Establishing a regression baseline for predicting satellite motion

Ismail Esack Dawoodjee
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp054033@mail.apu.edu.my

Mandava Rajeswari
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
prof.dr.mandava@staffemail.apu.edu.my

***Abstract*—Determining the orbital paths of space objects is a critical task in astronomy. In particular, knowledge of satellite trajectories is essential to avoid costly and hazardous collisions between satellites in space. However, due to the amount and complexity of variables affecting a satellite's orbit, it is no small feat to accurately predict its position. Moreover, it was only recently that novel alternatives to physics-based models have been proposed, namely machine learning (ML) models that can learn from historical data and make improvements to orbit prediction accuracy. Motivated by the hope that ML models can capture the underlying pattern of satellite orbital trajectories, the goal of this paper is to apply a supervised ML model called non-linear regression, to predict the position and velocity of a single satellite in orbit around the Earth. The study establishes a simple non-linear regression baseline for predicting satellite motion three days in advance, from which more complex ML models can be applied. Obtained forecasts were within acceptable error margins and the overall result shows promise in applying ML to predict satellite motion.**

***Keywords*—satellite, resident space object, orbit propagation, supervised machine learning, non-linear regression**

I. Introduction

During the past two decades, the number of Resident Space Objects (RSOs)[1] has nearly doubled, from around 11,000 objects in the year 2000 to around 19,500 objects in 2019. This number is expected to rise even higher as more satellites are put into space, thanks to improvements in satellite technology and lower costs of production. On the other hand, the increase in the number of RSOs also indirectly increases the risk of collision between them [1]. The awareness and identification of risky situations such as these are referred to as Space Situational Awareness (SSA) [2]. More specifically, SSA refers to "the ability to view, understand and predict the physical location of natural and man-made objects in orbit around the Earth, with the objective of avoiding collisions" [2, p. 23]. An important issue in SSA is the reliable and accurate orbit determination (or orbit tracking) of satellites over long periods of time. Failure to address this issue has led to incidents such as the collision between the active Iridium-33 US communication satellite and the inactive Kosmos-2251 Russian communication satellite in February 2009 [1]. In fact, this accident increased the amount of space debris by 13%, as shown in Fig. 1.

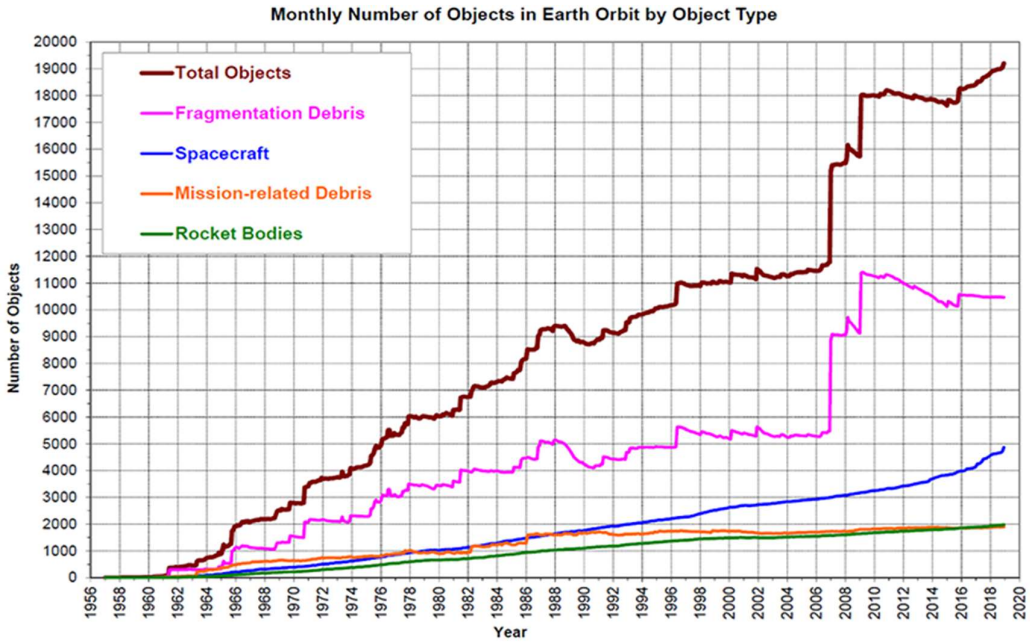

Fig. 1. Number of resident space objects in Low Earth Orbit [3]

[1] The term "resident space object" is most commonly used for referring to artificial objects that are in orbit around the Earth. Please refer to Section VII: Appendix for a complete list of all the abbreviations used in this paper.

More accidents will result in more debris being produced, and through a chain reaction of collisions (if left unchecked), may lead to a dire situation in which it becomes difficult or downright impossible to put a satellite into orbit due to the large accumulation of space debris surrounding the Earth. This scenario is known as the Kessler Syndrome [4]. Thus, considering the gravity of the situation at hand, it is imperative to prevent such catastrophic collisions from ever happening again.

The demand for accurate orbit prediction methods is on the rise, as more objects are put into space. Section I introduced the reasons on why this is a concerning state of affairs. Section II explains why current prediction models are lacking in the timeliness, accuracy, and practicality of predicting orbital paths, and why alternative methods such as machine learning (ML) techniques should be considered. Additionally, the research question that will be addressed in this paper is discussed. Section III introduces the related works in the literature involving both ML and non-ML techniques while assessing their strengths and limitations as well. Section IV presents the data sets, methodology, metrics and the data preparation and exploration phases. Section V presents the obtained results and a short discussion on them. Section VI concludes the paper with a summary and directions for further research.

II. BACKGROUND

A. The Need for Alternative Prediction Methods

The main cause of the Iridium-Kosmos incident[2] was attributed to the limited capability of orbit determination techniques at that time, together with the inherent uncertainty of data used in these techniques [1]. Likewise, most of today's orbit prediction methods are physics-based, which still possess the flaws of methods from 2009, albeit at a less severe degree. These methods require good information on the initial conditions of the satellite at the start of calculating its trajectory, as well as the space environment around it [5]. Environmental conditions such as solar radiation pressure, which is the force of extremely fast subatomic particles from the Sun hitting a satellite's surface, is unnoticeable over short time intervals but adds up to a noticeable effect on the satellite's orbit over long periods of time. This variable is difficult to estimate due to the Sun's constantly evolving surface and due to a lack of information about the satellite's size, mass and geometrical proportions, captured within a parameter called the area-to-mass ratio [6].

Similar difficulties are faced when considering the effect of Earth's atmospheric drag, which is the resistance of air against a satellite's movement when it moves around the Earth at a low enough orbit. Calculating this parameter requires knowledge of the air density at a given position of the satellite, how fast the satellite is moving and how much of its surface area is exposed to the air moving against it. Moreover, information about the change in altitude and state of the satellite when it is being maneuvered by operators from other countries may not always be readily available for use when orbit predictions are needed [5].

The list of variables to be estimated and updated are many and complicated, while the methods used for estimating and updating them are few and even more complex. Current orbit determination methods, especially physics-based ones, fall into one or more of the following shortcomings:

- They do not consider all of the variables affecting a satellite's orbit due to unattainable or inherently uncertain data. For example, the area-to-mass ratio and maneuver information.
- They require expensive monitoring tools with limited resources (such as ground-based observatories) or time-consuming models requiring impractical computing costs [5].
- They may not be generalizable to other types of orbits. For example, a model requiring atmospheric drag data to track Low Earth Orbit (LEO) satellites may not always be applicable to tracking High Earth Orbit (HEO) satellites, where the effect of Earth's atmosphere is effectively non-existent.

Given the limitations described above, errors in physics-based models may be too high to be useful for practical situations, namely collision avoidance and task scheduling (for example, assigning a weather satellite to watch over a particular patch of the Earth). As an alternative to these methods, a different approach to orbit prediction based on machine learning techniques was proposed in the literature.

B. Machine Learning for Space Situational Awareness

Machine Learning (ML) has made it possible to automate repetitive tasks in the scientific, financial and technical industries. Implementing ML and sophisticated artificial intelligence (AI) technologies can extract information and recognize patterns in data that usually require a considerable amount of time for humans to discover [7]. In addition, the volume and velocity of data can hinder humans from being able to process the data within the time frames necessary for practical decision-making. In the context of Space Situational Awareness (SSA), mundane jobs such as predicting orbital paths, assigning scheduled tasks and evaluating collision risks are best suited for machine learning, with operators stepping in when human intervention is required [7]. Hence, the ML technique provides a new framework for improving the capabilities of existing physics-based methods in carrying out these jobs.

In stark contrast to physics-based methods for predicting an RSO's orbit, ML techniques do not require explicit modelling to be done for the RSO's structure (such as its shape and area-to-mass ratio) and the local space environment around it, nor does it need information about the RSO's maneuvers [5]. Rather, the models are learned from large amounts of historical data, which is in some ways similar to how humans learn to predict future events through experiences from their past [5]. The three most common types of machine learning are supervised learning, unsupervised learning, and reinforcement learning. Reinforcement learning is used for making optimal decisions. Unsupervised learning is used for recognizing patterns and structures without having to provide labels for the input data. Supervised learning, however, learns a function that maps labelled input data to labelled output data based on example training data [5].

[2] Funnily enough, the irony of this incident was that both of them were communication satellites.

Due to the accessibility of labelled historical data concerning the satellites' previous positions and velocities, the supervised machine learning method is deemed to be most the appropriate approach [5]. Among the various supervised ML techniques available, the Non-Linear Regression (NLR) method will be used for the specific problem of improving orbit prediction accuracy for satellites. For a complete description of the NLR method, its mathematical formulation in one dimension, the reasons for choosing it to address this problem, as well as the justifications for setting up the dataset structure and error metrics (determined through an initial exploratory data analysis stage), please refer to Section IV: Methodology.

The essence of the ML technique for improving the error in orbit prediction is shown in Fig. 2 below. At time t_i, the ground station (indicated by the small orange dish on the Earth's surface) observes the RSO (indicated by the blue-and-yellow satellite) and estimates its position. This estimated state will undoubtedly have measurement errors associated with it. Because of these measurement errors and because of the assumed physics-based model, the propagated orbit prediction will deviate further from the true orbit at a later time, as shown in the predicted state at the time $t_j > t_i$. However, the ML technique modifies the predicted state, so that the ML-modified state is learned to be closer to the true state of the RSO [5].

C. *Research Hypothesis, Goal, and Scope*

At the very heart of the ML approach described above is the **hypothesis** that even though the scientist may not have all

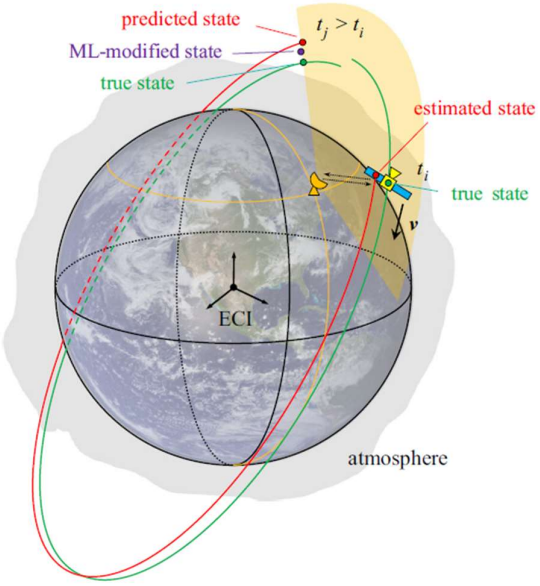

Fig. 2. ML approach for improving orbit prediction accuracy [5]

of the knowledge necessary for a physics-based forecast, some of this information is indirectly contained within the RSO's historical data [8]. This includes estimated states, measurement data and prediction errors for the RSO's particulars and space environment [8].

Given the above hypothesis, the **goal** of this paper is to apply a supervised ML technique called Non-Linear Regression (NLR) to predict, over a period of three days, the position and velocity components of a single satellite using simulated data obtained from a physics-based model.

The data was provided by the Russian Astronomical Science Centre for the first stage of the International Data Analysis Olympiad (IDAO) in January 2020 [9].

Due to the broad field of study in orbit determination, the **scope** of this research will be limited to only one outcome, that of predicting the target variables (positions and velocities) specified as per the instructions given to IDAO participants. Other measurements such as the area-to-mass ratio, which can also be inferred using ML techniques, will not be done. Moreover, only one ML technique will be applied in this paper, i.e. the NLR model. Finally, only the official datasets given to IDAO participants will be used for applying the NLR model. No external data shall be used.

III. Related Work

This section first presents the common themes found in the literature concerning orbit prediction, such as the techniques used and the types of orbits that they are applied to. Also presented are the data types and sources used for making these predictions, as well as the metrics used in evaluating orbit propagation models. Finally, a concise synthesis and analysis of the presented works were done.

A. *Orbit Propagation Techniques*

Until a few years ago, most orbit determination procedures were done using non-ML-based methods. These methods include the commonly used models based on the natural laws of physics, mathematical expressions such as Polynomial Chaos Expansion and Gaussian Processes [10], and analytical and semi-analytical solutions for describing orbital motion [11]. The physics-based methods were highly dependent on initial conditions, which may not always be available or reliable (asX demonstrated by the Iridium-Kosmos collision). Moreover, the mathematical expressions were inconsistent in modelling RSO behavior under higher-order effects (called "perturbations" in the scientific literature) such as atmospheric drag, solar radiation pressure and the gravitational pull from the Sun, Moon and other planets. As improvements to the mathematical expressions, analytical and semi-analytical solutions were explored to take into consideration the quantitatively small but important perturbations required for an accurate orbit prediction [1].

The supervised machine learning approach, on the other hand, does not require explicit models of the RSO and its space environment to infer initial conditions and propagate its orbit to a predicted position in the future. Rather, it has a basis on recognizing the inherent patterns within the observed historical data of the RSO and learning the mapping function between independent and dependent variables.

Moreover, there are also other data-driven approaches such as artificial neural networks [12], reinforcement learning [13], and artificial intelligence, in which the latter not only

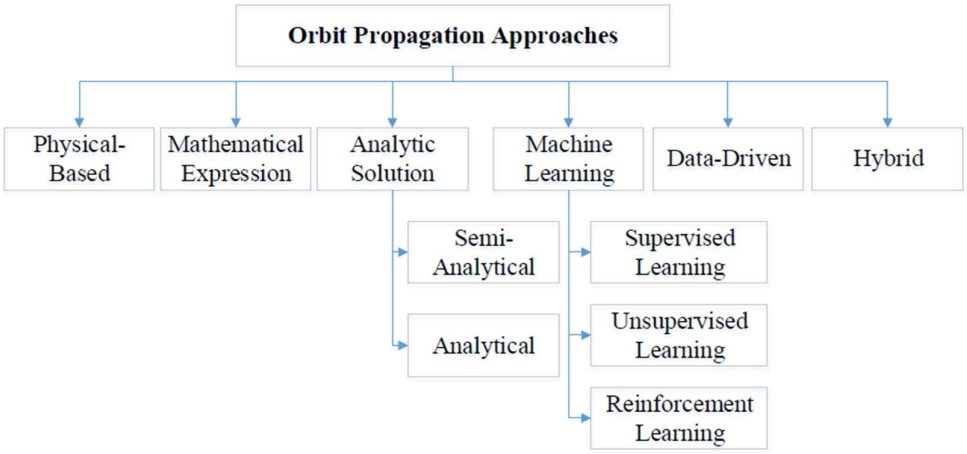

Fig. 3. A summary of orbit propagation techniques in the literature [1]

predicts orbital paths but also assists in the associated decision-making process [1].

Finally, the latest state-of-the-art techniques are called hybrid propagation methods. This method combines a classical propagation method (e.g. a semi-analytical method) with either a statistical time series forecasting model or an ML technique [14, 1]. A summary of all the orbit propagation techniques is given in Fig. 3.

B. Types of Satellite Orbit

Depending on their purpose and missions in space, satellites are put into different altitudes of orbit around the Earth. These orbit types are classified according to their relative distance from the Earth's surface: Low Earth Orbit (LEO) for altitudes below 2000 km, Medium Earth Orbit (MEO) for altitudes below 35786 km, Geosynchronous Orbit (GSO) at an approximate altitude of 35786 km, and High Earth Orbit (HEO) for altitudes above 35786 km [15]. Other types of orbit also exist, such as Sun Synchronous Orbits and GPS orbits. A summary of common RSO orbit types is shown in Table I on the next page.

Due to the commercial, civil, meteorological, military and scientific needs of different nations, most satellites are put into orbits at or below the geosynchronous altitude, in the LEO and MEO regions [14]. Unfortunately, it is also in these regions that the accumulation of space debris is the highest, making it one of the main dangers for functional satellites [16]. Thus, accurate RSO tracking not only includes determining satellite orbits but also the orbits of dangerous space debris as well.

The differences between orbit types are significant enough that there is no single master technique that can account for all the particular characteristics of each type of orbit. For example, a common physics-based model called SGP4 (Simplified General Perturbations-4) was made for tracking satellites at Low Earth Orbit, in which atmospheric drag is important to consider but solar radiation pressure is less important [2]. This is because LEO satellites travel fast, around 7-8 km/s, so the effect of the Sun's radiation (which occurs only over certain paths during the orbit) is eclipsed by the effect of the Earth's atmosphere (which exists throughout the entire orbit). On the contrary, MEO and HEO satellites travel much slower and at a much higher altitude than LEO satellites, so atmospheric drag becomes less important while solar radiation becomes more important. Hence, SGP4 becomes inaccurate and other models need to be used.

For cases such as the Sun Synchronous Orbit, in which a satellite is arranged to face the Sun at all times throughout its orbit, both atmospheric drag and solar radiation pressure become highly important variables. As they are mainly put into LEO regions (around 600-800 km altitudes), SSOs are popular for imaging, spying, meteorological and scientific uses due to having constant sunlight for its solar panels and more importantly for having constant illumination on the Earth's surface below them. Due to their purpose at low orbits, predicting SSO paths inherently require both solar radiation data as well as atmospheric drag data, which means more complex models need to be used.

Therefore, when applying ML models for predicting RSO trajectories, it is important to specify the type of orbit an RSO is in. Based on the orbit type, more or fewer features can be specified as input data for the ML model. For example, Peng and Bai [20] used additional drag coefficient data to develop their Support Vector Regression (SVR) approach to improving the orbit prediction accuracy of LEO, SSO and MEO satellites. The ultimate goal is that powerful data-driven (or otherwise) techniques will be developed that are generalizable and can be applied to all types of orbit, given the appropriate input data.

C. Data Types and Sources

The two main sources of publicly available data are the TLE (Two-Line Element, so-called because of its original encoding format in two punch cards) data from Kelso [17] and ILRS (International Laser Ranging Service) data from NASA [18]. To put it simply, the ILRS ground-based stations fire laser pulses at satellites, which gets reflected back to the station. By measuring the two-way time of flight of the laser pulse, the distance from the station to the satellite can be calculated [18].

The TLE data format, on the other hand, uses physics-based SGP4 computational models to simulate the state (position and velocity) of space objects. The CelesTrak website at Kelso [17] provides data for space objects that are larger than 10 cm in diameter, which is useful for tracking

Table I. Summary of common RSO orbit types

Name	Orbit	Description
LEO	Low Earth	Orbit altitudes below 2000 km
MEO	Medium Earth	Orbit altitudes below 35786 km
GSO	Geosynchronous	Orbit altitude at 35786 km; orbital period of 24 hours
GEO	Geosynchronous Equatorial	A GSO placed directly above the Earth's equator
GPS	Global Positioning System	Orbit altitude at 20200 km; orbital period of 12 hours
SSO	Sun Synchronous	An LEO satellite that faces the Sun during its orbit
HEO	High Earth	Orbit altitudes above 35786 km

space debris [10]. In addition, novel concepts for tracking space debris were also put forward, such as [16]. This concept uses an LEO satellite equipped with a telescope that visually observes space debris moving within the GSO region.

Their method is superior to ground-based telescopes because adverse weather conditions do not affect data collection [16]. While the data from ILRS is precise and more closely represents the "true state" of RSOs, ground-based stations are expensive, and data can only be collected when the RSO passes across its hemisphere of vision. In other words, the station cannot send laser pulses to a satellite while it is travelling across the opposite half of the Earth. An operational satellite may also contain its own GPS tracker which provides much more accurate data than TLE catalogues. However, the disadvantage is that this data belongs to the owner/operator of the satellite only and is not publicly available for use. As a result, Peng and Bai [10] used public ILRS data as the true state of the investigated satellites and the TLE data as the less accurate, estimated state.

D. Metrics and Evaluation

For the purpose of evaluation, papers in the orbit determination literature used a variety of metrics. Some of the pioneers in applying ML techniques such as Peng and Bai [10] exclusively prefer using a single performance metric that they introduced in their previous work [5], while others like [19] used the common metrics MSE (Mean Squared Error), MAE (Mean Absolute Error) and APE (Absolute Percentage Error), and [17] used the AUC (Area Under the Curve) metric with ROC (Receiver Operating Characteristic) curves to compare the performance of several ML techniques. Reference [6] also used RMSE (Root Mean Squared Error) to measure the performance of their random forest model in recovering the area-to-mass ratio of space objects. Most papers that used physics-based methods also employed RMSE and other common metrics.

A particularly intuitive and easy to interpret metric would be the unnamed performance metric introduced by Peng and Bai in [5]. The authors used the concept of the MAPE metric (Mean Absolute Percentage Error), common in forecasting applications, to define this performance metric as follows:

$$P_{ML} = 100\% \frac{\sum_{i=1}^{n} |e_{res}|}{\sum_{i=1}^{n} |e_T|} = 100\% \frac{\sum_{i=1}^{n} |e_T - \hat{e}_{ML}|}{\sum_{i=1}^{n} |e_T|} \quad (1)$$

The performance of an ML technique is defined as the ratio of the sum of absolute residual errors to the sum of absolute true errors, where *n* is the number of observations in the test dataset. For example, if a satellite's true *x*-coordinate is 10000 km and estimated *x*-coordinate is 10100 km, then the true error (the independent variable) is $e_T = 100\ km$. After applying an ML technique, which predicts the ML-modified error (the target variable) to be $\hat{e}_{ML}$ = 90 km, the performance metric can be calculated to be 10%. The lower the metric, the better the learning capability of the ML model. Also, the authors claimed that this metric directly quantifies the learning performance of the ML model because the remaining errors (e.g. the remaining 10 km that the ML model failed to correct in the above example) "represent the information that cannot be modelled with the available learning variables" [20, p.14].

Henceforth, the performance metric defined above will be referred to as the Mean Absolute Percentage Residual Error (MAPRE).

E. Synthesis and Analysis

Machine learning applications to orbit determination and propagation[3] for the purpose of SSA are still in its infancy [7]. Peng and Bai, the two pioneers in this field, have published several papers investigating the use of SVR, beginning with their paper in [22]. In subsequent papers such as [5, 8], they refined their method and applied it to different sets of data and RSOs. Their final paper on SVR in 2019 is the most up to date and provides a comprehensive account of their methodology and application to four LEO, three SSO, and four MEO satellites, achieving a MAPRE error of less than 50% in most of the 11×6 position and velocity components. In addition to SVR, they also explored the Gaussian Process (GP) improvements to their ML models in order to generate uncertainty information. This is because the ML approach only provides a point estimate without information about how uncertain that measurement is. Finally, they compared the performance of SVR, ANN (Artificial Neural Networks) and GP models, concluding that ANN provides the best estimates but is prone to overfitting, while SVR is the least likely to overfit but its performance is overshadowed by ANN and GPs [23].

The beauty of ML techniques is their universal data-driven approach. Not only can one use them to predict RSO orbits but also to discover their physical properties, given the appropriate data. Reference [6] used a regression random forest model to recover the area-to-mass ratio of 135 space objects in Sun Synchronous Orbit, achieving a discrete prediction accuracy of around 85%. Moreover, [7] used decision trees, random forest, logistic regression, and SVMs (linear, sigmoid and Gaussian kernels) to classify satellite stability state (whether a satellite is rotating or not), achieving a prediction accuracy of 89%. A slightly different application was in sending spacecraft to asteroids, and Gaussian Process Regression was used for estimating the parameters involved [19]. Yet another application involved the detection and

[3] Here, orbit determination refers to estimating discrete observations of a RSO's state (position and velocity) at certain points of its orbit, while orbit propagation refers to estimating the future state of the RSO, given historical data [21].

prediction of RSO maneuvers, which was done using inverse reinforcement learning [13]. Another study in maneuver detection used historical orbit data but without the ML approach, opting to use mathematical procedures instead [24]. Finally, a study on GPS and BeiDou System (the set of Chinese GPS satellites) used Convolutional Neural Networks (CNNs) with image data to reduce orbit prediction error by 45%. Overall, the use and application of ML techniques for SSA seem to be gaining in traction, especially in the last two years.

IV. METHODOLOGY

A. The Datasets and Approach Taken

The two datasets (in CSV format) used in this study were provided to participants of the 2020 International Data Analytics Olympiad by the Russian Astronomical Centre [9]. The first dataset is the Train dataset, which contains 649913 observations and 15 variables. The second is the Test dataset, which contains 284072 observations and 9 variables. The Train dataset consists of true and simulated coordinates (measured in *km*) and velocities (measured in *km/s*) of 600 satellites in the month of January 2014, while the Test dataset consists of only simulated coordinates and velocities of 300 satellites in February 2014. The variable names and descriptions for both datasets are summarized in Table II.

Table II. Variable names and descriptions

Name	Type	Description
id, sat_id	integer	unique row and satellite identifier
epoch	datetime	timestamp at the time of measurement
x, y, z	float	measurements of true position coordinates
Vx, Vy, Vz	float	measurements of true velocities
x_sim, y_sim, z_sim	float	measurements of simulated position coordinates (present in both Train and Test datasets)
Vx_sim, Vy_sim, Vz_sim	float	measurements of simulated velocities (present in both Train and Test datasets)

The *epoch* variable, as used in the astronomical literature, refers to the moment in time that the elements of a space object, such as its coordinate and velocities, are specified. This should not be confused with the number of epochs of training done for an ML model. The set of *simulated* coordinates and velocities were obtained using the less accurate SGP4 physics-based simulation model while the set of *true* coordinates and velocities were obtained using a more accurate (but unspecified) simulation model. To illustrate this information using an example satellite, the x- and x_{sim}-coordinates of Satellite 1 from both Train and Test datasets are plotted in Fig 4.

The green plot displays the real x-coordinates of Satellite 1 in January 2014, obtained from the Train dataset. The blue and orange plots are the simulated x_{sim}-coordinates in January 2014 (obtained from the Train dataset) and February 2014 (obtained from the Test dataset), respectively.

Several findings can be deduced by inspection from Figure 2 and from plots of other kinematic states as well (the set $\{x, y, z, V_x, V_y, V_z\}$ can be collectively called the *kinematic states*). Firstly, satellite kinematics display a "seasonal" pattern, which makes sense from a physical point of view, because a full orbit around the Earth occurs over a fixed period of time. Secondly, there are no cyclical patterns displayed by the data. To distinguish between cyclical and seasonal patterns, an extract from the book *Forecasting: Principles and Practice* says "If the fluctuations are not of a fixed frequency then they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal" [30, p. 31]. Since the fluctuations occur over a fixed orbital period, the pattern should be seasonal.

Finally, the simulated coordinates x_{sim} were seen to be initially accurate in predicting true/real coordinates but becomes increasingly inaccurate at future epochs. These findings suggest applying classical forecasting techniques, namely Seasonal Holt-Winters and Seasonal ARIMA methods. However, the focus of this study is on developing a regression-based machine learning model, so applying classical forecasting methods can be done as an extension to this paper.

Based on the plots and the above information, three approaches can be taken:

- Ignore the simulated states (coordinates and velocities) and directly predict real states for February using

Fig. 4. Plots of true and simulated x-coordinates for Satellite 1

training data of real states and epochs in January. This is equivalent to simply forecasting/extrapolating the green curve into the future:

Predictor variables $\{epoch, x\}$ → ML Model → Output $\{\hat{x}\}$

- Use both real and simulated states in January as training variables to fit the ML model, which can then predict the real states for February:

Predictor variables $\{epoch, x, x_{\text{sim}}\}$ → Model → Output $\{\hat{x}\}$

- Prepare the data so that the true errors (e.g. $e_x = x_{sim} - x$) and the simulated states in January become predictor variables for the ML model, which can then produce the ML-modified errors for February (e.g. $\hat{e}_x$). This is similar to the approach used in [20]:

Predictors $\{epoch, e_x, x_{\text{sim}}\}$ → Model → Output $\{\hat{e}_x\}$ → $\{\hat{x}\}$

In this paper, only Approach 1 will be used for fitting a Non-Linear Regression model, leaving the others for further research. In addition to classical forecasting methods, the application of modern neural network models, such as RNNs and LSTMs, will also be left for further study.

B. Evaluation Metrics

The three metrics that were chosen to evaluate the ML models are RMSE (Root Mean Square Error), SMAPE (Symmetric Mean Absolute Percentage Error) and MAPRE (Mean Absolute Percentage Residual Error). The RMSE is a commonly used metric in regression models, defined by the equation below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}, \quad (2)$$

where $\hat{x}_i$ is the predicted value, x_i is the actual value, and n is the number of observations in the dataset.

The SMAPE metric is less well-known, but it is specialized for use in forecasting applications. It is defined by the equation below:

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|\hat{x}_i - x_i|}{|\hat{x}_i| + |x_i|}, \quad (3)$$

where the terms in the equation are the same as those in RMSE. Sometimes, the model's performance Score (out of 100%) can also be defined as:

$$Score = 100\% - SMAPE. \quad (4)$$

The MAPRE metric is a metric introduced by Peng and Bai [5], in which they used the concept of MAPE (Mean Absolute Percentage Error) to define the MAPRE as follows:

$$MAPRE = 100\%\frac{\sum_{i=1}^{n}|e_x - \hat{e}_x|}{\sum_{i=1}^{n}|e_x|}, \quad (5)$$

where e_x is the true error in the x-coordinate and $\hat{e}_x$ is the ML-predicted error. The authors claimed that the learning performance of the ML model is directly quantified by this metric [20].

C. Non-Linear Regression

When faced with a periodic pattern in the data, an intuitive mathematical function for fitting it would be the Fourier series. It is known from mathematics that any periodic pattern that is continuous (need not be differentiable) can be approximated by a Fourier series.

However, based on the periodic pattern observed from Fig 2. in the previous section, a simple sinusoidal curve would be good enough to model the data. Also, due to the slight inclination seen from the figure, a combination of the sine function together with an added linear trend was considered for the regression model. Modelling the x-coordinate then turns out as follows:

$$x(t) = A\sin\left(\frac{2\pi}{T}t + \phi\right) + mt + c, \quad (6)$$

where A is the amplitude, T is the period, ϕ is the phase shift, m and c are the slope and intercept of the linear trend, t is the independent time variable and $x(t)$ is the time-dependent x-coordinate. For this model, the built-in `nls` or non-linear least squares function in R will be used to fit the above equation.

D. Data Preparation

Out of the 300 satellites in the Train dataset, Satellite 372 was chosen for implementing the NLR model outlined in the previous section. This satellite was chosen on the basis of having the largest number of observations, so as to provide the most training data for the ML model.

Satellite 372 has 6320 observations and 15 variables, with no evidence of missing values. However, an anomaly was discovered in which it seems as though the simulation data was updated within one millisecond. This is because while x_{sim} changes by a lot (from -1766 to 804), x changes very little (from 8581.835 to 8581.831) within that millisecond, as shown in Fig. 5 below. The engineered feature $delta\_t$ is the time difference between two successive epochs. Exploring this feature led to detecting the anomaly in the first place.

The solution to this anomaly was to keep the updated observation (the row highlighted in red) and to delete the old observation (the row above the highlighted red).

Other than this anomaly, kinematic data was seen to be sampled at equally spaced intervals of 7.06 minutes. In addition to $delta\_t$, other features such as the absolute time starting from 01 January 00:00, and error variables were also created for both exploration and analysis purposes.

Finally, the Satellite 372 dataset was split into training,

	epoch	sat_id	x	delta_t	x_sim
2014-01-19 17:39:14.809		372	9931.653	4.238690e+02	-4192.864
2014-01-19 17:46:18.680		372	8581.835	4.238690e+02	-1765.593
2014-01-19 17:46:18.680		372	8581.831	9.999275e-04	804.142
2014-01-19 17:53:22.549		372	6811.510	4.238690e+02	3319.096
2014-01-19 18:00:26.418		372	4678.618	4.238690e+02	5619.019

Fig. 5. Detected anomaly highlighted in red

validation and testing components. The training component was taken from Day 1 to Day 25 (25 days), validation component was taken from Day 26 to Day 28 (3 days), and the testing component was taken from Day 29 to Day 31 (3 days). Reference [31] states that a period of 2-3 days is typically considered before initiating maneuvers to avoid collision between space objects, while [12] and [20] extended this period to a maximum duration of 7 and 14 days, respectively.

In this study, the validation and test components were taken to be 3 days, as illustrated in Fig. 6.

E. Data Exploration

Firstly, all 6319 observations (excluding the anomaly) for the x-coordinate were plotted over 31 days to see how it varies, in Fig. 7.

Fig. 6. Splitting dataset into components

Fig. 7 indicates a somewhat periodic pattern, but it is difficult to see the detailed features of the graph, so only the first 204 observations will be plotted in subsequent visualizations.

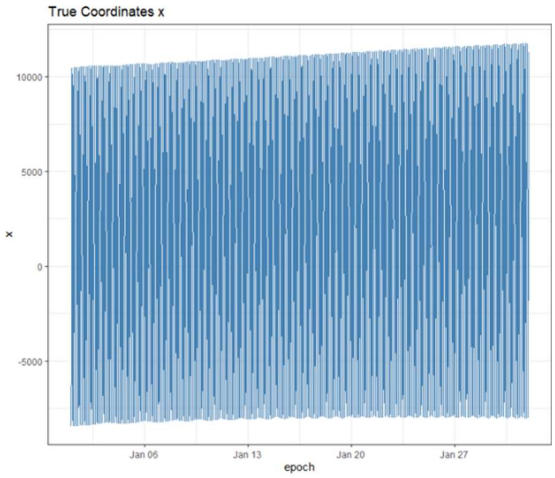

Fig. 7. Plot of true x-coordinates over 31 days

There are 6319 observations over 31 days, so one day is approximately 204 observations, assuming equal time interval between observations. Moreover, to see the comparison between the true and simulated coordinates, both of them were plotted on the same figure, in Fig. 8.

At first glance, it looks as if the simulated values are very

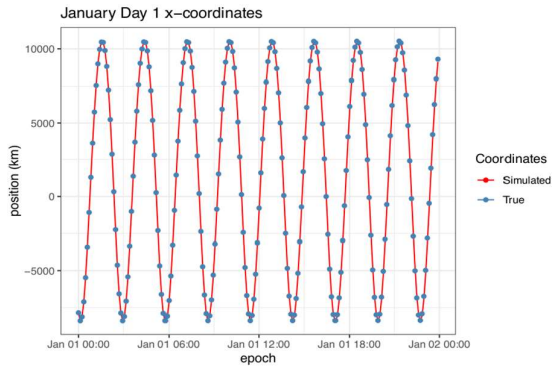

Fig. 8. Plots of true and simulated x-coordinates on 01 January

good at approximating the true coordinate. To see if this accuracy persists, the same plot on Day 22 was done, in Fig. 9.

A clear lag (or phase shift) between the true and simulated

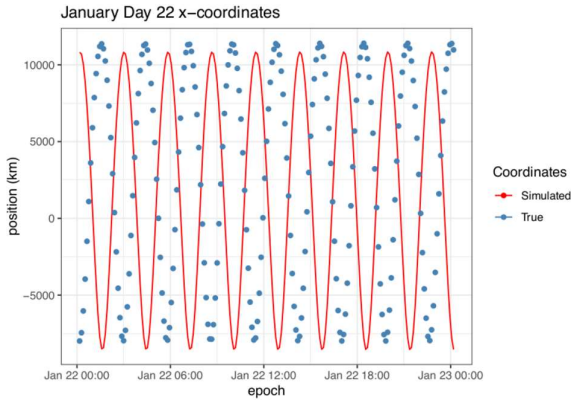

Fig. 9. Plots of true and simulated x-coordinates on 22 January

coordinates can be observed. Moreover, the overall point-

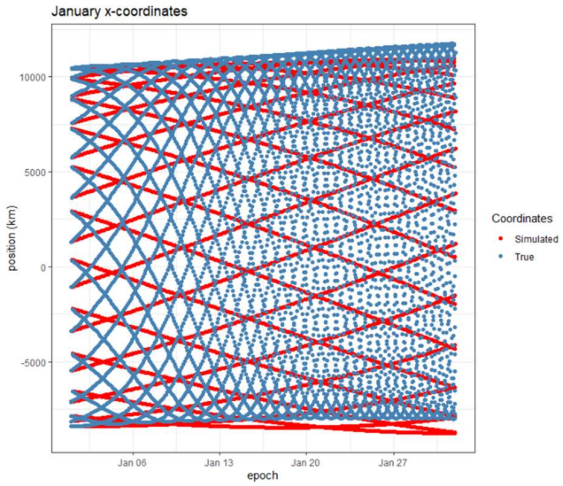

Fig. 10. Plots of true and simulated x-coordinates over 31 days

point plot for the entire 31 days was plotted, in Fig. 10.

Again, it is somewhat difficult to make out the details, but the fitting capacity of the simulated values on the true coordinates becomes poorer as time passes. This brings the discussion to visualizing the errors themselves, for instance, the error in x given by the variable $e_x = x - x_{sim}$, in Fig. 11.

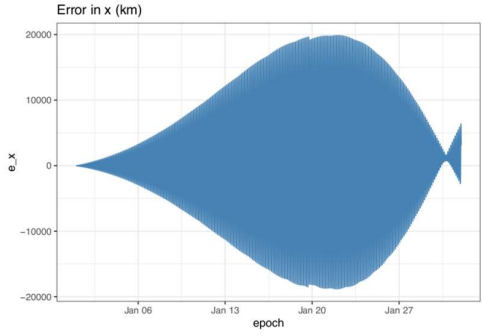

Fig. 11. The error in x, plotted over 31 days

The e_x variable looks like a spade-shaped object – in which the error starts out small and increases (this was expected) but peaks around Day 21-23 and then decreases to a small value at Day 30 (this was unexpected). After this, the error increases over Day 31. Visualizing the other errors also indicates the same spade-shaped pattern, albeit with the peaks occurring on different days.

Thus, it can be presumed that the simulated coordinates lag behind the true coordinates by some phase shift, which then corrects itself over 30 days. In fact, the plot above indicates two kinds of errors: the error in predicting the true coordinate, and the failure to do so at the correct epochs. Similar visualization can also be done for the remaining five kinematic states.

Finally, the orbit type of Satellite 372 was determined by calculating its radius of orbit according to

$$r = \sqrt{x^2 + y^2 + z^2}. \quad (7)$$

The orbit radius ranges from a minimum of 8166 km to a maximum of 12027 km, with a mean radius of 10248 km (Earth's radius is 6371 km). This orbit is on average, greater than the 2000 km threshold for Low Earth Orbit but less than the 35786 km lower limit for High Earth Orbit. Hence, Satellite 372 is an MEO (Medium Earth Orbit) object. A three-dimensional plot of the satellite orbit (in orange) using the 3-day validation data is plotted in Fig. 12. The model of Earth centered at the origin is shown in a blue mesh.

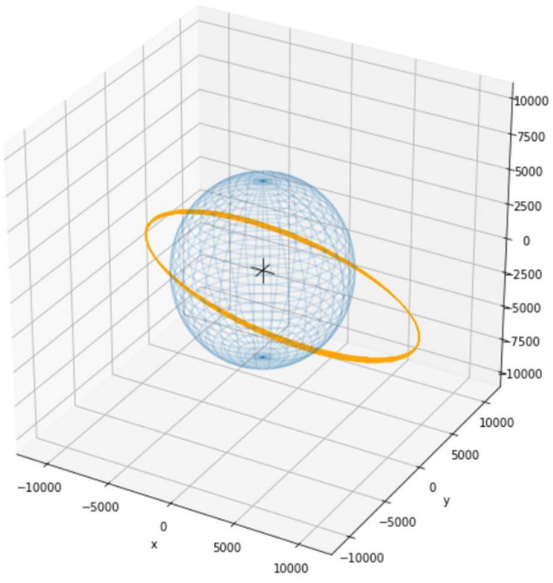

Fig. 12. 3D plot of satellite validation data over 3 days

F. *Implementing the Non-Linear Regression Model*

When implementing a non-linear fit, there is no need to use normalized data. Therefore, the non-normalized prepared datasets, including the split Validation and Test, were loaded onto R. The x-coordinate was first fitted, then y and so on, producing six separate models for each of the kinematic states.

One problem arose while implementing the non-linear regression on the first coordinate. Fitting on all observations in the Train dataset resulted in non-sensical values for the model parameters, which was checked by comparing with a quick x-coordinate plot shown in Fig. 14. Trying out various amounts of observations were found to be impractical because of the granularity, e.g. a comparison between using the last 950 observations or the last 951 observations is not very practical. Instead, this difficulty was implemented as a hyperparameter called "split" or *spl*, in which the Train dataset was divided at a certain fraction, and only the last $(100/spl)\%$ of data was used to fit the model. For example, if the Train dataset has 6000 observations and $spl = 10$, only the last 600 observations are used for fitting. If $spl = 1$, then all the observations are used. This is illustrated in Fig. 13.

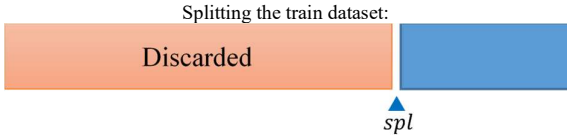

Fig. 13. Introducing the hyperparameter *spl*

In addition to reducing the amount of data required for fitting the model and making predictions, the spl hyperparameter was also chosen such that it produced statistically significant values for the model parameters. Moreover, the final selection of spl was made so that it resulted in the lowest possible SMAPE value while also respecting the significance of the model parameters.

Fitting the `nls` function requires the user to provide starting values. Very rough estimates of the amplitude and period were obtained by inspecting a plot of the final day (Day 25), while the gradient and intercept can be obtained by looking at the entire 25 days. Data for Day 25 from the Train dataset is plotted in Fig. 14.

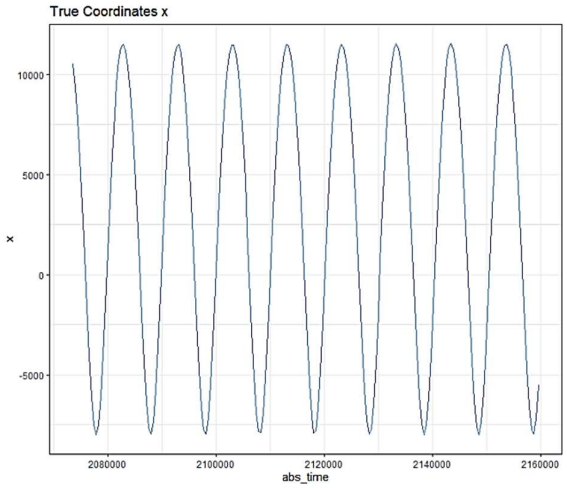

Fig. 14. A quick plot of Day 25 x-coordinates

In the above plot, the x-coordinate varies between $-7{,}000$ to $12{,}000$, so the amplitude A was estimated as $10{,}000\ km$. Likewise, the time difference between every other peak (or trough) appears to be separated by $20{,}000$ seconds, so the period T of the curve was estimated to be $10{,}000$ seconds. The phase shift ϕ and intercept c was just provided as 1 while the very gentle upward slope m was given the small positive starting value 0.01. The starting values need not be accurate as the model will automatically find the best-fitted parameters. Also, these parameters are different for each kinematic state,

so the starting values need to be updated and verified with a quick plot when fitting a different state.

After this inspection, the non-linear regression model was implemented. Values of the hyperparameter spl were tried out from 1 until 15 with a step size of 0.1. Both the model parameters and spl were fitted on the Train dataset and optimized on the Validation dataset. In cases where there were several possible spl values to choose from, the model was fitted on the combined Train and Validation dataset and optimized on the Validation dataset. The split producing the lowest SMAPE metric was chosen based on this validation procedure. Finally, using the best spl, the optimum model was fitted on the combined Train and Validation datasets and its performance was evaluated on the Test dataset.

V. Results and Discussion

There are 25 days in the Train dataset but only the last section split by spl was used for training. Duration of this section is given by the "Required" column in Table III. The 3-day Validation dataset, from Days 26-28 was used for finding the optimal spl and the final 3-day Test dataset, from Days 29-31 was used for evaluating the performance of each of the models on three metrics: RMSE, SMAPE and MAPRE, outlined in Section IV-B. The results obtained from this evaluation are summarized in Table III below. A summary of all the model parameters for each kinematic state is also summarized in Table IV.

Table III. Summary of evaluation results on Test data

Name	spl	Required	RMSE (km, km/s)	SMAPE (%)	MAPRE (%)
x	5.4	5d 4.5hrs	863.21	11.56	10.96
y	5.4	5d 4.5hrs	830.99	11.80	11.23
z	7.5	3d 17hrs	285.71	9.38	10.34
V_x	9.5	2d 23hrs	0.847	14.00	18.99
V_y	5.0	5d 14hrs	0.887	18.79	20.21
V_z	5.4	5d 4.5hrs	0.329	18.69	19.93

Table IV. Summary of model parameters

Name	A (km, km/s)	T (s)	ϕ	m (km/s, km/s^2)	c (km, km/s)
x	9.65×10^3	1.01×10^4	41.3	3.70×10^{-4}	1.86×10^3
y	9.55×10^3	1.01×10^4	14.7	-4.7×10^{-4}	6.15×10^2
z	3.58×10^3	1.01×10^4	15.9	3.81×10^{-4}	-1.4×10^3
V_x	6.01	1.01×10^4	14.3	2.1×10^{-8}*	-4×10^{-2}*
V_y	5.95	1.01×10^4	16.3	-3×10^{-8}*	5.9×10^{-2}*
V_z	2.22	1.01×10^4	14.5	6.9×10^{-9}*	-2×10^{-2}*

* Not statistically significant, $p \gg 0.05$

Firstly, all kinematic states required no more than 5 days and 14 hours of training data in order to predict 3 days in advance. It is also much easier to predict positions instead of velocities, as indicated by the lower SMAPE and MAPRE values. The z coordinate produced the most accurate forecasts, while also requiring less than 4 days of training data, whereas the V_y coordinate produced the least accurate ones. In addition, all model parameters were obtained with a significance of $p < 0.05$ except for the velocity component slopes and intercepts. This could be attributed to their low magnitudes of amplitude A compared to the position components. However, even after converting time from seconds to hours (to increase the order of magnitude value of the slope), the m and c parameters could not be estimated with statistical significance. The standard errors were too high compared with their estimates.

Secondly, the RMSE metric has units of km (or km/s for velocities), which are the same units as the kinematic states themselves. In this sense, the RMSE value can be interpreted as the average interval around the predicted value within which the true coordinate can be found. Also, the estimated value of the period T is the same for all kinematic states. This was expected from the visualizations done previously, and over 3 days, the period of the satellite is going to stay relatively unchanged. The data confirm the period of orbit as constant.

Thirdly, the optimal split was found by plotting the SMAPE metric against the spl hyperparameter and choosing the spl with the lowest validation SMAPE value, which also produces significant model parameters. This process is shown in Fig. 15.

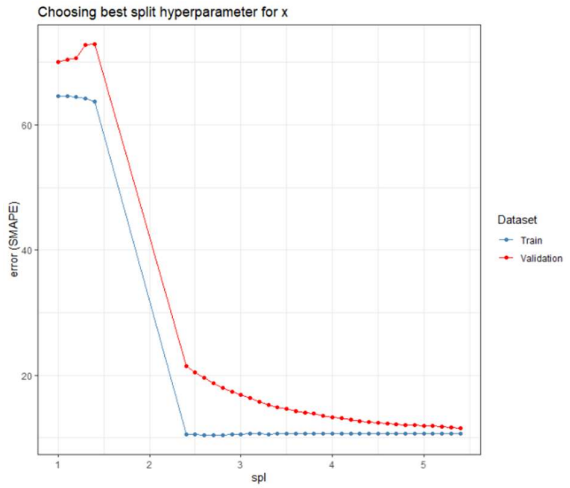

Fig. 15. Train and Validation curves for hyperparameter optimization

In Figure 15, further splits of the dataset resulted in non-significant values for the model parameters, so in this case, the best spl was chosen as 5.4 for the x-coordinate. For the velocity components however, no splits from 1 to 15 (step-size 0.1) produced significant m and c values, so the criteria for choosing the best spl was based only on the minima of the validation curve.

VI. Conclusion

To recap, this study looked at data on satellite kinematics and introduced a simple non-linear regression technique that can be used as a baseline for predicting the kinematic states of an MEO satellite. This satellite (Satellite 372 from the IDAO dataset) was chosen based on having the highest number of observations, however, not all of the data was required. The prediction was done 3 days into the future and used no more than 5 days and 14 hours of historical data. The lowest error achieved was for the z-coordinate with a SMAPE value of 9.38% and the highest was for the V_y velocity coordinate with a SMAPE value of 18.79%. The average SMAPE value

across the six kinematic states is 14.0%, resulting in an overall model score of 86.0%.

In comparison with Peng and Bai's work, where they applied Support Vector Regression to 11 satellites in different orbits (3 SSOs, 4 LEOs and 4 MEOs) [20], this paper only applied non-linear regression to one MEO satellite. However, the SVR technique in [20] required 300 days of training data to predict 14 days in advance while this paper used less than 5 days 14 hours of data to predict 3 days in advance. Reference [20] also used additional features in their training (such as drag information) that were not included in the present study. In addition, [20] also investigated different training data lengths, testing data lengths, different random seeds and different satellite orbits. Similar to this study, their results indicated a higher performance of the model on position coordinates as opposed to velocity coordinates. As for accuracies, it would not be fair to compare the performance between two studies that used different amounts of training and test data. That being said, the best performances in [20] range from MAPRE values between 13% and 25% while the worst ones were above 100%.

Finally, now that a non-linear regression baseline has been established, application of more complex ML models can be investigated. These include RNNs and LSTMs that can capture the periodic pattern of data and extrapolate into the future. Classical forecasting techniques such as the ARIMA and Holt-Winters exponential smoothing techniques may also be applied. In addition, the performance of prediction models on different satellite orbits and longer forecast horizons (between 7 to 14 days) can be investigated.

VII. APPENDIX

Table V. Table of Abbreviations

Abbreviation	Complete Form
ML	Machine Learning
AI	Artificial Intelligence
RSO	Resident Space Object
SSA	Space Situational Awareness
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
HEO	High Earth Orbit
NLR	Non-Linear Regression
IDAO	International Data Analysis Olympiad
SSO	Sun Synchronous Orbit
GSO	Geosynchronous Orbit
GEO	Geosynchronous Equatorial Orbit
GPS	Global Positioning System
SGP4	Simplified General Perturbations 4
TLE	Two-Line Element
ILRS	International Laser Ranging Service
RMSE	Root Mean Square Error
SMAPE	Symmetric Mean Absolute Percentage Error
MAPRE	Mean Absolute Percentage Residual Error
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory

VIII. ACKNOWLEDGMENTS

The author thanks the Russian Astronomical Science Center (and the hosts, partners and organizers of the International Data Analysis Olympiad) for providing the datasets used in this study.

REFERENCES

[1] Salleh, N. Asnilawati, Yuhaniz, S.S., Azmi, N.F. and Sabri, S.F. (2019). Enhancing simplified general perturbations-4 model for orbit propagation using deep learning: A review. In: *ACM International Conference Proceeding Series*. 2019, Association for Computing Machinery, pp. 27–32.

[2] Luo, Y. Zhong and Yang, Z. (2017). A review of uncertainty propagation in orbital mechanics. *Progress in Aerospace Sciences*. 89. pp. 23–39.

[3] NASA, 2019. ARES | Orbital Debris Program Office. [Online] Available at: https://orbitaldebris.jsc.nasa.gov/modeling/legend.html [Accessed 13 February 2020].

[4] Adilov, N., Alexander, P.J. and Cunningham, B.M. (2018). An economic "Kessler Syndrome": A dynamic model of earth orbit debris. *Economics Letters*. 166. pp. 79–82.

[5] Peng, H. and Bai, X. (2018). Improving orbit prediction accuracy through supervised machine learning. *Advances in Space Research*. 61 (10). pp. 2628–2646.

[6] Peng, H. and Bai, X. (2018). Recovering area-to-mass ratio of resident space objects through data mining. *Acta Astronautica*. 142. pp. 75–86.

[7] Mital, R., Cates, K., Coughlin, J. and Ganji, G. (2019). A machine learning approach to modelling satellite behaviour. In: *Proceedings - 2019 IEEE International Conference on Space Mission Challenges for Information Technology, SMC-IT 2019*. 1 July 2019, Institute of Electrical and Electronics Engineers Inc., pp. 62–69.

[8] Peng, H. and Bai, X. (2018). Exploring capability of support vector machine for improving satellite orbit prediction accuracy. *Journal of Aerospace Information Systems*. 15 (6). pp. 366–381.

[9] Higher School of Economics and Yandex, (2020). IDAO | International Data Analysis Olympiad. [Online]. Available at: https://idao.world/ [Accessed 21 January 2020].

[10] Peng, H. and Bai, X. (2019). Gaussian processes for improving orbit prediction accuracy. *Acta Astronautica*. 161. pp. 44–56.

[11] Gong, B., Geller, D.K. and Luo, J. (2016). Initial relative orbit determination analytical covariance and performance analysis for proximity operations. *Journal of Spacecraft and Rockets*. [Online]. 53 (5). pp. 822–835. Available from: https://doi.org/10.2514/1.A33444.

[12] Peng, H. and Bai, X. (2018). Artificial neural network-based machine learning approach to improve orbit prediction accuracy. In: *Journal of Spacecraft and Rockets*. 2018, American Institute of Aeronautics and Astronautics Inc., pp. 1248–1260.

[13] Linares, R. and Furfaro, R. (2017). Space objects manoeuvring detection and prediction via inverse reinforcement learning. In: *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2017. [Online]. Available from: https://amostech.com/TechnicalPapers/2017/SSA/Furfaro.

[14] Perez, I., San-Martín, M., López, R., Vergara, E., Wittig, A. and San Juan, J. (2019). Extending the hybrid methodology for orbit propagation by fitting techniques. *Neurocomputing*. 354.

[15] Matricciani, E. (2020). Geocentric spherical surfaces emulating the geostationary orbit at any latitude with zenith links. *Future Internet*. *12* (16).

[16] Lupo, R., Albanese, C., Bettinelli, D., Brancati, M., Minei, G. and Pernechele, C. (2018). Lighthouse: A spacebased mission concept for the surveillance of geosynchronous space debris from low earth orbit. *Advances in Space Research*. 62 (12). pp. 3305–3317.

[17] Kelso, T., 2020. CelesTrak: Current NORAD Two-Line Element Sets. [Online] Available at: https://www.celestrak.com/NORAD/elements/ [Accessed 19 February 2020].

[18] NASA, 2020. ILRS Home Page. [Online] Available at: https://ilrs.cddis.eosdis.nasa.gov/index.html [Accessed 19 February 2020].

[19] Shang, H., Wu, X., Qiao, D. and Huang, X. (2018). Parameter estimation for optimal asteroid transfer trajectories using supervised machine learning. *Aerospace Science and Technology*. 79. pp. 570–579.

[20] Peng, H. and Bai, X. (2019). Machine learning approach to improve satellite orbit prediction accuracy using publicly available data. *Journal of the Astronautical Sciences*.

[21] Shou, H.-N. (2014). Orbit propagation and determination of low earth orbit satellites C.-L. Chang (ed.). *International Journal of Antennas and Propagation*. [Online]. *2014*. 12. Available from: https://doi.org/10.1155/2014/903026.

[22] Peng, H. and Bai, X. (2017). Limits of machine learning approach on improving orbit prediction accuracy using support vector machine. In: *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. January 2017, p. 15.

[23] Peng, H. and Bai, X. (2019). Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy. *Astrodynamics*. 3 (4). pp. 325–343.

[24] Li, T., Li, K. and Chen, L. (2018). New maneuver detection method based on historical orbital data for low Earth orbit satellites. *Advances in Space Research*. 62 (3). pp. 554–567.

[25] Chen, J., Du, J. and Sang, J. (2019). Improved orbit prediction of LEO objects with calibrated atmospheric mass density model. *Journal of Spatial Science*. 64 (1). pp. 97–110.

[26] Kato, H., Akiyama, K., Akiyama, Y., Sakamoto, T., Matsumoto, T., Ikeda, S., Hinagawa, H. and Nakamura, S. (2018). Machine Learning for Atmospheric Drag Prediction of LEO satellites. [online]. In: *AIAC18: 18th Australian International Aerospace Congress*. pp. 1016-1021.

[27] Li, B., Sang, J. and Chen, J. (2018). Achievable orbit determination and prediction accuracy using short-arc space-based observations of space debris. *Advances in Space Research*. 62 (11). pp. 3065–3077.

[28] Li, T., Li, K. and Chen, L. (2018). New maneuver detection method based on historical orbital data for low Earth orbit satellites. *Advances in Space Research*. 62 (3). pp. 554–567.

[29] Pihlajasalo, J., Leppäkoski, H., Ali-Löytty, S. and Piché, R. (2018). Improvement of GPS and BeiDou extended orbit predictions with CNNs. In: *2018 European Navigation Conference, ENC 2018*. 10 August 2018, Institute of Electrical and Electronics Engineers Inc., pp. 54–59.

[30] Hyndman, R.J., and Athanasopoulos, G. (2018). Forecasting: Principles and Practice, 2nd Edition, OTexts: Melbourne, Australia. [Online] Available at: otexts.com/fpp2/. [Accessed 3 March 2020].

[31] M. Hülsmann, R. Kahle and M. Schneller et al. (2019). Debris collision avoidance by means of attitude control-in-flight demonstration with TET-1, *Journal of Space Safety Engineering*. Available at: https://doi.org/10.1016/j.jsse.2019.09.003