mathematics

MDPI

Article

Satellite Orbit Prediction Using Big Data and Soft Computing Techniques to Avoid Space Collisions

Cristina Puente [1,*], Maria Ana Sáenz-Nuño [2], Augusto Villa-Monte [3] and José Angel Olivas [4]

[1] Computer Science Department, ICAI School of Engineering, Comillas Pontifical University, 28015 Madrid, Spain
[2] Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, 28015 Madrid, Spain; msaenz@iit.comillas.edu
[3] School of Computer Science, National University of La Plata, La Plata 1900, Argentina; avillamonte@lidi.info.unlp.edu.ar
[4] Department of Information Technologies and Systems, University of Castilla-La Mancha, 13071 Ciudad Real, Spain; joseangel.olivas@uclm.es
* Correspondence: cristina.puente@comillas.edu

Abstract: The number of satellites and debris in space is dangerously increasing through the years. For that reason, it is mandatory to design techniques to approach the position of a given object at a given time. In this paper, we present a system to do so based on a database of satellite positions according to their coordinates (x,y,z) for one month. We have paid special emphasis on the preliminary stage of data arrangement, since if we do not have consistent data, the results we will obtain will be useless, so the first stage of this work is a full study of the information gathered locating the missing gaps of data and covering them with a prediction. With that information, we are able to calculate an orbit error which will estimate the position of a satellite in time, even when the information is not accurate, by means of prediction of the satellite's position. The comparison of two satellites over 26 days will serve to highlight the importance of the accuracy in the data, provoking in some cases an estimated error of 4% if the data are not well measured.

Keywords: orbit prediction; error position estimation; debris; data accuracy

check for updates

Citation: Puente, C.; Sáenz-Nuño, M.A.; Villa-Monte, A.; Olivas, J.A. Satellite Orbit Prediction Using Big Data and Soft Computing Techniques to Avoid Space Collisions. *Mathematics* **2021**, *9*, 2040. https://doi.org/10.3390/math9172040

Academic Editor: Fernando Sánchez Lasheras

Received: 5 July 2021
Accepted: 17 August 2021
Published: 25 August 2021

1. Introduction

The number of satellites and artificial objects orbiting in space around the earth has grown exponentially in recent decades due to the reduction of production costs and technological advances that facilitate their launch. Thus, the number of artificial objects orbiting the Earth in the year 2000 has increased from 11,000 to 19,500 objects in the year 2019, with a much higher forecast for the coming years [1], as seen in Table 1 published by the European Space Agency (ESA) about the debris figures [2]. Consequently, the disproportionate increase in space debris over the years, as seen in Figure 1, generates a greater risk of collision between satellites, which in turn generates more debris and more risk of collision. This growth highlights the need for positional control of the satellite in space so that it does not collide with any other object and cause major damage.

The debris problem has been a matter of study and concern since the United Nations published a report in 1999 warning about this forthcoming problem [3], focusing its information on orbital debris measurements, modeling, risk assessments, and mitigation measures.

The United States as the lead country, and NASA, has a specific government organization designated to control space debris, The United States Government Orbital Debris Mitigation Standard Practices (ODMSP) [4]. Among its tasks is the mission of issuing reports and generating control mechanisms to minimize the discharge of space debris in case of collision as well as a guide of good practices for space operations to promote efficient and effective space safety practices.

Table 1. Esa estimations of debris in space [2].

Subject	Amount
Number of rocket launches since the start of the space age in 1957	About 6050
Number of satellites these rocket launches have placed into Earth orbit	About 11,370
Number of debris objects regularly tracked by Space Surveillance Networks and maintained in their catalogue	About 28,160
Estimated number of break-ups, explosions, collisions, or anomalous events resulting in fragmentation	More than 560
Total mass of all space objects in Earth orbit	More than 9300 tonnes

The loss of control of satellites in their orbits can generate collisions like the 2009 accident where the American satellite US Iridium-33 communication satellite and the inactive Russian Kosmos-2251 collided with each other [5], or more recently, in 2021, where the Chinese rocket Long March 5B entered the Earth's atmosphere uncontrolled and at risk of causing numerous damages [6].

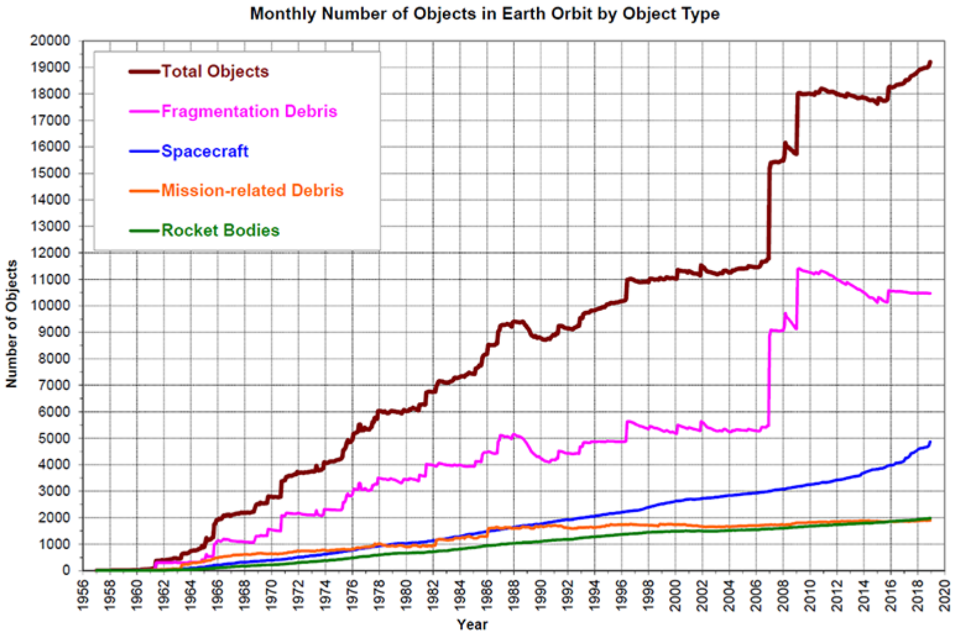


Figure 1. Evolution of space objects and debris through the years (Credit: NASA ODPO) [7].

The prediction of orbits in space objects and the calculation of their error has been a topic addressed by several studies applying numerous artificial intelligence techniques. The ease of finding reliable data on object positions in space provided by numerous space agencies [8–10], makes more reliable the use of machine learning techniques combined with Big Data that increase the reliability of the results obtained. In this scope, machine learning algorithms provide data modelling based on large historical data, providing in some cases a different prediction from physical evidence [11].

In this regard, it is worth mentioning the study presented in [12] on the orbit prediction of 11 satellites using machine learning techniques, specifically a support vector machine (SVM) model applied to data obtained from TLE catalog. In this case, the measurements were uniform, so results demonstrated an improvement in the orbit prediction accuracy. SVM has been widely used to predict orbit errors as in [13,14], improving the orbit prediction accuracy learning from historical prediction errors.

Other machine learning techniques have been used to improve orbit prediction accuracy as in [15] implementing LSTM networks to provide atmospheric density predictions, in [16] convolutional neural networks to work with special images to detect space objects, or in [17–19] where neural networks were used, for example, to explore the solar gravity-driven orbital transfers in the Martian system.

These studies started with a uniform, process-ready dataset. This is usually not the real case, where measurements and data require deep filtering, or even, as in our study, an intermediate data prediction process to complete the dataset added to these machine learning techniques, we have used prediction techniques based on median values from the dataset to fill in those gaps of missing measurements. This way we will be able to provide more accuracy when calculating our orbit's parameters, and so the highest accuracy on our error calculation and prediction model.

2. Data Collection and Preprocessing Stage

For this experiment, we have used the satellite data available for download at Kaggle website [20]. For each satellite, we know its ID and the sequence of positions (x,y,z) through which it passed over time with the timestamp of the moment it did so. We also have the velocity of each of the dimensions that will be used in future works.

The dataset has the positions of 600 satellites between 1 and 24 January, 2014. It has in total about 503,227 rows and one would expect to have about 839 positions for each satellite. However, this is not the case and each satellite has a different number of positions (some more some less) and even the positions of the satellites are recorded at different times during the passage of time. This makes it difficult, for example, to calculate the error in the orbit of a satellite or even to detect collisions between satellites as we will tackle in the next sections.

For this reason, in this article we decided to create a prediction algorithm to estimate those measurements missing and help to better accuracy in the ellipse calculations along with the estimated error provided among all ellipses.

In this article, we will work with the positions (x,y,z) of 137 satellites with all their positions recorded in no more than 30 min. This is achieved by calculating the differences between consecutive timestamps. This is how we managed to have from each satellite about 50 samples per day (2 every one hour). Figure 2 shows the mean ± deviation and the box plot corresponding to the number of records that the satellites have per day for (a) the total sample, (b) the 137 selected satellites and (c) for the rest. As seen in (b), by having at least one record every half hour, we ensure that we have the most complete track of the satellites to be analyzed.

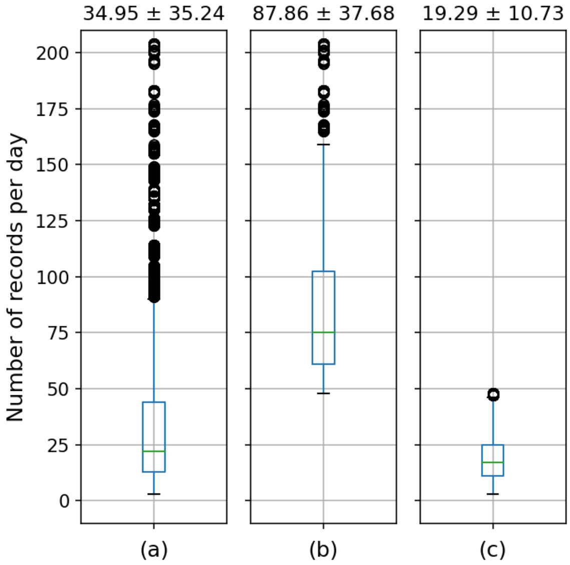


Figure 2. Mean ± deviation and box plot corresponding to the number of records per day for different subsets of satellites: (**a**) the total sample, (**b**) selected satellites and (**c**) non-sectioned satellites.

As seen in Figure 3, for those 137 satellites that have been selected, the differences in time between records are somewhat more stable. They do not exceed 30 min and are around 16.38 ± 6.45 min. On the other hand, for the remaining 463 satellites of the total set, the differences vary greatly with outliers. In the experimental section, we will reflect the differences between these 137 satellites with stable measures, with the others to minimize the prediction error in our position's estimation.

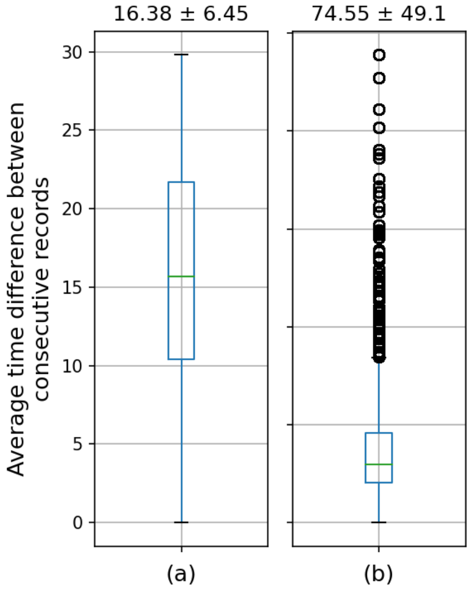


Figure 3. Mean ± deviation and frequency box plot with which the positions of (**a**) the selected satellites and (**b**) the remaining satellites were recorded.

Figure 4a shows the differences in (x,y) of the satellite with id 123 at the beginning (blue), at an intermediate point (red) and at the end of the timeline (green). The differences in this particular case are due to the times at which the positions are recorded. However, these differences may also be due to errors during the satellite orbit as seen in Figure 4b.

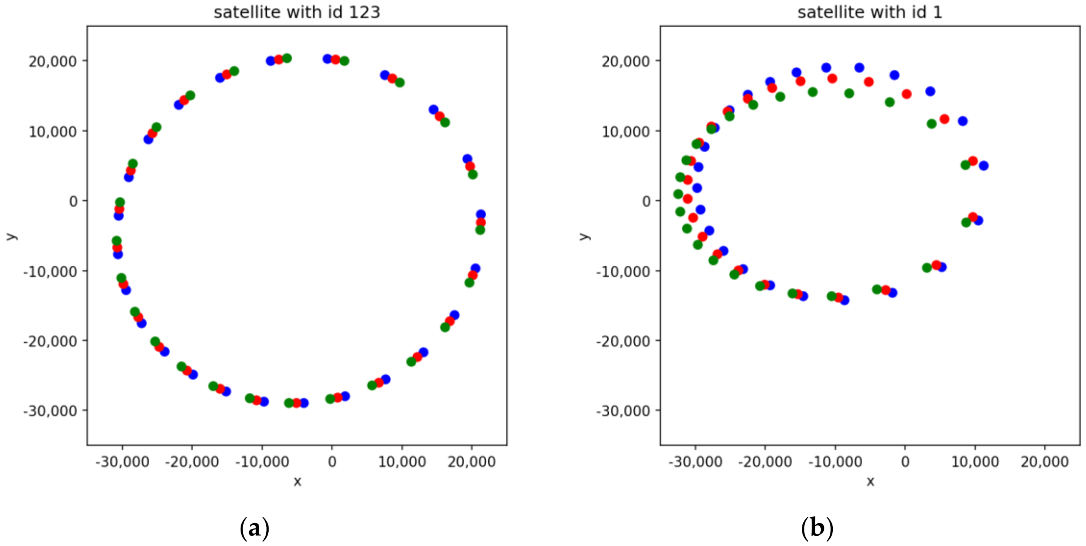


Figure 4. (**a**,**b**). Differences in (x,y) positions over time; at the beginning (in blue), in the middle (in red) and at the end of time (in green) for two satellites.

These differences may even be due to accelerations that the satellite undergoes during the passage of time. Figure 5 shows the differences of the same two satellites of Figure 4, taking into account in this case the positions in three dimensions.

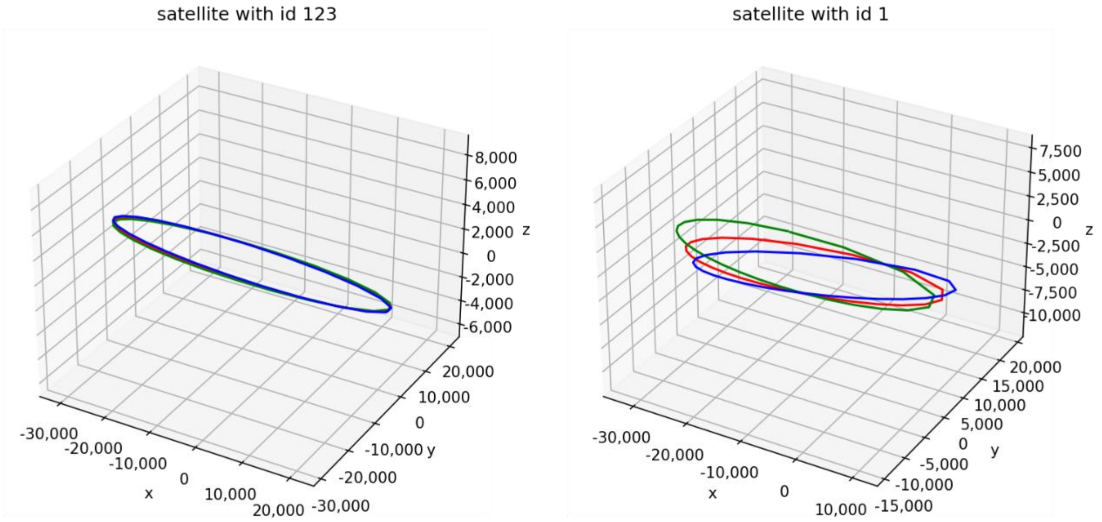


Figure 5. Differences in the orbit at the beginning (blue), at the midpoint (red) and at the end (green) of two satellites.

To calculate the error of the satellite orbits, it is essential to approximate the ellipse of the satellite motion with the available data (see Section 3.2) and to compare all the satellite cycles with the ellipse approximation. Figure 6 shows in signal form the motion of a satellite in its three dimensions at three different times. As seen in the figure, the positions of the same satellite on three different days are not comparable. Therefore, the key is to determine the "cycle" of the satellite to slice the signal and this can be solved by obtaining the inflection points. The difficulty lies in the triple dimension of the data, as obtaining the 3D inflection points is not an easy thing to do [21]. Figure 6 shows the signal (x,y,z) of a satellite scaled between (0,1)for easy visualization and the position of the satellite in each dimension per time instant. As seen, it is not the same instant in all three dimensions when the satellite is returning to its position. Both aspects (those mentioned in Figures 6 and 7) make data processing and satellite position prediction difficult.

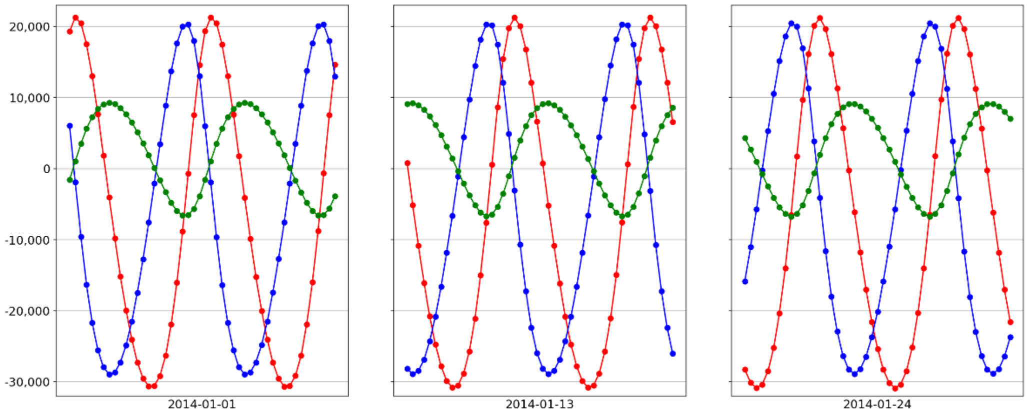


Figure 6. Satellite motion with id 123 on three different days (1-January, 13-January and 24-January, 2014) for each dimension: x in red, y in blue and z in green). Note that the satellite locations at similar times in the cycle do not coincide, for example, at the upper and lower peaks.

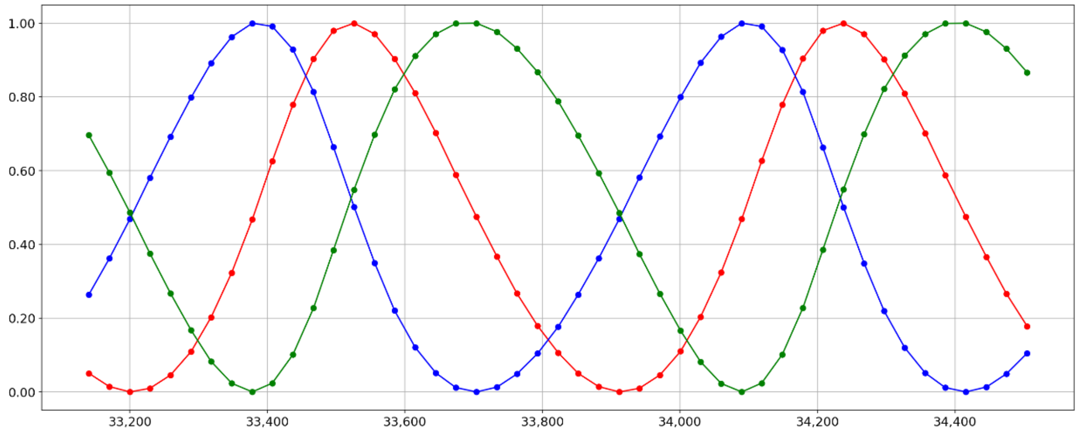


Figure 7. Signal (x,y,z) of a satellite over time, whose positions are scaled between 0 and 1 so that it is clear that the moment when the satellite turns around does not coincide in its three dimensions.

However, in the case of satellites, it is sufficient to obtain the peaks of the signal in one of the three dimensions (e.g., x), obtaining the local maxima by a simple comparison of neighboring values. In this way, the satellite cycles can be obtained and the beginning and end of the signal can be discarded to obtain complete cycles of the satellite path.

3. Fitting the Points and Finding the Errors

Once the data have been filtered and preprocessed as we mentioned in the previous section, we used the minimum squared method to fit the points to an ellipse and find the center and the axis of said ellipse.

To do so, we applied python and Matlab languages as we will see in the next sections.

3.1. Least Squared Method

Least squares is a numerical analysis technique framed within mathematical optimization, in which, given a set of ordered pairs—independent variable, dependent variable—and a family of functions, an attempt is made to find the continuous function, within that family, that best approximates the data (a "best fit"), according to the criterion of least squared error.

In its simplest form, it attempts to minimize the sum of squares of the differences in the ordinates (called residuals) between the points generated by the chosen function and the corresponding values in the data. Specifically, it is called least mean squares (LMS) when the number of measured data is 1 and the gradient descent method is used to minimize the squared residual. It can be shown that LMS minimizes the expected squared residual, with the minimum number of operations (per iteration), but requires a large number of iterations to converge.

From a statistical point of view, an implicit requirement for the least squares method to work is that the errors of each measurement are randomly distributed. The Gauss–Markov theorem proves that least squares estimators are unbiased and that the data sampling does not have to conform, for example, to a normal distribution.

The way used to minimize the geometric distance of the points to the ellipse, for which it will be necessary to use the parametric representation of the ellipse [22].

Let $X = \{x_1, x_2, \ldots .x_n\}$ a set of points in $\mathbb{R}^2$,ε a representation of the ellipse, and $D_G(X, \varepsilon)$ the geometric distance of the points to the ellipse, then we want to find the ellipse such that $min_{\varepsilon ellipse} D_G(X, \varepsilon)$ Another possibility is to represent the ellipse by means of its parametric equations:

$$\begin{aligned} x &= z_1 + a\ cos\theta \\ y &= z_2 + b\ sin\theta \end{aligned} \tag{1}$$

Which in vector form, we can write as:

$$x = z + \left(\frac{a\ cos\theta}{b\ sin\theta} \right) \tag{2}$$

where $0 \leq \theta \leq 2\pi$. As the axes can be any axes, we will use the representation:

$$x = z_1 + Q(\alpha)\gamma(\theta)\left(\frac{a}{b}\right) \tag{3}$$

where $Q(\alpha)$ is the rotation matrix and $\gamma(\theta)$ is a diagonal matrix. Both are written below:

$$\begin{aligned} Q(\alpha) &= \begin{pmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{pmatrix} \\ \gamma(\theta) &= \begin{pmatrix} cos\theta & 0 \\ 0 & cos\theta \end{pmatrix} \end{aligned} \tag{4}$$

Then, in terms of the chosen representation, the ellipse is characterized as follows:

$$\varepsilon(\alpha, a, b, z) = \left\{ z + Q(\alpha)\gamma(\theta)\left(\frac{a}{b}\right) \middle| 0 \leq \theta \leq 2\pi \right\} \tag{5}$$

Thus, the distance from a point to the conic can be defined as:

$$\begin{aligned} D_G(X_i, \varepsilon(\alpha, a, b, z) &= min_{p\epsilon\varepsilon} \|p - x_i\|^2 \\ &= min_{\theta\epsilon\mathbb{R}} \|p(\theta) - x_i\|^2 \end{aligned} \tag{6}$$

In our case, the limitation presented with this method is basically that it has an error in itself due to the way it is calculated. There are different methods that could be contrasted when there is enough data from the same satellite and above all that we only have information of what is supposed to be the CM of the satellite, and not its physical extensions. The mathematical programming of the least squares is based on the code written in Matlab by Richard Brown [23], who freely allows its use with slight variations in order to get the expected results

3.2. Least Squared Method Application

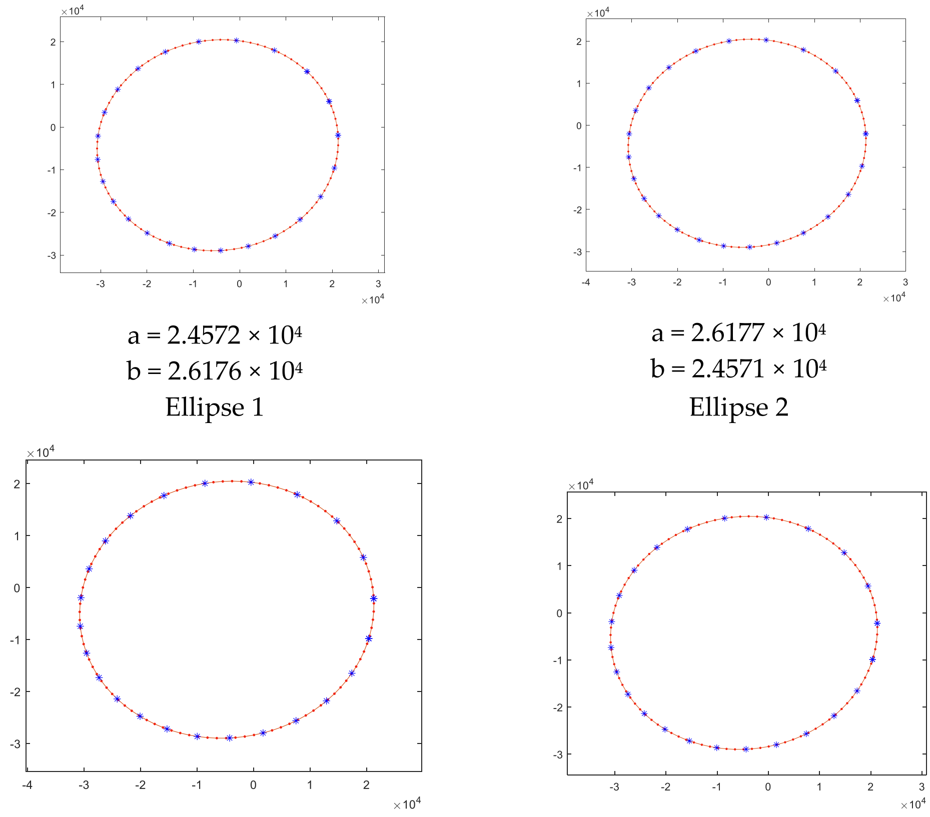
According to all this, we have applied the fitEllipse(x,y) function of python which receives a set of points with its x and y coordinates and returns the parameters of the ellipse that best fits into these points and the coordinates of the center of the ellipse and compared its results with the Matlab results.

For every trajectory, we computed the center position of the 3D elliptical trajectory and the size of the two axis, a and b, as the more robust measurement of the quality of the least square approximation. These data will be not too sensitive to small changes in any spurious data.

The same satellite evaluated its trajectory in different moments dealt with these results presented in Figure 8.

Because of the changes in the trajectory, we evaluated for the same satellite, its trajectory in different moments and evaluated those parameters. Based on the database information and the exploratory analysis performed, one satellite lasts 12 h in performing a full orbit, so we have in 26 days, 48 full orbits to compute the error among them.

Then, we plotted the center position coordinates (x,y) of the ellipse trajectory versus the day of measurement. Approximating the points to a straight line results in two graphs with practically opposite slopes, which shows an opposite but homogeneous trend in the two coordinates. When one varies, the other varies with the same relationship in the opposite direction. This is shown in Figure 9.



a = 2.4572 × 10⁴
b = 2.6176 × 10⁴
Ellipse 1

a = 2.6177 × 10⁴
b = 2.4571 × 10⁴
Ellipse 2

a = 2.6179 × 10⁴
b = 2.4570 × 10⁴
Ellipse 3

4569 × 10⁴

Ellipse 4

Figure 8. Trajectories of satellite 123 in different days.

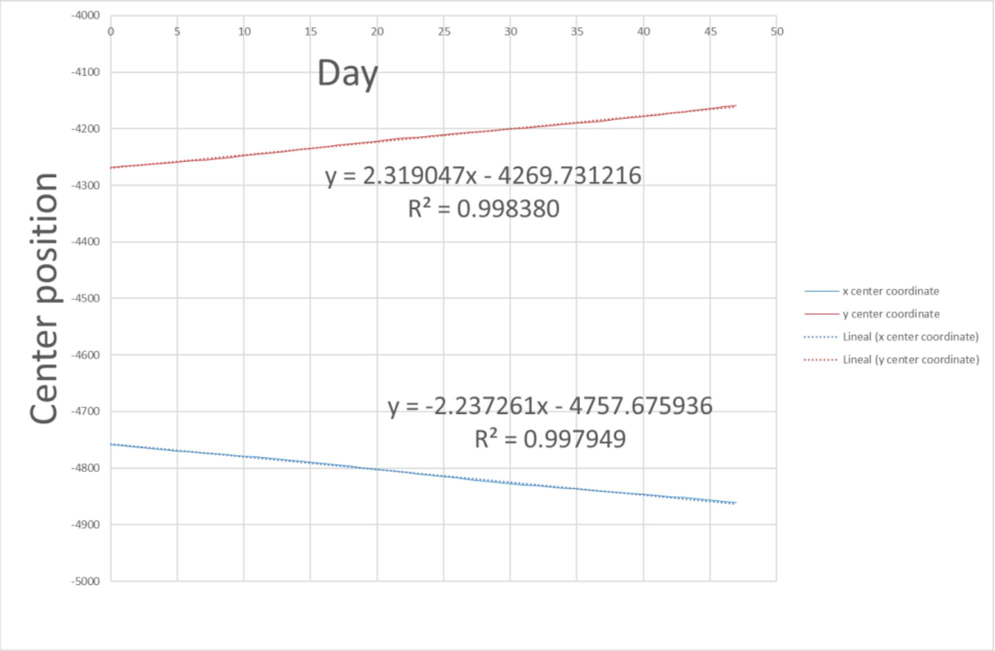


Figure 9. Position coordinates of each trajectory, being x—Day of measurement, y—center position [km] and R—the correlation between the predicted values and the observed values.

As seen in Figure 9, a linear dependence is observed in the slope of the two coordinates of the center of the ellipse, the trajectory of the satellite. This dependence is possibly caused by the satellite's own control. Although these data are known to the owners of the satellite, they are not public. A satellite tracking and position prediction will help to avoid collisions. Therefore, the goodness of the determination of such an approach is determined. To begin with, we represent the dependence of these coordinates on each other as seen in Figure 10.

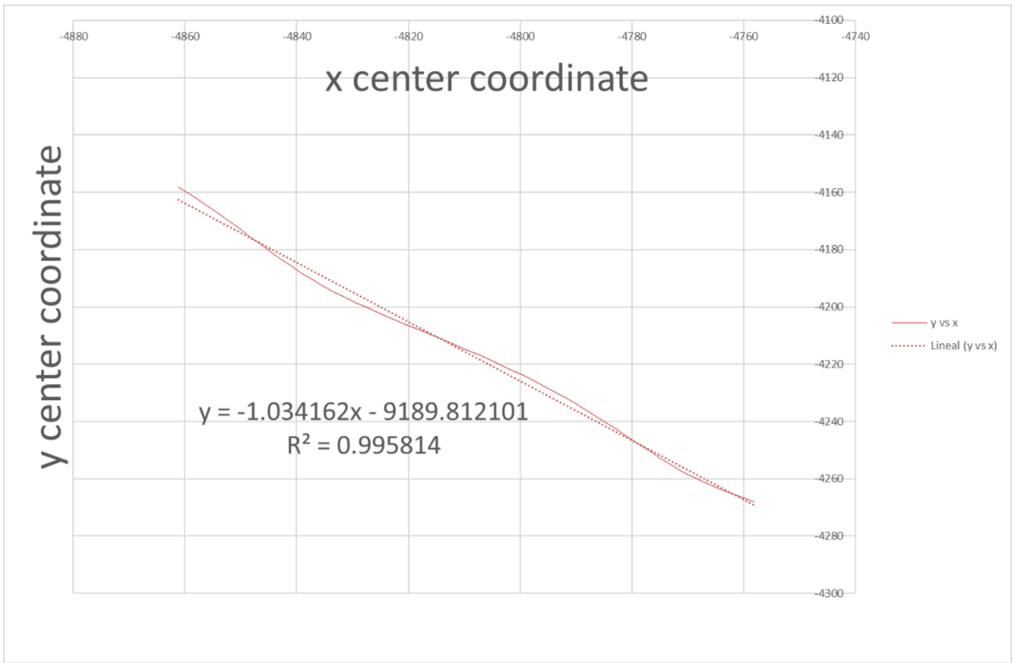


Figure 10. Variation of the position of the center of the trajectory.

Since the latter equation has a slope of 1, it indicates that the two coordinates x and y vary approximately equally. What one increases is what increases the other. However, when the semi-axes of the ellipse are represented, it turns out as seen in Figure 11.

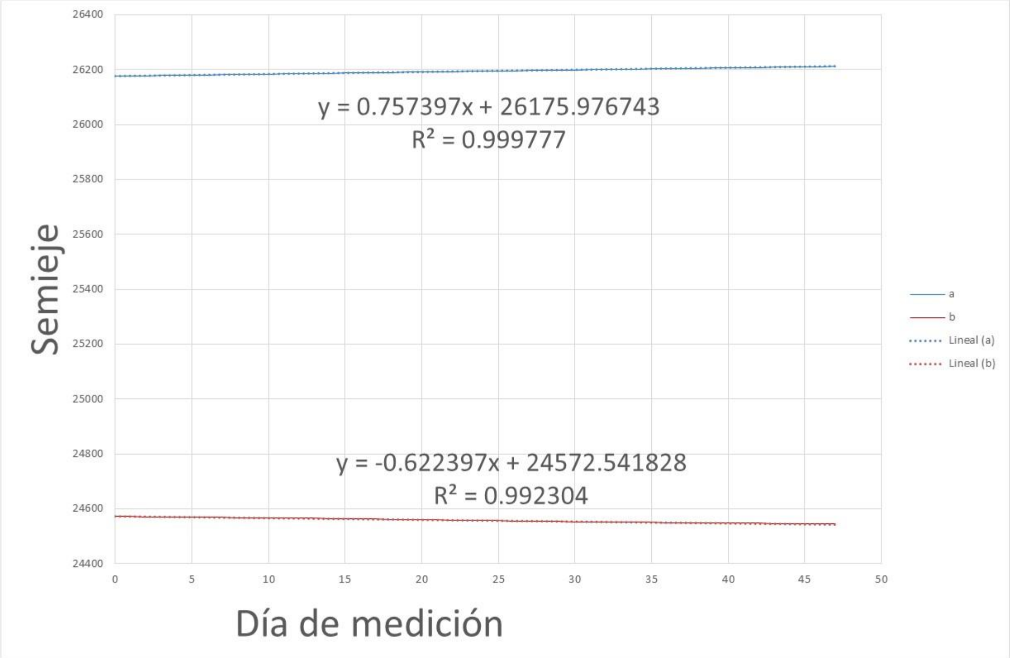


Figure 11. Semi-axes of the ellipse.

There is also a dependence between the two semi-axes of the satellite trajectory. Therefore, we are determining one of the parameters that can best help us to predict the position of the satellite based on the public data. Representing the two semi-axes graphically, it follows as seen in Figure 12.

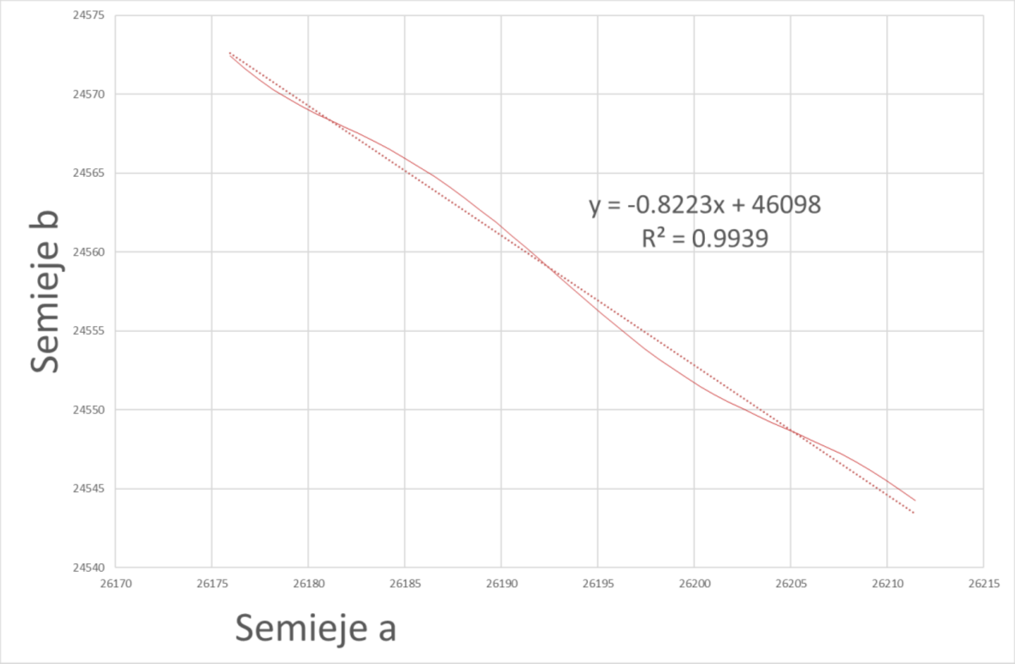


Figure 12. Representation of one axis versus the other.

Where it can clearly be seen a pulsation on the fitted line. Reorienting the axes so that the straight line of the linear fit becomes the x-axis results in a pulsation as Figure 13 represents.

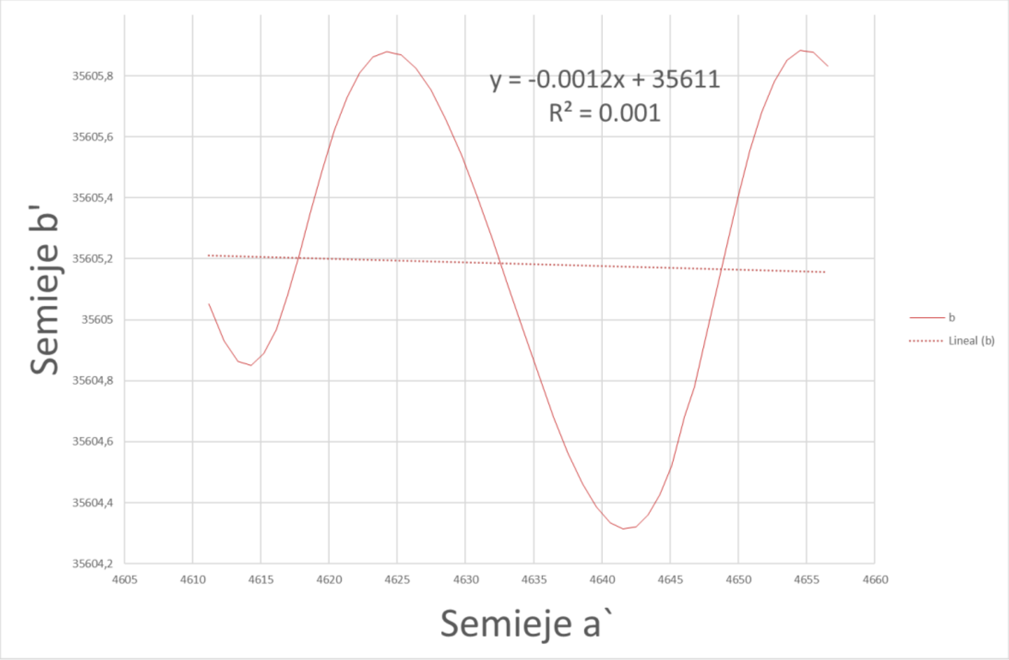


Figure 13. Representation of one axis versus the other in a new reference system.

3.3. Prediction of Missing Positions

Among the data relating to the positions of the 600 satellites, we found that only 137 had uniform measurements made at 30-min intervals, which gave a degree of reliability to the results. The remaining 463 satellites had imprecise intervals, with gaps between measurements of different temporal sizes, which led to somewhat irregular orbit calculations on most occasions, and made the calculation of the estimated error of the satellite under study very imprecise, as seen in the orbit of Figure 14, where there is no closing point, taking the next temporal measurement already in the course of the next orbit.

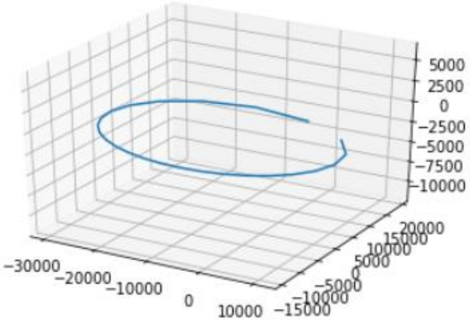


Figure 14. Satellite 1, first orbit representation with no closing point. The next data available belongs to the second orbit of the satellite.

To solve this problem, a prediction has been made based on the historical measurements that allows to recreate smaller time windows, in this case of 1 min (the value of this parameter can be changed) in a uniform way. In this way, the closing of one orbit and the beginning of another can be estimated.

In this paper the use of median to estimate the different positions in each satellite dimension has been evaluated. The approach for the prediction exploits the pairs of time consecutive positions. The pseudocode is detailed as follows:

```
for each satellite:
-- for each measurement and its next
-- -- if the difference between the timestamps is greater than 2 min
-- -- -- calculate median between timestamps
-- -- -- for each dimension (x, y, z)
-- -- -- -- calculate the median between the positions
-- -- -- add estimated measure
```

The complexity of the algorithm is linear as a function of the number of measurements, except when there are missing measurements, where a prediction must be made about the missing measurements, with the complexity increasing the greater the number of missing gaps between the data.

In any case, when extrapolating it to the 600 satellites, the complexity will be a function of n^2 (being n the number of elements) as seen in the source code shared (), with a computational time of less than one minute for all satellites during one month.

This technique is able to give accurate estimates the more times it is applied to fill in the gaps of missing measurements. The application of the repeated median method is better than the single median.

Using this method, we can extrapolate a more reliable approximation of the orbit described by the satellite, as shown in Figure 15. Thus, the calculation of the error between the different orbits corresponding to the 26 days in the same satellite will be more accurate as shown in the following section.

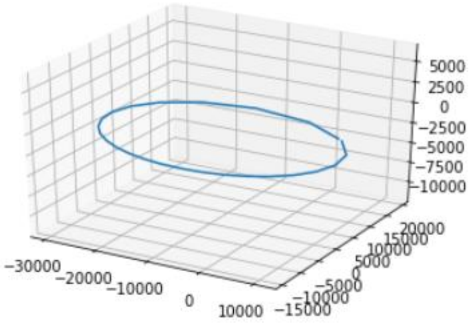


Figure 15. Satellite 1 first orbit representation with predicted values.

3.4. Error Aproximation in Each Approximation and Comparison

While the a-axis seems to remain constant, in its drift the ellipse pulses on its semi-axis b, possibly due to corrections of the orbit.

The satellite seems to travel on a trajectory as represented on Figure 16.

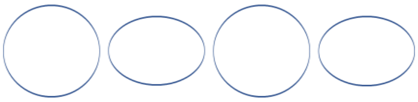


Figure 16. Representation of the variation on the orbit's trajectory.

As the period of time over which it has been analyzed is 26 days, it seems that every 12 h it completes a cycle, and every 6 days or so its semi-axis has gone from maximum to minimum or vice versa.

As demonstrated, the trajectory of the satellite is clearly captured in an elliptical shape whose parameters, center and semi-axis can be calculated relatively easily from objective measurements.

When analyzing the evolution of these parameters, it is surprising that they do not remain constant over time, as might have been thought at first, but rather vary.

The center and the semi-major axis do remain constant, while the semi-minor axis oscillates sinusoidally in a stable manner. This leads to the conclusion that such changes are intentional and therefore predictable, they cannot be considered a measurement error.
The center and the semi-major axis do remain constant, while the semi-minor axis oscillates sinusoidally in a stable manner. This leads to the conclusion that such changes are intentional and therefore predictable, they cannot be considered a measurement error.
If this variation were taken as a rectangular distribution, such an error would give a conservative contribution of about 1.626/root(3) km, somewhat less than 0.94 km.

However, if the least squares fit with learning is used, we can approximate the position with 1.569/root(12) km, which is about 0.45 km standard deviation.

Consequently, the error in the position decreases and the error security is twice as high in the case of the adjustment compared to the raw prediction data.

The position of the satellite is predicted twice as reliably when the least squares adjustment is performed.

Analyzing the variation between the maximum value of the semi-axis b versus a, it turns out to be, in each method, 1569 km in the least squares method compared with the 1626 km as seen in Table 2.

Table 2. Real trajectory parameter error versus predicted one.

	Real Value	**Predicted Value**
Deviation	1569 km	1626 km

Therefore, the mathematical method proposed in this paper is sufficiently accurate for satellite trajectory prediction and estimation, taking into account the safety space maintained around each satellite.

4. Conclusions

As the number of satellites and spatial objects is growing exponentially, the amount of debris is increasing in the same proportion. This provokes a huge long term problem that must be controlled by providing reliable prediction methods to estimate the position of an object in space at a given time. In this paper, we have analyzed a set of data belonging to 600 satellites for 26 days. According to this information, we noticed some inconsistencies in the measurement intervals, for which we have presented a prediction algorithm that allows us to correct these inconsistencies and allows us to calculate an estimation of the satellite position in the future with higher reliability. With this prediction, we have estimated an error and limited the maximum error we can have in the prediction. It allows us to know the current position of the satellite, to predict the final position with a bounded error, and as new positions are acquired, the result can be updated.

As a general evaluation, the pros of this paper are the rigorous study that we have performed to predict new information in base to the data that we had with a high accuracy level, limiting the error in such prediction. Otherwise, we have applied a method that works and is able to locate the position of an object in a given time with a low margin of error. On the other hand, we have still work to do as we still have a lack of information to provide a statistical study as not all the information of the satellites was uniform, so it makes this process more complex. One key point can be the information related to the velocity of the object. In this study, we have not used this information but for future work, we will add this information to try to improve the positional error estimation despite being less than 2 km in both methods, which demonstrates the accuracy of the estimation.

Author Contributions: Conceptualization, M.A.S.-N. and C.P.; methodology, C.P., J.A.O.; software, A.V.-M. and C.P. and M.A.S.-N.; validation, M.A.S.-N. and C.P.; formal analysis, A.V.-M. and C.P.; investigation, C.P. and A.V.-M. and J.A.O.; resources, J.A.O.; data curation, C.P.; writing—original draft preparation, C.P. and A.V.-M. and M.A.S.-N.; funding acquisition, J.A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This This research was funded by FEDER and the State Research Agency (AEI) of the Spanish Ministry of Economy and Competition under grant SAFER: PID2019-104735RB-C42 (AEI/FEDER, UE).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work has been partially supported by FEDER and the State Research Agency (AEI) of the Spanish Ministry of Economy and Competition under grant SAFER: PID2019-104735RB-C42 (AEI/FEDER, UE).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geospatial World. Available online: https://www.geospatialworld.net/blogs/how-many-satellites-orbit-earth-and-why-space-traffic-management-is-crucial/ (accessed on 3 June 2021).
2. European Space Agency. The Impact of Space Debris Report. Available online: https://www.esa.int/Safety_Security/Space_Debris (accessed on 3 June 2021).
3. UN. Committee on the Peaceful Uses of Outer Space. Scientific and Technical Subcommittee. In *Technical Report on Space Debris: Text of the Report/Adopted by the Scientific and Technical Subcommittee of the United Nations Committee on the Peaceful Uses of Outer Space*; UN: New York, NY, USA, 1999; ISSN 9211008131.
4. The United States Government Orbital Debris Mitigation Standard Practices (ODMSP). Available online: https://orbitaldebris.jsc.nasa.gov/ (accessed on 1 June 2021).

5. Achenbach, J. Debris from Satellites' Collision Said to Pose Small Risk to Space Station. Available online: https://www.washingtonpost.com/wp-dyn/content/article/2009/02/11/AR2009021103387.html (accessed on 1 June 2021).
6. LeBlanch, P.; Chinese Rocket Debris Is Expected to Crash into Earth Soon. It's Not the First Time. Available online: https://edition.cnn.com/2021/05/05/politics/chinese-rocket-debris-space-earth-explainer-scn/index.html (accessed on 5 June 2021).
7. Space Track. Available online: https://www.space-track.org (accessed on 25 June 2021).
8. Institute of Astronomy of the Russian Academy of Sciences. Available online: http://www.inasan.ru/en/scientific_activities/databases/ (accessed on 20 June 2021).
9. In the Sky–Livemap of Satellite Positions. Available online: https://in-the-sky.org/satmapworldmap.php (accessed on 15 June 2021).
10. JPL Small-Body Database Search Engine. Available online: https://ssd.jpl.nasa.gov/sbdbquery.cgi (accessed on 10 June 2021).
11. Mitchell, T.M. "Machine Learning". In *McGraw-Hill Series in Computer Science*, 1st ed.; McGraw-Hill Education: New York, NY, USA, 1997.
12. Peng, H.; Bai, X. Machine learning approach to improve satellite orbit prediction accuracy using publicly available data. *J. Astronaut. Sci.* **2019**, *67*, 762–793. [CrossRef]
13. Peng, H.; Bai, X. Improving Orbit Prediction Accuracy through Supervised Machine Learning. *Adv. Space Res.* **2018**, *61*, 2628–2646. [CrossRef]
14. Peng, H.; Bai, X. Artificial Neural Network–Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *J. Spacecr. Rocket.* **2018**, *55*, 1248–1260. [CrossRef]
15. Barton, K.E.; McLaughlin, C.A. Long short term memory neural networks for the prediction of localized atmospheric density for orbit determination. In Proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT, USA, 19–23 August 2018.
16. Ertl, C.A.; Christian, J.A. Identification of partially resolved objects in space imagery with neural networks. In Proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT, USA, 19–23 August 2018.
17. Smet, S.D.; Scheeres, D.J.; Parker, J.S. Systematic exploration of solar gravity driven orbital tranfers in the Martian system using artificial neural networks. In Proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT, USA, 19–23 August 2018.
18. Jia, S.Y.; Shan, J.J. Neural network-based adaptive sliding mode control for gyroelastic body. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 1519–1527. [CrossRef]
19. Xia, X.; Jia, Y.; Xu, S.; Wang, X. An adaptive nonsingular terminal sliding mode tracking control using neural networks for space manipulators actuated by CMGs. In Proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT, USA, 19–23 August 2018.
20. Kaggle. Available online: https://www.kaggle.com/idawoodjee/predict-the-positions-and-speeds-of-600-satellites (accessed on 10 April 2021).
21. Acebes, F.; Pereda, M.; Poza, D.J.; Pajares, J.; Galán, J.M. Stochastic Earned Value Analysis using Monte Carlo Simulation and Statistical Learning Techniques. *Int. J. Proj. Manag.* **2015**, *33*, 1597–1609. [CrossRef]
22. Figueroa, B. Algunos Metodos Para Ajustar Elipses. Universidad Nacional Autónoma de México. Available online: http://132.248.9.41:8880/jspui/handle/DGB_UNAM/TES01000340915 (accessed on 10 April 2021).
23. Brown, R. Repository. Available online: https://www.mathworks.com/matlabcentral/profile/authors/870621 (accessed on 10 April 2021).