Machine Learning for LEO and MEO Satellite Orbit Prediction

Author(s):	Selvan, Kannan; Siemuri, Akpojoto; Prol, Fabricio S.; Välisuo, Petri; Kuusniemi, Heidi
Title:	Machine Learning for LEO and MEO Satellite Orbit Prediction
Year:	2024
Version:	Publisher's PDF
Copyright	©2024 Author(s). Published by Institute of Navigation.

Please cite the original version:

Selvan, K., Siemuri, A., Prol, F, S., Välisuo, P., & Kuusniemi, H. (2024). Machine Learning for LEO and MEO Satellite Orbit Prediction. *Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024), (pp. 3556-3571).* Institute of Navigation. https://doi.org/10.33012/2024.19765

Machine Learning for LEO and MEO Satellite Orbit Prediction

Kannan Selvan[1], Akpojoto Siemuri[1], Fabricio S. Prol[1,2], Petri Välisuo[1], Heidi Kuusniemi[1,2]
[1]*School of Technology and Innovations, University of Vaasa, Finland*
[2] *Finnish Geospatial Research Institute, National Land Survey, Finland*

BIOGRAPHY

Kannan Selvan received his B.Sc.(tech) degree in Electronics and Communication Engineering from Anna University, India in 2012, the M.Sc.(tech) degree in Communications and Systems Engineering from the University of Vaasa, Finland in 2020. He is currently pursuing a Ph.D. degree in automation technology at the University of Vaasa. From 2018 to 2020, he was a Research Assistant in the Digital Economy Research Platform at the University of Vaasa, Finland. He is currently a Project Researcher at the University of Vaasa. His research interest includes GNSS technologies, LEO-PNT, satellite-data analysis, machine learning, satellite communication, smart devices, and embedded Systems.

Akpojoto Siemuri received his B.Sc.(tech) degree in electrical and computer engineering from the Federal University of Technology Minna, Nigeria in 2010, an M.Sc.(tech) degree in wireless industrial automation with a minor study in industrial management from the University of Vaasa, Finland in 2019. He is currently pursuing a Ph.D. degree in automation technology at the University of Vaasa. From 2018 to 2019, he was a Research Assistant in the Smart Energy Systems Research Platform (SESP) Project at the University of Vaasa, Finland. He is currently a Project Researcher at the University of Vaasa. His research interest includes machine learning, GNSS technologies, smart devices, embedded systems, communication systems, and game theory.

Fabricio dos Santos Prol is a Senior Researcher at the Finnish Geospatial Research Institute (FGI) in the National Land Survey of Finland (NLS) - Finland. He is also a Docent Fellow in the University of Vaasa. The focus of his current research lies in GNSS positioning, LEO-PNT systems, ionospheric modeling, and data assimilation.

Petri Välisuo is currently working as an Associate Professor (tenure track), in sustainable automation, at the School of Technology and Innovations, University of Vaasa, Finland. He received M.Sc.(tech) degree in computer science from the Tampere University of Technology, Finland, and a D.Sc.(Tech) degree in automation technology from the University of Vaasa, in years 1996 and 2011 respectively. He has authored and co-authored 27 peer-reviewed and more than 10 other scientific publications. His research interests cover machine learning, IoT, positioning methods, and other technologies relevant to industrial automation. He has been working for 10 years in the telecommunication industry before his research career at the University of Vaasa.

Heidi Kuusniemi is a professor of computer science and director of Digital Economy at the University of Vaasa in Finland. She is also a part-time research professor in satellite navigation at the Finnish Geospatial Research Institute. She has an M.Sc. (Tech.) degree (with distinction) from 2002 and a D.Sc. (Tech.) degree from 2005 in information technology, respectively, from Tampere University of Technology, Finland. She is the President of the Nordic Institute of Navigation. Her technical expertise and interests include GNSS reliability and resilience, estimation and data fusion, mobile precision positioning, indoor localization, and PNT in new space.

ABSTRACT

Accurate orbit prediction plays a significant role in many space geodesy applications, including space situational awareness, orbital maneuvers, and satellite navigation in real-time scenarios. The traditional orbit prediction approach includes analytical and numerical algorithms to accurately propagate the satellites' state and associated uncertainties. However, these methods are based on limited dynamic models and may have limitations in accurately capturing the complex dynamics of satellite motion. With the rapid growth in computing power and the development of advanced machine learning (ML) and deep learning (DL) algorithms, there has been growing interest in leveraging ML algorithms to enhance orbit prediction accuracy. In this study, we investigate the potential of using different ML algorithms for the orbit prediction of low Earth orbit (LEO) satellites. We also extend our analysis to medium Earth orbit (MEO) satellites. LEO Swarm-A satellite's precise orbit products and MEO GNSS satellites' final ephemeris products are used. The datasets are pre-processed and used in training various ML models. The ML models are then used to estimate the position and velocity of the LEO and MEO satellites. The accuracy of both LEO Swarm-A satellite and MEO GNSS satellites' using various ML models are compared and discussed. Overall, the standalone ML-based method holds significant promise for improving orbit prediction accuracy and reliability for SWARM-A satellite and GNSS satellites, ultimately enhancing the performance and efficiency of space-based applications and services.

Keywords - Orbit prediction; Low-Earth-Orbit; Medium-Earth Orbit; Global Navigation Satellite Systems; Machine Learning; Ephemeris

I. INTRODUCTION

Highly accurate orbit prediction is indispensable for a wide range of applications in space geodesy [1]. With the advancements in ground-based and space-based technology, near-real-time and real-time services have been developed requiring high-accuracy orbit prediction. Mission design, orbit determination, payload data analysis, satellite navigation and positioning, and other applications such as real-time navigation, atmospheric monitoring, and precise point positioning require a high-precision satellite orbit prediction [2], [3], [4], [5], [6].

The traditional method for satellite orbit prediction relies on physics-based models, which include solving the differential equations of motion using either analytical or numerical methods. The numerical method involves integrating the equations of motion to predict the satellite orbit [7]. The main advantage of using the numerical method is that the orbit prediction is highly accurate compared to other methods, being suitable for complex scenarios but computationally demanding. On the other hand, the analytical method provides the position and velocity of the satellite at any point in time [8]. It is a commonly used method for orbit prediction and is computationally efficient. However, the disadvantage of the analytical method is the challenge to derive higher-order solutions, leading to less accuracy in predicting the satellite orbit compared to the numerical methods. In addition, a combination method known as the semi-analytical method involves complex perturbation effects, which are then transferred by the analytical method [9]. Although faster computational results can be obtained compared to the numerical method, a trade-off between accuracy and agility should be analyzed.

Various analytical and numerical method propagators are used to propagate the state of the satellites and its associated uncertainties [10]. Multiple sources of perturbing forces such as Earth's non-uniform gravitational field, solar radiation pressure, and attraction of the sun and moon are taken into account to various extents [11]. The simplified general perturbations (SGP4) algorithm is an analytical propagator which uses the two-line element (TLE) files to generate the ephemeris. TLE includes a set of mean orbital elements that defines position and velocity of the satellite at a reference epoch without needing to calculate each intervening state [12]. Higher order terms are omitted in SGP4 due to the intricacy of the equations of motion of orbital mechanics. Therefore, with analytical methods purely based on limited dynamic models, it is not suitable for all positioning, navigation and timing (PNT) applications requiring high accuracy [11]. On the other hand, high-precision orbit propagators (HPOP) are numerical propagators incorporating detailed force models, taking into account the higher-order terms that are omitted by SGP4. Though the numerical integration technique yield orbit predictions accurate to within meters, it requires the computation of the object's state vector for each time step increment. Thus, numerical methods require large quantities of accurate data and computation capabilities, which are often not existing for many satellites. [13] Therefore, it is challenging to establish an accurate physics-based model using analytical and numerical methods and predict the satellite orbit with high accuracy.

Considering the rapid growth in computing power, advanced machine learning (ML) algorithms can be used for predicting the LEO and MEO satellites' orbit. ML offers new modeling strategies for orbit prediction methods. ML models learn the underlying pattern based on a large amount of data and predict future events [14]. This data-driven approach enables ML models to capture dynamic and unstable environments, handle nonlinear relationships, and improve accuracy by identifying correlations that are difficult to model using the conventional techniques. The ML approach has been beneficial in a wide range of applications [15, 16] and especially in the aerospace field [17, 18, 19, 20].

In this paper, we aim to apply ML techniques for the orbital accuracy of LEO satellites. A simple yet efficient method for modeling the relationship between input variables are provided by utilizing ML model. Based on the training of various ML models using historical satellite orbit data, we develop a predictive model capable of estimating the positions and velocities of the LEO satellite, thereby reducing the root-mean-square error (RMSE) of orbit predictions. Therefore, the primary objective is to assess the feasibility and efficacy of utilizing different ML techniques for orbit predictions of LEO satellites and improving the prediction accuracy quantified by RMSE. To evaluate the robustness and versatility of our proposed method using different ML models, we also extend our analysis to MEO satellites and highlight any potential differences between LEO and MEO satellite predictions. We implemented various ML algorithms such as KNeighbors Regressor (KNR), Random Forests Regressor (RFR), linear regression (LR), and polynomial regression (PR), and aimed to demonstrate the effectiveness of different ML models and compare their performances.

The remaining parts of this article are organized as follows. In Section II, the related works are presented with respect to the orbit prediction carried out using various approaches. Section III presents the strategy used in this research work to assess the ML models for orbit prediction of LEO and MEO satellites. In Section IV, detailed information on the data preprocessing and preparation is provided. Section V discusses the ML models implemented to improve the orbit prediction accuracy, while section VI presents the results and evaluation of the ML models for both the LEO and MEO satellites. In Section VII, conclusions are summarized, and future research are suggested.

II. RELATED WORKS

From the existing studies, different approaches have been implemented to improve the orbit prediction accuracy. In [13], the authors proposed a batch least-squares method, in which differential corrections are applied to objects in the two-line element (TLE) catalog. Using a high-precision numerical propagator, the orbit is fitted to the state vectors obtained from consecutive TLEs. The predicted range error increases at a typical rate of 100 m per day. In [5], the authors proposed a method to improve the computational efficiency of numerical propagators for orbit propagation of Molniya satellite while maintaining the required accuracy levels. It aimed towards optimal balance between accuracy and computation by adjusting the number of geopotential spherical harmonics retained during integration, thereby ensuring that the numerical propagator achieves the required accuracy with minimum computational cost. In [21], the authors utilized an analytical orbit model to analyze the precision of orbit prediction for MEO and LEO satellites. The orbit prediction was validated using a high precision laser ephemeris and obtained an accuracy of several hundred meters in a 1-day orbit prediction and not more than 10 km in 7 days. In addition to the traditional methods, [22] introduced a novel approach in orbital modeling by utilizing data-driven techniques to forecast the orbital parameters. Using historical orbital data, the novel predictor system upon training constructs new TLEs near the desired prediction time. The initial results using a simple predictor system demonstrate comparable or better accuracy than traditional SGP4, particularly when predicting two weeks beyond the available data.

In [23], the authors discuss the Earth rotation parameters (ERPs) predicted by different services to analyze their impact on ultra-rapid ephemeris prediction of GNSS. A real-time correction method is proposed to reduce the impact of the ERPs on GNSS ultra-rapid orbit prediction products. Experimental results demonstrated significant accuracy improvements when using the proposed correction method. In [24], the authors introduced an error analysis approach based on historical data and the periodic characteristics of the orbit prediction error. The error fitting is carried out by introducing the Poisson series. In [25], two methods aimed to overcome the challenges caused by the memory and computing time required for orbit prediction procedures by providing an analytical representation of numerical orbits.

Based on ML approaches, the three most common types that have been studied include supervised learning, reinforcement learning, and unsupervised learning [15]. In [26], the authors discuss challenges in achieving accurate orbit prediction due to limited information about space object's trajectories and operating environments. It highlights the shortcomings of physics-based models and the lack of accuracy in TLE-based prediction techniques. The approach for orbit prediction makes use of ML techniques, particularly curve fitting and Long Short-Term Memory (LSTM) models trained using TLE data. In [27], a novel hybrid model was implemented combining the SGP4 with two ML estimators, the autoencoder and random forest. This approach aims to reduce errors in SGP4 propagators caused by incomplete perturbation forces and low-order of series expansions. The applied ML estimators model enabled the correction of the time-series nature of error patterns, resulting in more accurate orbit predictions. The hybrid model, tested on three satellite objects with corresponding TLE data, achieved a 20-30% improvement in orbit prediction accuracy over a 30-day period. In [28], orbit prediction on GNSS constellations in the medium Earth orbit (MEO) using different ML and DL algorithms was analyzed for improving the accuracy of the ultra-rapid products from IGS. Utilizing Long Short-Term Memory and Gated Recurrent Unit, an average improvement of 40% in 3D RMS was obtained within the 24-hour prediction interval of the ultra-rapid products.

In [29], the Support Vector Machine (SVM) was used to improve the satellite orbit prediction accuracy making use of two publicly available data to validate the proposed ML approach, namely the TLE catalog and the International Laser Ranging Service (ILRS) catalog. In most cases, the results showed that the designed dataset structure and SVM model can improve the orbit prediction accuracy with good performance. Further, in [30], three machine learning approaches are investigated to improve the orbit prediction accuracy in a simulation environment, namely SVM, artificial neural network (ANN), and Gaussian processes (GP). ANN showed better approximation capability compared to SVM and GP.

In addition to the aforementioned studies, numerous other research focused on enhancing the precision of orbital propagation models through the application of ML techniques [31, 32, 33]. These investigations have demonstrated that ML methods can predict accurately while maintaining the orbit propagation model using historical data with limited resources. Nevertheless, each of the techniques have its own strengths and limitations based on factors such as data types, sizes, and dataset behavior.

ML has also been extensively used for the orbit determination and orbit prediction of LEO satellites. LEO satellites have the potential to deliver several benefits over MEO satellites in terms of PNT, as well as location-enabled communications [34]. In [11], a hybrid analytical-ML framework was developed for improved LEO satellite orbit prediction. It includes three stages as follows: 1) initial estimation of LEO satellite's states with SGP4-propagated TLE data and subsequently estimating with extended Kalman filter (EKF) during satellite visibility; 2) utilization of a nonlinear autoregressive with exogenous inputs (NARX) neural network for orbit propagation when the satellite is not in view; and 3) estimation of the terrestrial receiver position using ML-predicted satellite ephemeris and carrier phase measurements. Experimental results demonstrated a significant reduction in ML-predicted ephemeris error by nearly 90% compared to SGP4 propagation. In [35], ML is used for predicting LEO satellites in a simultaneous tracking and navigation (STAN) framework. In this work, a time delay neural network (TDNN) is developed, which is shown to improve the LEO satellite tracking performance over an extended Kalman filter (EKF). The EKF-STAN achieved 3D position RMSE of 71 m and 26 m for the two LEO Orbcomm satellites, while the proposed EKF-TDNN-STAN framework achieved 3D position RMSE of 6 m and 26 m, respectively. In [36], the paper employed HPOP in conjunction with decoded Orbcomm satellite ephemeris messages to train a neural-network (NN) model capable of estimating the satellite's position with meter-level precision in a short time period.

These existing studies implemented various approaches to improve the orbit prediction accuracy. Traditional methods have been employed to predict the satellite orbits with varying degrees of accuracy. Subsequently, ML approach has shown capability compared to the dynamic models. Furthermore, existing studies also implemented a hybrid approach which combines ML and traditional methods, demonstrating significant improvements in the orbit prediction accuracy. However, there remains value in further exploring standalone ML-based method for orbit prediction. In this paper, we utilize various ML models and assess the accuracy of each of the models in predicting the LEO and MEO GNSS satellites. The decision to employ a standalone ML-based method is purely based on the need to provide a simplified and accessible alternative to traditional or hybrid methods while maintaining predicted accuracy. By focusing solely on the ML-based method, this approach aims to alleviate the intricacies associated with traditional and hybrid methodologies, thus facilitating simplified implementation and interpretation. Therefore, while various approaches have demonstrated notable success, the use of ML-based method provide an additional means of addressing orbit prediction challenges with simplicity, efficiency and interpretability. The models will be trained using the precise ephemeris data obtained from the European Space Agency's Swarm Data Access [37] for LEO satellites and from the office of Geomatics of the National Geospatial-Intelligence Agency (NGA) [38] for MEO GPS satellites.

III. STRATEGY

Orbit prediction involves the estimation of future state of motion of an object based on the orbit determined in past observations. Applications of orbit prediction include satellite navigation, orbital maneuvers, and space situational awareness. A set of approximate equations of motion is used to describe the motion of the object. The degree of approximation varies depending on the intended use of orbital information. However, a better orbit prediction can be made with highly accurate dynamic models [39].

In this work, the primary orbit prediction strategy involves implementing ML-based algorithms to estimate the LEO SWARM-A satellite and MEO GPS satellites' orbit. The accuracy of ML models are evaluated in terms of RMSE. By using different ML models such as KNR, RFR, LR and PR models, we aim to demonstrate that a standalone ML-based approach can yield high accuracy in orbit prediction.

The steps involved in orbit prediction are as follows:

1. Data preprocessing and preparation;
2. Proposed ML-based prediction method;
3. Performance comparison of the ML models;
4. Evaluation of the implemented ML models.

IV. DATA PREPROCESSING AND PREPARATION

With respect to LEO, the data used to train the model are the SWARM-A satellite precise ephemeris data in the SP3 format. SWARM-A is one of the three satellites in the ESA's SWARM constellation, operating in a circular near-polar orbit at an altitude of 460 km [40]. The ephemeris data includes time of observation, position, velocity, and satellite clock parameters. In this study, we used the time series of dynamic position and velocity of the SWARM-A satellite in the International Terrestrial Reference Frame (ITRF) related to the center of mass. The sampling period is 10 seconds, i.e., the positions and velocities are observed for every 10 seconds. The ephemeris data from 02/10/2023 to 31/01/2024 are collected from the ESA portal [37].

In MEO, the data used to train the model is the GNSS satellites' final ephemeris data provided by the International GNSS service (IGS) in the SP3 format. The GNSS satellites' here refers only to the 32 GPS satellites orbiting in different orbital planes. The ephemeris data includes time of observation, position, velocity, and satellite clock parameters. Time series of dynamic position and velocity of GPS satellites were used in the analysis. The sampling period is 5 minutes, i.e., the position and velocity of all 32 GPS satellites are measured every 5 minutes. Data from GPS week 2086 to 2251 were collected from the NGA portal for the analysis [38].

The initial step in the data processing pipeline involves defining the number of days of data to be used by the ML models. We consider different number of days of data such as 1, 3, 10, 30 and 50 days for the analysis of the ML models. In case of MEO, the algorithm processes only one single GPS satellite at a time, so we include GPS PRN number in the processing pipeline. The next step involves reading the time, position and velocity data for SWARM-A satellite or a single GPS satellite. In order to ensure that the data are in a consistent and appropriate coordinate system, the function not only retrieves the necessary satellite tracking data but also converts the time-series position and velocity data from ITRF to the Geocentric Celestial Reference System (GCRS). Following the coordinate transformation, Lagrange interpolation is used to improve the temporal resolution of the data. A polynomial is generated by using the interpolation technique that passes over a group of measured data points and takes certain values randomly, providing a denser and more continuous dataset. For SWARM-A satellite, considering the 10-second sampling interval, 24 hours consist of 8639 actual data points of position and velocity. Upon interpolation process, there are a total of 86390 points as shown in figure 1 for step size of 1 s. In other words, there is an actual data point or interpolated data point for every one second. Similarly, for step size 3, there is an actual data point or interpolated point for every three seconds. The step size defines the prediction time of the satellites. In this analysis, for SWARM-A satellite, different prediction time intervals such as 1, 3, 5, 7, 9 and 10 seconds are also considered in addition to the number of days of data used by the model. The red-colored dots are the actual data points obtained from SP3 files at a certain sampling time (seconds), while black-colored dots indicate the interpolated points, providing denser data points upon interpolation process as seen in the figure 1.

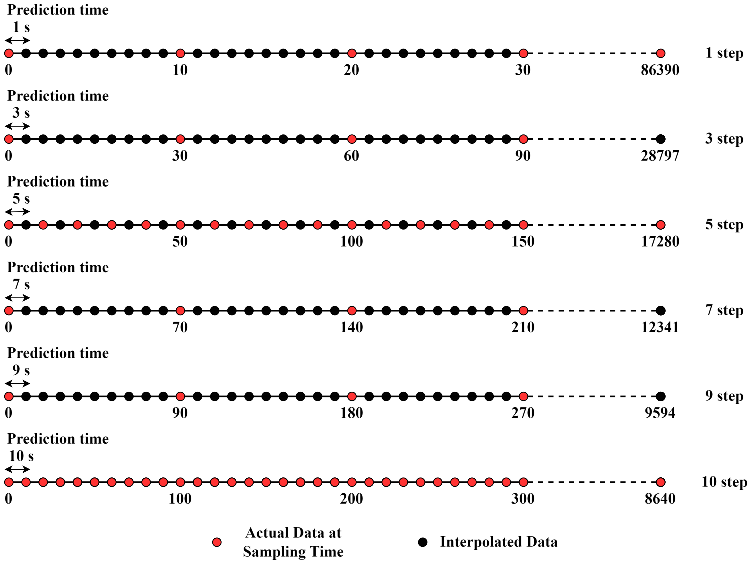

Figure 1: Interpolating points between the observed data points for LEO SWARM-A satellite data using Lagrange interpolation method.

For each GPS satellite, considering a 5-min sampling interval, there are a total of 288 data points in 24 hours. Upon interpolation process, a single satellite selected with 288 data points in 24 hours will have a total of 86100 points for step size of 1 as shown in figure 2. There are 299 interpolated points between every 5-min sampling interval when the step size is 1. Similarly, there are 29 interpolated points between every 5-min sampling interval for a step size of 10. In this analysis, for GPS satellites, the prediction time interval of 10, 20, 30, 40, 50 and 60 seconds were considered in addition to the number of days of data used. Figure 2 shows how the interpolation process works, displaying the actual data for every 5-minutes (300 seconds) with red color and interpolated points with black color.

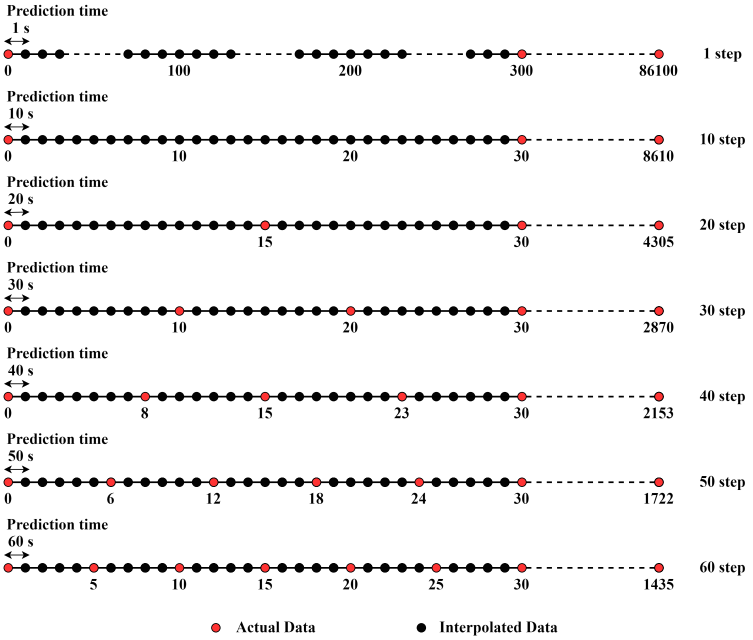

Figure 2: Interpolating points between the observed data points for a single GPS satellite using Lagrange interpolation method.

After the completion of the interpolation process, the dataset now contains the observed data points and interpolated points. Based on this dataset, our developed algorithm is processed to generate the input features and labels to train the ML models. The input features are the historical positions Px, Py, Pz and velocities Vx, Vy, Vz of the SWARM-A and GPS satellites.

For i^{th} timestamp, the position and velocity are given as follows: $Px_i,\ Py_i,\ Pz_i$ and $Vx_i,\ Vy_i,\ Vz_i$

Similarly, the position and velocity for the next timestamp, i.e., the next data point is given by: $Px_{i+1},\ Py_{i+1},\ Pz_{i+1}$ and $Vx_{i+1},\ Vy_{i+1},\ Vz_{i+1}$

The labels are the true position offsets and velocity offsets computed based on the positions and velocities value. True position offset, y_p is given by the difference between the current position data (Px_i, Py_i, Pz_i) and next position data $(Px_{i+1}, Py_{i+1}, Pz_{i+1})$, given as follows:

$$y_p = C_p - N_p \tag{1}$$

where y_p is the true position offsets, C_p is the current position and N_p is the true next position.

Similarly, the true velocity offset, y_v is given by the difference between the current velocity data (Vx_i, Vy_i, Vz_i) and next velocity data point $(Vx_{i+1}, Vy_{i+1}, Vz_{i+1})$, given as follows:

$$y_v = C_v - N_v \tag{2}$$

where y_v is the true velocity offsets, C_v is the current velocity and N_v is the true next velocity.

The true position and velocity offsets calculated as shown in the equations 1 and 2 using the positions and velocities value will be used as the reference data to validate the different ML models. In general, the true offsets are represented using the following equation.

$$y = C - N \tag{3}$$

where y is the true offset which can be either a positive or negative value, C is the current position or velocity, N is the true next position or velocity.

Equation 3 can be rearranged as follows:

$$N = C - y \tag{4}$$

where N is the true next position or velocity, C is the current position or velocity and y is the true position or velocity offset.

Following the generation of the input features and labels, the dataset is now divided into training (70%) and test (30%) subsets. Training the ML model with training data enables the model to learn the underlying pattern from the historical positions and velocities obtained. The predictive performance of the model is then evaluated using the test data. This comprehensive data preparation approach ensures that the ML model is trained using a high-quality dataset.

V. PROPOSED METHOD

1. ML-based Prediction Method

ML models are utilized to estimate the satellite orbits using precise ephemeris data. We implemented different ML models, namely the KNR model, RFR model, LR model and PR model. The models are trained using the input features generated as discussed in the section IV. The trained model then predicts the position and velocity offsets based on the test data fed into the trained model. Similar to equation 3, the predicted position and velocity offsets, in general, are given using the following equation.

$$\hat{y} = C - \hat{N} \tag{5}$$

where $\hat{y}$ is the predicted offset, C is the current position or velocity, $\hat{N}$ is the estimated position or velocity.

From the predicted offsets for the position or velocity, the estimated position or velocity is obtained by rearranging the equation 5:

$$\hat{N} = C - \hat{y} \tag{6}$$

where $\hat{N}$ is the estimated position or velocity, C is the current position or velocity and $\hat{y}$ is the predicted position or velocity offset from the ML model.

The loss function used for evaluating the performance of different ML models is the root-mean-square error (RMSE). RMSE is a common metric used in ML and statistics to measure the accuracy of a predictive model due to its ability to quantify the average magnitude of the prediction error. By minimizing the RMSE, we aim to ensure that the predicted positions and velocities correspond with actual positions and velocities, thereby improving the accuracy of the orbit predictions.

The RMSE is given by the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(N_i - \hat{N}_i)^2} \tag{7}$$

where n is the number of observations, N_i represents the true values and $\hat{N}_i$ is the predicted values.

The various ML models implemented are as follows:

*a). **KNR Model***

The KNR model is a versatile and widely used non-parametric algorithm that predicts the dependent variable, given by y based on the average of the nearest neighbors in the input space [41]. The model captures the non-linear relationships by considering the structure of the data. The general formula for the KNR model is given by:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{8}$$

where the target variables are the predicted position and velocity offsets, given by $\hat{y}$. k represents the number of nearest neighbors considered in the regression and y_i are the target variable value or the outputs of the nearest neighbors. From the predicted position and velocity offsets based on the KNR model, using equation 6, the positions and velocities can be estimated.

*b). **RFR Model***

The RFR model is a commonly used model due to its ability to work well with a large and diverse dataset. An ensemble learning method that combines multiple decision trees is used to make predictions. Based on the prediction of the individual trees, the model captures the non-linear relationships in the data. [42] The general formula for the RFR model is given by:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x) \tag{9}$$

where the target variables are the predicted position and velocity offsets, given by $\hat{y}$. T represents the number of decision trees considered in the regression, and $f_t(x)$ are the prediction or outputs of the t^{th} decision tree for the input x. From the predicted position and velocity offsets based on the RFR model, using equation 6, the positions and velocities can be estimated.

*c). **LR Model***

The LR model is one of the most widely used modeling technique due to its simplicity and interpretability [43]. It describes the relationship between a dependent variable, y, and one or more independent variables, X. The general formula for LR model is given by:

$$y = \beta X + \epsilon \tag{10}$$

The input features are the historical positions and velocities of the satellite, given by X and the target variables are the predicted position and velocity offsets, given by y. β represents the coefficients and ϵ is the error term. From the predicted position and velocity offsets based on the LR model, using equation 6, the positions and velocities can be estimated.

*d). **PR Model***

The PR model is a regression algorithm that models the relationship between a dependent variable, y, and one or more independent variables, X as n^{th} degree polynomial [44]. The PR model is implemented as the data is non-linear in nature. The general formula for PR model is given by:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \epsilon \tag{11}$$

The input features are the historical positions and velocities of the satellite, and the target variables are the predicted position and velocity offsets given by y. In the PR model, the input features are transformed into polynomial features of required degree (2, 3, ... n) and then modeled using a linear model. The model then fits a polynomial function to the data, allowing for non-linear relationships to be captured. $\beta_0, \beta_1, \beta_2, ...\beta_n$ represents the coefficients of the polynomial term, and ϵ is the error term. From the predicted position and velocity offsets based on the PR model, using equation 6, the positions and velocities can be estimated.

VI. RESULTS

A comprehensive evaluation of different ML models for estimating the orbits of LEO SWARM-A were conducted and extended the proposed method to MEO GPS satellites. We assess the efficacy of ML models in predicting the satellite orbits, compare the performance of each model, identify their strengths and limitations, and enhance the orbit prediction accuracy. Therefore, this section presents the performance comparison of ML models, evaluation of each ML models implemented, and discussion to interpret the findings and highlight the implications.

1. Performance Comparison of ML Models

As stated in section IV, we define the number of days of data to be used by the ML models such as 1, 3, 10, 30 and 50 days. For clarity and focus, we present and discuss the performance comparison of the ML models that showed the best performance when using a certain number of days of data. In this case, the ML models obtained best and reliable results when using 3-days of data. Therefore, we highlight the performance comparison of ML models using 3-days of data. The performance of the models were assessed using the RMSE as the primary evaluation metric.

Figure 3 shows that PR model demonstrated significant performance overall compared to KNR, RFR and LR model for both position RMSE (m) and velocity RMSE (m/s) of LEO SWARM-A satellite. In position RMSE (left), PR model obtained the lowest RMSE for different prediction intervals, followed by LR model. Since the KNR and RFR models obtained high position RMSE, the LR model position RMSE is not visible in the plot, which is as close as to PR model results. In case of velocity RMSE (right), PR model obtained lowest RMSE compared to the KNR, RFR and LR models. Additionally, the position and velocity RMSE analysis shows that RFR model collectively performed better when compared with KNR model. Overall, PR model exhibited best RMSE, indicating superior performance in estimating the position and velocity of the LEO SWARM-A satellite.

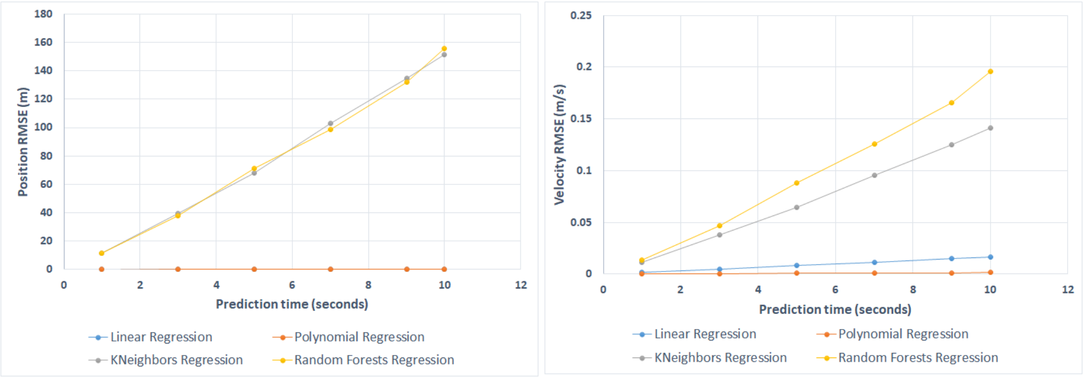

Figure 3: LEO SWARM-A satellite position RMSE (left) and velocity RMSE (right) of implemented ML Models for different prediction time intervals.

Similarly, different ML models were evaluated for estimating the position and velocity of MEO GPS satellites based on the prediction of position and velocity offsets. Figure 4 shows that PR model demonstrated significant performance overall compared to KNR, RFR and LR model for both position RMSE (m) and velocity RMSE (m/s) of GPS satellite PRN-14. Overall, PR model exhibited lowest RMSE, indicating superior performance in estimating the position and velocity of the MEO GPS satellite PRN-14.

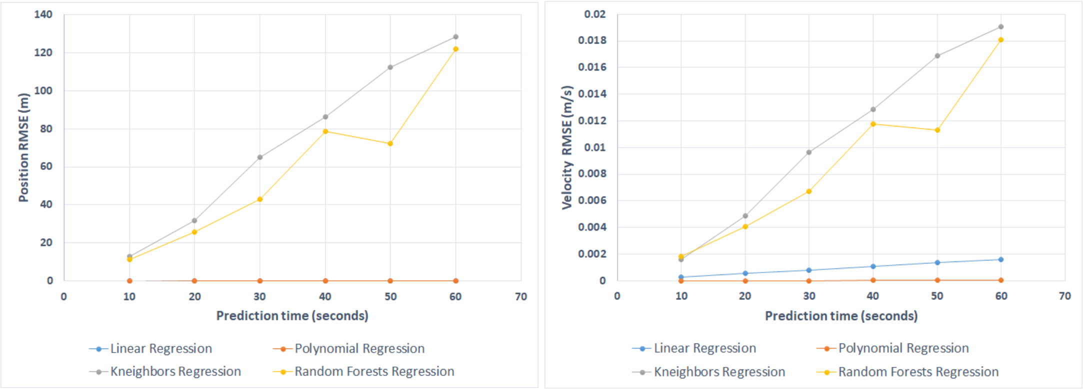

Figure 4: MEO GPS satellite position RMSE (left) and velocity RMSE (right) of implemented ML models for different prediction time intervals.

2. Evaluation of ML Models

In the evaluation of ML models, we present the results of various ML models used in estimating the position and velocity of the LEO SWARM-A satellite. Though we extended our analysis to MEO GPS satellites to evaluate the robustness and versatility of our proposed method, only the best RMSE results obtained by different ML models for MEO GPS satellite is presented as table in the discussion of results section VI.3. Therefore, this section presents the results of various ML models implemented for LEO SWARM-A satellite.

a). KNeighbors Regressor (KNR) Model

KNR model is a regression technique that works by averaging the target values of K-nearest neighbors in the feature space to make prediction. In the KNR model, the total number of neighbors denoted by K, as stated in section V.1 a), can significantly influence the performance of the model. KNR model with K = 2 is considered to be the optimal parameter, yielding to position RMSE of 11.652 m and velocity RMSE of 0.01173 m/s when prediction time is one second and uses three days of data. When K = 1, the model offers better accuracy, but it overfits, has high variance and does not generalize to new datasets. When K is higher, the model was unable to learn the patterns in the data, which resulted in a highly biased model and yielded higher RMSE. Upon cross validation, using K = 2 yielded the best score, indicating that the model predicts the offsets with better accuracy. Therefore, using K = 2 provided an optimal balance between the bias and variance.

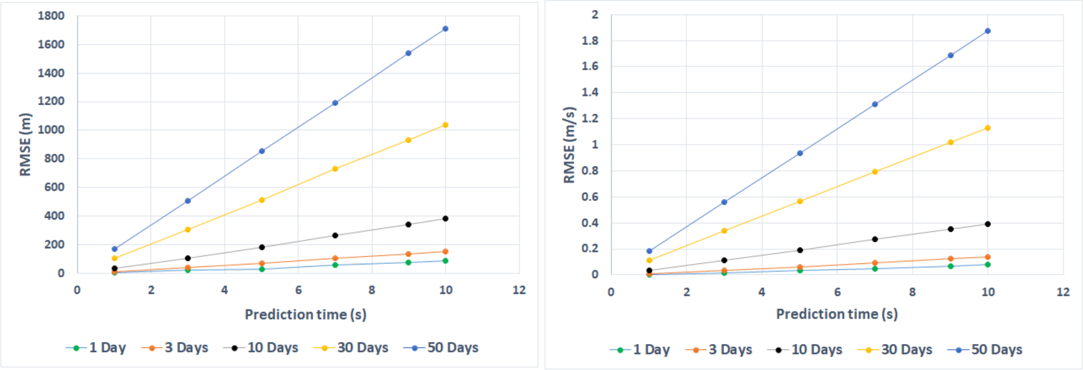

Figure 5: LEO position RMSE (left) and velocity RMSE (right) using KNR Model for different prediction time intervals and according to different days of data used.

Figure 5 shows the position and velocity RMSE obtained by KNR model for different prediction time intervals and according to the number of days of data used. Though best position and velocity RMSE results were achieved using one day of data, the KNR model is considered to be reliable when using three days of data, obtaining a better position and velocity RMSE. RMSE increases when the prediction time increase and amount of data used by the model increase. In general, deep learning models perform better when large amount of data is fed to the model. However, position and velocity RMSE for KNR model increased when more number of days of data were used, as seen in figure 5. Overall, the evaluation of KNR model shows that highest RMSE values are obtained for predicting the position and velocity offsets.

b). Random Forests Regressor (RFR) Model

RFR model is a supervised learning algorithm which operates by building a multitude of decision trees and outputs a mean prediction of the individual trees. In the RFR model, total number of decision trees denoted by T as stated in V.1 b), can significantly influence the accuracy, bias, variance and overall performance of the model. In general, more number of decision trees ($n\_estimators$) will lead to improved accuracy, as the forest averages the prediction of the individual trees. However, at a certain point, with more number of trees, the improvement becomes negligible and computational costs increases. Thus, the RFR model with number of trees, T = 100 is considered to be the optimal number of estimators, yielding to position RMSE of 11.45 m and velocity RMSE of 0.0134 m/s when prediction time is one second and uses three days of data. With few number of trees, i.e., when T < 100, the model was unable to learn the patterns in the data, which lead to high variance and causes the model to overfit. Upon cross validation, when T is higher, i.e., T > 100, the model yielded the best scores. However, the improvement becomes negligible and increased the computational costs. Therefore, T = 100 is considered to be the optimal parameter in predicting the position and velocity offsets via cross validation, providing a balance between the accuracy and computational costs. In addition, feature scaling and hyperparameter tuning are done during the cross-validation to ensure an effective process for optimizing the model performance.

Figure 6 shows the position and velocity RMSE obtained by RFR model for different prediction time intervals and according to the number of days of data used as stated in section IV. Though best position and velocity RMSE results were achieved using one day of data, the RFR model is considered to be more reliable when using three days of data, obtaining a better position and velocity RMSE. Similar to the KNR model, position and velocity RMSE increases when the prediction time and days of data used increase, as seen in figure 6. Overall, the evaluation of RFR model shows that higher RMSE values are achieved in predicting the position and velocity offsets.

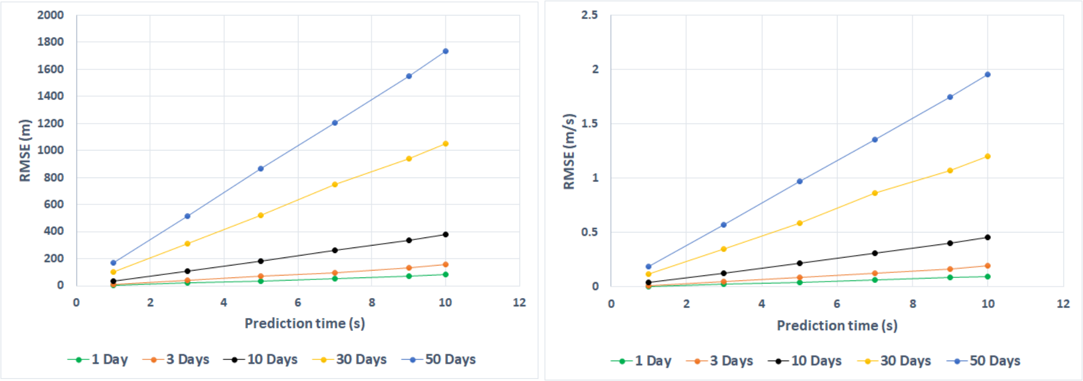

Figure 6: LEO position RMSE (left) and velocity RMSE (right) using RFR Model for different prediction time intervals and according to different days of data used.

c). Linear Regression (LR) Model

LR model is one of the simplest and most interpretable ML model implemented for estimating the position and velocity of the satellites. The simplicity of the LR model lies in its straightforward approach. LR model achieved position RMSE of 0.00077 m and velocity RMSE of 0.00155 m/s for prediction time of one second when using one day of data. Though best position and velocity RMSE results were achieved using one day of data, the LR model is considered to be reliable and robust when using three days of data, obtaining a better position and velocity RMSE. Figure 7 shows the position and velocity RMSE obtained by LR model for different prediction time intervals and according to the number of days of data used. Similar to the KNR and RFR models, position and velocity RMSE increases when the prediction time and days of data used increase. Overall, the evaluation of LR model shows that lower RMSE values are achieved in predicting the position and velocity offsets, significantly better when compared to the KNR and RFR models.

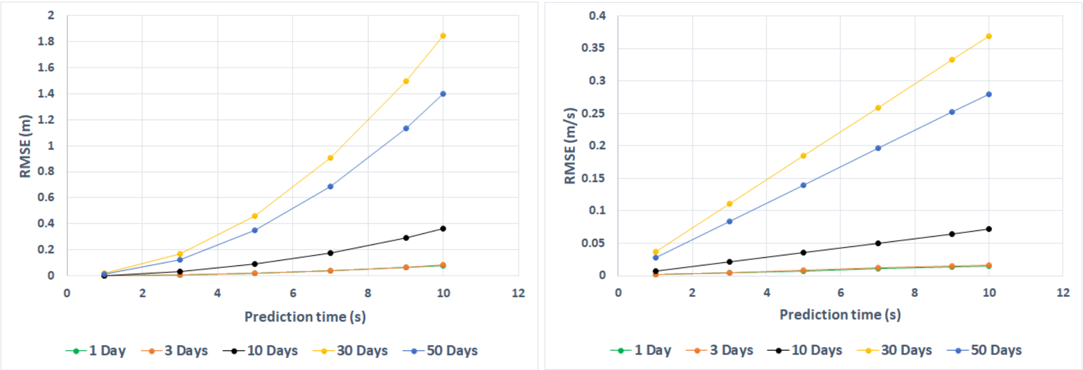

Figure 7: LEO position RMSE (left) and velocity RMSE (right) using LR Model for different prediction time intervals and according to different days of data used.

d). Polynomial Regression (PR) Model

In addition to the KNR, RFR and LR models, we have also implemented the PR model. PR model, by utilizing polynomial features of required degree, the model improved over the LR model by capturing nonlinear relationships in the data. As stated in the section V, PR model with degree 3, that is, third-degree polynomial regression was implemented to predict the position and velocity offsets. PR model performed better when using degree-3 with position RMSE of 0.00007 m and velocity RMSE of 0.00014 m/s for LEO SWARM-A satellite when using three days of data and the prediction time is one second. Higher or lower polynomial degrees resulted in increased RMSE, indicating overfitting or underfitting respectively.

Figure 8 shows the position and velocity RMSE obtained by PR model for different prediction time intervals and according to the number of days of data used. Unlike the other ML models, the best position and velocity RMSE results were not achieved when using one day of data. Instead, PR model achieved best RMSE when using three days of data. This indicates why RMSE results obtained using three days of data were considered to be reliable when best RMSE was obtained using one day of data

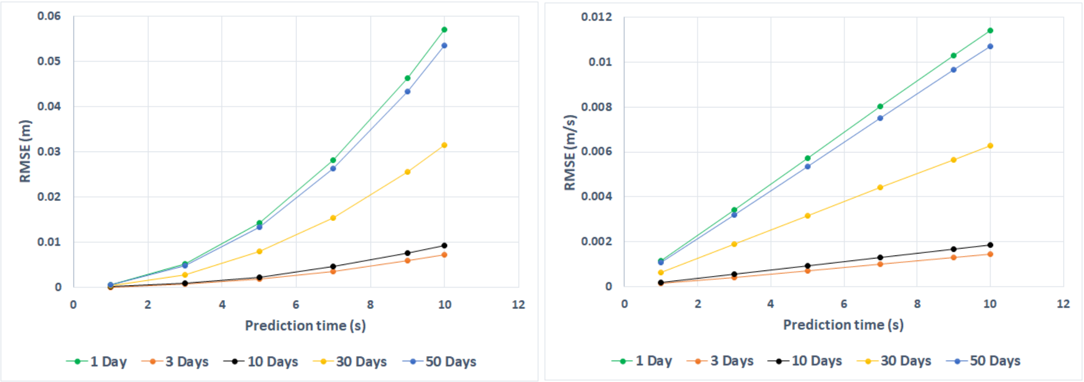

Figure 8: LEO position RMSE (left) and velocity RMSE (right) using PR Model for different prediction time intervals and according to different days of data used.

by other models. In addition, position and velocity RMSE using the PR model increases when prediction time and days of data used increase, similar to other models, as seen in the figure 8. Overall, PR model is more robust, reliable and efficient when using three days of data. The evaluation of PR model shows that lowest RMSE values are achieved in predicting the position and velocity offsets. PR model significantly outperformed all the other ML models such as KNR, RFR, and LR models.

In addition to the RMSE, since PR model outperformed other ML models, the difference plot for position (top) and velocity (bottom) are illustrated in figure 9. The position difference is the difference between the true position values and estimated positions values with respect to time in hours. Similarly, the velocity difference is the difference between the true velocity values and the estimated velocity values. The estimated position and velocity values are obtained using equation 6 based on the predicted position and velocity offsets by the PR model. The x-axis in figure 9 shows the date and hours of the test data. In this case, when using three days of data, first 70% of the data were used as the training data and last 30% of the data were used as test data. Therefore, the x-axis illustrates that test data from 04/10/2023 03:00 hour to 05/10/2023 00:00 hour were used in estimating the position and velocity values and their difference to actual values are plotted.

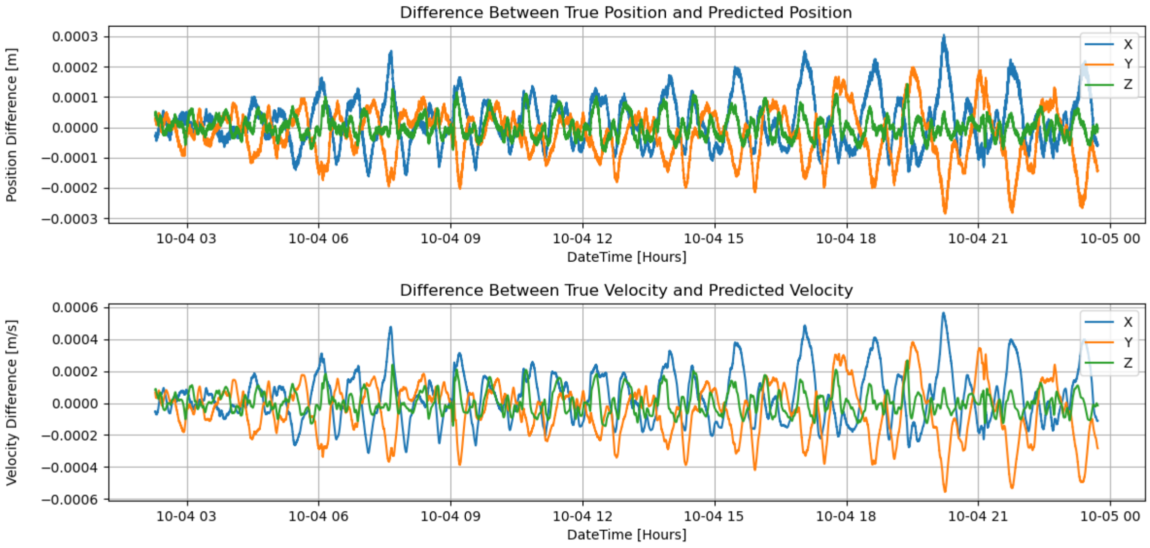

Figure 9: Differences of the true positions and velocities to that of estimated positions and velocities for SWARM-A satellite test data points.

3. Discussion of Results

From the results, a comprehensive evaluation of different ML models for estimating the position and velocity of LEO SWARM-A satellite and GPS satellites, based on the prediction of the offsets, reveal distinct and significant performance in terms of accuracy, measured using RMSE. The performance of KNR model can be significantly influenced by the total number of neighbors, given by K. An optimal K value is determined by using cross validation for the optimized performance of KNR model. The analysis shows that $K = 2$ provided an optimal balance between the bias and variance. KNR model yielded higher RMSE overall compared to the other ML models despite optimizing the model performance using cross validation. The implementation of RFR model performed slightly better than KNR model. The performance of the RFR model can be influenced by the T value, i.e., the total number of trees. The best optimal T value is determined using a grid search with cross validation. In addition, feature scaling and hyperparameter tuning were done to optimize the performance of the model. The analysis shows that $T = 100$ provided an optimal balance, yielding to better RMSE. However, the RMSE values were higher than LR and PR models. LR model performed better for estimating the position and velocity of the LEO SWARM-A and GPS satellites, as evidenced from the RMSE provided in Table 1. A significantly lower RMSE was obtained when compared to the RFR and KNR models. In PR model, a third degree polynomial includes the original input features as well as their squared terms and cubic terms. The analysis shows that best RMSE results are obtained when using a third degree polynomial for LEO SWARM-A satellite. In case of MEO GPS satellite, second degree polynomial yielded the best RMSE results. The model demonstrated an outstanding performance compared to the other ML models due to incorporating higher-order terms, thereby effectively capturing the non-linear dynamics present in the satellite orbits.

Overall, PR model introduces adaptability by allowing the model to fit the data more accurately compared to LR model. The adaptability of the model leads to significant outperformance with lower RMSE when the underlying relationship between variables is non-linear. Though the RFR model is a powerful algorithm for capturing non-linear relationships, the model may find it difficult to capture certain non-linear relationships that a polynomial model capture directly. The non-linearity of the satellite orbit data used in this study might be better approximated without overfitting by a third degree polynomial than by an ensemble RFR model. On the other hand, KNR model is a local approximation method, making predictions on the nearest data points. While PR model uses a global approximation, fitting a polynomial equation to the entire dataset leading to better generalization for non-linear trends. In conclusion, for both LEO and MEO satellites, PR model outperformed other ML models such as KNR, RFR and LR models due to its ability to effectively model the non-linear characteristics in the satellite data with an optimal degree of complexity as determined by cross validation.

The study provides a comprehensive analysis of different ML models used in estimating the position and velocity of the satellites. The research focused on providing a simplified and alternative approach based on standalone ML-based method compared to traditional or hybrid methods. The models were trained and tested using real-world data, ensuring that the results are applicable to actual satellites. The study implemented ML models, considering also the computational costs and robustness of the models. Thus, it provides a balanced assessment of the trade-offs associated with selecting a model by evaluating both accuracy and computational costs. On the other hand, the study did not extensively address the computational costs involved in implementing ML models such as RFR and KNR models. The performance of the ML models might vary significantly when applied to different satellites in the LEO. The study mainly implemented ML models for short-term predictions of few seconds, which may be a limitation with respect to MEO as it requires accurate long-term predictions to ensure the reliability of PNT services. However, short term predictions are crucial in the context of LEO satellites.

In this study, different ML models were analyzed for different number of days of data (1, 3, 10, 30 and 50 days) and prediction time (1, 3, 5, 7, 9 and 10 seconds for LEO; 10, 20, 30, 40, 50 and 60 seconds for GPS) and evaluated using RMSE. The table 1 summarizes the performance of ML models which obtained best RMSE results.

Table 1: Best RMSE results obtained by different ML models for LEO SWARM-A and MEO GPS Satellite PRN-14.

LEO SWARM-A Satellite				
Number of Days Used	**ML Method**	**Prediction Time (s)**	**Position_RMSE (m)**	**Velocity_RMSE (m/s)**
3	KNeighbors Regression	1	11.65206606	0.011731784
3	Random Forests Regression	1	11.45781611	0.013498317
3	Linear Regression	1	0.000841591	0.001683116
3	Polynomial Regression	1	7.23E-05	0.000144123

MEO GPS Satellite PRN-14				
Number of Days Used	**ML Method**	**Prediction Time (s)**	**Position_RMSE (m)**	**Velocity_RMSE (m/s)**
3	KNeighbors Regression	10	12.9881978	0.001578546
3	Random Forests Regression	10	11.48159881	0.001818075
3	Linear Regression	10	0.001323894	0.000265484
3	Polynomial Regression	10	9.71E-05	3.35E-06

VII. CONCLUSION AND FUTURE WORK

This study aimed to evaluate the feasibility and effectiveness of using various machine learning models to predict the orbits of low Earth orbit (LEO) satellites, with a focus on improving prediction accuracy as measured by RMSE. Additionally, we extended our analysis to medium Earth orbit (MEO) GPS satellites to identify any potential differences in prediction accuracy between LEO and MEO satellites.

The study utilized different ML models, such as KNeighbors regressor (KNR), Random Forests regressor (RFR), linear regression (LR) and polynomial regression (PR). Despite the availability of more complex models, such as KNR and RFR, the polynomial regression model outperformed all the other ML models implemented, highlighting the significance of model selection for a certain application. This is evident from the primary evaluation metric, RMSE, used in assessing the performance of the ML models. Overall, a significantly lower RMSE was obtained by using PR, followed by LR, while higher RMSE were obtained using RFR and KNR models.

Furthermore, the estimation of position and velocity of the LEO SWARM-A and MEO GPS satellite was carried out based on the offset predictions. The ML models predicted the offsets for different time intervals such as 1, 3, 5, 7, 9 and 10 seconds in case of LEO satellite, while for MEO satellites, the prediction time intervals are 10, 20, 30, 40, 50 and 60 seconds. The study, therefore, highlights the critical importance of ML-based precise short-term predictions in LEO satellites, due to the satellites orbiting at high speed and dynamic space environment.

In future work, we will explore more advanced machine learning models for long-term orbit predictions and further enhance the orbit prediction accuracy. Additionally, investigation on the influence of other features such as atmospheric drag, solar position and satellite acceleration on the model performance will be assessed. In summary, this study contributes to the significance of standalone ML-based technique in estimating the position and velocity of especially the LEO satellite at millimeter levels, ultimately enhancing the performance and efficiency of space-based applications and services.

REFERENCES

[1] K. Selvan, A. Siemuri, F. S. Prol, P. Välisuo, M. Z. H. Bhuiyan, and H. Kuusniemi, “Precise orbit determination of LEO satellites: a systematic review,” *GPS Solutions*, vol. 27, no. 4, p. 178, 2023.

[2] O. Montenbruck, *Space Applications*, P. J. Teunissen and O. Montenbruck, Eds. Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-42928-1_32

[3] T. Springer and U. Hugentobler, “IGS ultra rapid products for (near-) real-time applications,” *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, vol. 26, no. 6, pp. 623–628, 2001, proceedings of the First COST Action 716 Workshop Towards Operational GPS Meteorology and the Second Network Workshop of the International GPS Service (IGS). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1464189501001119

[4] T. Hadas and J. Bosy, “IGS RTS precise orbits and clocks verification and quality degradation over time,” *GPS Solutions*, vol. 19, no. 1, pp. 93–105, 2015. [Online]. Available: https://doi.org/10.1007/s10291-014-0369-5

[5] R. Flores, B. M. Burhani, and E. Fantino, “A method for accurate and efficient propagation of satellite orbits: A case study for a molniya orbit,” *Alexandria Engineering Journal*, vol. 60, pp. 2661–2676, 2021.

[6] A. Nowak, R. Zajdel, and K. Sośnica, “Optimization of orbit prediction strategies for GNSS satellites,” *Acta Astronautica*, vol. 209, pp. 132–145, 2023.

[7] O. Montenbruck, “Numerical integration methods for orbital motion,” *Celestial Mechanics and Dynamical Astronomy*, pp. 59–69, 1992. [Online]. Available: https://doi.org/10.1007/BF00049361

[8] F. R. Hoots, P. W. Schumacher Jr., and R. A. Glover, “History of analytical orbit modeling in the U.S. space surveillance system,” *Journal of Guidance, Control, and Dynamics*, 2004. [Online]. Available: https://doi.org/10.2514/1.9161

[9] J. F. San-Juan, I. Pérez, M. San-Martín, and E. P. Vergara, “Hybrid SGP4 orbit propagator,” *Acta Astronautica*, vol. 137, pp. 254–260, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576516311535

[10] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Inc., 2004. [Online]. Available: https://www.sciencedirect.com/book/9780126836301/statistical-orbit-determination#book-info

[11] J. Haidar-Ahmad, N. Khairallah, and Z. M. Kassas, “A hybrid analytical-machine learning approach for LEO satellite orbit prediction,” *25th International Conference on Information Fusion (FUSION), Linköping, Sweden*, pp. 1–7, 2022.

[12] D. A. Vallado and P. Crawford, “SGP4 orbit determination,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii*, 2008.

[13] C. Levit and W. Marshall, “Improved orbit predictions using two-line elements,” *Advances in Space Research*, vol. 47, pp. 1107–1115, 2011.

[14] H. Peng and X. Bai, “Improving orbit prediction accuracy through supervised machine learning,” *Advances in Space Research*, 2018. [Online]. Available: https://arxiv.org/pdf/1801.04856.pdf

[15] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. AMLBook, 2012.

[16] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997. [Online]. Available: http://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf

[17] C. Ampatzis and D. Izzo, “Machine learning techniques for approximation of objective functions in trajectory optimisation,” in *Proceedings of the IJCAI-09 workshop on artificial intelligence in space*, 2009, pp. 1–6.

[18] J. Hartikainen, M. Seppanen, and S. Sarkka, “State-space inference for non-linear latent force models with application to satellite orbit prediction,” in *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012.

[19] S. Sharma and J. W. Cutler, “Robust orbit determination and classification: A learning theoretic approach,” *IPN Progress Report*, vol. 42, p. 203, 2015.

[20] F. S. Prol, M. Z. H. Bhuiyan, S. Kaasalainen, E. S. Lohan, J. Praks, K. Çelikbilek, and H. Kuusniemi, “Simulations of dedicated leo-pnt systems for precise point positioning: Methodology, parameter analysis, and accuracy evaluation,” *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–19, 2024.

[21] B. Wang, H. Peng, and K. Li, “Precision analysis of an analytical method in space debris orbit prediction,” *International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2016)*, pp. 238–242, 2016. [Online]. Available: https://doi.org/10.2991/iccia-16.2016.43

[22] A. R. Muldoon, G. H. Elkaim, and I. Rickard. (2009) Improved orbital debris trajectory estimation based on sequential TLE processing. [Online]. Available: https://api.semanticscholar.org/CorpusID:8407890

[23] Q. Wang, C. Hu, T. Xu, G. Chang, and A. H. Moraleda, “Impacts of earth rotation parameters on GNSS ultra-rapid orbit prediction: Derivation and real-time correction,” *Advances in Space Research*, vol. 60, pp. 2855–2870, 2017.

[24] C. Lei, B. Xian-Zong, L. Yan-Gang, and L. Ke-Bo, “Orbital error analysis based on historical data,” in *Orbital Data Applications for Space Objects*, 2016, pp. 77–105.

[25] J. Sang, B. Li, J. Chen, P. Zhang, and J. Ning, “Analytical representations of precise orbit predictions for earth-orbiting space objects,” *Advances in Space Research*, vol. 59, pp. 698–714, 2017.

[26] G. Jadala, G. N. Meedinti, and R. Delhibabu, “Satellite orbit prediction using a machine learning approach,” *Workshops at the 5th International Conference on Applied Informatics*, pp. 28–46, 2022.

[27] Z. Y.-C. Liu, S. Tarlow, M. Akbar, Q. Donnellan, and D. Senkow, “Improved orbital propagator integrated with SGP4 and machine learning,” *Small Satellite Conference, Utah*, 2021.

[28] J. Gou, C. Rösch, E. Shehaj, K. Chen, M. Kiani Shahvandi, B. Soja, and M. Rothacher, “Improving the accuracy of GNSS orbit predictions using machine learning approaches,” *EGU General Assembly 2022, Vienna, Austria*, 2022. [Online]. Available: https://doi.org/10.5194/egusphere-egu22-1834

[29] H. Peng and X. Bai, “Machine learning approach to improve satellite orbit prediction accuracy using publicly available data,” in *J Astronaut Sci*, vol. 67, 2020, p. 762–793.

[30] H. Peng and X. Bai, “Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy,” *Astrodynamics*, vol. 3, no. 4, p. 325–343, 2019. [Online]. Available: https://doi.org/10.1007/s42064-018-0055-4

[31] H. Peng and X. Bai, “Improving orbit prediction accuracy through supervised machine learning,” *Advances in Space Research*, vol. 61, pp. 2628–2646, 2018.

[32] I. E. Dawoodjee and M. Rajeswari, “Establishing a regression baseline for predicting satellite motion,” *Journal of Applied Technology and Innovation*, vol. 5, pp. 47–58, 2021.

[33] H. Peng and X. Bai, “Artificial neural network–based machine learning approach to improve orbit prediction accuracy,” *Journal of Spacecraft and Rockets*, vol. 55, pp. 1248–1260, 2018.

[34] F. S. Prol, R. M. Ferre, Z. Saleem, P. Välisuo, C. Pinell, E. S. Lohan, M. Elsanhoury, M. Elmusrati, S. Islam, K. Çelikbilek, K. Selvan, J. Yliaho, K. Rutledge, A. Ojala, L. Ferranti, J. Praks, M. Z. H. Bhuiyan, S. Kaasalainen, and H. Kuusniemi,

"Position, navigation, and timing (PNT) through low earth orbit (LEO) satellites: A survey on current status, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 83 971–84 002, 2022.

[35] T. Mortlock and Z. M. Kassas, "Assessing machine learning for leo satellite orbit determination in simultaneous tracking and navigation," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–8.

[36] S. E. Kozhaya, J. A. Haidar-Ahmad, A. A. Abdallah, Z. M. Kassas, and S. S. Saab, "Comparison of neural network architectures for simultaneous tracking and navigation with LEO satellites," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri*, 2021, pp. 2507–2520.

[37] ESA. (2024) European Space Agency, SWARM Data Access. [Online]. Available: https://swarm-diss.eo.esa.int/#

[38] NGA Public Affairs. (2023) National Geospatial-Intelligence Agency, Office of Geomatics. [Online]. Available: https://earth-info.nga.mil/index.php?action=home

[39] H. Shou, "Orbit propagation and determination of low earth orbit satellites," *International Journal of Antennas and Propagation*, vol. 2014, 2014. [Online]. Available: https://doi.org/10.1155/2014/903026

[40] E. Friis-Christensen, H. Lühr, and G. Hulot, "Swarm: A constellation to study the earth's magnetic field," *Earth, Planets and Space*, vol. 58, pp. 351–358, 2006. [Online]. Available: https://doi.org/10.1186/BF03351933

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] L. Breiman, "Random forests," *Springer Link*, vol. 45, pp. 5–32, 2001.

[43] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *Linear Regression*. Cham: Springer International Publishing, 2023, pp. 69–134. [Online]. Available: https://doi.org/10.1007/978-3-031-38747-0_3

[44] E. Ostertagová, "Modelling using polynomial regression," *Procedia Engineering*, vol. 48, pp. 500–506, 2012, modelling of Mechanical and Mechatronics Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705812046085