BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

LOW EARTH ORBIT TRAJECTORY PREDICTION USING A MACHINE LEARNING APPROACH

VYUŽITÍ STROJOVÉHO UČENÍ PRO PREDIKCI TRAJEKTORIE OBJEKTU NA NÍZKÉ OBĚŽNÉ DRÁZE

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **PŘEMEK JANDA**
AUTOR PRÁCE

SUPERVISOR **Ing. JIŘÍ NOVÁK**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

148537

Institut: Department of Computer Graphics and Multimedia (UPGM)
Student: **Janda Přemek**
Programme: Information Technology
Specialization: Information Technology
Title: **Low Earth orbit trajectory prediction using a machine learning approach**
Category: Artificial Intelligence
Academic year: 2022/23

Assignment:

1. Study the basic principles of orbital mechanics and describe the influence of aerodynamic drag and non-spherical shape of Earth on orbital motion and trajectory.
2. Compose a training dataset consisting of ephemerides from a database of tracked objects in Earth orbit.
3. Train a machine learning model to predict the object motion with higher accuracy.
4. Implement the motion model and evaluate the achieved results.

Literature:

- AGGARWAL, Charu C. *Neural Networks and Deep Learning*. Cham: Springer International Publishing. ISBN 3319944622. Dostupné z: doi:10.1007/978-3-319-94463-0
- CURTIS, Howard D. *Orbital mechanics for engineering students*. Amsterdam ; Boston: Elsevier Butterworth-Heinemann, 2005, xv, 673 s. : il. ; 26 cm. ISBN 0-7506-6169-0.

Requirements for the semestral defence:
1. a 2. assignment points.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Novák Jiří, Ing.**
Head of Department: Černocký Jan, prof. Dr. Ing.
Beginning of work: 1.11.2022
Submission deadline: 10.5.2023
Approval date: 8.3.2023

Abstract

Precise orbital predictions are vital for every-day Low Earth orbit satellites operations to evade collision. The study aims to analyse historical trajectory data in a form of the classical orbital elements. By leveraging machine learning, data are aggregated, processed and finally used for orbit prediction. The best forecast results have been achieved using Long Short-Term Memory Neural Network.

Abstrakt

Přesné orbitální predikce orbit satelitů na nízé oběžné dráze Země jsou nezbytné pro každodení provádění operací, tak aby se předešlo jejich srážkám. Tato práce se zaměřuje na analýzu histrorických dat trajektorií, tvořených z klacických dráhových elementů. S využitím strojového učení, data jsou sdružena, zpracována a na závěr užita pro predikci dráhy orbity. Nejlepších výsledků předpovědí bylo dosaženo s neuronovou sítí s doulodobou a krátkodobou pamětí (LSTM).

Keywords

Machine Learning, Deep Learning, Orbital Mechanics, Satellites, Low-Earth Orbit, Collision avoidance, Orbit Prediction, Perturbations, Simulation

Klíčová slova

Strojové učení, Hluboké Učení, Orbitální Mechanika, Satelity, Nízká Obježná Dráha, Předcházení Kolizím, Perturbace, Simulace

Reference

JANDA, Přemek. *Low Earth orbit trajectory prediction using a machine learning approach.* Brno, 2023. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Jiří Novák

Rozšířený abstrakt

Práce se zabývá predikcí drah těles na nízké oběžné dráze Země. Cílem práce je sestrojit nástroj schopný dostatečně přesně předpovědět budoucí trajektorii tělesa. Pohyb tělesa ovšem není vůbec jednoduché popsat. Existuje mnoho matematických modelů, které jsou závislé na mnoha parametrech a tvoří soustavu rovnic, která by vystihla všechna specifika orbity jako jsou například perturbace. Dalším řešením vyjma analytických a numerických metod je strojové učení. To pracuje pouze s poskytnutými daty. Dle povahy poskytnutých dat se bude odvíjet kvalita predikce, proto je třeba data nejdříve zpracovat, odstranit chybné záznamy apod. Práce je rozdělena do sedmi kapitol, kde první poskytne náhled na problematiku a motivaci k jejímu vypracování. Další 2 kapitoly obsahují teoretickou část práce, kapitola čtvrtá až šestá se zaměřuje na praktickou část vlastní implementace. V závěrečné sekci jsou shrnuty výsledky a je zmíněn případný následný vývoj.

Druhá kapitola se zabývá do detailu orbitální mechanikou. Jsou zde vysvětleny základní pojmy týkající se vlastností elipsy, referenčních systémů, souřadných systémů, či klasických orbitálních elementů (známé také pod názvem dráhové elementy). Důležitým bodem této kaitoly je samotné završení, kde je zaveden pojem stavového vektoru.

Následuje kapitola popisující pohyb tělesa na nízké oběžné dráze Země. V úvodu je představen problém dvou těles osvobozený od vnějších vlivů. Ten dostatečně přesně nastíní, jak by mohla fungovat dynamika satelitu. Kapitola poté pokračuje představením hlavních (perturbačních) vlivů ovlivňujících dráhu takového tělesa jako je např. aerodynamický atmosferický odpor, vliv slunečního záření či gravitační působení Měsíce a jiných těles.

Další kapitola se zaměří na práci s daty. Konktrétně se jedná o jejich získání, agregaci, zpracovaní, seřazení a filtraci. S přihlédnutím k množství dat je možné provést selekci dat a vybrat pouze ty, které budou vhodné pro následné strojové učení. Cílem zde tedy bylo vyfiltrovat zdroje, které obsahovaly vysoké množství chybných měření, dlouhé prodlevy mezi jednotlivými pozorováními, byly ovlivněny propulzí raketového motoru apod. Dále je v práci rozebráno shlukování dat (clustering). Vzhledem k velkému počtu parametrů jsou zde použity a popsány metody redukce dimenzionality/parametrů.

Kapitola simulace v úvodu popisuje různé způsoby prezentace získaných dat ve formě 2D grafu či 3D simulace. Krátce je zde také zmíněna efektivita jedné z hlavních komponent, která se počítá např. při převodu souřadnic. Záměrem poslední sekce je prezentace předpovědi počtu slunečních skvrn na Slunci, která do velké míry koreluje se sluneční aktivitou. K tomu bylo využito několik metod strojového učení jako jsou rozhodovací stromy (Decision trees) či náhodný les (Random forest).

Předposlední kapitola se věnuje samotnému strojovému učení pro predikci dráhy. Postupně jsou zde zavedeny pojmy vysvětlující některé důležité charakteristiky zvoleného přístupu. Okrajově je nastíněn Gaussovský proces (Gaussian process), Poté se přistoupí k implementaci zvolené neuronové sítě s dlouhou a krátkou pamětí (LSTM). Mimo jiné jsou zde rozebrány různé modely neuronové sítě a jejich kvalita. K závěru kapitoly jsou získané výsledky prezentovány a zhodnoceny.

Low Earth orbit trajectory prediction using a machine learning approach

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Ing. Jiří Novák. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

....
Přemek Janda
May 8, 2023

Acknowledgements

I would like to thank my supervisor Ing. Jiří Novák for patience and professional guidance throughout the creation of the work. I am equally grateful to my family for their support.

Contents

Nomenclature

A Ascending node. 5, 8

D Descending node. 5

E Eccentric anomaly. 9, 10

M Mean anomaly. 7, 9, 10, 12, 21, 29, 31, 42

P_a Furthest point on the ellipse. 5

P_p Closest point on the ellipse. 5, 9

Ω Right ascension of ascending node RAAN. 7, 8, 29, 31, 42

α Right ascension. 5, 6

Υ Vernal/spring/march equinox. 5, 8

δ Declination. 5, 6

ν True anomaly. 7, 9, 10, 12

ω Argument of perigee ARGP. 7, 8, 21, 29, 31, 42

a Semi-major axis. 5, 7, 9, 34, 42

b Semi-minor axis. 5

c Linear eccentricity. 5

e Eccentricity. 21, 42

i Inclination. 7, 8, 21, 28, 34, 42

p Semi-latus rectum. 5, 18

Chapter 1

Introduction

With the growing number of objects launched onto the Earth's orbit [8], collision avoidance and satellite navigation systems have become increasingly more important. Since the inception of space exploration in 1957, number of resident space objects (RSOs) has gradually increased. It is estimated that 36 500 space debris objects greater than 10 cm are currently in the Earth's orbit, with likelihood for more than 600 break-ups [4]. Thus Space debris and the potential for collisions with satellites has become a major concern in the field of space exploration.

As the number of satellites in the orbit grows, the risk of collision also increases. Possible crashes not only cause loss of crucial equipment and money, but also generate additional fragments in the course of an impact. Consequentially chained reactions [13] occur creating more debris with each revolution. These remains, yet very small (less than 1 cm), can account for irreparable damage or even repeal entire mission.

To reduce such events, high-accuracy orbit prediction model is absolutely essential. Present methods for orbit prediction are based on dynamic models, founded on physics and analytical solutions, approximate features like perturbations, RSO shape, mass etc. subsequently diverging from the correct trajectory. Current methods are capable to obtain reasonable forecast up to 5 days in the future after which error deviation reaches orders of kilometers [1], thus making it irrelevant.

There are also several studies with machine learning approach in mind. Such approaches neglect physical features of the RSOs and focus primarily on large amounts of historical data. There are studies focusing on two-line elements (TLE) as the input data [16], or propagation methods such as Support Vector Machines (SVM) [26], Gradient Boosting (GB) [17], Gaussian Process (GP) [27] and finally Long Short-Term Memory (LSTM) [32], selected for the prediction in this work.

The thesis will begin by theoretical overview of orbital mechanics, including different Earth-centered reference systems, the role of orbital elements in obit determination and prediction. In the next chapter, RSO movement in Low-Earth orbit will be probed more in detail with focus on forces acting on the body and deviation from its orbit. The importance of probabilistic and empirical models will be then discussed. Following chapters describe practical part of the thesis. Successively, reader will be well acquainted with obtained dataset and its preparation for machine learning. In the last chapters, prediction method using machine learning and its result will be presented.

Chapter 2

Orbital mechanics

Centuries ago, humanity struggled to resolve principles of celestial bodies motion. Presently, we are endowed with laws of force, gravitational attraction, Kepler's laws, general relativity and many other principals, by means of we describe processes above, how planets, stars, moons, satellites etc. revolve. In this section, reference systems used in this work and fundamentals of Kepler's and Newton's laws will be discussed.

2.1 Reference systems

The Earth's orbit can not be regarded as a stable system, for couple of reasons. In short term and smaller scale, there are many factors influencing RSOs orbit called perturbation (more in section 3.1). For longer periods, one has to take into account precession, nutation, polar motion and others. For instance precession is the process of rotation of the Earth's axis orientation, yet for the time span of human life quite insignificant, for precise measurements non-negligible. Main reason is the gravitational pull of the Moon and the Sun on the Earth's equatorial mass imbalance. One full cycle takes about 25 772 years, which corresponds to an 8' arc during one decade [7]. Other influences are much smaller in magnitude. To evade this effect, an inertial system must be accordingly locked in space and time.

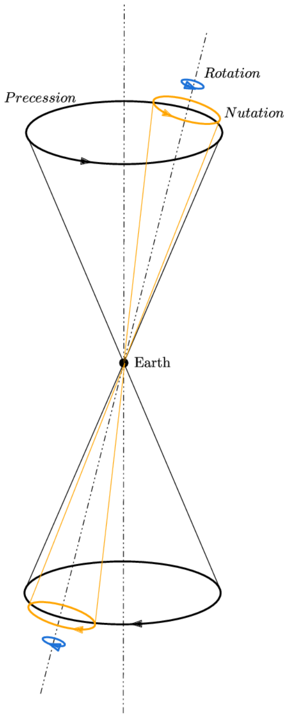


Figure 2.1: The three Earth rotations.

2.1.1 BCRS vs GCRS

There are two main approaches regarding coordinate systems. First one is the Barycentric celestial reference system (BCRS) suitable for objects located outside the gravitational vicinity of Earth. It is used in this work to obtain location from ephemerides of the Planets to count third-body perturbations. The center is defined by the barycenter of the entire Solar system. Other being Geocentric celestial reference system (GCRS) which is preferred for bodies closer to the Earth such as space debris and satellites. The origin of the coordinate system lies in the center of mass of the Earth.

Location of the RSO can be described in various ways. For purposes of orbital mechanics, it is suitable to choose ECI (Earth-centered inertial). Such a frame does not accelerate or rotate in space and stays stable with respect to the stars. The one used primarily throughout this work is the most profound one – ECI J2000 (J2K) defined by orientation of the Earth rotation axis and Equatorial plane on January 1, 2000, at 12:00 TT (Terrestrial Time). Another reference frame referred to as perifocal frame rotates system as if RSO orbited on the XY plane (from ECI) with periapsis 2.1.2 aligned with X axis.

2.1.2 Equatorial reference frame

This reference frame is suitable for static display of the Earth. All of the Earth's motions are evaded, thus making it the best option to display Low-Earth orbit (LEO) trajectories: X – vernal point, Y – points 90° east on the equatorial plane and Z – north celestial pole

Many notations describing position of the object on the sky (such as ephemerides) use right ascension α, declination δ and others. Below, there is summary of the terms used during the next chapters:

- Vernal/spring/march equinox Υ – fictional point in the sky where equatorial plane and ecliptic (orbital plane of planets) intersect.
- Right ascension α – angular distance of the body measured from the Υ.
- Declination δ – angular distance of the body measured from the equatorial plane.
- Ascending node A – point where body crosses equatorial plane on its way upwards.
- Descending node D – opposite to the A.

Related to the ellipse:

- Periapsis P_p – closest point on the ellipse of the orbiting body to the central body.
- Apoapsis P_a – furthest point.
- Semi-major axis a – half of the distance between periapsis and apoapsis.
- Semi-minor axis b – distance from the center to its closest point on the ellipse.
- Linear eccentricity c – distance from the center to any of the two foci.
- Semi-latus rectum p – length of the chord through one of the foci, perpendicular to the major axis [23].

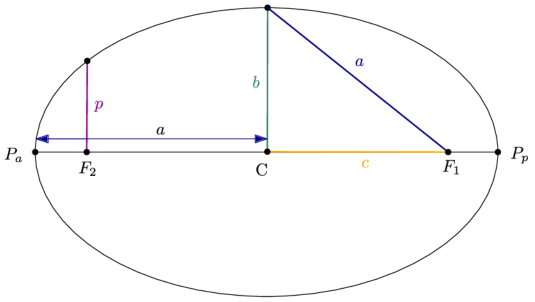


Figure 2.2: Figure depicts ellipse features, if central body was in F_1.

2.2 Inertial position and velocity

Given a frame of reference (such as ECI), the position of RSO in time t is determined by position vector $\mathbf{r} \in \mathbb{R}^3$ comprised of x, y and z coordinates stated as

$$\mathbf{r}(t) = x(t)\,\hat{\mathbf{X}} + y(t)\,\hat{\mathbf{Y}} + z(t)\,\hat{\mathbf{Z}} = x\,\hat{\mathbf{X}} + y\,\hat{\mathbf{Y}} + z\,\hat{\mathbf{Z}} \tag{2.1}$$

where $\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$ and $\hat{\mathbf{Z}}$ are the unit vectors $(1,0,0), (0,1,0), (0,0,1)$, respectively [3]. Velocity of a given object is first time derivative of the position vector

$$\mathbf{v} = \frac{dx}{dt}\hat{\mathbf{X}} + \frac{dy}{dt}\hat{\mathbf{Y}} + \frac{dz}{dt}\hat{\mathbf{Z}} \tag{2.2}$$

Acceleration can be stated as second derivative of the position vector $\mathbf{r}$ or first derivative of its velocity $\mathbf{v}$, thus

$$\mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{r}} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{r}}{dt^2}. \tag{2.3}$$

Another way to represent RSO in space is to use equatorial coordinates. Unlike Cartesian coordinates, equatorial components are defined as angles α and δ, with distance r.

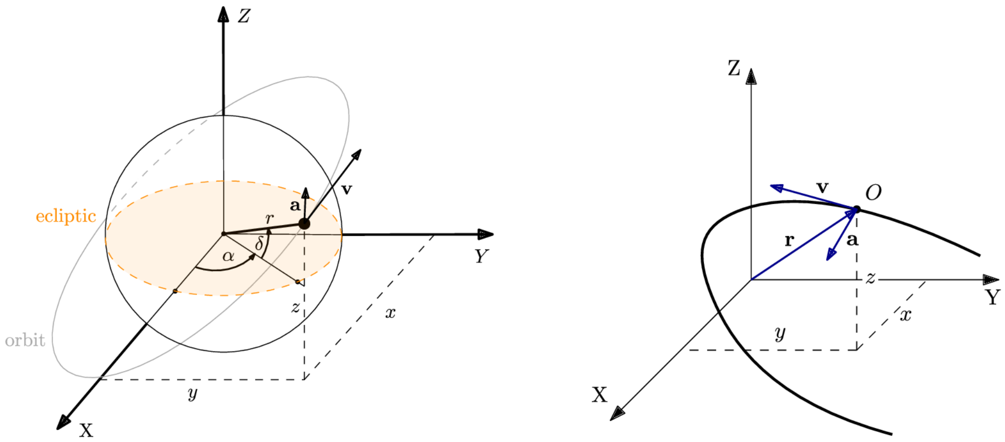


Figure 2.3: Left-hand graphic shows RSO position determination using equatorial coordinates using right ascension α, declination δ and distance from origin r. Figure shows position of object O in Cartesian coordinates XYZ with indicated velocity and acceleration vectors $\mathbf{v}$ and $\mathbf{a}$, respectively.

It should be noted here that both systems are convertible. It is sometimes more convenient to leverage equatorial rather than Cartesian and vice versa, thus conversion:

$$\begin{aligned} x &= r\cos\delta\cos\alpha & & & r &= \sqrt{x^2+y^2+z^2} \\ y &= r\cos\delta\sin\alpha & &\leftrightarrow & \delta &= \arctan\left(\frac{\sqrt{x^2+y^2}}{z}\right) \\ z &= r\sin\delta & & & \alpha &= \arctan\left(\frac{y}{x}\right) \end{aligned}$$

2.3 Keplerian elements

Johannes Kepler based his work on the study of celestial measurements acquired by Tycho de Brahe. His approach substituted previous circular orbits based on epicycles with elliptical trajectories. His work resulted in 3 laws:

1. The orbit of each planet is an ellipse with the Sun at one focus.
2. The heliocentric radius vector of each planet sweeps over equal area in equal time.
3. The square of the orbital period is proportional to the cube of the ellipse's semi-major axis.

The first one describes the ellipse – a somewhat oblique circle with a a connecting the most distant and the closest points. Perpendicular to it lies a semi-minor axis. The two points are called aspis. According to the reference body they can be called differently such as (perihelion, aphelion). The Second law used for calculation of the position of the body on its trajectory. Keplerian elements are used to uniquely describe position and orbit trajectory of the body in space. It is sometimes called (classical) orbital elements and can be used interchangeably with term Keplerian elements or orbital elements throughout this work. There are 4 groups with different functionalities.

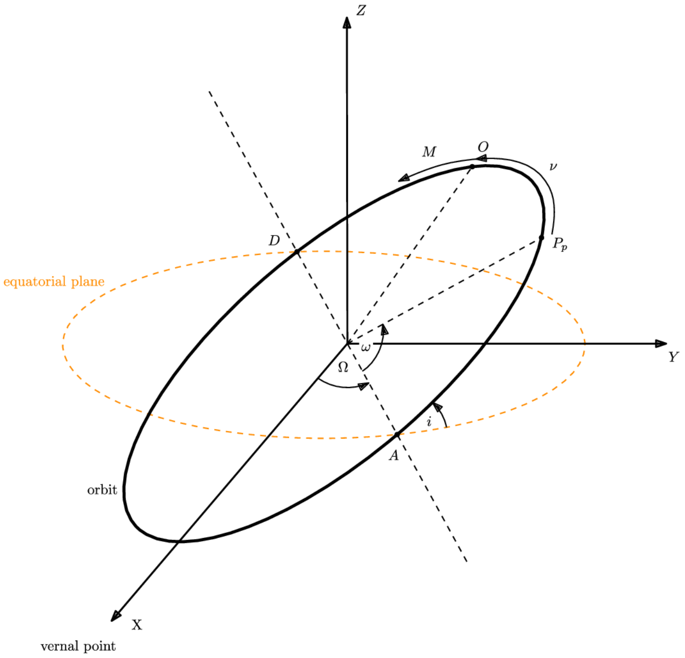


Figure 2.4: This is a XYZ_{ECI} coordinate system with the Earth in the center, more precisely in the focal point. Points P, A, D, O represent preiapsis, ascending node, descending node and current position of the RSO respectively. Other symbols are Keplerian elements ω argument of periapsis, Ω – right ascension of the ascending node, i – inclination, M – mean anomaly and ν – true anomaly. Orange dashed line represents ecliptic plane.

2.3.1 Size and shape

These elements describe the shape and size of the orbit.

1. Semi-major axis a – half of the length of the longest line, which goes through both foci.
2. Eccentricity e – nonnegative number defines rate of obliqueness of the ellipse, zero means perfect circle, $e \in (0; 1)$ represent ellipse, $e = 1$ denote parabola and finally hyperbola is determined by $e > 1$.

2.3.2 Keplerian angles

These define the orientation of the trajectory plane with respect to the ♈.

3. Argument of periapsis ω – the angle between the ascending node and the periapsis, describes rotation of the plane.
4. Inclination i – the angle subtending the plane of orbit and reference plane (usually equator or ecliptic), describing plane tilt.

$$i = \begin{cases} \text{Posigrade} & \dots \quad 0 < i < 90^\circ \\ \text{Polar} & \dots \quad i = 90^\circ \\ \text{Retrograde} & \dots \quad 90^\circ < i < 180^\circ \end{cases}$$

5. Right ascension (longitude) of the ascending node Ω – the angle between the vernal point ♈ and the ascending node A, depicting horizontal rotation of the plane.

As mentioned above, these parameters symbolize in 3D euclidean space Yaw-roll-yaw sequence also known as ZXZ Euler rotation [10].

$$R_z(\omega) = \mathrm{Yaw}_1(\omega) = \begin{bmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$R_x(i) = \mathrm{Roll}(i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \tag{2.5}$$

$$R_z(\Omega) = \mathrm{Yaw}_2(\Omega) = \begin{bmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

Applying gradually 2.4, 2.5 and 2.6 to the initial trajectory plane results in dot product, consequently forming desired transformation

$$
\begin{aligned}
R(\Omega, i, \omega) &= \begin{pmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{pmatrix} \begin{pmatrix} \cos\omega & -sin\omega & 0 \\ sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\
&= \begin{pmatrix} \cos\Omega\cos\omega - \cos i\sin\Omega\sin\omega & -\cos\Omega\sin\omega - \cos i\sin\Omega\cos\omega & -\sin i\sin\Omega \\ \sin\Omega\cos\omega + \cos i\cos\Omega\sin\omega & \cos i\cos\Omega\cos\omega - \sin\Omega\sin\omega & \sin i\cos\Omega \\ -\sin i\sin\omega & -\sin i\cos\omega & \cos i \end{pmatrix}
\end{aligned}
\tag{2.7}
$$

which in a sense describes conversion from perifocal reference frame to ECI reference frame.

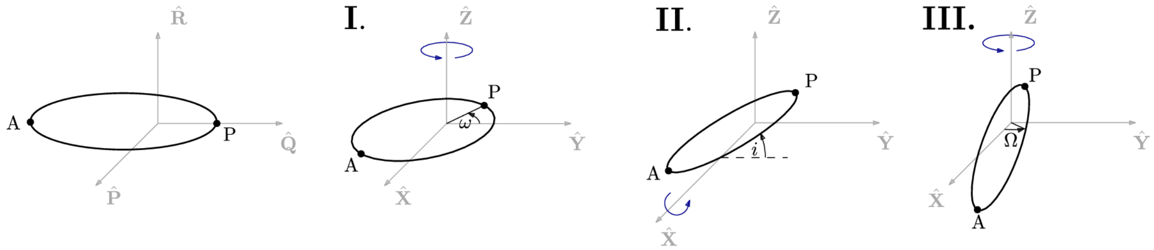


Figure 2.5: In the very left picture initial ellipse in perifocal reference frame $\hat{\mathbf{P}}\hat{\mathbf{Q}}\hat{\mathbf{R}}$ is depicted. Then gradually $Z\,X\,Z \rightarrow \omega\,i\,\Omega$ transformations are applied. This results in trajectory in ECI reference frame.

2.3.3 Anomalies

In astronomy, anomalies are apparent motions of the planets. They are expressed in as angles in degrees or radians. There are three types of anomalies, which serve to obtain the current position of the body on the elliptical path. There is usually given a M together with a, however, sometimes they can be replaced by true ν and angular momentum.

6. Mean anomaly M – a fraction of an elliptical orbit as if trajectory was not an ellipse rather a circle with radius of a with center in focal point (where the central body is located). Thanks to its linear increment throughout entire revolution, it suits well to many computations.

7. True anomaly ν or θ – an angle between periapsis P_p and current position of the body on the orbit.

8. Eccentric anomaly E – more complex compared to the previous anomalies. It is defined as an angle between periapsis and point, where orbiting body is not located, but perpendicular to the major axis on the auxiliary circumscribed circle.

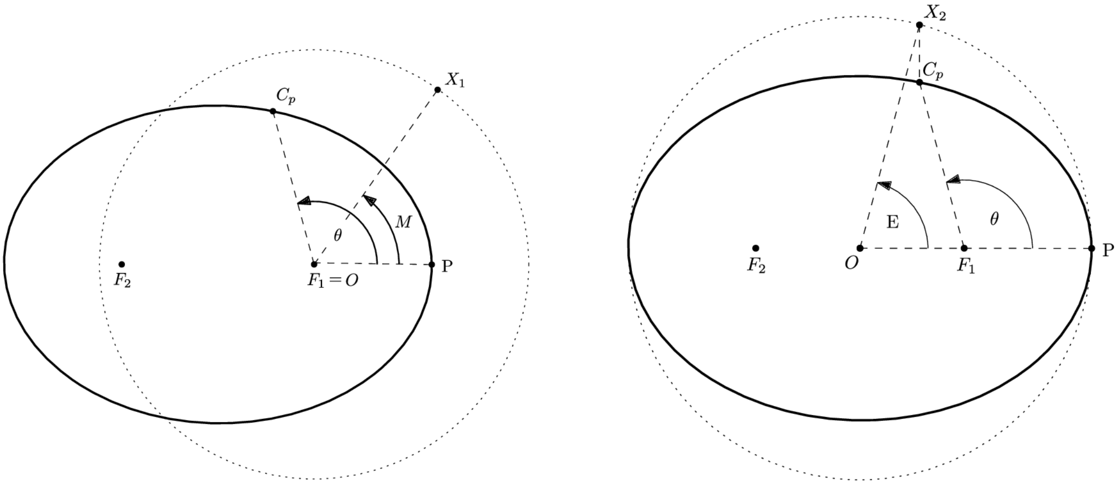


Figure 2.6: The left picture shows mean anomaly. Eccentric and mean anomaly are depicted in the right picture. O stands for the center of an auxiliary circle. Points X_1 and X_2 represent the imaginary position on the auxiliary circle for mean anomaly and eccentric anomaly resectively. Point C_p marks current position of the RSO in time t. P_p is periapsis. Foci points are F_1 and F_2, whith central body located in the F_1 point.

Unlike M, both ν and E are non-linear. Getting eccentric anomaly is very handy because it is closely related to the M by Kepler's equation 2.8.

$$M = E - e \cdot sin(E) \quad \rightarrow \quad 0 = M - E + e \cdot sin(E) \tag{2.8}$$

Only M can be obtained from space-track.org [31]. However, there is no closed-solution form to obtain E given M. By rearranging the equation to the form equal to 0, it is possible (by root finding) to numerically approximate the value of E. Rewritten form of the Kepler's equation can be found in the right-hand side of the equations 2.8.

One of the root finding algorithms is Newton-Raphson method. Its iteration step is described in equation 2.9. This method only needs initial guess and required precision ϵ. Trajectories of the satellites are generally close to circular orbits (meaning e is close to zero) thanks to that, M is highly suitable as initial starting point x_0 for the method. Due to the continuous nature of the function, desired root doesn't usually conform to 0 and number of steps or precision must be defined beforehand. Accuracy 10^{-15} has been chosen for its best performance-precision ratio (more on this in section 5.1).

$$E_{n+1} = E_n - \frac{f(n)}{f'(n)} \tag{2.9}$$

There are couple ways of setting initial guess. Typical value is $E_0 = M$ or $E_0 = \pi$ [22]. Faster and more robust possibility suggests [21] where initial estimate

$$E_0 = \frac{\pi}{2}\bar{e}\left(\operatorname{sgn}(\bar{e})\sqrt{1 + \frac{M}{e\,\bar{e}^2}} - 1\right) \quad \text{where} \quad \bar{e} = \frac{\pi}{4e} - 1 \tag{2.10}$$

combined with Halley's equation

$$E_{n+1} = E_n - \frac{f}{f'\left(1 - \frac{f\,f''}{2f'^2}\right)} = E_n - \frac{2ff}{2f'^2 - f\,f''} \tag{2.11}$$

should converge usually the fastest.

Having that done, it is possible to make use of E to compute the position of a point moving in a Keplerian orbit.

$$f \equiv E - e\sin E - M \qquad f' \equiv 1 - e\cos E \qquad f'' \equiv e\sin E \tag{2.12}$$

In many computations (such as anomaly conversions 2.15–2.19), it is convenient to use trigonometric functions sin and cos instead of deriving the explicit values, for their mutual conversions can be expressed eg. by infinite series. This comes in handy when converting from perifocal to ECI frame 2.21–2.22. Such equations and conversions between orbital elements are listed below:

$$\cos(\omega + \nu) = \frac{r_x}{r}\cos\Omega + \frac{r_y}{r}\sin\Omega \tag{2.13}$$

$$\sin(\omega + \nu) = \frac{r_z}{r\,\sin i} \tag{2.14}$$

$$\cos\nu = \frac{\cos E - e}{1 - e\,\cos E} \tag{2.15}$$

$$\sin\nu = \frac{\sqrt{1 - e^2}\sin E}{1 - e\,\cos E} \tag{2.16}$$

$$\cos E = \frac{\cos\nu + e}{1 + e\,\cos E} \tag{2.17}$$

$$\sin E = \frac{\sqrt{1 - e^2}\sin\nu}{1 + e\,\cos E} \tag{2.18}$$

$$r = a(1 - e\,\cos E) \tag{2.19}$$

$$p = a(1 - e^2) \tag{2.20}$$

$$\mathbf{r} = r\,(\cos\nu\hat{\mathbf{P}}, \sin\nu\hat{\mathbf{Q}}) \tag{2.21}$$

$$\mathbf{v} = \sqrt{\frac{\mu}{p}}(-\sin\nu\hat{\mathbf{P}} + e + \cos\nu\hat{\mathbf{Q}}) \tag{2.22}$$

2.3.4 Epoch

Due to the precession and other influences 2.1, there is a need to set a reference time for coordinates to be aligned properly. Changing the epoch results in conversion of the coordinates. There are several possible coordinate conversions, however, they won't be discussed here for their exceeding lengths.

2.4 State vector

In previous two sections 2.2, 2.3, different approaches to define any object in space have been discussed. In order to capture RSO in time and space and simplify notation, state vector X_{rv}, $\dot{X}_{force}$ and X_{COE} is introduced. As seen in section 2.3, using position vector **r** together with velocity vector **v** and acceleration vector **a**, a state vector and a force vector in Cartesian coordinate system could be defined as

$$\mathbf{X}_{rv} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \quad (2.23) \qquad \dot{\mathbf{X}}_{force} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T \quad (2.24)$$

It has been proven that Cartesian vectors 2.23 and 2.24 can be leveraged to define set of classical orbital elements in form of 2.25 or similarly by replacing M with ν in equation 2.26:

$$\mathbf{X}_{coe} = \begin{bmatrix} a & e & i & \omega & \Omega & M \end{bmatrix}^T \quad (2.25) \qquad \mathbf{X}_{coe} = \begin{bmatrix} a & e & i & \omega & \Omega & \nu \end{bmatrix}^T \quad (2.26)$$

It is rather useful to be able to express state vector in various forms and convert between them, since every resource offers different set of orbital elements. Moreover various prediction methods, for instance analytical based ones use usually X_{rv}, $\dot{X}_{force}$ state vectors, while ML X_{COE}.

Chapter 3

Movement of an object in LEO

A two body problem together with its implications for this work and Earth-like environment will be explained in this chapter. It began by Newton in 1687 in his work `Principia` described his 3 well-known laws of motion. Based on them he established the force of gravitational attraction between two point masses as follows

$$F = \frac{GM_1M_2}{r^2} \tag{3.1}$$

where G is the Gravitational constant, M_1 and M_2 are two point masses and r is the distance between them. This relation is of use in two body problem. Common example being indeed one dominant body — the Earth — and much less massive object — satellite. Let us impose $\mathbf{R_1}$ and $\mathbf{R_2}$ positional vectors pointing from the origin of reference system to the objects M_1 and M_2 respectively. To keep track of position of the M_2 relative to the M_1, vector $\mathbf{r} = \mathbf{R_2} - \mathbf{R_1}$ will be used.

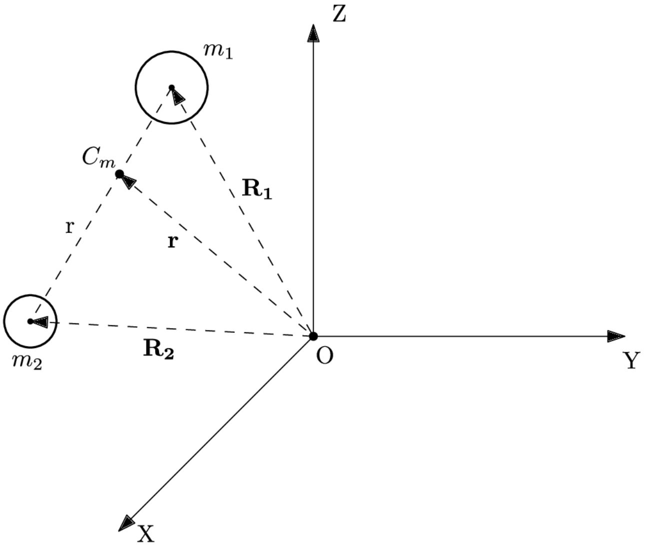


Figure 3.1: Three-component XYZ system with center in point O comprises of two bodies M_1 and M_2 with different masses $m_1 > m_2$. Therefore center of mass also called barycenter is shifted closer to the M_1.

Therefore, force of gravitational attraction posed on M_2 by M_1 is

$$F_{21} = -\frac{GM_1M_2}{r^2} \cdot \frac{\mathbf{r}}{r} = -F_{12} \tag{3.2}$$

Rewriting the equation 3.2 with acceleration relative to the inertial frame of reference using Newton's second law $F = ma$ gives

$$F_{12} = M_1\ddot{\mathbf{R}} \tag{3.3}$$
$$F_{21} = -M_2\ddot{\mathbf{R}} \tag{3.4}$$

Multiplying both equations 3.3 and 3.4 by M_1 and M_2 respectively and after several modifications and rearrangements, it eventually results in

$$\ddot{\mathbf{r}} = -\frac{G(M_1 + M_2)}{r^2} \cdot \frac{\mathbf{r}}{r} = -\frac{\mu}{r^3}\mathbf{r} \tag{3.5}$$

which depicts the motion of M_2 relative to M_1. This 3.5 differential equation can be solved numerically and results in undisturbed/unperturbed acceleration.

There are several takeaways from the two body problem which will be used in this work. Firstly, total sum of kinetic and potential energy is constant according to the expression 3.6. Secondly, by adding the 3.3 and 3.4 together means that the acceleration of the center of mass of the system (barycenter) is zero (see 3.7), thus it is convenient to reference to it as the origin of the coordinate system. Finally, generalization of the system to the Earth-satellite one yields $\mu = G(M_1 + M_2) \cong GM_1$. Therefore barycenter becomes the geometrical center of the Earth.

$$\zeta = \frac{\dot{r}\dot{r}}{2} - \frac{\mu}{r} \tag{3.6}$$
$$0 = M_1\ddot{\mathbf{R}} + M_2\ddot{\mathbf{R}} \tag{3.7}$$

3.1 Orbit Perturbations

As mentioned in section 2.3, every object revolving around its central body moves along elliptic trajectory. Ideally, disregarding all possible outer influences, the object could go infinitely along the same path. However, there are subtle forces pushing, pulling, decelerating, dragging, ... object from its undisturbed ideal path. Depending mainly on orbit altitude, forces other than central gravitation pull are called perturbations and can be more or less profound.

There are two types of perturbing forces:

- Non-conservative – kinetic and potential energy is lost or transformed into thermal energy, radiant energy, ...
  - Atmospheric drag.
  - Non-spherical Earth shape.
  - Solar Radiation Pressure (SRP).

  - Solid Earth tides.
  - Ocean tides.
  - Albedo.

- Conservative – total kinetic and potential energy remains the same
  - Main (central) body gravity force.
  - Third body gravitational influence.

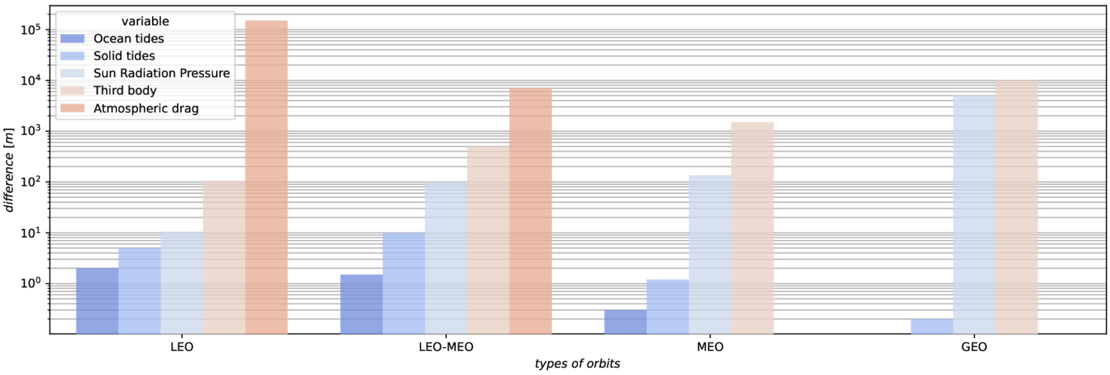


Figure 3.2: Significance of perturbation forces based on the distance form the RSO and the Earth [40].

In the picture 3.2, four distance domains are displayed, ordered by orbital distance from the closest to the furthest. Clearly the main influence regarding LEO RSOs is atmospheric drag, followed by third body perturbations, sun radiation pressure (SRP) and finally tidal forces. For the reasons explained more into detail below 3.1.2, atmosphere reaches only few hundreds of kilometers above the ground, thus with rising altitude, drag significance decreases. Tidal forces also lose its magnitude due to the quadratic orbit radius r^2 from 3.1.

3.1.1 Third-Body Perturbations

These forces are caused by Solar system objects, primarily Earth. Next two most dominant influencers are the Moon and the Sun. The rest of the planets play insignificant role for the LEO RSOs, however, higher the altitude, more significant the influence is. It can be seen in many papers with machine learning approach eg. by Peng and Bai [27], Liu [18] that Solar system planets are neglected completely and only two main protagonists Moon and Sun (Luni-Solar) attraction is taken into account. However, in other works [33], based on robust analytical solutions, planets are usually included into computation. Perturbing third-body acceleration can be expressed very conveniently in the ECI reference frame as [33]

$$\mathbf{a}_{3-body} = \sum_i \mu_i \left(\frac{\mathbf{r}_i - \mathbf{r}}{|\mathbf{r}_i - \mathbf{r}|^3} - \frac{\mathbf{r}_i - \mathbf{r}_e}{|\mathbf{r}_i - \mathbf{r}_e|^3} \right) \tag{3.8}$$

$$= \sum_i \mu_i \left(\frac{\mathbf{r}_i - \mathbf{r}}{|\mathbf{r}_i - \mathbf{r}|^3} - \frac{\mathbf{r}_i}{|\mathbf{r}_i|^3} \right) \tag{3.9}$$

where μ_i is gravitational parameter of the third body, $\mathbf{r}_i, \mathbf{r}$ and $\mathbf{r}_e$ are position vectors of the third-body, the RSO and the Earth. There are two fractions inside bracket in 3.8. First one could be described as the gravitational acceleration exerted by the third body on the satellite. Second term means acceleration implied by the same body on the Earth. Distance $\mathbf{r}_e$ is in ECI reference frame null, thus, can be omitted, resulting in 3.9.

Position vectors of the Solar system planets in different reference frames can be obtained from high precision ephemerides JPL DE430 through skyfield API (more about that in section 4.1.3).

3.1.2 Atmospheric drag

Atmospheric drag is caused by exchange of a momentum between spacecraft or space debris, which is in direct contact with surrounding air. Aerodynamical forces can be decomposed into three parts: lift, side slip and drag, which is the most dominant of the three. For LEO, atmospheric drag is the most profound perturbing force up to about 2 000 km above the ground [24], however, quickly dissipates and over 5 000 km its effect is negligible [40]. It can be expressed via equation

$$\mathbf{a}_{drag} = \frac{\rho}{\rho_0} B^* \mathbf{v}^2 \tag{3.10}$$

B^* is used in Simplified general perturbations model 4 referred to as SGP4 predictor (more on this topic in subsequent chapter 4.1.1). It is essentially empirical measure of the satellite's drag. It is usually derived from couple of measurements and then averaged to fit more than just atmospheric drag [40]. Its value is ordinarily very close to zero. $\mathbf{v}$ denotes RSO speed relative to the surrounding particles, ρ is air density and ρ_0 similarly but measured from perigee. B^* is described via equation

$$B^* = \frac{1}{2} \cdot \rho \cdot \frac{C_D A}{m} \tag{3.11}$$

where C_D defines drag coefficient, A is frontal area and m denotes RSO mass. Regarding the drag coefficient, the lower the number, less resistance does the surface impose to the frontal facing air. For satellite, C_D is relatively high, around 2.2 up to 3.0 [22], however, for space debris, whose proportions are not always known, shape could be approximated to sphere, implying the Cannonball model [15]. Cannonball model proposes scenario, where RSO is spherical body with homogeneous weight distribution, it can be leveraged in cases when enough data are not provided or known. Drag coefficient for a sphere is lower relative to the satellite's one. For perfect sphere 0.47 up to 0.8 for uneven, irregular shapes. Due to applied approximations, model by itself is not very accurate, thus it is only applicable for LEO trajectories, higher altitude could produce incorrect results.

Modeling of the atmospheric phenomena, such as density, is very difficult task. It depends on many factors like the Earth's pressure, temperature, day-night phase, solar activity, tides, etc. [40] Therefore, there are no high-precision analytical formulas describing the Earth's atmosphere, only empirical or highly-simplified expressions.

There are two types of air density models: static and dynamic. Static can be further divided into: altitudional – caused by the Earth's oblateness and longitudinal – accounting for mountain ranges. Another category are dynamic models. These are more complicated and requires more detailed information to construct. One of them is to determine air density as exponential method

$$\rho = \rho_0 \, e^{-\frac{r}{H}} \tag{3.12}$$

where ρ_0 is the atmospheric density at some reference height h from the ground and H is the density scale height of the atmosphere [22].

In several models, sun radio activity $F_{10.7}$ is included in computation. It is exactly 10.7 cm wavelength radio emission flux (2 800 MHz frequency). It covers several phenomena such as solar magnetic activity, solar ultraviolet and X-ray emissions, solar irradiance etc. [9] It serves for atmospheric density modeling, since it directly influences outer Earth's ionosphere layer. Due to this parameter, Earth's atmosphere experiences pressure and squashes a little. Side effect is increased magnetic phenomena occurring in the atmosphere, resulting in aurora borealis. There are couple more indicators of solar activity including K_p, sunspots etc, however, they do not play such significant role as $F_{10.7}$ (more on that in section 3.1.4).

In case of RSO predictions, atmospheric state, solar activity forecast are included. The are couple of complex models such as `DTM2000` and `NRLMSISE-00`. Drag temperature Model (DTM) is a semi-empirical model describing the temperature, density and composition of the Earth's thermosphere [27]. On the other hand, `NRLMSISE-00` is an empirical model, modeling temperatures of the Earth's atmosphere.

3.1.3 Non-spherical Earth

Keplerian equations assume that the Earth is perfect sphere, however surface as well as mass and density varies throughout the planet. The shape of the Earth is therefore called Geoid rather than perfectly round sphere. Oblateness is a measure of difference between equatorial R_{eq} and polar R_{pol} radius. Generally speaking, it is caused mainly by tidal effects of the orbiting Moon and centrifugal force acting outwards perpendicular to the rotational axis. Closer to the equator, this process is more substantial, creating bulge with difference of approximately 21 km [44] between Earth's polar and equatorial radius. The formula for the rate of radial irregularity is defined as follows

$$f = \frac{R_{eq} - R_{pol}}{R_{eq}} \tag{3.13}$$

thus, for perfect sphere f is equal to zero.

Consequently, as a result of the above described irregularity, RSO experiences higher gravitational pull near the equatorial plane. From mathematical perspective, it is convenient to leverage polar or equatorial coordinates since it could easily describe change in altitude or longitude only by one parameter, while preserving other elements. Perturbation caused by spherical oblateness could be refactored into 3 components as follows [25]

$$U(\delta, \alpha, r) = \frac{\mu}{r} + U_{zonal}(r, \delta) + U_{sectorial}(r, \alpha) + U_{tesseral}(r, \delta, \alpha) \tag{3.14}$$

where $\mathbf{x} = [\,\delta\ \alpha\ r\,]^T$ denotes position in equatorial coordinates (see section 2.2). The most influential from the triplet above is zonal harmonics also known as J_2. It is often used solely for simplicity since the next other effect is approximately three orders of magnitude smaller. It is evident from the equation 3.14 that J_2 varies only with distance r and declination δ, thus resulting gravitational potential must have rotational symmetry about the Earth's spin axis. Neglecting sectorial and tesseral components, Earth's gravitational potential would result in an infinite series

$$U(\delta, \alpha, r) = \frac{\mu}{r}\left(1 + \sum_{n=2}^{N_z} \frac{J_n P_n \sin\delta}{r^{n+1}}\right) \tag{3.15}$$

from which P_n denotes Lagendre polynomials and J_k contain zonal harmonics and μ stands for Gravitational constant. The equation 3.15 can be the split into 3 components in ECI reference frame as J_X, J_Y and J_Z

$$J_X = -x\,\frac{\mu}{r^3}\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(5\frac{z^2}{r^2} - 1\right) \tag{3.16}$$

$$J_Y = -y\,\frac{\mu}{r^3}\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(5\frac{z^2}{r^2} - 1\right) \tag{3.17}$$

$$J_Z = -z\,\frac{\mu}{r^3}\frac{3}{2}J_2\left(\frac{R_{eq}}{r}\right)^2\left(3 - 5\frac{z^2}{r^2}\right) \tag{3.18}$$

It is possible then from Lagrange's planetary equations [12], describing the change rates of orbital elements with the partials of the disturbing forces, to derive 3.20, 3.21, 3.22, respectfully, nodal regression, apsidal and mean anomaly rotation. Other Keplerian elements are static. Equations 3.19–3.22 are averaged expressions, therefore, short-periodic variations happen in all of the six orbital elements. However, average change of parameters a, e, i are zero, which means, they won't differ after one full revolution.

$$\dot{a}_{J_2,av} = \dot{e}_{J_2,av} = \dot{i}_{J_2,av} = 0 \tag{3.19}$$

$$\dot{\Omega}_{J_2,av} = -\frac{3}{2}n\,J_2\left(\frac{R_{eq}}{p}\right)^2\cos i \tag{3.20}$$

$$\dot{\omega}_{J_2,av} = -\frac{3}{2}n\,J_2\left(\frac{R_{eq}}{p}\right)^2\left(2 - \frac{5}{2}\sin^2 i\right) \tag{3.21}$$

$$\dot{M}_{J_2,av} = n + \frac{3}{2}n\,J_2\left(\frac{R_{eq}}{p}\right)^2\left(1 - \frac{3}{2}\sin^2 i\right)\sqrt{1 - e^2} \tag{3.22}$$

where p is the semi-latus rectum (see 2.2) and n is the mean motion defined as

$$\begin{aligned} p &= a(1 - e^2) \\ n &= \sqrt{\frac{\mu}{a^3}} \end{aligned}$$

In work by Vallado [40], several gravitational models are presented. The one worth a mention is DMA. It consists of WGS (primarily WGS-84) and lately improved by EGM (the most recent EGM-2020) cite2015AGUFM.G34A..03B.

Planet	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
f	0.000	0.000	0.00335	0.00648	0.06487	0.09796	0.02293	0.01708
$J_2 \times 10^{-6}$	60	4.458	1 082.63	1 960.45	14 736	16 298	$3\dot{3}43.43$	3 411

Table 3.1: Table shows comparison between each planet's approximate oblateness f [44] and second zonal harmonics J_2 [5]. Due to a structure of gaseous planets, they tend to have higher rate than solid terrestrial planets.

3.1.4 Solar Radiation Pressure (SRP)

Solar radiation pressure refers to the force exerted on an object due to the momentum transfer of solar wind particles and photons radiating from the Sun. Direction of this force is, thus, always opposite to the Sun. Part of the radiation also bounces off the Earth's surface, referred to as Earth radiation pressure (more on that in following section 3.1.5). Acceleration implied by SRP can be stated as [15]

$$\mathbf{a}_{srp} = \nu P \frac{C_R A_S}{m} (AU)^2 \frac{\mathbf{r} - \mathbf{r_S}}{|\mathbf{r} - \mathbf{r_S}|^3} \tag{3.23}$$

where C_R, A_S, m are RSO's properties: radiation pressure coefficient, surface area exposed to solar radiation and mass. ν stands for illumination/shadow factor, which is unitless and lies in range $\langle 0, 1 \rangle$, 0 means that it is covered entirely by the Earth's shadow or 1 otherwise. Nominal solar radiation pressure at 1AU is defined by P. $\mathbf{r} - \mathbf{r_S}$ is opposite positional vector from RSO to the Sun.

The equation 3.23 describes SRP by the Cannonball model, which assumes that the RSO is perfect homogeneous sphere (or set of spheres) that is completely opaque, absorbs all photons and has a constant mass. Such precondition implies the effects of reflection deffusion are treated as negligible.

Satellite in contrast to space debris has much more incident area, since it has solar panels which are flat. They usually face the Sun to gain as much power as possible. In case of a satellite, the coefficient of reflection C_R is dependent on type of utilized solar panel and the types of materials used for coverage of the surface of the spacecraft. Since majority of the surface comprises of solar panels, similarly to atmospheric drag, the area is relatively wide and account for large amount of drag.

It should be noted that C_R, A_S and m parameters are not always known. Perturbation characteristics, therefore, can be approximated by sphere (Cannonball model), or entire SRP effect neglected.

As previously mentioned in section about atmospheric drag perturbation 3.1.2, solar radio flux $F_{10.7}$ is included in many models. However, it can be used not only for that, it may be employed in many more fields of interest like climate modeling, meteorology, geophysics, communications, satellite systems and so on [9]. The $F_{10.7}$ correlates well with the sunspot number as well as a number of UltraViolet (UV) and visible solar irradiance records [24].

Its main advantage is its robustness and long history since 1950s. It can be also measured in any weather conditions from the Earth's surface. This provides valuable data set for machine learning (ML) which requires long records for training phase.

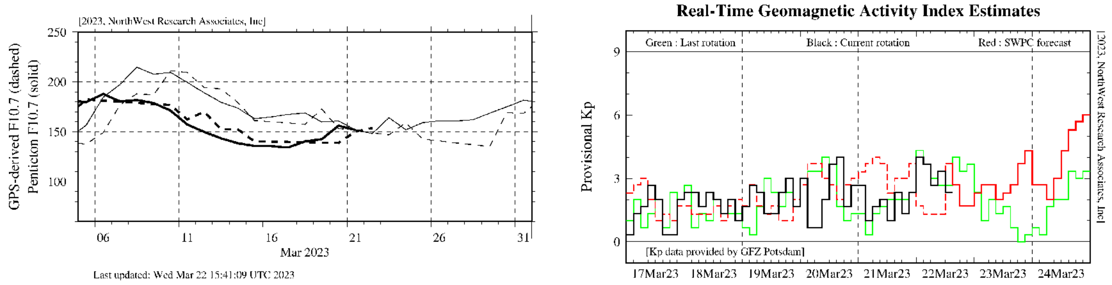


Figure 3.3: First graph shows heavy (dark) lines as the current 27-day Bartel rotation, light lines are for the last rotation [35]. $F_{10.7}$ "solar flux units", (s.f.u.), the F10.7 can vary from below 50 s.f.u., to above 300 s.f.u., over the course of a solar cycle [24]. In the bottom picture the black curve stands for the current solar rotation, and the green curve is for the previous rotation (27 days prior), and the red curve is Kp as predicted by the NOAA Space Weather Prediction Center [34]. Second picture is due to relevance and space saving reasons cropped.

SRP force can be also leveraged as propulsion technique, similar to sail boats. Thanks to the fact that outer space is nearly perfect vacuum, firing solar wind particles can boost spacecraft significantly barely without any opposite drag. It is apparent from the equation 3.23 that the lighter, spacier and closer to the Sun the RSO is, stronger the effect. SRP is also more pronounced at higher altitudes as seen in figure 3.2.

3.1.5 Earth Radiation Pressure (ERP) / Albedo

In astronomy, albedo denotes the fraction of the incident sunlight that the surface (or atmosphere) of the surface of some space object reflects [2]. For instance the Earth or Venus have much higher albedo than Mars. It is mainly caused by composition of the surface and atmosphere. In case of Venus, thick barely impenetrable atmosphere is responsible for very high rate of reflectivity. On the other hand, the Earth's surface (71 % [42]) is mainly covered by water [42], whose albedo is low (on average only 0.06). Nonetheless, thanks to presence of clouds (approximately half of visible surface) in the Earth's atmosphere, having much higher albedo relative to Martian-like surface, results in better reflective properties.

Planet	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune	Enceladus	Moon
Albedo	0.142	0.689	0.434	0.170	0.538	0.499	0.488	0.442	1.38	0.14

Table 3.2: Geometric albedo values of selected Solar system bodies. Highest rate has Enceladus [41], the sixth-largest Saturn moon, for its snow-covered surface. The lowest is the Earth's Moon with value reaching only 0.14 [19].

The albedo radiation effects on satellites are caused by reflected or re-emitted sunlight from the Earth's or Moon's surface [11], hence Earth Radiation Pressure (ERP), making it a by-product of the SRP. Considering this fact, relative positions of the satellite and the Earth and the Moon are of the utmost importance influencing magnitude of the albedo perturbation force. Another important factor is inclination i. According to a recent study [11], three orbital elements (e, ω and M) are influenced. It states that perturbing force increases with decreasing i of the orbital plane. For high locally specular nature of the water, the force has the most extensive impact on objects orbiting near ecliptic plane.

Every surface has different properties of reflection. Simulation of each would be computationally unfeasible. Therefore, surface is reduced to only few dominant components and both the spectral and angular distributions of the reflectances for the individual components are highly simplified [2].

3.1.6 Tides

The are two typical types of tidal forces: Solid Earth tides and Ocean tides. Solid Earth tides (as the name suggests) relate to the Earth's crust and mantle movements, and ocean tides to the water volume. Similarly to the non-spherical Earth influence, displacements of a mass cause variations in the Earth's gravitational potential.

The Solid tides produce vertical and horizontal displacements that can be expressed by spherical harmonics expansion (m, n), characterised by the Love and Shida numbers h_{mn}, l_{mn} [38], thus equation [28]

$$\Delta\mathbf{r} = \sum_{j=2}^{3} \frac{G\,M_j\,R_{eq}^4}{G\,M_e\,R_j^3} \left\{ h_2\,\hat{\mathbf{r}} \left(\frac{3}{2} (\hat{\mathbf{R}}_j \cdot \hat{\mathbf{r}})^2 - \frac{1}{2} \right) + 3\,l_2\,(\hat{\mathbf{R}}_j \cdot \hat{\mathbf{r}}) \left[\hat{\mathbf{R}}_j - (\hat{\mathbf{R}}_j \cdot \hat{\mathbf{r}})\,\hat{\mathbf{r}} \right] \right\} \tag{3.24}$$

where:

$\Delta\mathbf{r}$ is a site displacement vector in a Cartesian coordinate system
GM_e, GM_2, GM_3 are gravitational parameters of the Earth, Moon and Sun, respectively
$\hat{\mathbf{r}}, R_j$ are the unit vector from the geocentre to moon or sun and the magnitude of that vector
$\hat{r}, r$ are the unit vector from the geocentre to the station and the magnitude of that vector
h_2 is the nominal degree 2 Love number (h_2=0.6078)
l_2 is the nominal degree 2 Shida number (l_2=0.0847)

The effects of the ocean tides are even one order weaker, therefore, usually omitted entirely from the predictions.

3.1.7 Other

Apart from gravitational attraction posed by the Solar system planets and the Sun, the zonal gravitational harmonics and tidal forces, also relatively minor perturbing forces can be taken into account. However, for computational efficiency, following factors are usually omitted.

First one to be mentioned is *general relativity.* As the Earth spins, the mass and angular momentum cause a curvature in four-dimensional space-time. This results in diverging time between an observer on the surface of the Earth and RSO revolving around. Therefore, small calibration in order of $10^{-9}\, {}^{m}/{}_{s^2}$ must be added. Such acceleration is the defined as [28]

$$\mathbf{a}_{rel} = \frac{G\,M_e}{c^2\,r^3}\left\{\left[(2\beta+2\gamma)\,\frac{G\,M_e}{r} - \gamma\,(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}})\right]\mathbf{r} + (2+2\gamma)(\mathbf{r}\cdot\dot{\mathbf{r}})\,\dot{\mathbf{r}}\right\} + \tag{3.25}$$

$$(1+\gamma)\,\frac{G\,M_e}{c^2\,r^3}\left[\frac{3}{r^2}(\mathbf{r}\times\dot{\mathbf{r}})(\mathbf{r}\cdot\mathbf{J}) + (\dot{\mathbf{r}}\times\mathbf{J})\right] + \tag{3.26}$$

$$\left\{(1+2\gamma)\left[\dot{\mathbf{R}}\times\left(-\frac{G\,M_S\,\mathbf{R}}{c^2\,R^3}\right)\right]\times\dot{\mathbf{r}}\right\} \tag{3.27}$$

where, c is the speed of light, r, $\mathbf{r}$ and $\dot{\mathbf{r}}$ are the geocentric RSO distance from geocentre, position and velocity vectors, respectively. $\mathbf{R}, \dot{\mathbf{R}}$ are the position and velocity vectors of the Earth with respect to the Sun. $\mathbf{J}$ denotes Earth's angular momentum per unit mass. Finally γ, β are parameterized post-Newtonian (PPN) parameters.

Equation above can be split into 3 parts. Compared to the main Newtonian acceleration, the first one, the Schwarzschild terms 3.25 are about 10^{10} (high orbits) to 10^{9} (low orbits) times smaller. The effects of Lense-Thirring precession 3.26 and the geodesic (de Sitter) precession 3.27 are around 10^{11} to 10^{12} smaller. The main effect of the Schwarzschild terms is a secular shift in the argument of perigee while the Lense-Thirring and de Sitter terms cause a precession of the orbital plane [28].

Probing further into solar wind influences, not only the incident light acts upon RSO as propelling force, however, also heating occurs on the object's surface. This process is called *Yarkovsky Thermal Effect* and causes a temperature difference across the RSO, which leads to a force that alters the RSO's trajectory over time. It is rather insignificant for massive objects like the Earth, nonetheless, smaller objects such as space debris, picosatellites etc. could be influenced over time. Despite that, resulting force is several orders of magnitude smaller than other primary perturbations, therefore, Yarkovsky Thermal Effect is considered rather during Solar system asteriods, comets, ... calculations.

From another point of view, in case of satellites, sometimes occasional thrusts for altitude correction happen. In such instances orbital elements are usually dramatically disturbed, thus measured data are compromised and unusable. For this reason human-made objects with these such signs are in majority of cases precluded.

3.2 Perturbation forces summary

Assembling previous findings into table containing perturbations and their models leads to following table

Perturbation	**Model**
N-body	JPL DE430
Atmospheric drag	NRLMSISE-00, DTM2000
Earth shape	WGS84, EGM2020
SRP	$F_{10.7}$, K_p, sunspots
Tides	GOT4, RE2014
Albedo	CERES
General relativity	Post-Newtonian correction

There are four main approaches for orbit propagation: analytical, numerical, hybrid and emerging NN based. Analytical solutions (aka. General perturbation techniques) are usually more physically accurate, however they are lacking precision if any approximation, such as series expansions, is being applied. Numerical (aka. Special perturbation techniques) is way to deduce future state incrementally, exclusively for a given initial condition, therefore special. Advantage of the analytical solution is that there is no need to calculate numerically step-by-step, which is computationally time consuming, but produces instant result given any initial conditions. On the other hand, having small iteration step yields smoother, thus more precise prediction. All other techniques, not purely analytical or numerical, are considered as hybrid or semi-analytical [40]. However, in recent years simultaneously with progress in ML, new methods have been proposed even outperforming current analytical and numerical methods (more in the chapter 6).

Based on the equation 3.5 for unperturbed motion of the satellite M_2 relative to the Earth M_1 by including perturbing accelerations a would yield

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_p \tag{3.28}$$

where every perturbation can be expressed as three-component acceleration vector $\mathbf{a_p}$ as compound of all perturbations.

$$\mathbf{a}_p = \mathbf{a}_{drag} + \mathbf{a}_{harmonic} + \mathbf{a}_{3-B} + \mathbf{a}_{SRP} + \mathbf{a}_{albedo} + \mathbf{a}_{tides} + \mathbf{a}_{other}$$

3.3 Trajectory interpolation

Since the orbital elements are measurements at given time (epoch), connecting them together produces so-called time series for each element. Composed time series is not, however, evenly spaced. This means that ticks have diverse spans between each other. Although some RSOs have these gaps small (in order of hours), some may be even months long. Such inconsistencies are not suitable for ML, since the missing data may influence model and incorrectly interpret these gaps. To eliminate this factor, records with overly oversized time spans are removed from learning dataset.

For many usages, it is handy to have time series with fixed step. Interpolation is one method to achieve this. Unequally spaced points are foremost fitted with curve that best describes it and then re-sampled with given time-step. Studied interpolation methods were:

- `interp1d` – Line interpolation (first order – line).
- `QuadraticSpline` – Produced unstable results (second order – polynomial).
- `BSpline` together with `splrep` – B-spline representation of a curve.
- `PchipInterpolator` – Piecewise Cubic Hermite Interpolating Polynomial.
- `Akima1DInterpolator` – Fit piecewise cubic polynomials.

The best results have been achieved using Pchip and Akima interpolators, since they merge advantages from both linear and spline fitting. Unlike linear interpolation, they are differentiable on the entire definition domain thus lack unwanted sharp peaks. To the other extreme, using polynomial fitting may produce Runge's Phenomenon [6], which is overshooting if the data is not smooth and large differences on small timesteps are present. This happens usually in cases when using high order Lagrange polynomials [14]. On the other hand, Pchip and Akima interpolators have in common that they are based on Cubic Hermite Spline. Similarly to Lagrange, but one must explicitly specify derivatives at each measured point, useful in combination with velocity and position vector. Pchip and Akima interpolators do this automatically, so it is very convenient to use.

Having required theory explained, it is time to shift to the practical part of this thesis, focused primarily on ML development.

Chapter 4

Libraries and obtained data

The purpose of this chapter is to provide an overview of the obtained/downloaded data and astronomical tools. In the first section, leveraged tools will be discussed. The rest of the chapter will be dedicated to the data analysis and the preprocessing.

4.1 Astronomy

Following three sections are primarily focused on astronomically related topics. The reader is going to get familiar with the structure of the data and their intricacies,

4.1.1 Space-track.org API

First step of the entire implementation process was to get data from space-track.org which provides REST (representational state transfer) API (application programming interface). It is then easy to build custom queries to obtain desired data such as general perturbations, which is the most recent set of several values for each man-made Earth-orbiting object (tracked by the 18th Space Defense Squadron) called SGP4 orbital element set [31]. The most important parameters are previously discussed Keplerian elements, object type, ID and date of creation. More detailed format description can be seen on space-track.org website. Following is a snippet from one general perturbations record of the ISS.

```
"CREATION_DATE": "2022-12-22T18:47:50",
"EPOCH": "2022-12-22T14:15:53.485920",
"NORAD_CAT_ID": "25544",
"MEAN_MOTION": "15.49508690",
"ECCENTRICITY": "0.00055300",
"INCLINATION": "51.6422",
"RA_OF_ASC_NODE": "123.2511",
"ARG_OF_PERICENTER": "175.6943",
"MEAN_ANOMALY": "277.6446",
"BSTAR": "-0.00015123000000",
"SEMIMAJOR_AXIS": "6796.299",
"TLE_LINE0": "0 ISS (ZARYA)",
"TLE_LINE1": "1 25544U 98067A   22356.59436905 -.00008932  00000-0 -15123-3 0  9998",
"TLE_LINE2": "2 25544  51.6422 123.2511 0005530 175.6943 277.6446 15.49508690374406",
...
```

4.1.2 Two-line element set (TLE) and SGP

Subset of the General perturbations are also TLE records. It is data format used to store orbital elements. Its most profound usage is for simplified general perturbation models (three SGP, SGP4, SGP8 for near-Earth satellites and SDP4, SDP8 for deep-space satellites) that calculate future position and velocity vectors of the Earth-orbiting RSOs. The most important is SGP4 which is most widely used as a mathematical based predictor developed by Ken Cranford in 1970.

However, there are few limitations regarding every analytical predictor such as SGP4. Firstly, TLEs provide only approximated figures, which pose problems for longer computations where every decimal place give more precision to the computation and influence final result. This is particularly visible when trying to forecast more than 3 days, after that TLE precision becomes insufficient. Secondly, SGP/SDP models by their nature provide simplification of the real world, in other words analytical solutions would be so computationally demanding thus inapplicable for daily assessments of thousands of RSOs. Also as previously discussed in section 3.1, there are pseudo-random systems like the Sun, which are for us analytically insolvable so far, hence only reliable option is to depend on probabilistic models describing its behaviour.

For comparison objective between developed prediction and official SGP4 predictor, Python implementation [36] have been used.

4.1.3 Skyfield API

Skyfield is Python library that computes accurate (order of milliarcseconds) positions of the Solar system bodies as well as of stars and satellites. There are also several helpful methods such as reference system transformations, Earth-centered or barycentric planetary distances, ephemerides etc.

4.2 Data analysis

Main focus of this work is to develop orbit predictor which derives its prediction from the series of state vectors discussed previously in chapter 3.2. First phase was to download the data. However, space-track.org imposes limitations on the number of requests per hour, thus maximal frequency is only 300 requests per hour. That means 83 hours for 25 000 files (requests) of pure download time, totaling 69.9 GB of disk memory. Each file is in JavaScript Object Notation (JSON) format and contains collection of orbit elements and other data described in the chapter 4.1.1. There are three main categories of RSOs: Satellites, rocket bodies and space debris. It can be then split further based on its present state: decayed or still on the orbit.

For the purpose of the machine learning (ML) training, the bigger the dataset, the better. Otherwise, learned model would likely produce biased, in other words misleading, results. This rule also applies to file sizes. Longer the orbital elements are measured, the better it can learn long-term characteristics. In the following graphs 4.1, records according to their length (measurement count) have been distributed. There are many inconsistencies in the data, such as duplicate records or coupled epochs. Since the epoch is a benchmark which determines time of the measurement, coupled data must be averaged or removed if high dissimilarity in the couple is encountered. Therefore, the original length may differ after previously discussed removal.

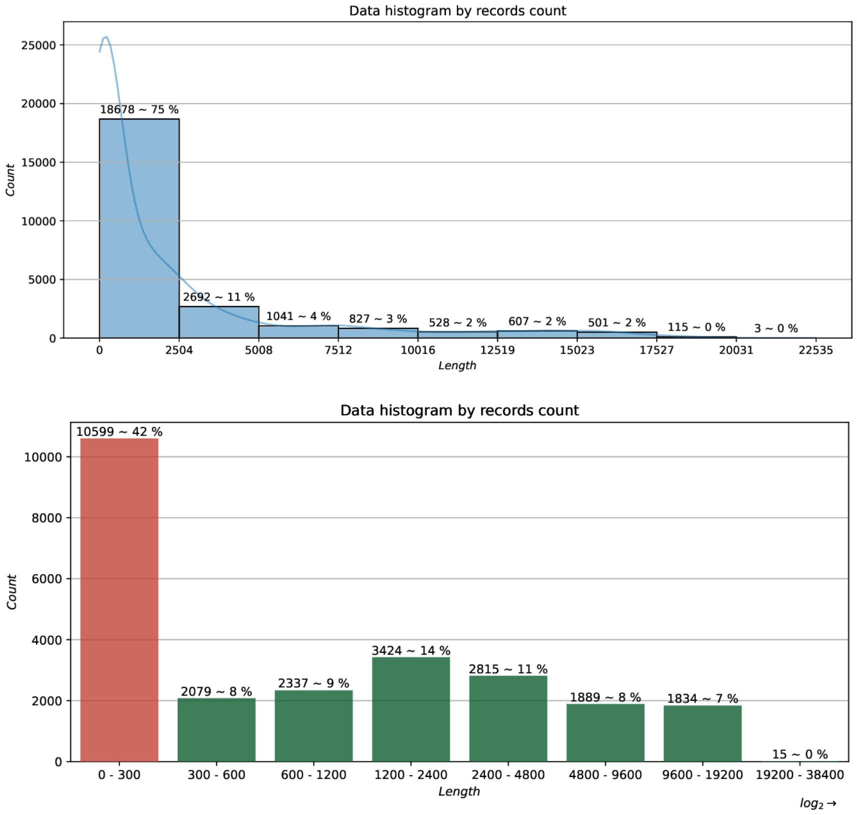


Figure 4.1: First graph shows histogram of file sizes fitted with smoother blue curve. The bottom picture shows log-scale distribution of the file lengths for better human perception. In the first red column (for its lacking magnitude) lie unusable data.

First graph reveals that majority of the records are smaller in size. The best data (in a size sense) could be found in the last three columns adding up to 15 %. Based on the findings by several studies, there is a considerable difference in the length of the input data. According to the study [18] at least 3 years of TLE data is required to produce meaningfully better orbital predictions than SGP4. Another study [17] shows different approach where only few hundreds state vectors of an RSO were enough to produce reasonable results. Admittedly, it depends on the method used for prediction. More robust methods utilize usually more data. On the other hand, dynamic methods support fewer inputs.

In order to get appropriate prediction, several criteria must be met such as afterwards already discussed duplicates removal the minimal measurements count must be 300, which roughly translates to several months of observations. Another requirements are low error rate (referred to as outliers count 4.2.1), minimal time-span of four days between each measurement, and elimination of self propelled satellites, causing arbitrary deviations within data.

4.2.1 Data filtering

Obtained data from space-track.org, as mentioned before, contain corrupted entries. Good example of weird-looking data file is depicted in the figure 4.2. It is clear from the picture that on several spots (eg. i curve around March 2011), high peaks emerged. These are called outliers and can represent incorrect measurements or another kind of errors. Since this data is obviously wrong, it is advised to remove them, or in case of high frequency within one file, delete the file entirely.

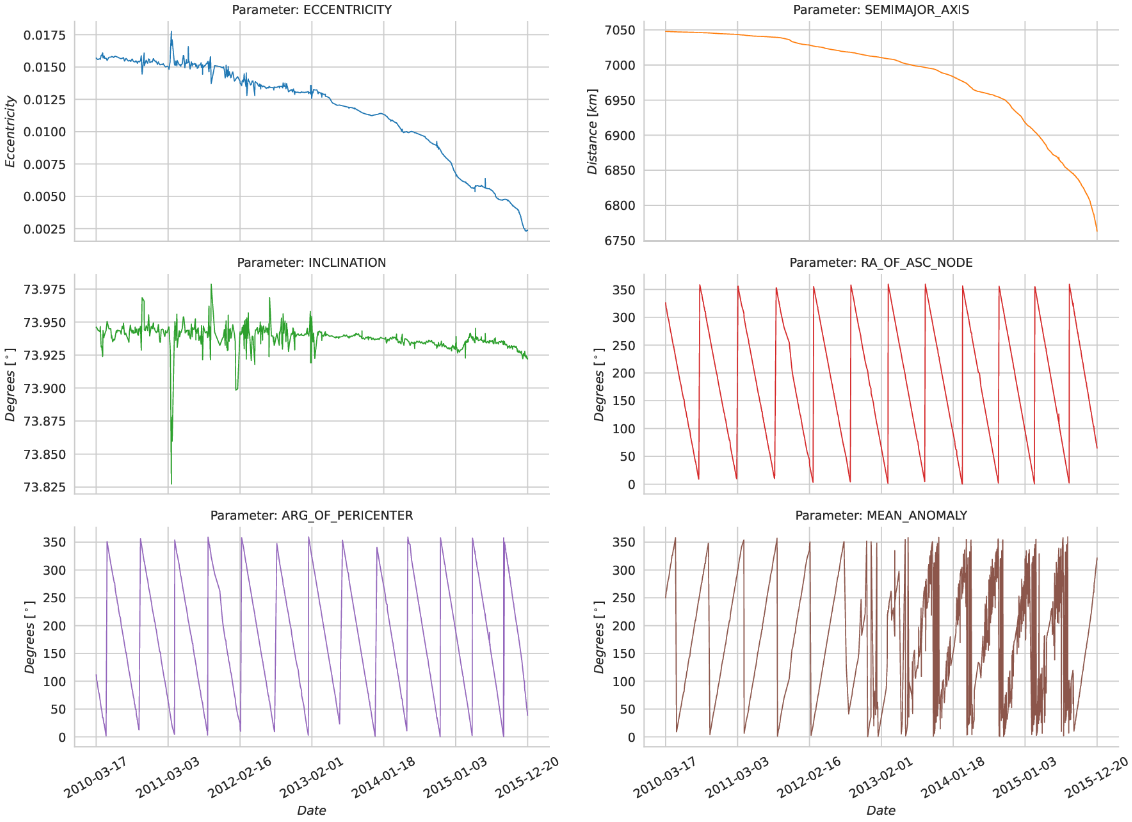


Figure 4.2: Graph shows unfiltered orbital elements of the RSO with ID 36474.

Outliers, as the name suggests, are point/measurements which lie outside of presumed mean curve. There are couple of ways to detect outliers. One of the simplest methods is to use normal probability distribution (depicted in the figure 4.3). However, there is one difficulty regarding that... portion of the whole data set must be always excluded. The amount depends on factor σ, which defines how wide distribution curve is. It is common practise to use rule of 3 σ. That means, that approximately 99.7 % of the data is correct and the rest is labeled as outliers. Therefore, each time, there will be exactly 0.3 % of the data labeled as incorrect.

Different way is by using smoothing functions. There are 3 tested: Savitzky-Golay filter (Savgol), rolling mean and convolution. Each function has its limitation and advantage. For instance, Savgol function perfectly smooths rough peaks in longer function course. On the other hand, rolling mean is useful in situations where big differences in values are encountered (eg. ω, Ω, M, see 4.2). For example, when anomalies change their value in radians, thus, leap from 360° to zero° is necessary. In such cases, Savgol is not ideal, creating too smooth curve, which does not correspond to the actual data. Moreover, due to its polynomial expansion nature it is not suitable for high frequency periodic signals. On the other hand, having quite stable function course with occasional peaks, Savgol would perform very well.

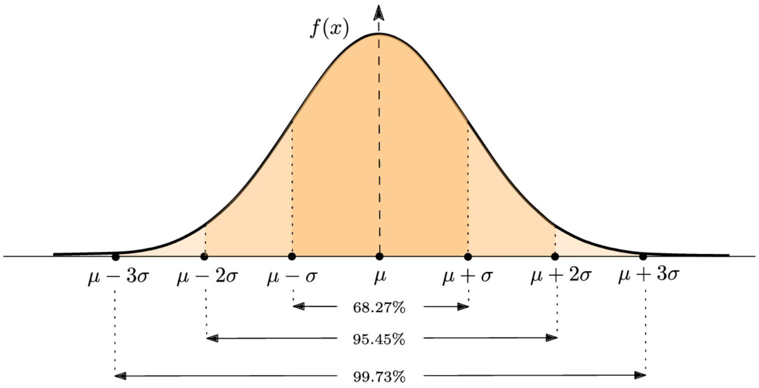


Figure 4.3: Graph shows on y axis normal probability probability distribution, on x axis values of measurements.

Convolution behaves in the middle of the two. It can be modified to serve different purposes. Convolution woks on basis of two sliding one-dimensional arrays/matrices, which multiply one another and sum up. Thus, one matrix contains input data and one array, sliding over the first one, is convolution matrix. In this case, for smoothing, matrix would be $\begin{bmatrix} 1/n & \dots & 1/n \end{bmatrix}$ where n is length of the matrix. This process averages step by step entire function. However this process is not particularly resistant to far outliers, since it takes only a mean of all values and sums them up. There could be many modifications applied to second matrix such as its length (longer the smoother) or indexes (creating eg. high-pass or low-pass filters). It is characteristic for its "slow" start and end, where not enough data are provided to fill entire convolution matrix, thus, low and incorrect values emerge (this phenomenon is visible in the figure 5.5, orange curve).

In order to exclude boundary exceptions such as min max value in linear function, moving window (also referred to as rolling window) have been used. First step is to create moving window, which could be comprised of, say 10 % of the entire data length. Then with defined step, apply outlier detection method and move the window by the given amount. Long step would be more suitable for faster run, however, to achieve better results, shorter step is a better option usually.

In this thesis, each method to discover outliers, described above, and moving window has been combined. Applying all of the methods together reduces error induced solely by one method and only those measurements, where each method defined the measurement as outlier, will be removed. The result can be seen in 4.4 or in appendix A where reader can compare side by side the filtering procedure outcome.

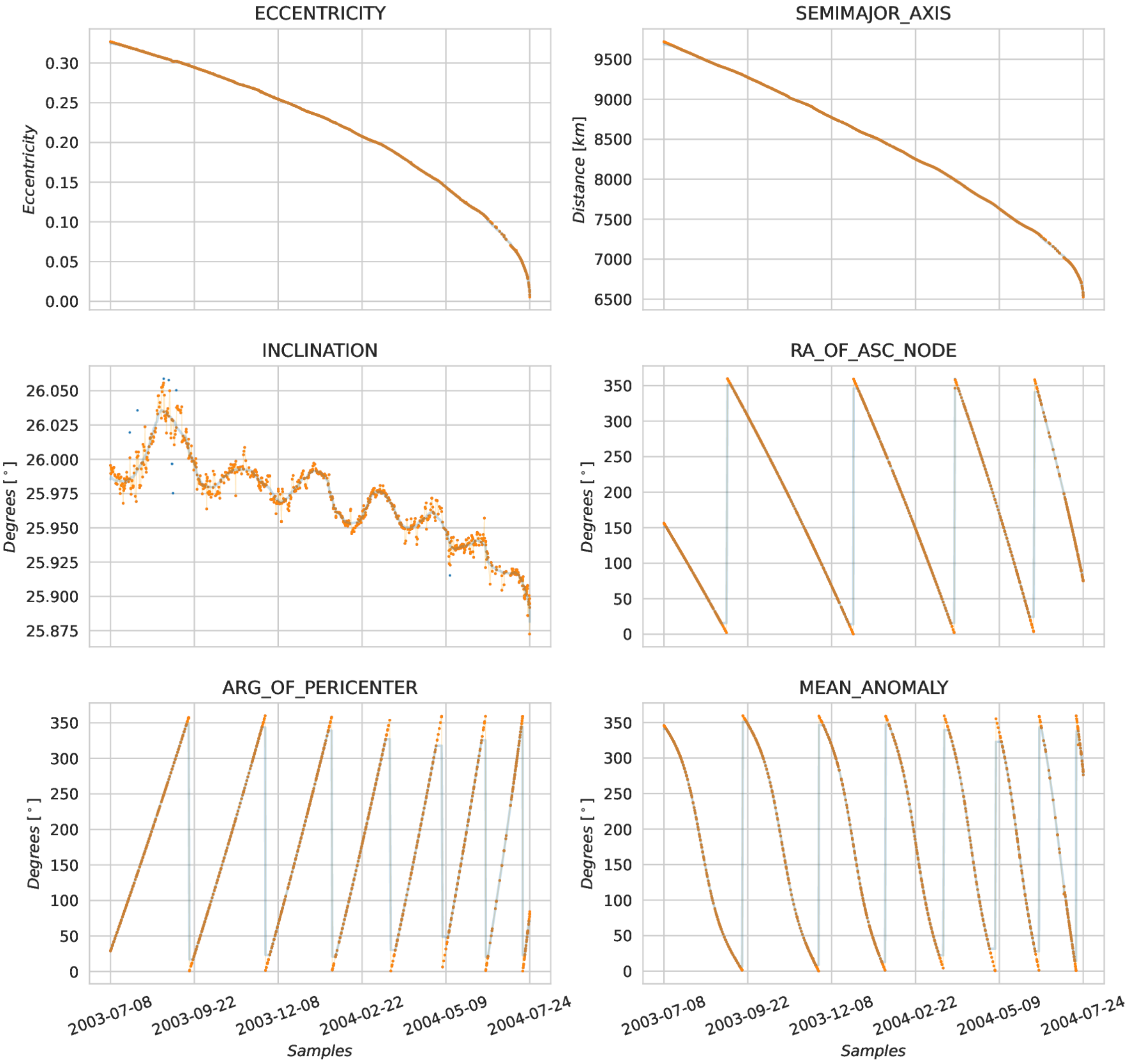


Figure 4.4: Graph shows unfiltered/original (blue dots) and filtered/updated (orange dots) orbital elements of object with catalogue ID 28192.

To summarize previously induced findings about data, unapproved files, which did not met following requirements, must be excluded from training data set:

1. Measurement count after duplicates removal > 300.
2. Duplicate records $< 50\,\%$.
3. Outliers count $< 5\,\%$.
4. Minimal timedelta between each measurement < 4 days.
5. No propulsion evidence.

4.2.2 Clustering

Clustering is a method which tries to group records with similar features together. This comes in handy when training model with distinctive behaviour, finding connections between data, or simply learning more about dataset.

There are several steps involved (besides clustering process itself) which could be divided into two halves. The first one is the data preprocessing, then the process of making clusters follows. The preprocessing stage incorporate data filtering (described in previous section 4.2.1) and features extraction, the data are then organized into dataframe and scaled to unit variance with mean equal to 0. After that, dimensionality reduction and clustering are applied. The very last operation of the entire process is to visualize the results.

Since orbital elements are expressed as time series functions, the objective here is to find features which can describe their course, development, rate of change, etc. while preserving important information about their nature. These parameters later serve for clustering. By looking at the plotted orbital elements (4.4), couple of traits stand out. The first obvious thing is that the value domain of ω, Ω, M is in degrees $\langle 0, 360)$. On that account, on first glance sawtooth wave plots denote that the elements are changing periodically. Other orbital elements look differently and do not exhibit same recurrent behaviour. On the other hand, there is usually tendency to raise or fall, hence, simple linear regression and its slope parameter a from $f(x) = a\,x + b$ is utilized. Rate of change a is also used in case of periodicity, since it doesn't usually remain constant and shrink or prolong itself. Next interesting feature is correlation between some elements (usually between couples of elements $e - a$, $e - i$, $a - i$, $\Omega - \omega$, $\Omega - M$, $\omega - M$). Last two measures define quality of the data by means of its length and measurement errors (number of outliers).

To conclude previous findings, these features were used for clustering (digit in the bracket represents its number of occurrences – 23 features in total):

- (6) Standard deviation of each orbital element.
- (3) Rate of change of e, a and i elements (using linear fit).
- (6) Mean periods and rate of change of ω, Ω and M elements.
- (6) Correlation rates between couples of orbital elements.
- (1) Length of observation (measured in days).
- (1) Number of outliers.

Although visualizing is the last step of the process, one must take into account that the most common example of clustering is in 2 D Euclidean space, where it is easy and convenient to imagine axis, distances, etc. Nonetheless, for more than 2 D (or 2 parameters) data, it is not possible to plot graph with more than 3 features. Due to the fact that discussed clustering is based on more than three parameters (23), data must be represented in a different way. Luckily, there have been developed methods for dimensional reduction such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS). Or in case of supervised learning: Linear Discriminant Analysis (LDA), Neighborhood Components Analysis (NCA) etc.

PCA and MDS have been used, since they fall into unsupervised category and dataset is not labeled yet. Their advantage lies primarily in narrowing number of features used for clustering. Byproduct of this process is speedup in clustering since incorporated components are diminished. For instance, PCA performs an orthogonal linear transformation that transforms the data to a new coordinate system with reduced dimension. Algorithm does not dump any features entirely, only re-expresses them in its own way. Resulting features are called principal components and differ in their variance relative to the total variance of the dataset. On the other hand, MDS works on distance preservation principle. While scaling the dimension down, distances between points are kept similar (to the best extend possible) with respect to original high-dimensional space.

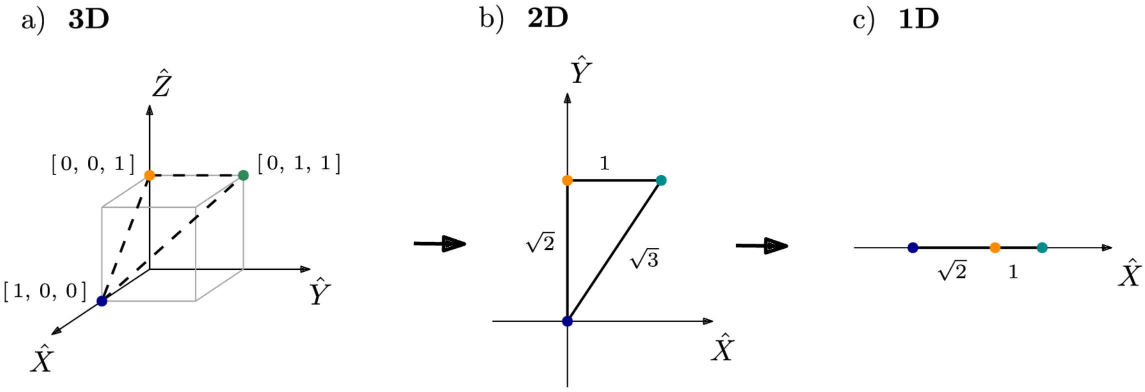


Figure 4.5: Examples of MDS process for mapping 3 points gradually from 3D to 2D to 1D space.

Reducing dimensions to a minimum is however not advised because some information is always lost or altered to meaningless extend. To achieve the best results, both MDS and PCA have their own measures of stress and explained variance, respectively. MDS's stress indicates sum of squared distance of the disparities and the distances for all measured pints. PCA uses variance of each principal component relative to the total variance referred to as explained variance ratio. Computing MDS stress and PCA explained variance ratio gradually for m-th component in n dimensions/components yields according to [39] equations

$$MDS = \sqrt{\sum_{i \neq j = 1}^{n} \Big(\|z_i - z_j\| - \|x_i - x_j\| \Big)^2} \qquad (4.1)$$

$$PCA = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2} \qquad (4.2)$$

where $x = [x_1 , \ldots , x_n]$ are original data and $z = [z_1 , \ldots , z_n]$ are their images.

However, there are no commonly agreed methods to determine right number of dimensions for both methods currently. One of the methods leveraged for PCA is setting threshold for cumulative variance of all principal components (in this case 80 %), higher the number, more components are included. For case of MDS, stress is increasing with each dimension reduction. Therefore, fewer components have high rate of stress and function is inverse, relative to PCA. For MDS's stress threshold 10 % has been chosen as a reasonable compromise between number of dimensions and precision. Both percentages could be increased or decreased to obtain more accurate results.

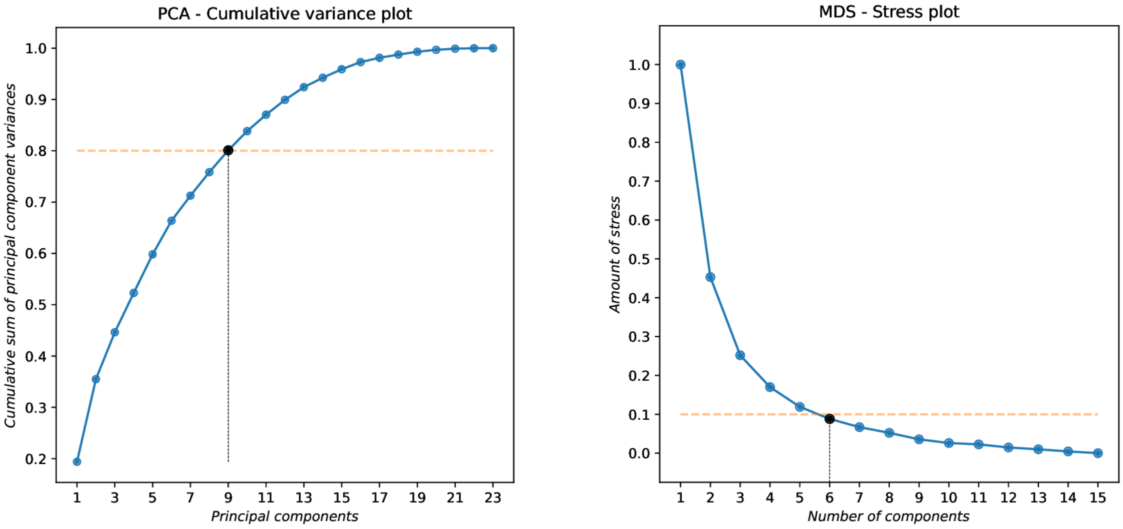


Figure 4.6: Left-hand side graph represents measure for PCA. Orange line means 80 % cumulative explained variance ratio. Number of principal components (black dot) surpassing the line is the sought one. Other figure is for MDS, depicting stress values. Black dot represents picked number of dimensions.

Clustering is an unsupervised machine learning method and can take at minimum only one important argument, which is the total number of clusters. Therefore, there are couple of methods designed to guess the best fitting number of clusters. The most popular one is Within-Cluster-Sum-of-Squares (WCSS). This way, sum of distances of measurements from their cluster centroids, is computed. The best fitted model is when the sum is low. The formula which WCSS relies on is

$$WCSS = \sum_{i=1}^{N} (X_i - Y_i)^2 \tag{4.3}$$

where X_i and Y_i are distances from X and Y coordinates of the cluster's centroids, respectively. It runs clustering and counts WCSS while increasing number of clusters. The process starts with one cluster, The worst scenario is always one cluster, therefore highest WCSS value. Higher the number of clusters, smaller the squared distance would be. However, that does not mean the best result, but rather overfitting. Hence, there could be applied methods to determine the best number of centroids, such as the Elbow method, which tries to find the biggest angle between three measurements as seen in the figure below (4.7). Angles are defined by two lines connecting three neighbouring points. The angle is then measured from the midpoint. The biggest angle (measured clock/wise) or the smallest (measured counter/clockwise) determines best number of centroids.

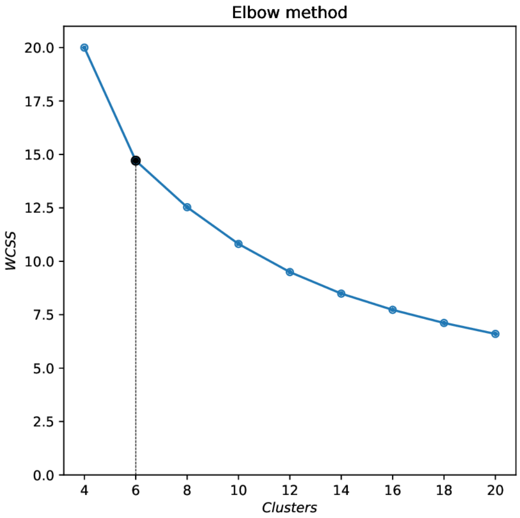


Figure 4.7: Graph shows on y axis WCSS and on x axis number of measurements. Black thick dot represents ideal number of centroids.

Finally, there are several options how to visualize the results obtained by clustering. Thanks to reduction in space dimensionality, there are not 23 axes anymore but based on the method used, 6 for MDS and 9 for PCA. It is possible to display so-called stripplot 4.8 for each feature. This allows viewer to better fathom the data grouping. Combination with violinplot in the background indicates data distribution even better. However, it only depicts one feature at one time. This is lacking bigger picture of the data and final results. Best way is to summarize all measurements into 2 D plot which is easier to read, yet it presents only two most dominant components. To get a whole picture, all components should be displayed, which is still not possible, since humans are constrained with perception of 3 dimensions. Good example of well clustered data is when clusters and centroids do not interfere and blend together.

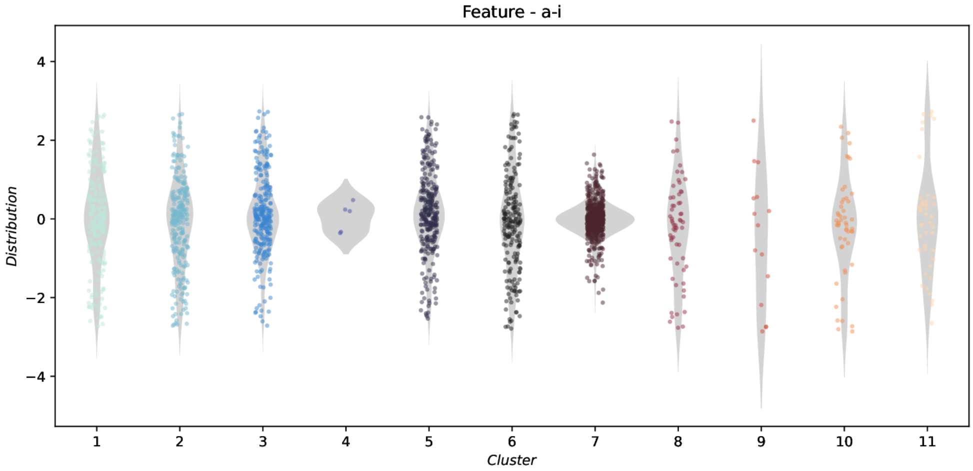


Figure 4.8: Graph (strippot + violinplot) displays one of the features – correlation between a and i. This is only one of all 23 features.

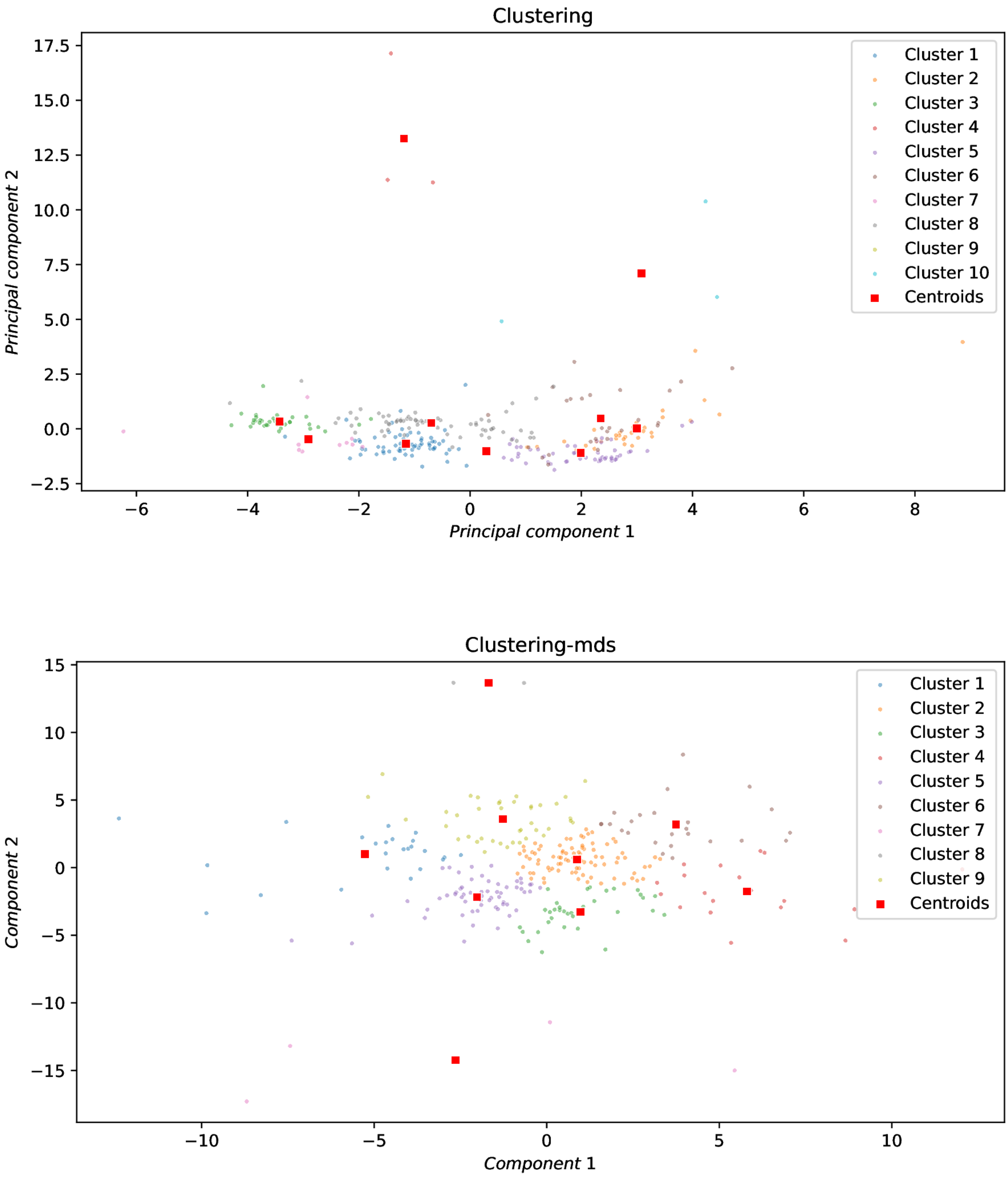


Figure 4.9: Figure shows clustering results. Each graph shows different approach for displaying the results. Upper one was produced using PCA and the bottom one by MDS. Cluster centroids can be recognized by red squares.

Chapter 5

Simulation

Simulations are critical tool for visual verification of the outcome of trajectory computations. In this section, several processes, which require simulation to be applied, will be probed.

First of all, basic simulation of the elliptic path of the RSO has been developed. The path conforms to one revolution of an RSO according to one orbital elements measurement. It offers viewer to judge on first glance, how properties transfer to reality, to which extend ellipse is oblate, how tilted is the trajectory and so on. It was the very next step when working on this thesis right after data acquirement from space-track.org.

Whole work is written in Python [29], therefore, one of Python's libraries – Vispy[1] has been used. It is a very fast library suitable for 2D or 3D briskly running GPU operations and simulations. Following picture depicts the occasion where the library has been used.

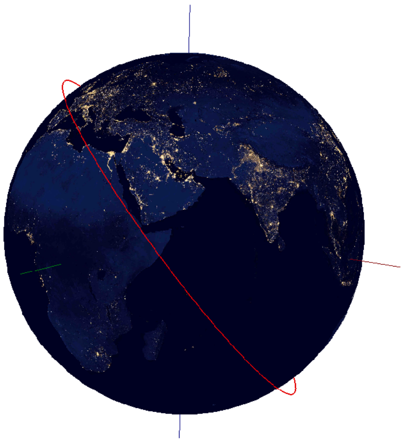


Figure 5.1: Here is the Earth spherical model (excluding Equator oblateness) with ECI coordinate axis X, Y, Z colored red, green and blue, respectively. Red ellipses capture 10 unperturbed Keplerian orbits computed from raw space-track.org data. Transparency indicates evolution in time, where opacity increases proportionally with time.

[1] https://vispy.org/

Secondly, similarly to the previous picture, trajectory is displayed. This time, however, with comparison between numerically propagated ordinary differential equation solution using Runge-Kutta (including only three dominant perturbibg actors: atmospheric drag, SRP and third-body influence).

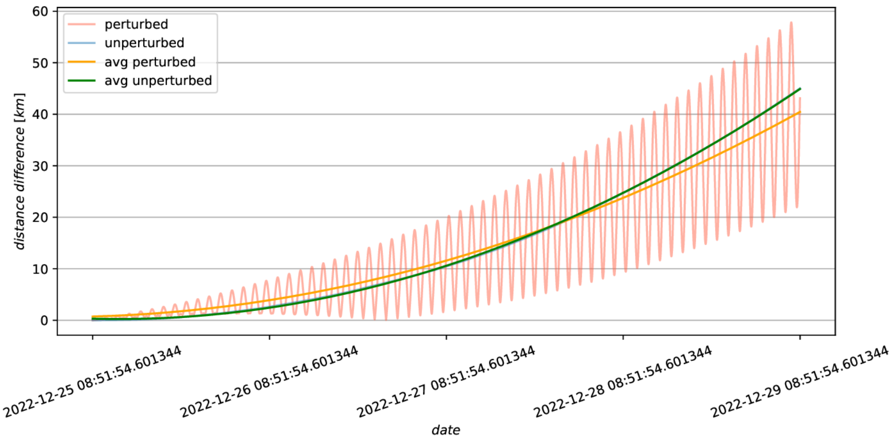


Figure 5.2: Both trajectories are computed using Scipy[1] ODE solver, `odeint`. There is very clear oscillation between perturbed and unperturbed trajectories. Both orbits are compared to the ideal elliptic keplerian orbit.

[1] https://scipy.org/

5.1 Performance

In this section, performance of a few key components is being analysed. These are process which occur frequently in code and influence general program swiftness. Testing device was the laptop `Lenovo IdeaPad L340-15IRH` with 8 cores i5-9300H @ 2.4 Ghz (4.1 GHz boost) CPU with 8 GB RAM and operating system Fedora Linux 35.

First probed metric is computation of the current position on the ellipse from classical orbital elements thanks to Kepler equation 2.8. Since the equation has no closed solution, it must be obtained iteratively using some root-finding algorithm such as Newton-Raphson. This method iterates until desired precission ϵ is exceeded. There had been several ϵ tested.

ϵ	10^{-5}	10^{-10}	10^{-15}	10^{-20}	10^{-25}
time	12.8 μ s	28.7 μ s	37.2 μ s	412 μ s	762 μ s
deviation	1.88 μ s	2.76 μ s	3.85 μ s	45.9 μ s	112 μ s

Table 5.1: Table shows 10 test runs for each ϵ with 1 000 000 loops using utility `%testit`.

As seen from the table 5.1, function with precision $\epsilon = 10^{-20}$ is significantly slower than previous $\epsilon = 10^{-15}$ also lower values can lack accuracy while taking similar computational time. It has been picked $\epsilon = 10^{-15}$ for the best time-precision ratio.

5.2 Sunspots

As mentioned before in section 3.1.4, sunspots are one of measures contributing to the SRP prediction. Based on their total sum and magnitude, Sun's activity and influence could be estimated. Moreover, according to [24], total number of sunspots correlates well with the amount of radiation emitted from the Sun. Nonetheless, due to the fact that the Sun is behaving chaotically, hence hardly predictable, SRP is considered as dynamic perturbation. Therefore, no analytical form to express amount of Sun's radiation exists, only empirical models do.

In this section machine learning methods Decision Tree (DT) and Random forest (RF) are proposed for prediction. They fall into machine learning algorithms category of Decision Forests (DF) used for supervised classification, regression and ranking. In general data are comprised of time series sunspot observations. Data that have been used here, have been gathered since 1818 capturing every-day observations. They are acquired from [37] as comma separated values (CSV) file type. Example of the data can be seen in the following snippet 5.2.

	Year	Month	Day	Date Fraction	Sunspots	Std	Observations	Indicator
0	1818	1	1	1818.001	-1	-1.0	0	1
1	1818	1	2	1818.004	-1	-1.0	0	1
2	1818	1	3	1818.007	-1	-1.0	0	1
...	...	...	...	...	...	...	...	...
74841	2022	11	28	2022.908	52	10.4	27	0
74842	2022	11	29	2022.911	29	5.8	15	0
74843	2022	11	30	2022.914	30	8.1	15	0

Table 5.2: Sunspots with value -1 denote missing observations, hence 0 in observation column. For space saving reasons, real column names are abbreviated `Date Fraction`, `Sunspots`, `Std` are titled respectively `Date In Fraction Of Year`, `Number of Sunspots`, `Standard Deviation`.

There are numerous ways to determine time series development. One of them is to use machine learning approach. This is comprised of two main phases. learning and prediction. Firstly, data are split into two groups. The first one (also the major one, about 70 % of the whole data amount) is designated for training/learning phase, the rest is then meant for testing and evaluating the result, to what extend is the model accurate. Higher the accuracy, the better the outcome is. It is expressed in percentages as fraction of correct predictions to all predictions. Nonetheless, it should be noted that accuracy very close to 100 % is suspicious and can indicate model overfitting.

Each day sunspots sum together with count of observations are given. However, sometimes there are no observations at all, hence zero in the observation column. Such data are unfortunately useless for learning phase. From the early decades until the beginning of 20th century, no more than handful of measurements per day had been conducted, this is rather small amount to deduce anything from, therefore, this entire stage is pruned. Utilized data for training phase begin with year 1 900 and end at the end of 20th century. The rest is dedicated to the evaluation.

Histogram below proves that significant portion of acquired measurements contain relatively low number of observations. To achieve high-precision training results, first column observation should be minimized as much as possible.

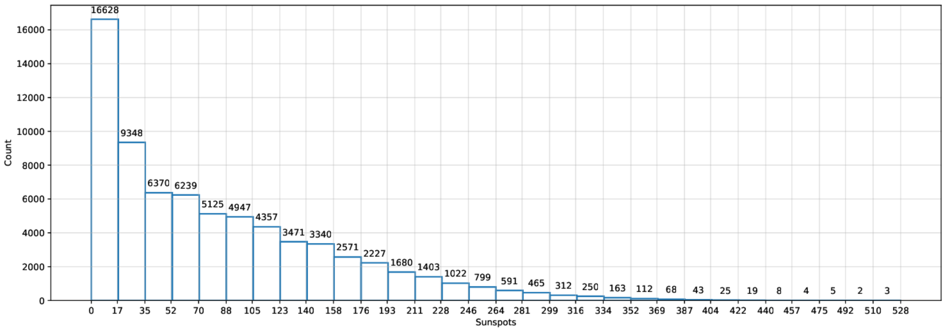


Figure 5.3: Histogram depicts observation counts per day. Most important are missing and low count observations gathered in the first column. Measurements with more than 17 occurrences are depicted in the rest of the graph. Each column consists of sum of 18 consecutive observation counts.

Until now, observation count has been mentioned several times. It should be, however, mentioned that it is not included in machine learning process, since it does not correlate at all to measured sunspots count. The only thing it influences is data reliability, because more measurements reduce occasional errors.

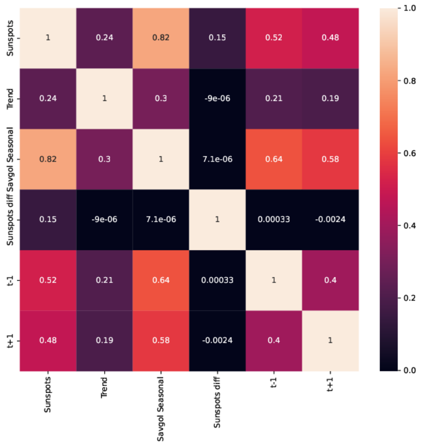


Figure 5.4: It is suitable having correlation table before training phase. Thanks to that, it is possible to detect overly correlated parameters whose usage can lead to model overfitting. In this picture, all parameters used for machine learning process are displayed. There are self correlated values on diagonal, therefore all ones. Every correlation pair is defined by column and row. Each couple is presented twice with transposed row and column.

5.2.1 STL

For proper prediction of the time series it is advised to develop seasonal and long/term trend curve. This procedure in known as Seasonal and Trend decomposition using Loess hereinafter referred to as STL. Loess stands for a nonlinear relationships estimator. Figure below serves as comparison between three different curves for seasonal trends. An objective of it was to find visually best parameters in order to gain the best result.

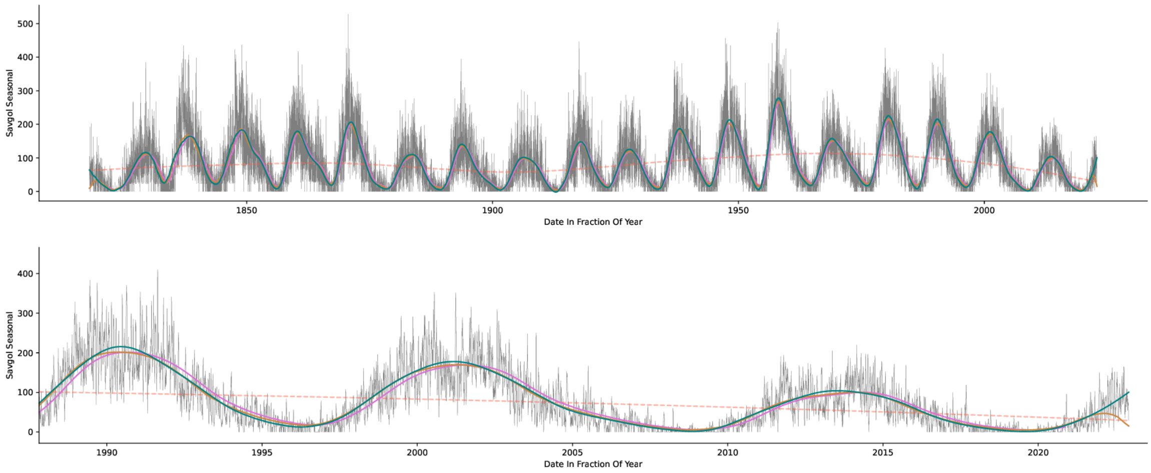


Figure 5.5: Comparison between three smoothing functions for seasonal curve.

5.2.2 Methods

Four different methods were leveraged for sunspots prediction. The simplest one is Decision Tree (DT). It works on basis of boolean (TRUE/FALSE) algebra. Method gradually develops a tree, whose leaves represent final guess of predicted values. With every iteration, tree structure modifies to improve itself. Step by step, tree nodes importance amplify or diminish until the learning phase is accomplished. DT method has been chosen for its relatively simple and doesn't require too much fine-tuning for reasonable results. In this case, DT reached accuracy of 40–50 %, which is, however, insufficient.

Next method is RF. It works in principal quite similarly to DT, since it comprises of set of DTs. Method then evaluates all or selected trees separately. In a sense it can be considered as upgrade to DT and that's is why it has been chosen as next approach. For regression purposes, results from all trees are averaged. This approach attained approximately 72 % accuracy. Which is significantly more than DT. For this reason RF is then probed deeper in an effort to obtain the best outcome.

Two remaining procedures are variations of previously discussed RF. First one improves input dataset by scaling each feature by standard deviation. Nonetheless, since the data are quite similar and do not have the aspect of diversity the improvements ware not significant, accounting only for around 1 percent.

The last one performs exhaustive search over specified hyper parameters and tries to find the best set of them. It is a process where ranges of tuning parameters are provided then `RandomizedSearchCV` randomly picks (with specified number of times) from the sets and performs K-fold cross-validation. Best parameters are determined after the computation is done. For even finer tuning, `GridSearchCV` with narrower ranges obtained from previous cross-validation is conducted. This time however, `GridSearchCV` unlike `RandomizedSearchCV` tries every combination from given parameter ranges.

Selected tuning hyper parameters for RF were: `max_depth`, `min_samples_leaf`, `min_samples_split`, `n_estimators`, `bootstrap`. The best score of 76.293 % was achieved with set of parameters (5, 2, 100, 1125, True). `RandomizedSearchCV` applied 50 combinations running each twice results in total of 100 iterations. `GridSearchCV` performs 162 runs (combination of each parameter).

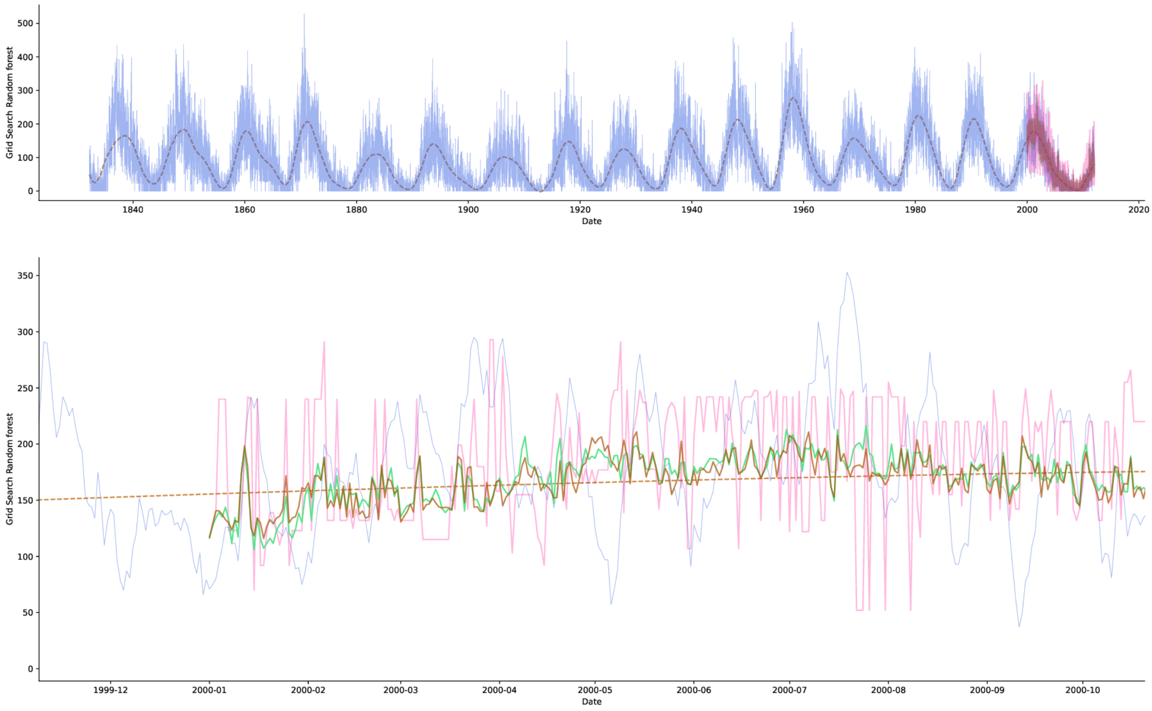


Figure 5.6: The picture above depicts sunspots (light blue curve), seasonal trend (dashed orange), prediction by DT (pink), RM (green] and grid search (red). The bottom picture is zoom in on the last part of the graph with tested outcome.

Chapter 6

Machine learning for prediction

This chapter gathers previous discoveries and utilizes them. The first part is all about selection process and discussing NNs and their intricacies. The chapter will then present selected machine learning approach, performance, consequences of various models and then carry on with implementation portion of the work. Implementation has been conducted using programming language Python, and primarily ML library Keras[1] running on top of the ML platform TensorFlow, and another ML module for machine learning – Sklearn[2]. As far as Gaussian processes, libraries GPflow[3] and Mogptk[4] have been used.

6.1 Choice of the approach

As mentioned already in previous chapters, orbital elements (namely eccentricity e, semi-major axis a, inclination i, argument of periapsis ω, RAAN Ω and mean anomaly M) are time dependent varying 6 features, dependent on one another. For that reason the problem of OE prediction requires sort of state persistence. Traditional NN are usually based on single or handful of inputs. In this case hundreds up to several thousands of input values are feed into the NN. Recurrent Neural Network (RNN) addresses this issue perfectly. It can handle information and prolong its existence within NN. This system is based on formation of loops (see figure 6.2) allowing the information to be passed from one step to the next one. Thanks to this ability, RNN are, thus, very popular in regression and prediction tasks where historical data play an important role in its subsequent progress.

There have been also several studies conducted based on different approaches such as Support Vector Machines (SVM) [26], Gradient Boosting (GB)[17], Gaussian Process (GP) [27], Long Short-Term Memory (LSTM) [32] etc. From which two methods (GP, LSTM) have been probed here into more detail. Researches in the preceding works focus primarily on prediction based on force state vector, meaning position and velocity vector. Therefore classical orbital elements (COE) have been chosen as a set of input parameter for the forecast. Since the ground truth timeseries of each element is known, the goal here would be to minimize deviation of the predicted values relative to the ground truth.

[1] https://keras.io/

[2] https://scikit-learn.org/stable/

[3] https://gpflow.github.io/GPflow/2.7.1/index.html

[4] https://games-uchile.github.io/mogptk/

Three accuracy measures are used for this purpose: Mean Squared Error (MSE) 6.1, Mean Absolute Error (MAE) 6.2 and Mean Absolute Percentage Error ($MAPE$) 6.3, defined as follows

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \tag{6.1}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{y}_i - \hat{\mathbf{y}}_i| \tag{6.2}$$

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} |\mathbf{y}_i - \hat{\mathbf{y}}_i| \tag{6.3}$$

All of the accuracy measures are based on the historical data and compare ground truth vector $\mathbf{y}$ to forecasted $\hat{\mathbf{y}}$, N denotes the number of predicted timesteps. The resulting values can be between $\langle 0; \infty)$.

Thanks to the NN ability of learning underlying connections between the data [45]. As mentioned in preceding study [27], motivation in this machine learning does not need to be purely ML-oriented, it may serve as a complement to an already existing system.

6.1.1 Gaussian processes

Regression based on GP is upon probabilistic Bayesian model. This introduces uncertainty to the ML which is desired. Hence, GP can be described as normal distribution over functions, which are defined by mean function and covariance function. The GP's objective is to model each orbital element as a mean function of time and extrapolate for given hours into the future. This also results in confidence region around mean (see the 6.1). Clear advantage of GP is that it does not require a fixed time step, unlike LSTMs, which do need additional parameter or layer to fulfill the varying time condition. In the work [27] conducted by Peng and Bai. The GP has been used to for orbit prediction modification based on estimation errors.

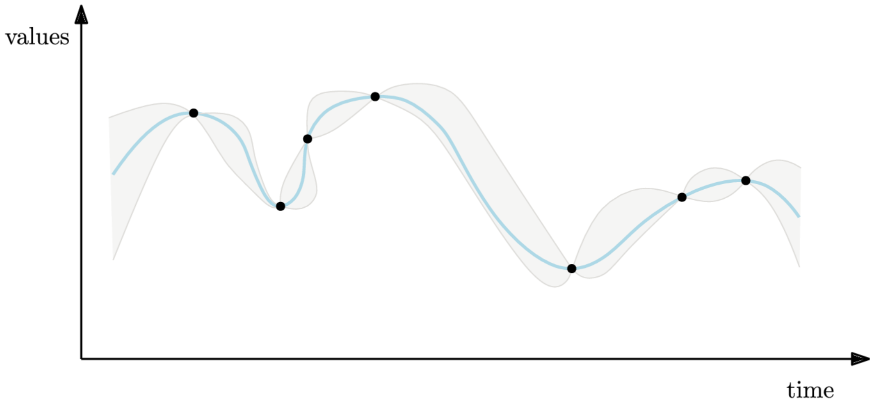


Figure 6.1: Example of a Gaussian process. Blue line denotes mean function and gray field represents 1σ confidence range. The graph does not contain any real values, as it is only an illustrative example.

6.1.2 Long short-Term Memory

The LSTM is a type of RNN, which performs very well at handling sequential data such as time series. Its best advantage lies in its capability of remembering previous states, based on which it decides how to adjust its inner parameters when learning from succeeding states. For that reason LSTMs are applied in various domains such as speech recognition, language models, image and video classification.

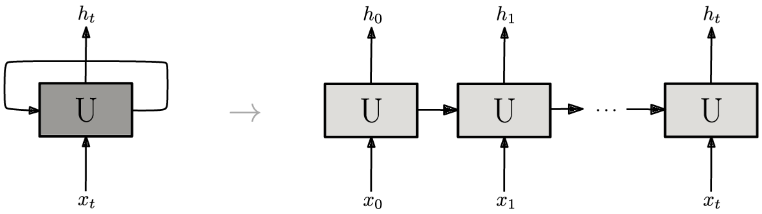


Figure 6.2: Unfold long short-term memory architecture.

Likewise to convolution or smoothing functions, sliding window is applicable also to LSTMs. Sliding window can be divided into two main parts containing: past and future timesteps. The vector (in fact in case of Tensorflow – tensor) of the past P measurements serves as an input into the LSTM model. By contrast, F denotes vector of future measurements. F vector for single-output contains 6 OE, or arbitrary consecutive sequence of those for the multi-output LSTM. The process works as follows. Sliding windows begins first iteration in the time t, P number of preceding are leveraged as an input, future F measurements are then compared with the LSTM output, improvements are back-propagated and sliding window moves one timestep forward. The process then repeats itself until the end of the entire timeseries.

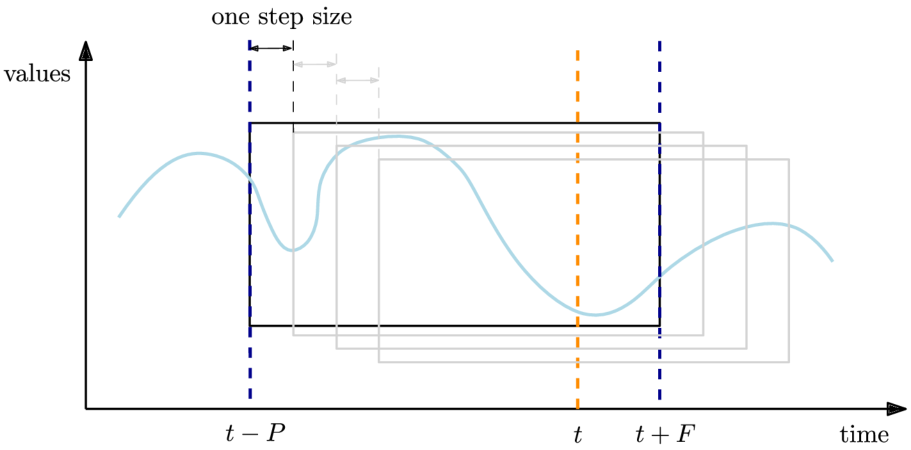


Figure 6.3: Sliding window illustration. P denotes number of the past measurements and F future forecasted ones.

6.2 Neural network implementation

Since the problem of COE time series prediction works with six COE, two possible ways how to deal with the size of an input vector come out. The first one is concerned with just a single time series, therefore is referred to as univariate regression. In contrast to that, multivariate LSTMs work with more than one input feature and the sequence is composed of vector of values. Dealing with only one input feature, however, encapsulates problem to only one OE, which is not intended in this case, therefore, multivariate approach must be chosen. GPs, however, didn't work out for multivariate use case, therefore the main focus of rest of the chapter will be dedicated to LSTMs.

Next question relates to the number of trained objects, whether to train only on one set of OE or multiple. Training on multiple RSOs is possible assuming that the objects are somehow related to each other, otherwise it would not make sense to train a model on several RSOs. Due to the fact that each object has a different properties and moves independently through the space, single model training has been chosen. This raises opportunity to leverage clustering. Since all RSOs are divided into groups with similar features, the training can be focused on one specific group of the RSOs. This approach can be performed only when it is certain that clustering is well performed. This may bring enhancement in specific types of RSO trajectories, on the contrary may lead to overly narrow conception of the variety of trajectories.

6.2.1 Preliminaries

Process of preparation of the object before feeding into the NN can be unfold into several steps:

1. Filtering – outliers and anomalous values removal.
2. Cut off – exclusively for decayed objects, lath portion of the data contains usually disturbed measurements, hence the learning and subsequent prediction would be incorrect.
3. Flattening – transformation of the saw-like courses of the last 3 OE (ω, Ω and M) to continuous curve, big leaps are caused by measuring in degrees (0°, 360°), continues function expands range of values to $(-\infty,\infty)$, In the postprocessing phase, these values are then converted back using modulo utility.
4. Interpolation – optional step, used for shorter timeseries, ensures fixed-sized timestep.
5. Scaling – necessary and arguably the most important part of the preprocessing phase, using raw data leads to biased assumptions about individual OE timeseries, scaling ensures that data lie in certain range, inverse transformation is performed after completion of the learning and prediction phase.

6.2.2 Model description

The key component of any LSTM and machine learning in general is its model. There are many models tailored specifically for LSTMs, each model has unique features, advantages and drawbacks. As mentioned before, LSTMs dispose of variety of applications, hence, different models work better and some worse.

There are multiple recurrent layers available in Keras and some of them will be mentioned here. The main structure comprises always of one Input layer and one Output layer. Based on the shape of the “past” part of the sliding window P and number of predicted features, the Input and Output layers, respectively, would have corresponding number of LSTM units. It should be noted here that increasing the unit count or number of layers does not necessarily enhance model positively or at all [30].

The simplest model is vanilla LSTM, which comprises of arbitrary stacked LSTM layers on top of each other. Little extension of the model can be done by using LeakyReLu (Rectified Linear Unit) is a function used in many types of NN along with *tanh*, *sigmoid*. It outputs the input directly if it is positive, otherwise, it will output zero.

This model has been adapted from [32]. It is an extension to the Vanilla LSTM with added Dropout layer. Its purpose is to prevent overfitting since given percentage of neurons/units will be displaced.

Bidirectional model is variation of Vanilla LSTM with difference that it processed the input in two directions (forth and back). Because of two ways of processing, bidirectional model includes two layers to carry out both directions. Outputs from both are afterwards concatenated and fed to the hidden Dense layer.

Another models are Encoder-decoder [45] or Sequence2sequence [20]. Both models are utilized primarily for translators and occasions where variable input and output lengths are encountered.

Lastly mentioned model here will be Convolutional Neural Network Encoder-decoder model. CNNs are usually used in pattern recognition tasks such as images. Here it tries to find underlying features of the data [43]. The system may be assembled from 2 CNN and one flattening layer. The rest is encoder-decoder based model.

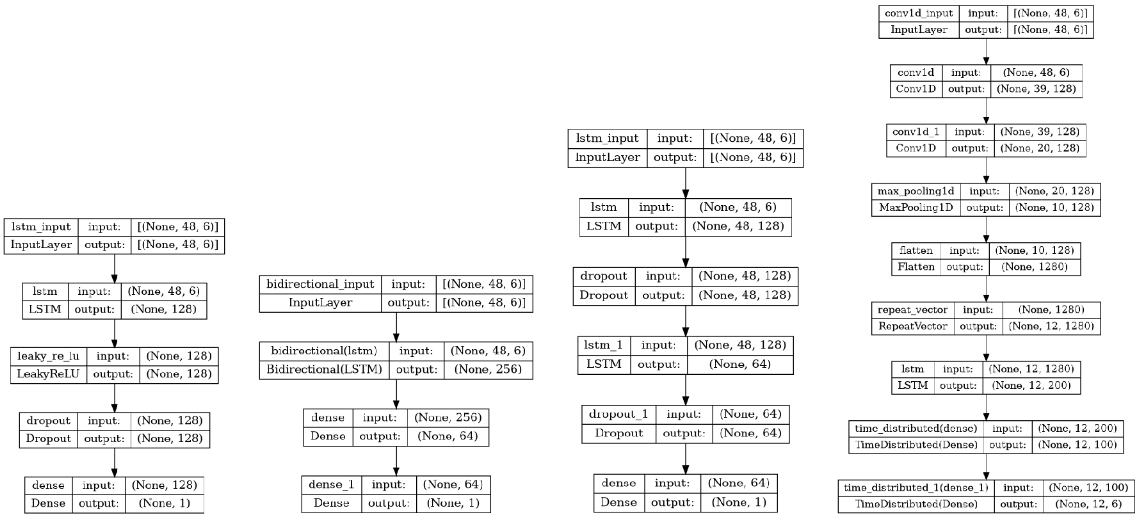

lstm_input | InputLayer | input: [(None, 48, 6)] | output: [(None, 48, 6)]
lstm | LSTM | input: (None, 48, 6) | output: (None, 128)
leaky_re_lu | LeakyReLU | input: (None, 128) | output: (None, 128)
dropout | Dropout | input: (None, 128) | output: (None, 128)
dense | Dense | input: (None, 128) | output: (None, 1)
bidirectional_input | InputLayer | input: [(None, 48, 6)] | output: [(None, 48, 6)]
bidirectional(lstm) | Bidirectional(LSTM) | input: (None, 48, 6) | output: (None, 256)
dense | Dense | input: (None, 256) | output: (None, 64)
dense_1 | Dense | input: (None, 64) | output: (None, 1)
lstm_input | InputLayer | input: [(None, 48, 6)] | output: [(None, 48, 6)]
lstm | LSTM | input: (None, 48, 6) | output: (None, 48, 128)
dropout | Dropout | input: (None, 48, 128) | output: (None, 48, 128)
lstm_1 | LSTM | input: (None, 48, 128) | output: (None, 64)
dropout_1 | Dropout | input: (None, 64) | output: (None, 64)
dense | Dense | input: (None, 64) | output: (None, 1)
conv1d_input | InputLayer | input: [(None, 48, 6)] | output: [(None, 48, 6)]
conv1d | Conv1D | input: (None, 48, 6) | output: (None, 39, 128)
conv1d_1 | Conv1D | input: (None, 39, 128) | output: (None, 20, 128)
max_pooling1d | MaxPooling1D | input: (None, 20, 128) | output: (None, 10, 128)
flatten | Flatten | input: (None, 10, 128) | output: (None, 1280)
repeat_vector | RepeatVector | input: (None, 1280) | output: (None, 12, 1280)
lstm | LSTM | input: (None, 12, 1280) | output: (None, 12, 200)
time_distributed(dense) | TimeDistributed(Dense) | input: (None, 12, 200) | output: (None, 12, 100)
time_distributed_1(dense_1) | TimeDistributed(Dense) | input: (None, 12, 100) | output: (None, 12, 6)

Figure 6.4: Here is the structure and the layers of some models visualized using Keras’ plot model utility. Right model is the only multi-output one.

6.2.3 Callbacks

There are two types of callbacks applied. Into the first category fall `EarlyStopping` and `TerminateOnNaN`. They are called after each epoch and can terminate fitting of the model if some requirement is met. `EarlyStopping` controls development of a training and validation loss. It prevents overfiting if the model stagnates for longer number of epochs. `TerminateOnNaN` stops training if it fails and encounters NaN value.

Second category are loggers. They are supposed to store model after the completion of the training. `CSVLogger` captures progress of loss function and `ModelCheckpoint` stores fitted model. Both callbacks are called after the last epoch.

6.3 Analysis and testing

The best results have been achieved using multivariate single-output LeakyRelu and Bidirectional models. After several iterations, 128 units for the first layer turned out to perform quite well.

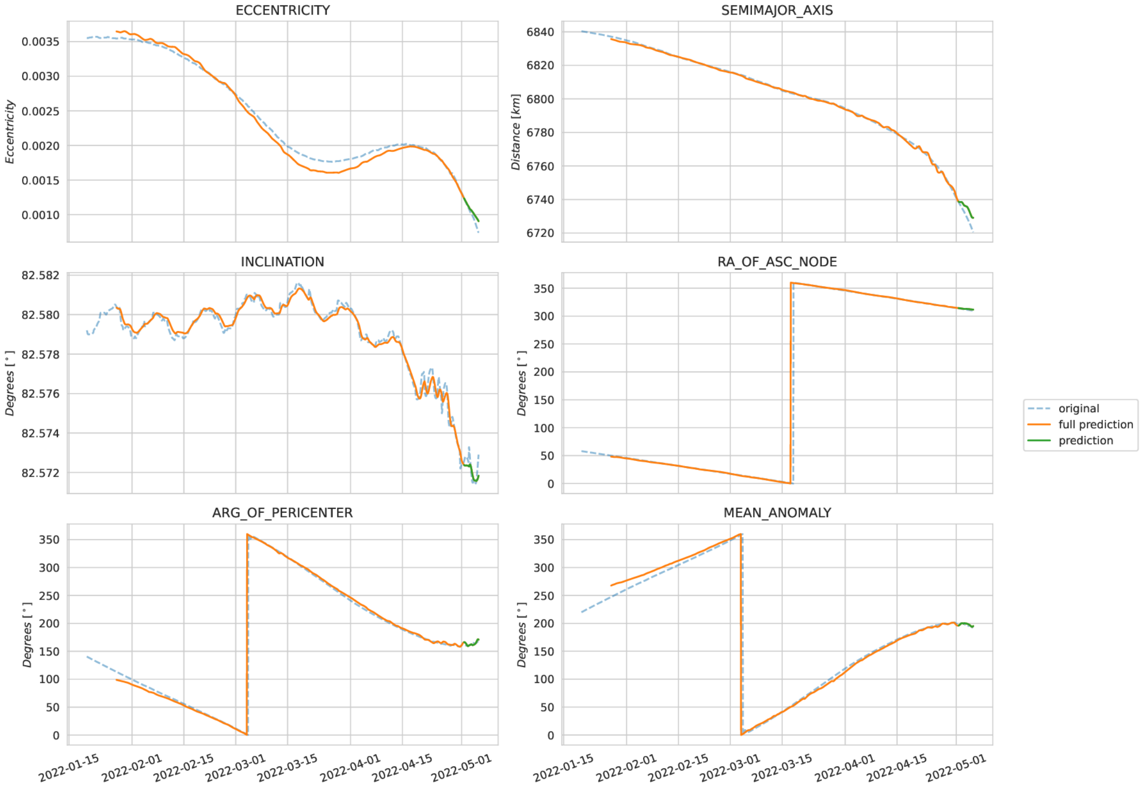


Figure 6.5: Successfully trained model on RSO 51236.

On the other hand, “dropout” model adapted from [32] performed worse relative to the already mentioned ones. Other models sometimes even failed to train the model and got terminated by callbacks, establishing them as unusable.

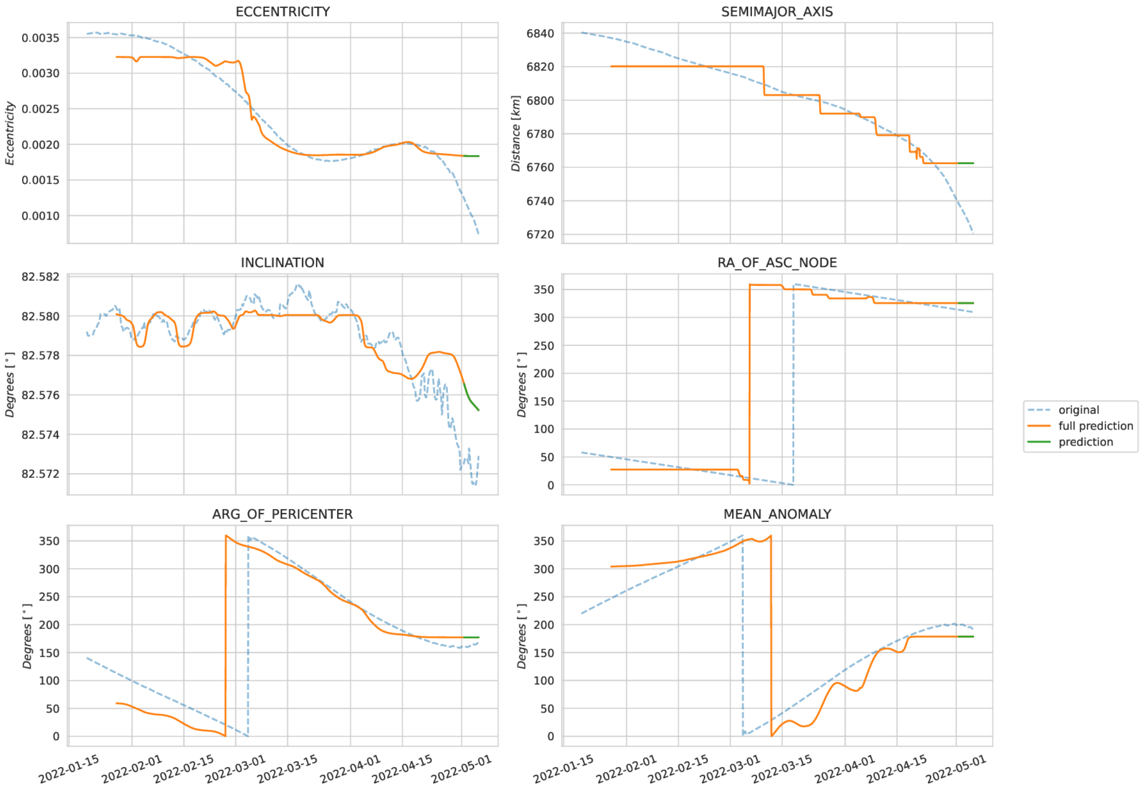


Figure 6.6: Wrong choice of the model lead to unsuccessful prediction on RSO 51236.

Insufficient model forecast performance with regard to the long timeseries (figure 6.7) can be caused by the nature of the Long Short-term memory itself, because the long-term part could loose the sense of the data from the beginning. Prediction is generally not bad in first OE threesome, yet delayed by a fraction of the period in case of last three OE.

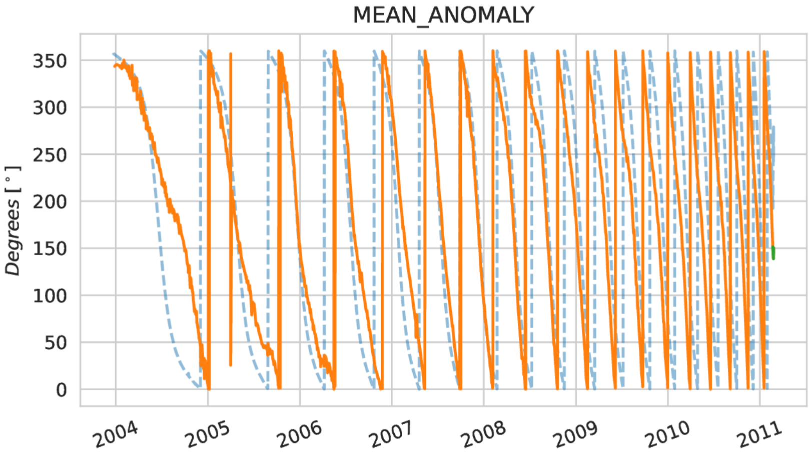


Figure 6.7: This is an example of long-term failed training. Majority of the seven-year course is shifted by a fraction of the period both to the past and future.

The ultimate goal in training is minimising the “gap” between the training and the validation loss curves as seen in the graph 6.8.

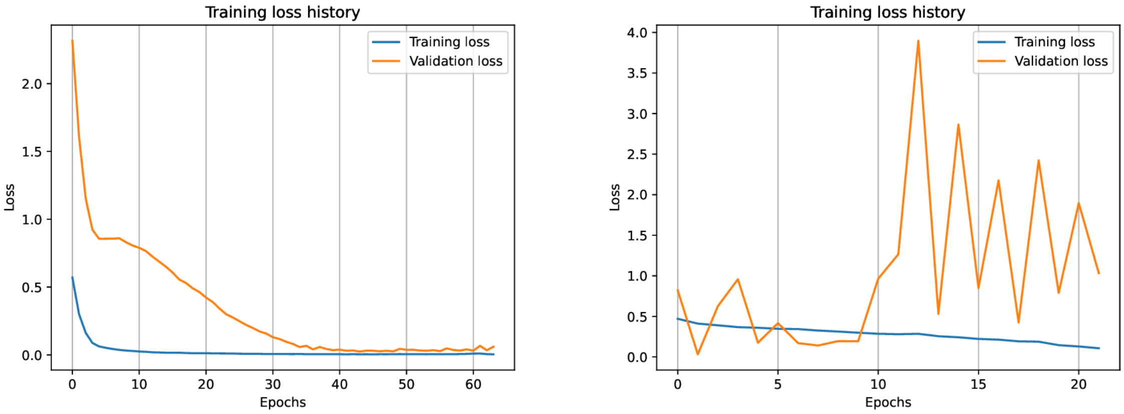


Figure 6.8: On the left-hand side, course of training and validation loss of the LeakyReLu model after each epoch is displayed. There were 100 epoch in total, but model managed to reach desired fitness and got terminated by the EarlyStopping callback. On the contrary the right one using CNN Enc-dec architecture was not suited for the prediction and didn’t accomplish learning phase and was due to the raised validation loss terminated.

6.3.1 Dead ends

It is clear from the course of OE, that they vary with time. This rate of change can be described analytically using Variation of parameters developed by Lagrange and later updated by Gauss [40]. The Lagrange and Gauss planetary equations (LPE and GPE) are two most recognized methods to describe the change rates of orbital elements with the partials of the disturbing forces, from which one can derive the dependence on other elements. The idea was to solve each ODE and its resulting curve use into a covariance function of a GP. However, that turned out to be wrong assumption. Similarly, since there are such relationships between some elements, multivariate LSTM was trained only on those coupled elements. Nonetheless, that lead to overfitting and resulting prediction resembled univariate timeseries forecast.

Adding solar activity to the existing features proved to be misleading. Due to the Sun’s stochastic nature, included features behaved more like a noise. Therefore, incorporating solar flux and sunspots number only decreased model performance.

Chapter 7

Conclusion

The aim of this thesis was to develop a system capable of future orbit prediction using a machine learning. Nonetheless, the Earth's orbit is very complex and puts many obstacles in the way. Hence, several steps including data preparation and filtering must be undergone. For even better grasp of the data, clustering was applied which divided objects further into groups. The goal was accomplished by training a neural network on single object at a time, and then predicting its future position in the space. In conclusion, proposed Long Short-Term Memory machine learning approach proved to be suitable choice for classical orbital elements predictions. Multivariate prediction captured some of the underlying connections between the elements and managed to model several days into the future satisfactorily.

A further development may be focused on incorporation of the perturbing forces. The study could lead into advancements in LSTM models with more complex structure and additional layers.

Bibliography

[1] Caldas, F. M. and Soares, C. Machine Learning in Orbit Estimation: a Survey. *CoRR*. 1st ed. 2022, abs/2207.08993. DOI: 10.48550/arXiv.2207.08993. Available at: https://doi.org/10.48550/arXiv.2207.08993.

[2] Coakley, J. REFLECTANCE AND ALBEDO, SURFACE. In: Holton, J. R., ed. *Encyclopedia of Atmospheric Sciences*. Oxford: Academic Press, 2003, p. 1914–1923. DOI: https://doi.org/10.1016/B0-12-227090-8/00069-5. ISBN 978-0-12-227090-1. Available at: https://www.sciencedirect.com/science/article/pii/B0122270908000695.

[3] Curtis, H. D. Chapter 1 - Dynamics of Point Masses. In: Curtis, H. D., ed. *Orbital Mechanics for Engineering Students (Third Edition)*. Third Editionth ed. Boston: Butterworth-Heinemann, 2014, p. 1–57. DOI: https://doi.org/10.1016/B978-0-08-097747-8.00001-3. ISBN 978-0-08-097747-8. Available at: https://www.sciencedirect.com/science/article/pii/B9780080977478000013.

[4] ESA. *ESA's Space Debris Office* [online]. ESA, december 2022 [cit. 2022-12-30]. Available at: https://www.esa.int/Space_Safety/Space_Debris/Space_debris_by_the_numbers.

[5] Folkner, W. M., Williams, J. G., Boggs, D. H., Park, R. S. and Kuchynka, P. The Planetary and Lunar Ephemerides DE430 and DE431. *Interplanetary Network Progress Report*. feb 2014, 42-196, p. 1–81.

[6] Geier, M. *Polynomial Curves in Euclidean Space* [online]. 2023 [cit. 2023-10-03]. Available at: https://splines.readthedocs.io/en/latest/euclidean/index.html.

[7] Gregersen, E., Bhutia, T. K. and Lotha, G. *Precession of the equinoxes* [online]. Encyclopedia Britannica, march 2022 [cit. 2023-03-01]. Available at: https://www.britannica.com/science/precession-of-the-equinoxes.

[8] Heather Cowardin, P. and Shoots, D. *National Aeronautics and Space Administration - Orbital Debris Quarterly News* [online]. 2022 [cit. 2022-12-30]. Available at: https://www.orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv26i2.pdf.

[9] Huang, C., Liu, D.-D. and Wang, J.-S. Forecast daily indices of solar activity, F10.7, using support vector regression method. *Research in Astronomy and Astrophysics*. 1st ed. jun 2009, vol. 9, no. 6, p. 694. DOI: 10.1088/1674-4527/9/6/008. Available at: https://dx.doi.org/10.1088/1674-4527/9/6/008.

[10] Ilian Bonev, E. *How is orientation in space represented with Euler angles?* [online]. July 2017. 2023-03-01. Available at: https://www.mecademic.com/en/how-is-orientation-in-space-represented-with-euler-angles.

[11] Izzet, A. The Influence of the Earth's Albedo on Low-orbit Satellite Motion. *NeuroQuantology*. may 2022, vol. 20, p. 408–418. DOI: 10.14704/nq.2022.20.5.NQ22189.

[12] Jo, J. H., Park, I. K., Choe, N. and Choi, M. The Comparison of the Classical Keplerian Orbit Elements, Non-Singular Orbital Elements (Equinoctial Elements), and the Cartesian State Variables in Lagrange Planetary Equations with J2 Perturbation: Part I. *Journal of Astronomy and Space Sciences*. The Korean Space Science Society. march 2011, vol. 28, no. 1, p. 37–54. DOI: 10.5140/JASS.2011.28.1.037. Available at: http://dx.doi.org/10.5140/JASS.2011.28.1.037.

[13] Kessler, D. J. and Cour Palais, B. G. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research*. 1978, vol. 83, p. 2637–2646.

[14] Kong, Q., Siauw, T. and Bayen, A. M., ed. Introduction to Numerical Methods. In: Kong, Q., Siauw, T. and Bayen, A. M., ed. *Python Programming and Numerical Methods*. Academic Press, 2021, p. xiv–xxiv. DOI: https://doi.org/10.1016/B978-0-12-819549-9.00006-3. ISBN 978-0-12-819549-9. Available at: https://www.sciencedirect.com/science/article/pii/B9780128195499000063.

[15] Kubo oka, T. and Sengoku, A. Solar radiation pressure model for the relay satellite of SELENE. *Earth, Planets and Space*. Sep 1999, vol. 51, no. 9, p. 979–986. DOI: 10.1186/BF03351568. ISSN 1880-5981. Available at: https://doi.org/10.1186/BF03351568.

[16] Levit, C. and Marshall, W. Improved orbit predictions using two-line elements. *Advances in Space Research*. 2010, vol. 47, p. 1107–1115.

[17] Li, B., Zhang, Y., Huang, J. and Sang, J. Improved orbit predictions using two-line elements through error pattern mining and transferring. *Acta Astronautica*. 2021, vol. 188, p. 405–415. DOI: https://doi.org/10.1016/j.actaastro.2021.08.002. ISSN 0094-5765. Available at: https://www.sciencedirect.com/science/article/pii/S0094576521004045.

[18] Liu, Z. Y.-C., Tarlow, S., Akbar, M., Donnellan, Q. and Senkow, D. Improved Orbital Propagator Integrated with SGP4 and Machine Learning. In: LLC, H. I., ed. *Proceedings of the AIAA/USU Conference on Small Satellites, Mission Lessons, SSC21-IX-02* [online]. 1st ed., verze 1.0. Conference on Small Satellites, July 2021, vol. 24, no. 9, chap. 2, p. 1–5, revised 9. 7. 2021 [cit. 2023-02-27]. Advanced Technologies III. Available at: https://digitalcommons.usu.edu/smallsat/2021/all2021/193.

[19] Mallama, A., Krobusek, B. and Pavlov, H. Comprehensive wide-band magnitudes and albedos for the planets, with applications to exo-planets and Planet

Nine. *Icarus*. 1st ed. Elsevier BV. jan 2017, vol. 282, p. 19–33. DOI: 10.1016/j.icarus.2016.09.023. Available at: https://doi.org/10.1016%2Fj.icarus.2016.09.023.

[20] Mariet, Z. and Kuznetsov, V. Foundations of Sequence-to-Sequence Modeling for Time Series. In: Chaudhuri, K. and Sugiyama, M., ed. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. PMLR, 16–18 Apr 2019, vol. 89, p. 408–417. Proceedings of Machine Learning Research. DOI: 10.48550/arXiv.1805.03714. Available at: https://proceedings.mlr.press/v89/mariet19a.html.

[21] Mathar, R. J. Improved First Estimates to the Solution of Kepler's Equation. *ViXra*. 2022.

[22] Montenbruck, O. and Gill, E. Force Model. In: *Satellite Orbits: Models, Methods and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, p. 53–116. DOI: 10.1007/978-3-642-58351-3_3. ISBN 978-3-642-58351-3. Available at: https://doi.org/10.1007/978-3-642-58351-3_3.

[23] Neumann, T. *Orbital basics* [online]. August 2019 [cit. 2023-02-03]. Available at: https://t-neumann.github.io/space/OrbitalBasics/.

[24] Oceanic, N. and Administration, A. *ABOUT SPACE WEATHER* [online]. March 2022 [cit. 2023-03-20]. Available at: https://www.swpc.noaa.gov/about-space-weather.

[25] Peet, M. M. *Spacecraft Dynamics and Control - Lecture 13: The Effect of a Non-Spherical Earth* [online]. April 2020. 2023-03-03. Available at: http://control.asu.edu/Classes/MAE462/462Lecture13.pdf.

[26] Peng, H. and Bai, X. Improving Orbit Prediction Accuracy through Supervised Machine Learning. *Advances in Space Research*. january 2018, vol. 61. DOI: 10.1016/j.asr.2018.03.001.

[27] Peng, H. and Bai, X. Fusion of a machine learning approach and classical orbit predictions. *Acta Astronautica*. 2021, vol. 184, p. 222 – 240. DOI: 10.1016/j.actaastro.2021.04.017. Cited by: 6; All Open Access, Bronze Open Access. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85104754441&doi=10.1016%2fj.actaastro.2021.04.017&partnerID=40&md5=8b9241c18b909af51d551c52147c1ad5.

[28] Petit, G. and Luzum, B. IERS Conventions (2010). *IERS Technical Note*. jan 2010, vol. 36, p. 1.

[29] Rossum, G. van. *Python* [online]. 2023 [cit. 2023-01-02]. Available at: https://www.python.org/.

[30] Ruder, S. *Deep Learning for NLP Best Practices* [online]. ruder.io, july 2017 [cit. 2023-04-20]. Available at: https://www.ruder.io/deep-learning-nlp-best-practices/.

[31] SAIC. *Space-track.org - Space, TLE, orbit, satellites and space objects* [online]. 2004 [cit. 2022-11-22]. Available at: https://www.space-track.org/.

[32] Salleh, N., Mohd Azmi, N. F. and Yuhaniz, S. S. An Adaptation of Deep Learning Technique In Orbit Propagation Model Using Long Short-Term Memory. In: IEEE. *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE).* 2021, p. 1–6. DOI: 10.1109/ICECCE52056.2021.9514264. ISBN 978-1-6654-3897-1.

[33] Sang, J., Li, B., Chen, J., Zhang, P. and Ning, J. Analytical representations of precise orbit predictions for Earth orbiting space objects. *Advances in Space Research.* 2017, vol. 59, no. 2, p. 698–714. DOI: https://doi.org/10.1016/j.asr.2016.10.031. ISSN 0273-1177. Available at: https://www.sciencedirect.com/science/article/pii/S0273117716306251.

[34] Secan, J. A. *Graphs of geomagnetic-activity Kp Index from Potsdam and DSCOVR Solar Wind* [online]. December 2021 [cit. 2023-03-22]. Available at: https://spawx.nwra.com/spawx/kpsw.html.

[35] Secan, J. A. *Plot of the 10.7cm solar radio flux (F10)* [online]. October 2021 [cit. 2023-03-22]. Available at: https://spawx.nwra.com/spawx/f10.html.

[36] Rhodes, B. *Track Earth satellites given TLE data, using up-to-date 2020 SGP4 routines* [online]. April 22 [cit. 2023-10-04]. Available at: https://pypi.org/project/sgp4/.

[37] SILSO World Data Center. *The International Sunspot Number.* Royal Observatory of Belgium, avenue Circulaire 3, 1180 Brussels, Belgium: [b.n.], march 1818-2023. Available at: https://www.sidc.be/silso/datafiles.

[38] J. Sanz Subirana, J. J. Z. and Hernández Pajares, M. *Solid Tides* [online]. 2012 [cit. 2023-03-08]. Available at: https://merlin.fit.vutbr.cz/wiki/index.php/English_SOS.

[39] Bravo, H. C. *Unsupervised Learning: Dimensionality Reduction* [online]. April 20 [cit. 2023-03-14]. Available at: https://www.hcbravo.org/IntroDataSci/bookdown-notes/unsupervised-learning-dimensionality-reduction.html.

[40] VALLADO, D. A. 1 - Perturbed Motion. In: Gurfil, P., ed. *Modern Astrodynamics.* Butterworth-Heinemann, 2006, vol. 1, p. 1–22. Elsevier Astrodynamics Series. DOI: https://doi.org/10.1016/S1874-9305(07)80003-3. ISSN 1874-9305. Available at: https://www.sciencedirect.com/science/article/pii/S1874930507800033.

[41] Verbiscer, A., French, R., Showalter, M. and Helfenstein, P. Enceladus: Cosmic Graffiti Artist Caught in the Act. *Science.* feb 2007, vol. 315, no. 5813, p. 815. DOI: 10.1126/science.1134681.

[42] Williams, M. *What percent of Earth is water?* [online]. December 14 [cit. 2023-02-28]. Available at: https://phys.org/news/2014-12-percent-earth.html.

[43] Widiputra, H., Mailangkay, A. and Gautama, E. Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complexity.* Hindawi. Oct 2021, vol. 2021, p. 9903518. DOI: 10.1155/2021/9903518. ISSN 1076-2787. Available at: https://doi.org/10.1155/2021/9903518.

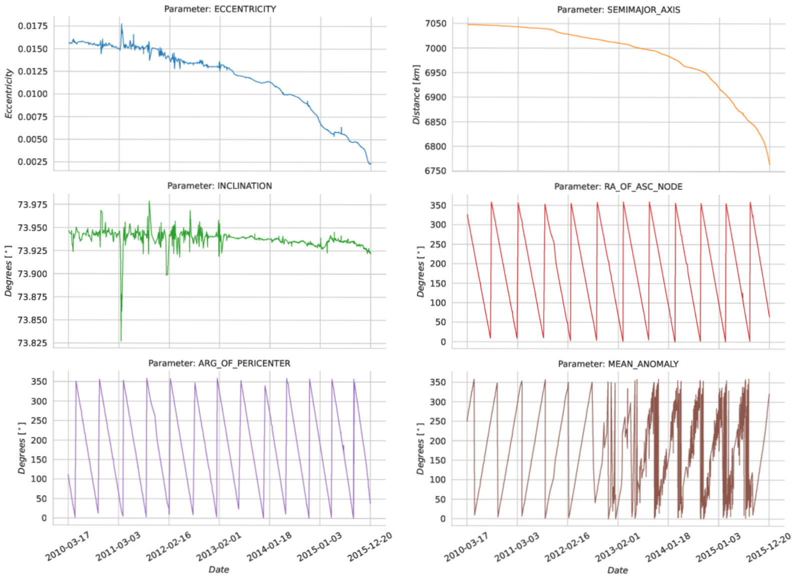
[44] Williams, D. D. R. *Planetary Fact Sheets* [online]. January 2016 [cit. 2023-02-03]. Available at: https://nssdc.gsfc.nasa.gov/planetary/planetfact.html.

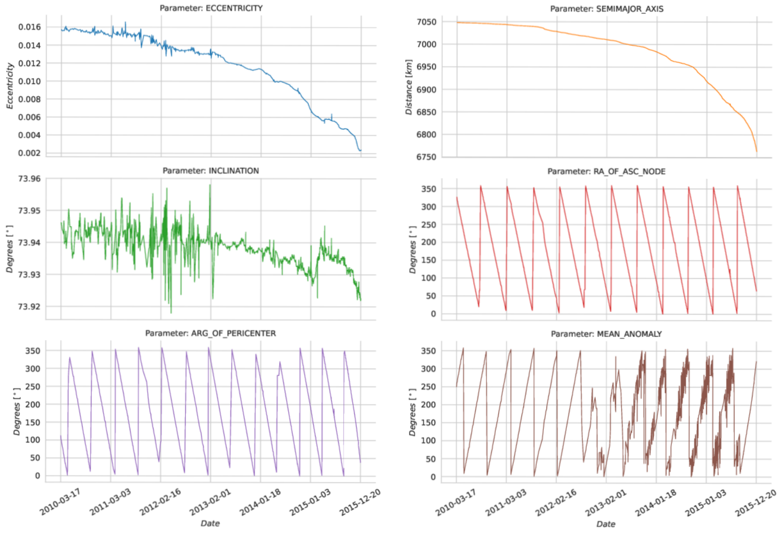
[45] Zhu, L. and Laptev, N. Deep and Confident Prediction for Time Series at Uber. In: IEEE. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, Nov 2017. DOI: 10.1109/icdmw.2017.19. ISBN 978-1-5386-3800-2. Available at: https://doi.org/10.1109%2Ficdmw.2017.19.

Appendix A

Data filtering

The following graphs show side-by-side comparison between unfiltered (up) and filtered (down) orbital elements.





Appendix B

Contents of the included storage media

Only following selected important files and folders are displayed. Capacity of the storage media is restricted to few GB of available memory. Therefore, only several hundreds of the RSOs are included.

```
/
└─ source ............................................ Source codes and
   └─ data ........................................... Input and output data
      └─ prediction ............................ Contains saved models and losses
      └─ selected ............... Downloaded data obtained from space-track.org
         └─ decayed
         └─ eccentric
         └─ gp_launched_debris
      └─ stars ........................ Contains coordinates and constellations.
      └─ sun .......................................... Data related to the Sun
         └─ sunspot_data.csv ........ Original sunspots and observations counts
      └─ analyzed_data.csv ...................... Detailed info about every file
      └─ clustereded_data.csv ............................... Clustering input
      └─ clustering_result.csv ............................. Clustering output
      └─ merged_results.csv .... Final output from the preprocessing and clustering
   └─ img ..................................... All image output goes here
   └─ clustering.py .................................... Clustering ensemble
   └─ constatnts.py ................................ Program constant values
   └─ earth.py ...................... Creates 3D mesh for the earth simulation
   └─ parse_data.py ..................... Transformer for any type of raw data
   └─ prediction.py ..................... Orbital elements predisction using ML
   └─ sim.py ............................................ 3D Vispy simulation
   └─ sky.py .......................... Display ISS on the sky relative to stars
   └─ sun_activity.py ................................. Sun activity prediction
   └─ trajectory.py ........................................ Deals with orbits
   └─ visualize.py ........................... Coniains all visualizimg methods
   └─ download_data_spacetrack ............ Shell script for space-track.org data
└─ thesis ....................................... Thesis latex source codes
└─ README.md ............................................... Thesis manual
└─ xjanda28-predicting-leo-trajectories.pdf ..................... Thesis text
```