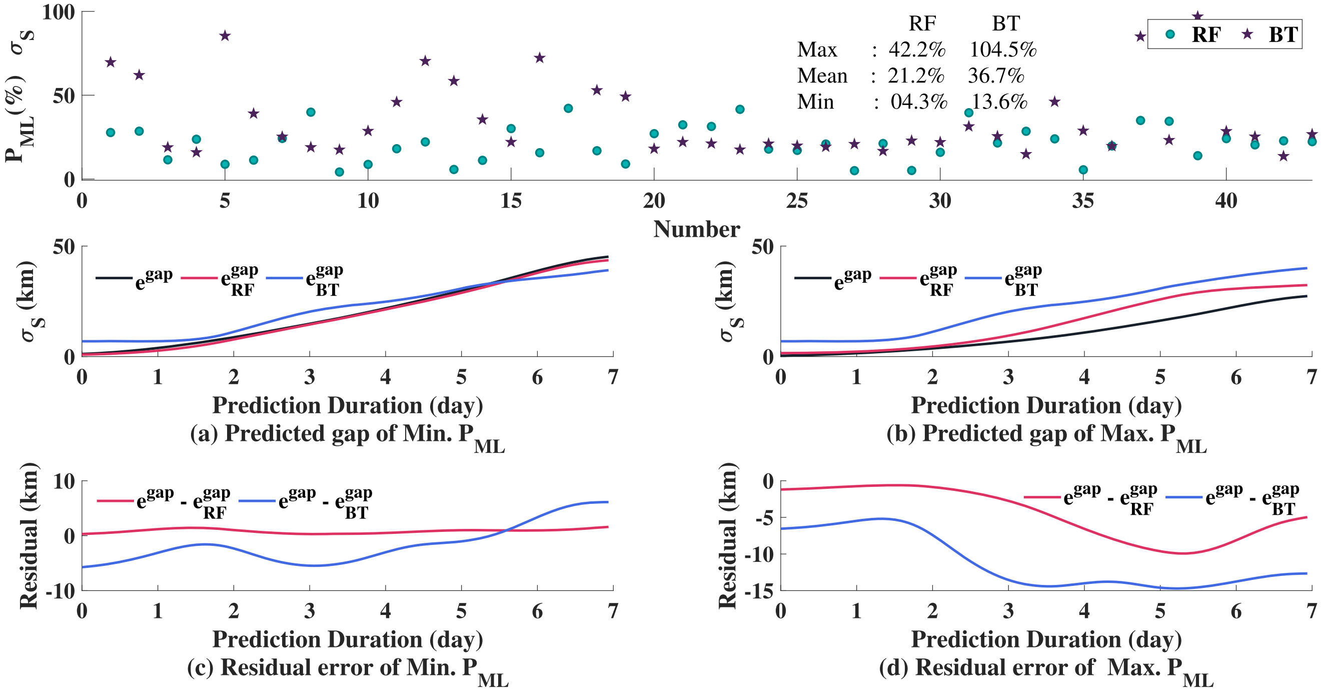
months. The performance of Swarm A model's generalization to Swarm B is summarized in Figure 12. It is observed that the mean metric P_{ML} exceeds 100%, reaching 282.1% and 400% for the RF and BT models, respectively, which indicates the trained models have very poor capability of generalization to Swarm B. As can be seen in Figure 12(a) and (b), the predicted gap, $e^{\mathrm{gap}}_{\mathrm{RF}}$ and $e^{\mathrm{gap}}_{\mathrm{BT}}$ given by the RF and BT models, are about 40 km at the end of the 7 days OP span, while the true inconsistency e^{gap} is about 10 km. An obvious reason is that the two satellites are at different altitudes, one at 460 km and the other at 520 km. This causes the data conditions and orbital environment, especially the atmospheric drag conditions, for the two satellites significantly different, and thus the inconsistency patterns of Swarm B would be significantly different from those of Swarm A. As a result, the trained models of Swarm A unlikely to contain all the patterns of Swarm B, and this causes large fluctuations of the predicted inconsistency when the Swarm A models are applied to Swarm B. This phenomenon would be further analyzed from the perspective of data similarity in Section 4.6.

On the other hand, the generalization performance of Swarm A models to Swarm C, shown in Figure 13, is much better. The mean P_{ML} values are 21.1% and 36.7% for RF and BT models, respectively, and the performance of the RF model is better than that of the BT model. It is also noted that the curves of the predicted gaps are smooth. As can be seen from subplot (a), the best performance of the RF model, with the P_{ML} equal to 4.2%, is achieved, which means more than 95% of the gap can be corrected by the trained RF model. Even in the worst validation case, as shown in (b), the predicted gap is larger than the true gap but the difference between them appears acceptable. For example, the propagated PC and RF-model predicted gap at the end of the 7 days OP span are 4.31 km and 33.46 km, respectively, while the PE is 33.76 km, making the difference reduced from 29.44 to 4.01 km. The rewarding performance of the trained Swarm A model's generalization to Swarm C is promising. This can be attributed to the fact that the two satellites are pair-flying such that they share similar data conditions and orbital environments.

4.6. Generalization Capability Analysis

The generalization of Swarm A models to Swarm B and Swarm C exhibits vastly different performances. An analysis is necessary to understand the possible causes.

Generally speaking, the generalization performance of the trained models is strongly dependent on the similarities between the training data and application (validation) data. That is, when the training data and application data have high similarities, good generalization performance can be expected. In real scenarios, this assumption may not be satisfied, rendering the generalization performance difficult to be guaranteed. In this section, we propose the use of Jaccard similarity to measure the similarity between training and validation data sets and to examine the relation between the data similarity and generalization performance.



(a) Predicted gap of Min. P_{ML}

(b) Predicted gap of Max. P_{ML}

(c) Residual error of Min. P_{ML}

(d) Residual error of Max. P_{ML}

Figure 13. Performance of Swarm A models generalization to Swarm C.

Jaccard similarity is a common proximity measurement that can be used to find the similarity between two data sets (Baroni & Buser 1976). Given two data sets, P and Q, their intersection is $P \cap Q$ with the size $|P \cup Q|$ (the number of members in the intersection set), and the union is $P \cup Q$ with the size $|P \cup Q|$. Then, the Jaccard similarity between them is defined as

$$J = \frac{|P \cap Q|}{|\,P \cup Q\,|}. \tag{9}$$

The Jaccard similarity will be 0 if the two sets do not have any member in the intersection set and 1 if the two sets are identical. It is intuitive to see that J gets higher when the two data sets are more alike. Additionally, this function can be used to find the dissimilarity between two sets by calculating $1 - J$. This metric is applicable to a wide range of applications due to its generality.

We now use the Jaccard similarity to measure the similarity between the training and validation data sets. To apply Equation (9), the training data set is defined as P, and the validation data set as Q, both of which are formed in the following way. Recalling that the training data set or the validation data set span over the same 3 months, and each data set has different numbers of OD runs. Each 3 days OD span is divided into several equal intervals. For every interval, the availability of the SLR track is examined. If there is a track in an interval, the track is coded as 1, otherwise as 0. For example, in the interval from 1.6 to 3.2 hr (assuming the interval length is 1.6 hr), there is a track of Swarm A, then the track is coded as 1. For a 3 days OD span, the data set is encoded as [1, 0, 1,...,0], where the number of encoded values is the number of intervals. In this way, the training or validation data set is constructed, as shown in Figure 14, where each row represents one 3 days OD span. Here, the 3 days OD span is divided into nine intervals of a length of 8 hr, actually, this figure can be regarded as the temporal distribution of the observation attributes.

The similarity between the training and validation data sets is examined in terms of temporal and spatial closeness. Consider the case that, there is a track from the training data set, and in the same interval, there is a track from the validation data set. The starting time of a track is denoted as t_t for the track from the training data set or t_v for the track from the validation data set. The azimuth and elevation at the point of maximum elevation of a track are denoted as $(az, el)_t$ or $(az, el)_v$, which are converted to the unit vector $\boldsymbol{r}_t$ or $\boldsymbol{r}_v$. The introduction of temporal and spatial closeness is to assist us in determining whether a model from data sets of one object (say Object A) can be generalized to an OD case of another object (say Object B). To do this, we first find the maximum J from all J values each is computed using Equation (9) with one data set of Object A and the data set of Object B. (For example, Object A has 30 data sets, we will have 30 J values.) Then, if the maximum J is larger than the threshold of J ($J_{\text{threshold}}$), we proceed to compute the angle between the data set of Object B and the data set of Object A which has the maximum J using Equation (10). Last, if the angle is larger than its threshold α, we decide that the model of Object A can be

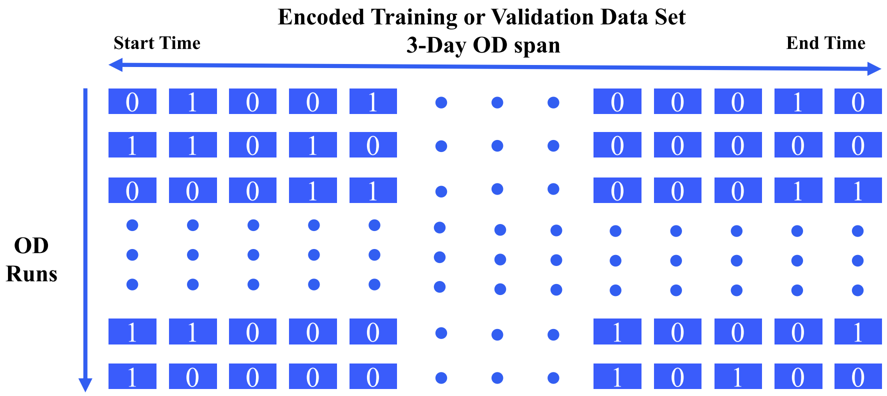


Figure 14. Illustration of the encoded data set.

generalized to the data set of Object B.

$$J \geqslant J_{\text{threshold}} \text{ and } \arccos\left(\frac{\boldsymbol{r}_t \cdot \boldsymbol{r}_v}{|\boldsymbol{r}_t||\boldsymbol{r}_v|}\right) \leqslant \alpha \tag{10}$$

For other cases where there is no or only one track from either the training or validation data set in an interval, the intersection in the interval is not available. The intersection set of the training and validation sets can now be formed, and the number of intersected tracks is counted. Figure 15 gives the schematic illustration of the spatial distribution of data.

When constructing the union set, only the interval having at least one track from either the training or validation data set is regarded as a member of the union, and thus the number of members in the union is obtained. The similarity between the training data set of Swarm A and the validation data set of Swarm B, as well as that between Swarm A and Swarm C, is presented in Table 3.

In Table 4, it is seen that different thresholds result in different similarity measures. When the thresholds for the time difference and angular difference increase, the similarity understandably increases as well. The similarity between Swarm A and Swarm C is always higher than that between Swarm A and Swarm B, indicating the observations of Swarm A and Swarm C are much closer to each other both temporally and spatially, which is favorable to generalize the Swarm A models to Swarm C. It is worth noticing that, in most cases, for a fixed time interval, the increase of angular threshold has no or little effect on the similarity. The higher similarity between Swarm A and Swarm C is understandable, because they are flying almost side-by-side, and the projected distance in the in-track direction between them is about 15 km, thus, the data condition and orbital environment for them are very similar. Different for the Swarm A and Swarm B, the similarity measures are significantly smaller than that between Swarm A and Swarm C, which leads to poorer generalization performance when the Swarm A models are applied to Swarm B.

The results provide some insights into whether the trained models of a satellite can be successfully generalized to other satellites. If the data of a satellite is temporally and spatially close to the training data, the trained models are more likely to have better generalization performance. In this sense, one may first perform the Jaccard similarity between the training data and application data, and then determine whether to apply the trained models to the application data.

4.7. Another Perspective of the Generalization Capability

Theoretically speaking, if the PEs are accurately described by the PCs at each propagation point, the set of ratios of the errors to the covariances expectations, calculated as Mahalanobis distances, should conform to a chi-squared distribution with 3 Degrees of Freedom (DoF) (Drummond et al. 2014). Usually, a Goodness-of-Fit (GOF) test is performed to assess how well the empirical distribution of Mahalanobis distances for each group of trajectories conforms to the expected 3-DoF chi-squared hypothesized parent distribution. Many GOF tests are proposed (Drummond et al. 2014; Aristoff et al. 2021), among these tests, Cramer von Misses (CvM) is widely used in space community.

In this section, the CvM GOF test is carried out to evaluate the effectiveness of the proposed compensation method. The evaluation procedure starts with the collection of bins of 3-DoF chi-squared statistics at each 60 s propagation point for a group of propagations. Accordingly, the number of 3-DoF chi-squared statistics in each bin is equal to the number of trajectories propagated in the group being tested. Ultimately, each bin of 3-DoF chi-squared statistics (at a common propagation point) is tested for realism using the aforementioned CvM test statistic.

Figure 16 shows the empirical and hypothesized parent 3-DoF chi-squared distributions before the trained model of Swarm A was generalized to Swarm C, the red curves represent empirical distributions at each propagation point whereas the blue square-dot curve represents the hypothesized parent distribution. Clearly, the empirical and hypothesized parent 3-DoF chi-squared distributions do not match at all, while in Figure 17 the empirical distributions are conforming much

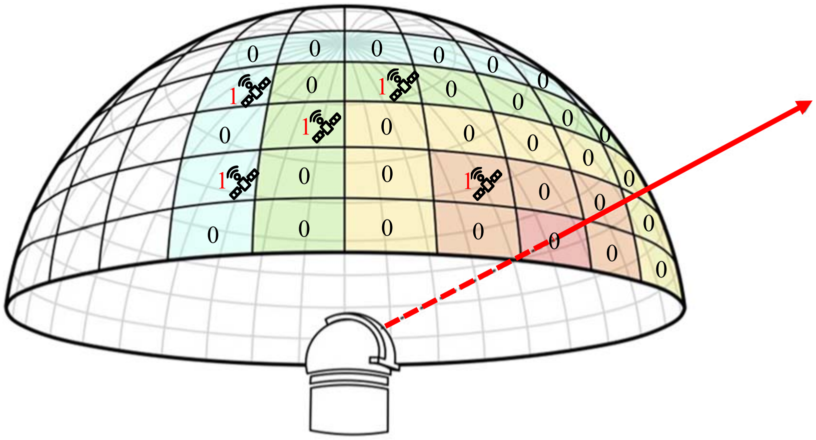


Figure 15. Schematic illustration of the spatial distribution of data set.

Table 4
Jaccard Similarity between Tracks of Swarm Satellites

Time Interval (hr)	Angular Interval (°)	Swarm A versus Swarm B	Swarm A versus Swarm C
	5	0.110	0.436
1.6	10	0.110	0.519
	15	0.169	0.519
	5	0.291	0.912
3.2	10	0.291	0.912
	15	0.310	0.325
	5	0.368	0.912
4.8	10	0.368	0.912
	15	0.368	0.912

closer to the hypothesized parent distribution after the trained model was generalized to Swarm C.

Figure 18 shows a p-value versus propagation time bar chart in which each p-value represents the likelihood the bin of covariances at a particular propagation point conforms to a 3-DoF chi-squared distribution. The red-dot line represents the 1% Confidence Interval whereas the blue bar represents the likelihood the empirical distribution conforms to its parent distribution at a particular propagation point (Sackrowitz & Samuel 1999). The covariance realism assessment in this study collects bins at each 1.5 hr propagation time step over a 7 days propagation timespan. As a result, 112 bins are assessed for covariance realism. After the trained model using Swarm A was generalized to Swarm C, 55.3% of the set of trajectories during the trajectory propagation timespan conform to a 3-DoF chi-squared hypothesized parent distribution. such a figure can be used to determine in which parts of the propagation timespan the covariance realism is faring well and in which it is faring poorly.

It is noted that covariance unrealism, however it may be evaluated, is unlikely consistent over the propagation time in the sense that, if the covariance is realistic at a particular propagation point, it cannot be expected to be realistic at all propagation points. When pursuing covariance realism remediation, one will therefore need to decide whether some sort of omnibus improvement over all propagation points is desired or whether it is preferable to focus on a particular part of the propagation timespan.

For our result, a Pass Percentage of 55.3% is not an ideal result, but a set of covariances produced by the generalized model exhibiting a Pass Percentage closer to 60% is still a significant step forward.

5. Conclusion

This work proposes an alternative approach to achieving better covariance realism of the predicted orbit state. The work has demonstrated the feasibility of using ML methods to capture the difference between the propagated covariance and the orbit prediction errors, and thus, applying the trained models to correct the propagated covariance will result in a more realistic characterization of the orbit prediction errors. Compared with our previous dynamic calibration method (Li et al. 2019), the proposed approach in this study is more automatic and general.

The BT and RF methods are implemented in the ML framework to learn the inconsistency pattern when only sparse SLR track data are available in the orbit determination. Using the data in one month as training and testing data, the RF methods can capture more than 98% of the underlying inconsistency pattern. Applying the learned models to the data in the subsequent two months (the generalization to the future), the propagated covariance can be effectively corrected for more than 50% of the inconsistency, resulting in the corrected covariance much closer to the errors of true predicted states.

The performance of applying Swarm A models to Swarm B and Swarm C is also investigated. The generalization of the Swarm A models to Swarm C is more successful than that to

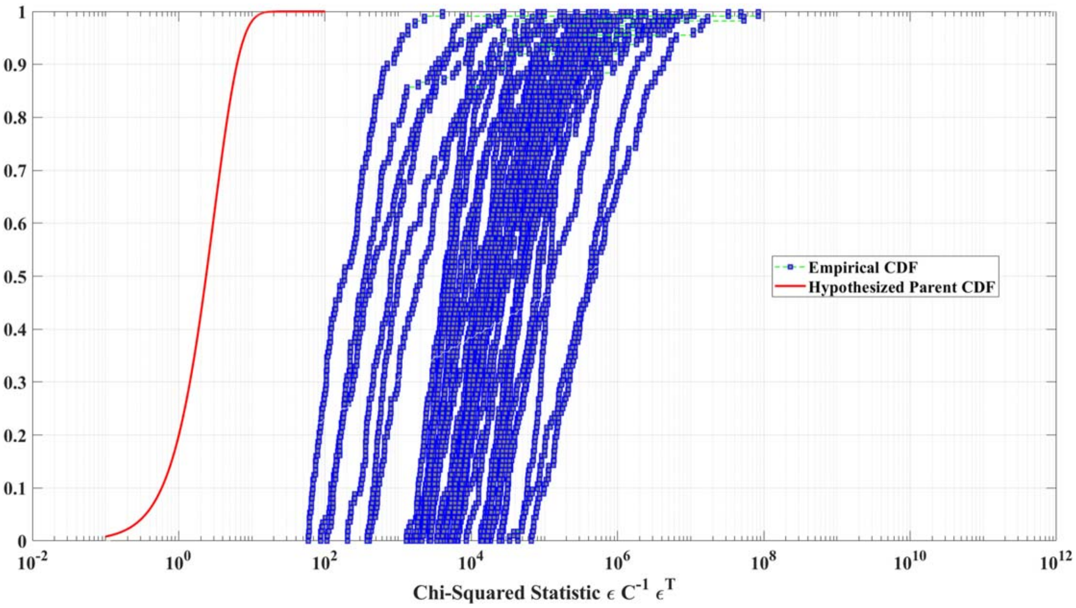


Figure 16. Empirical and hypothesized parent 3-DoF chi-squared distributions (before the trained model of Swarm A was generalized to Swarm C).

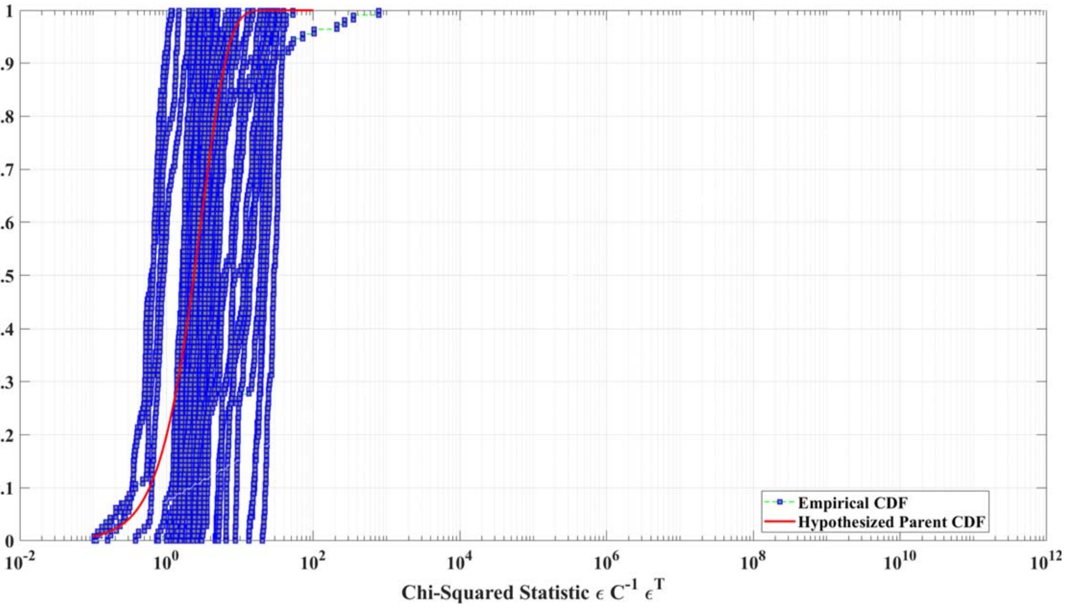


Figure 17. Empirical and hypothesized parent 3-DoF chi-squared distributions (after the trained model of Swarm A was generalized to the Swarm C).

Swarm B. Because Swarm A and Swarm C are flying side-by-side, about 79% and 64% of the inconsistency of Swarm C can be corrected on average when the Swarm A models are applied. To further explore the behind causes, the Jaccard similarity is introduced to examine the temporal and spatial similarity between the training data and application data. It is found that the Jaccard similarity is a proper metric that can guide one on whether and when is the right time to generalize the trained models of one satellite to another.

Furthermore, a traditional CvM GOF test is performed to evaluate the covariance realism. A Pass Percentage of 55.3% has been achieved, which further demonstrates the effectiveness of the proposed machine learning algorithm in improving the covariance realism.

From an operational perspective, the trained models, once optimized, could be deployed to run alongside traditional methods to perform near-real-time calibration to achieve better covariance realism. Further research is suggested to apply the

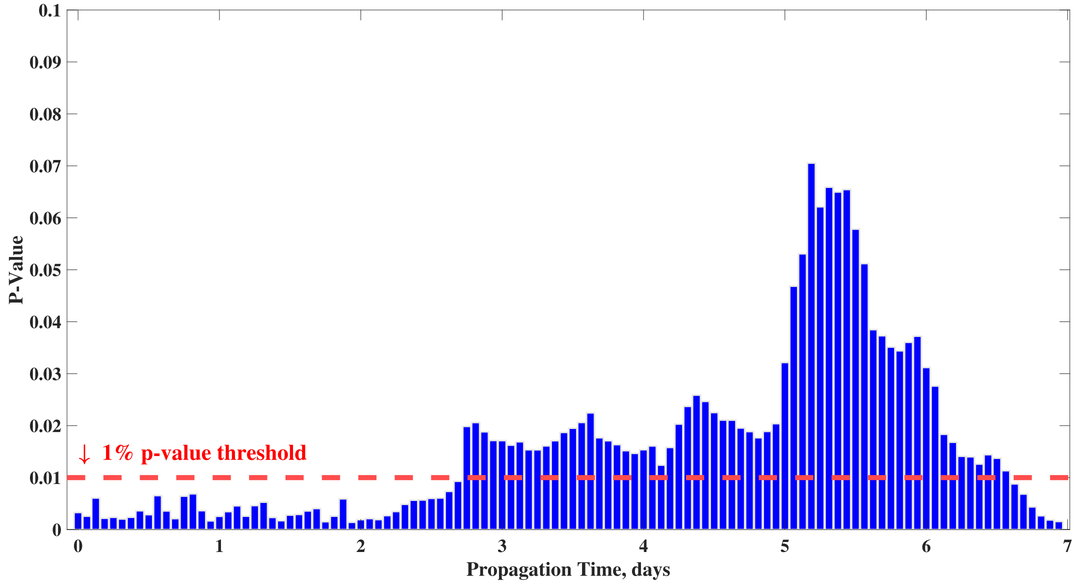


Figure 18. *p*-value versus propagation time bar chart (after the trained model of swarm A was generalized to Swarm C).

established ML framework to more tracking data scenarios and more RSOs.

Acknowledgments

This research has been supported by the National Natural Science Foundation of China (grant No. 12103035), the Special Fund of Hubei Luojia Laboratory (grant No. 230600003), and the Fundamental Research Funds for the Central Universities (grant No. 2042023gf0007). The authors are grateful to anonymous reviewers whose constructive and valuable comments greatly helped us to improve the paper.

ORCID iDs

Bin Li https://orcid.org/0000-0001-5548-0353

References

Aristoff, J. M., Horwood, J. T., & Alfriend, K. T. 2021, CeMDA, 133, 1
Baroni, U. C., & Buser, M. W. 1976, Systematic Zoology, 25, 251
Bobrinsky, N., & Del, M. L. 2010, CosRe, 48, 19
Curzi, G., Modenini, D., & Tortora, P. 2022, CEAS Space Journal, 14, 197
Drummond, O. E., Horwood, J. T., Aristoff, J. M., Singh, N., & Poore, A. B. 2014, Signal and Data Processing of Small Targets, 66, 92
Drummond, O. E., Ogle, T. L., & Waugh, S. 2007, Signal and Data Processing of Small Targets, 99, 395
Duncan, M., & Long, A. 2006, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 155, 6293
Friedman, J., Tibshirani, R., & Hastie, T. 2000, AnSta, 2, 28
Jiang, H., Liu, J., Cheng, H. W., & Zhang, Y. 2017, RAA, 17, 30
Junkins, J. L., Akella, M. R., & Alfrined, K. T. 1996, JAts, 44, 541
Li, B., Huang, J., Feng, Y. M., Wang, F. H., & Sang, J. Z. 2020, IEEE Trans. Aerosp. Electron. Syst., 3, 1
Li, B., Sang, J. Z., & Liu, H. K. 2019, AdSpR, 63, 2422
Li, B., Zhang, Y., Huang, J., & Sang, J. Z. 2021, AcAau, 5, 18
Liu, L., Li, B., Chen, J. Y., et al. 2021, RAA, 21, 301
Liu, T., & Schreiber, K. U. 2021, AcAau, 185, 161
Lopez, J. S., Pastor, A., & Escobar, D. 2021, AcAau, 181, 679
Mereta, A., Izzo, D., & Wittig, A. 2017, in Int. Conf. Hybrid Artificial Intelligence Systems, ed. T. Flohrer, R. Jehn, & F. Schmitz (Darmstadt)
Molnar, C. 2020, Interpretable Machine Learning (Lulu. com)
Montenbruck, O., & Gill, E. 2000, Satellite Orbits Model, Methods, and Applications (1st ed.; Berlin: Springer), doi:10.1007/978-3-642-58351-3
Pawloski, J., Duncan, M., Hesar, S., Lechtenberg, T., & Wysack, J. 2018, in SpaceOps Conference, ed. J. W. McMahon (San Diego: Univelt), 2699
Peng, H., & Bai, X. L. 2018, AdSpR, 121, 2628
Peng, H., & Bai, X. L. 2019, AcAau, 161, 44
Peng, H., & Bai, X. L. 2021, AcAau, 184, 68
Poore, A. B., Aristoff, J. M., Horwood, J. T., et al. 2016, Covariance and Uncertainty Realism in Space Surveillance and Tracking, Tech. Rep.1Numerica Corporation Fort Collins United States
Russell, S. J., & Norvig, P. 2003, in Artificial Intelligence: A Modern Approach, ed. S. J. Russell & P. Norvig (4th ed.; New York: Pearson)
Sackrowitz, H., & Samuel, C. E. 1999, The American Statistician, 53, 326
Shen, D., Lu, J., Chen, G., et al. 2019, in IEEE National Aerospace and Electronics Conf., ed. A. Vananti (NewYork, US), 565
Vallado, D. A. 2007, Fundamentals of Astrodynamics and Applications (3rd ed.; Hawthorne: Springer)
Vallado, D. A., & Seago, J. H. 2009, AdAnS, 135, 49
Xu, X. L., & Xiong, Y. Q. 2014, RAA, 14, 601
Zaidi, W., & Hejduk, M. D. 2016, AIAA/AAS Astrodynamics Specialist Conf., 1, 5628
Zhang, Y., Li, B., Liu, H. K., & Sang, J. Z. 2022, Geo-spatial Information Science, 25, 104