Indian Institute of Technology

Kanpur

DATA MINING

CS685A :: COURSE PROJECT

Data Mining from Two Line Element(TLE) sets

Group No.- 45
Aaryansh Mohan Bansal
190013, arynshmb@iitk.ac.in
Third Year Undergraduate
Department of Aerospace, IITK

Prof. Arnab Bhattacharya
Department of Computer Science,
IITK

Data Mining from Two Line Element(TLE) sets

November 20, 2021

Abstract

Space Situational Awareness is a field which deals with development of tools and technology aimed towards getting information and insights about our space environment. Awareness about space environment has been very crucial and is becoming a concern, since rapid developments and much easier access to space has resulted in numerous private and government agencies launching their satellites into space. Over the last decade, their has been exponential increase in satellites, space debris and management of these man-made objects revolving around our planet is very important. Thus developing useful data-models from available raw data of these space objects is the need of the hour. Two-Line-Elements(TLE) are the source of data for getting estimates of positions of these space objects. This project deals with data mining from this data and using data analysis techniques to get meaningful insights into our space environment.

1 Introduction

Over the past few decades, the increase in number of space objects have led to crowding of orbits around the earth. As orbit grows more crowded, a human-made of space-debris has become of serious concern. Today, more than 21,000 debris objects measuring > 10 cm in diameter whiz overhead in excess of 20,000 km/h. These are joined by an estimated 500,000 1-10 cm diameter particles. Colliding with any one of these objects can debilitate or even destroy a satellite. Any single collision in a densely occupied area of orbit can even lead to cascading effect of destruction of satellites in that area.

Space Situational Awareness(SSA) is the field which provides the users estimates about positions, nature, and movement of space objects. This requires continuous updates in the form of observational measurements. Identifying and tracking minuscule objects moving at bullet-like velocities thousands of kilometers away is inordinately difficult, even with sophisticated astrometry equipment.

In this project, we deal with data mining from one of the class of data of SSA, i-e Two-Line-Elements(TLEs). We develop models using this data to get useful insights about our orbits. There are mainly five models that have been built:-1)Error Analysis, 2)Catalogue Information, 3)Clustering and Patterns, 4)Debris Prediction and 5)Maneuver Detection.

2 Background

2.1 Space Situation Awareness(SSA)

Getting SSA data is not an easy task and most satellite operators rely on third-party SSA data. This is because observational measurements are required to track space objects. These measurements require sophisticated astrometry platforms like radar-based technologies, that require ground tracking stations. However, a single ground station cannot reliably track objects. Instead, many observations are needed from sites distributed across the Earth. The cost of such a system is immense. This makes SSA data very hard to be obtained.

By far, the dominant source of SSA is the United States Space Surveillance Network(SSN). The SSN comprises more than 20 locations. It is believed to be the only system capable of tracking smaller objects measuring 5-10 cm in LEO and 1 m in GSO. The US military publicly posts SSA through Space-Track.org in the Two-Line Element Set (TLE) format.

2.2 Two-Line-Elements(TLEs)

This data standard was developed in the 1970's to facilitate the sharing of an object's ephemeris (its projected orbital path and position) using two 80-column punch cards[Fig-2]. The format of TLEs was chosen so that future and past estimates of the orbital path and position could be obtained by propagating the orbital elements(data fields in TLEs)[Fig-1] to the de-

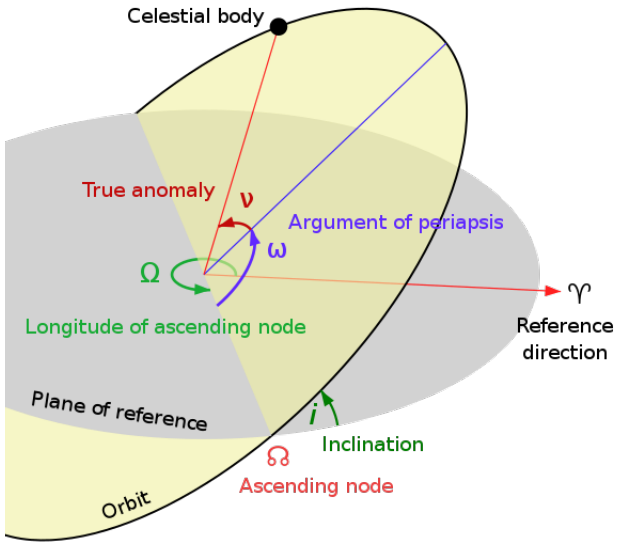

Figure 1: Orbital Elements used to determine position of object in orbit

sired epoch. The physical model chosen for propagation was Simplified Perturbation Model(SGP4), which basically performs astronomical integrations assuming a simpler physical model of the space environment.

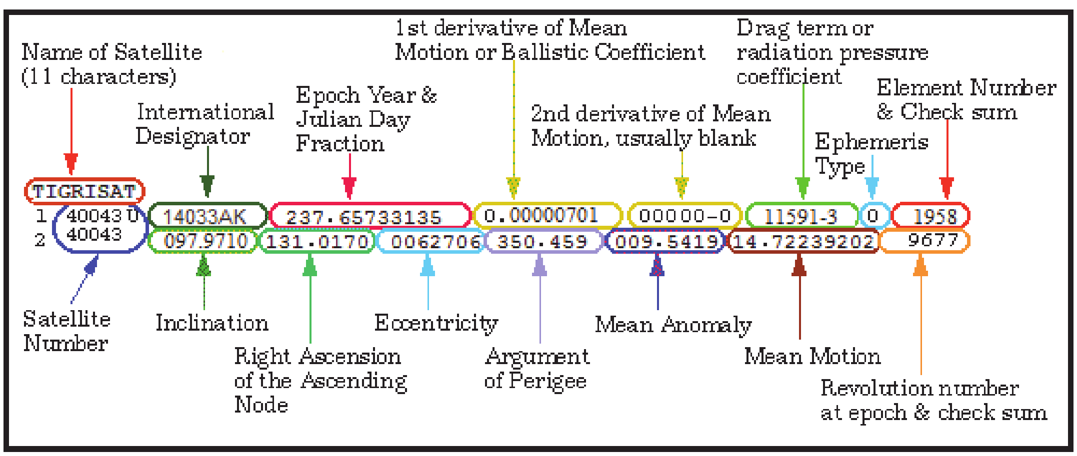

Figure 2: an example of a TLE for "TIGRISAT" satellite.

3 Project Work

As mentioned earlier, we have developed five modules based on principles of data mining. Purpose of each model is to gain some insights, predict future events or analyse significance of present data. We will elaborate on each of the model separately. The data for all the modules has been taken from www.space-track.org.

3.1 Error Analysis

In this module, we analyze the errors or accuracy of the predictions done using SGP4 model and TLEs. The predictions comprises of position, velocity estimates and error propagation. We have used the TLEs for International Space Station(ISS-ZARYA) and the online available python package for SGP4. The results are analyzed through various plots given below[Fig-3,4,5].

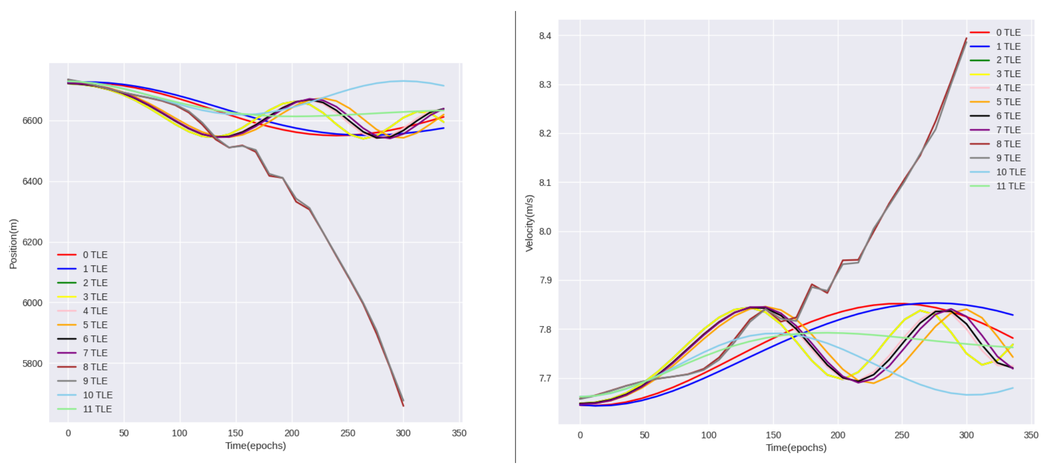

Figure 3: The graphs are plots of future estimates of position and velocity respectively. The variation is observed across 11 TLEs of the same object(ISS-ZARYA) on a timespan of 350 epochs

By the "Error Analysis" module for TLE propagation, we can conclude that predictions done by TLEs for future epochs is not good enough. The errors in positions are order of 100kms, which will be not good enough to prevent or predict possible collisions. Therefore, there is a need of sophisticated ML/AI models that can improve on their accuracies.

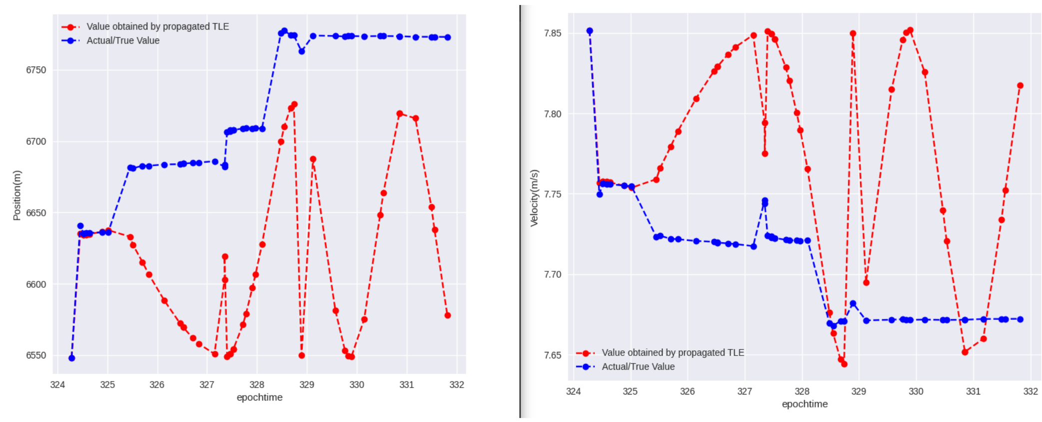

Figure 4: These graph analyses the error in positions and velocity for a single or specific TLE propagation over time.

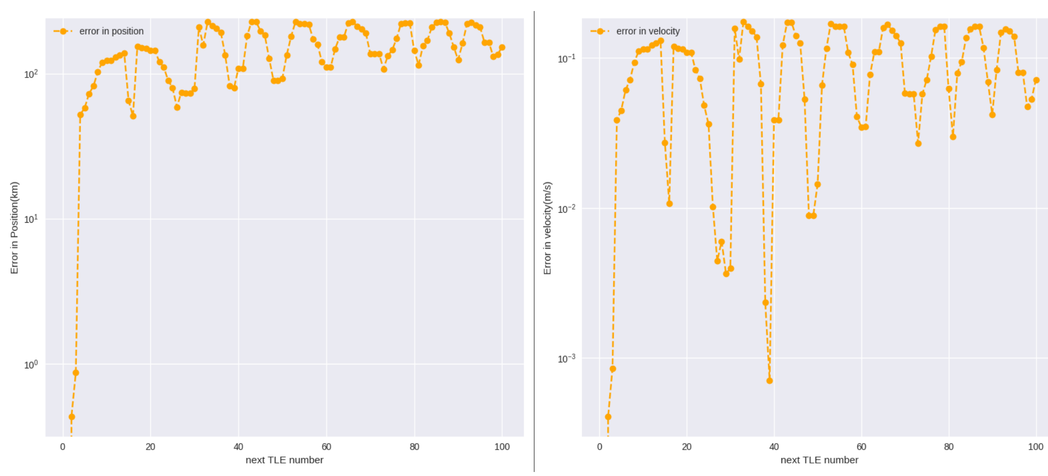

Figure 5: Plot showing variations in magnitudes of errors of positions and velocity as compared to next TLEs. A TLE is taken at a particular epoch and then it is propagated to future epochs, where we have the instantaneous TLE for that future epoch. Then the error between the position/velocity is calculated and this process is repeated for 100 such future epochs. Notice the initial exponential rise in errors and then remaining roughly within a specific region.

3.2 Catalogue Formation

This module converts the TLE data into useful and interpretable data ,i-e forms a catalogue. As mentioned above the format in which TLEs are published by the space agencies is very compact[Fig-2]. Therefore, it would be much better if we could convert the compact form of TLEs into a detailed tabular form, which would have columns contain the fields or information given by the TLEs. One can analyse the orbital elements and all other associated parameters from the tables generated by this module. Following are the fields present in the table :-

- **Satname:** Name of Satellite
- **Epoch of TLE (python date-time format)**
- **Mean motion (rev/day)**
- **Orbit period (s)**
- **Semi-major axis (m)**
- **Semi-latus rectum (m)**
- **Inclination (deg)**
- **Right Ascension of Ascending Node (Deg)**
- **Argument of Perigee (Deg)**
- **Eccentricity (unitless)**
- **Perigee altitude (km)**
- **Apogee altitude (km)**
- **Circular Orbit flag: based on eccentricity (as per threshold)**
- **Orbit Class: based on altitude (LEO <= 2000km; 2000 < MEO <= 35,636; 35,636 < GEO <= 35,836; DSO > 35,836)**
- **Annual precession of orbit per year (rad/yr)**
- **Sun-synchronous orbit flag (as per threshold)**

- **Mean Local Time of Ascending Node (hour)**
- **Ground-track repeat cycle (days)**
- **Mean anomaly (deg) or Mean longitude (deg)**

The features such as:- Sun-synchronous flag, Ground-track repeat cycle etc. have been calculated from the elementary features/field given by the TLE data. Thus, a user can also derive or calculate other related parameters as per the need. Moreover above feature representation would be helpful in developing other higher level data-science/ML models.

Note:- "Ground-track repeat cycle" refers to the time period after which the satellite revolving around the planet would exactly coincide with its initial position in reference to earth. i-e time interval in which the longitude, latitudes would both become exactly same. Derived this parameter from TLE propagation to maximum of 1 day(since errors are not significant until few(2-3) days). To our amazement, the results were quite accurate to actual cyclicity of some famous satellites like CARTOSAT, ENVISAT etc.

3.3 Clustering and Patterns

Clustering is an unsupervised Machine-Learning technique that helps us to find out patterns in data in form of clusters or groups. Without knowing actual classes or groups to which the data belongs, we are able to find out possible groups based on similarity between features of data.

In case of SSA data, we know that there are constellation of satellites(eg-STARLINK satellites) that have similarities in orbits. Satellites are also categorised in LEO(Low-earth orbits),MEO(middle-earth orbits),GEO groups depending upon the altitude at which they revolve. Thus, there are definitely some patterns or groups that can be indentified from TLE data. Since we do not have labels for our data, we therefore use unsupervised techniques such as Clustering to find desirable patterns.

The Clustering module can be broadly classified into three sub modules:-

3.3.1 Orbit Similarity

Satellites sharing the same or similar orbit(s) are most likely to be related either in ownership, intent or function. Finding these satellites, their relationships and distribution is key to obtaining a first guess of their intent and function. At times, satellites with clear similarity in orbits may be launched from different locations, with very different (official) names, by different space actors; but be performing activities on concert with each other. The objective here, is to detect these relationships purely based on publicly available tracking data to build the initial hypothesis, and pass on data in structured form for further analysis.

With the knowledge of semi-major axis (SMA), we partition the near-Earth RSOs into multiple shells: Tight altitude clusters containing objects within 2km of a notional shell SMA - outliers from which are passed to the next step Loose altitude clusters containing objects within 10km of a notional SMA - outliers from which are passed to the next step Broad altitude groups containing objects within 25km of a notional SMA outliers from the above, last step are marked as such to denote no relationships with other objects

For each altitude shell defined by a centroid SMA and accompanying set of RSOs, we can extract unique clusters of objects based on the planes they inhabit. Since the orbital plane is defined by the 2 angular parameters of inclination(i) and Right Ascension of Ascending Node(RAAN), clusters of objects with similar [i, RAAN] pairs can be considered to be in the same plane. As an extension, planes sharing the same value of inclination, but with RAANs displaying a pattern of $RAAN_j = RAAN_i + 360/\text{n}$, for every $RAAN_j > RAAN_i$, and where n = number of planes sharing the same inclination, within the same altitude shell.

There are multiple programs that link together to perform the above tasks. We have described in the sequence that they follow.

- **filter.py** This program simply removes objects from recent elset data that are of no concern, such as debrisis, or satellites that are non-functional etc.

- **kmeans+dbscan-catalogue.py** This program clusters the filtered objects based on their sma values in order to get the 3 types of altitude based clustering as mentioned in the documentation above. **Kmeans clustering** is performed for grouping based on sma values.

- **add-noradid-kmeans-data.py** In order to reduce computation, only some of the fields from the detailed catlogue were used in the above programs. Therefore, this program simply adds the rest of the required fields to the clustered data.

- **cluster-raan-inclination.py**This program finds clusters based on inc, raan values for objects that within sma labels, i-e objects within same altitude groups. The algorithm used is **DBSCAN** for optimal results.

	Norad_ID	Name	EPOCH	Label	dbscan_labels	SMA_Class	SMA_Cluster	Semi-major axis(m)	Inclination	Raan
0	46043	0 STARLINK-1582	2021-02-11 18:01:11	0.008	0	0	8	6925363.90984498	53.0564189710675	160.280257310456
1	46032	0 STARLINK-1555	2021-02-11 17:55:53	0.008	0	0	8	6925347.51282318	53.057018971282	160.290957314282
2	46038	0 STARLINK-1567	2021-02-11 17:39:57	0.008	0	0	8	6925371.77738504	53.0568189712105	160.345257333698
3	46031	0 STARLINK-1544	2021-02-11 17:13:25	0.008	0	0	8	6925356.85145155	53.0553189706742	160.425357362339
4	46030	0 STARLINK-1534	2021-02-11 17:08:06	0.008	0	0	8	6925326.90788087	53.0568189712105	160.441957368274
5	46045	0 STARLINK-1591	2021-02-11 17:02:48	0.008	0	0	8	6925364.58718199	53.0577189715323	160.459657374603
6	46041	0 STARLINK-1580	2021-02-11 17:01:27	0.008	0	0	8	6925376.9049368	53.0567189711748	160.463857376105
7	46028	0 STARLINK-1523	2021-02-11 16:52:11	0.008	0	0	8	6925332.68816009	53.0576189714966	160.483257383042
8	46029	0 STARLINK-1526	2021-02-11 16:41:35	0.008	0	0	8	6925397.19456528	53.060718972605	160.516157394806
9	46044	0 STARLINK-1584	2021-02-11 16:46:52	0.008	0	0	8	6925349.10961299	53.0608189726408	160.512457393483
10	46042	0 STARLINK-1581	2021-02-11 16:36:16	0.008	0	0	8	6925358.06820328	53.0606189725692	160.52995739974
11	46027	0 STARLINK-1522	2021-02-11 14:55:25	0.008	0	0	8	6925355.07996005	53.0569189712463	160.867657520489
12	46039	0 STARLINK-1569	2021-02-11 14:39:30	0.008	0	0	8	6925376.95091011	53.0563189710317	160.908257535007
13	46566	0 STARLINK-1740	2021-02-11 22:00:01	0.008	1	0	8	6925357.77704097	53.0572189713535	189.527267768135
14	46563	0 STARLINK-1730	2021-02-12 06:00:00	0.008	1	0	8	6925416.50356422	53.0564189710675	188.03006723279
15	46582	0 STARLINK-1728	2021-02-12 06:00:00	0.008	1	0	8	6925402.55816259	53.0549189705311	188.034667234435
16	46584	0 STARLINK-1732	2021-02-12 06:00:00	0.008	1	0	8	6925454.30361797	53.0561189709602	188.028867232361
17	46571	0 STARLINK-1753	2021-02-12 06:00:00	0.008	1	0	8	6925402.82174536	53.0556189707814	188.024767230895
18	46569	0 STARLINK-1747	2021-02-11 22:00:01	0.008	1	0	8	6925366.00621887	53.0554189707099	189.524867767277
19	46564	0 STARLINK-1733	2021-02-11 22:00:01	0.008	1	0	8	6925373.36499419	53.0627189733201	189.489967754798
20	46570	0 STARLINK-1748	2021-02-12 12:00:02	0.008	1	0	8	6925336.0012524	53.0581189716754	186.90366683003
21	46567	0 STARLINK-1741	2021-02-12 06:00:00	0.008	1	0	8	6925447.22353945	53.0773189785406	187.936667199394
22	46541	0 STARLINK-1687	2021-02-11 22:00:01	0.008	1	0	8	6925296.06358314	53.057018971282	189.526467767849
23	46562	0 STARLINK-1714	2021-02-12 12:00:02	0.008	1	0	8	6925315.2002473	53.0553189706742	186.915866834392
24	46591	0 STARLINK-1755	2021-02-12 06:00:00	0.008	1	0	8	6925432.42899224	53.0848189812223	187.892067183446
25	46565	0 STARLINK-1735	2021-02-12 06:00:00	0.008	1	0	8	6925435.98433509	53.0405189653822	188.106267260037
26	44747	0 STARLINK-1042	2021-02-12 07:30:38	0.008	2	0	8	6925377.47194102	53.0565189711032	257.757592164846

Figure 6: Screenshot of data-table resulting from above clustering programs. The "dbscan-labels" are same for objects within same orbital plane. "SMA-Class" represents wether they are in within tight altitudes of 2kms, 10kms or farther apart whereas "SMA-Cluster" refers to class number based on closeness of semi-major axis(SMA). Notice how our algorithm gives same "dbscan-labels" to Starlink-1500 series and different labels to Starlink-1700 series which indeed matches the ground truth.

3.3.2 Inclination-Based Maneuver Detections

The purpose of this exercise was to find maneuver dates from TLE data of space objects.

We have used a statistical based method to detect these types of maneuvers. The input is the inclination values of the object of concern and the corresponding epochs. Following is our methodology used: We analyzed the data in the form of an "Inclination-vs-Epoch" graph. We observed that the maneuver dates (as per the ground truth data) were occurring near to significant peaks in the graph. However there were many peaks that had small abrupt changes to which were not the actual maneuver dates. Therefore, the data needed to be refined in order to remove such noises. We used **Savitzky–Golay filter** to smoothen out the graph, thus removing noisy elements. The parameters are 101 for window size and 13 for polynomial order. These parameters have been decided based on optimal response obtained by tuning.

The general trend is to do an inclination based maneuver whenever the inclination has to be increased. This increase can be discovered when the average of inclination values of a set threshold number of next consecutive data points from the current data pointer is greater than the current data pointer's inclination value. The magnitude of threshold and the averages depend on how the data has been refined. Different results would be obtained depending upon the type of refinement. For example if one uses Savitzky-Golay filter or even if no filter has been used, different thresholds would have to be set for each type respectively. Along with this condition we have added that, inclination values of say 4-5 next consecutive data pointers are in ascending order. By doing this, one would obtain many consecutive dates around the actual maneuver dates. To find out unique dates out of these, we took the median of these consecutive dates as the predicted maneuver date.

Results: To test our performance, we compared the ground truth data with the TLE data of ENVISAT and other few satellites with ground truth available. Labels were created for those dates in TLE data that were nearest to the predicted maneuver dates(a threshold can be defined for this). The performance score was then calculated by a confusion matrix. **The performance score(F1 score) was 0.88 for ENVISAT**.

3.3.3 Arrangements in Orbits

RSOs working together, when sharing the same orbit (SMA, i, RAAN) may or may not be actively controlling their relative positions. Based on this all patterns observed will fall under one of the following types:

- No control at all (free dispersion, evolution)
- Passive management
- Active propulsive management at individual RSO level.

For both the passively actively managed intra-plane intra-altitude RSO groups, an analyst needs to look for all types of persistent patterns, including (but limited) to the following:

- Equal angular spacing, covering the full orbit.
- Equal spacing, without covering the full orbit.
- Equal spacing, with 1 or more RSOs either misplaces or missing from their designated slots.

3.4 Debris Prediction

SSA data is crucial to the safe orbiting of satellites through an ever-growing field of orbital debris. However, lack of independent measurement or observational tools for space debris makes rely on a handful of centralized repositories operated by major space powers.

In this module, we aim to build a classification model for debris and satellite prediction. The TLE data has suffix-"DEB" added to name of objects that are confirmed pieces of debris. Therefore, we create a pre-labeled TLE data that has two classes-"Satellites" and "Debris". We then develop and evaluate a supervised machine-learning based classification tool which could then possibly predict an object's nature based on the data fields from TLE data.

We have taken TLE data of year-2019 and then classified each TLE into "Sat" and "Deb" labels as per their suffix in names. Having done this, we obtain a data-set which consists of orbital elements(TLE data fields) as features and target as "Sat/Deb" labels. To visualize the distribution of these features , we plot probability distributions of these features[Fig-7].

From the plots, we can observe that there are some non-overlapping regions for some orbital elements. We therefore probably should able to build a classification model that can identify these regions and make appropriate boundaries.

For our problem, we have used **Random-Forest Tree Classifier** since, it has shown very good results in classifying non-linear boundaries and performs superior to vanilla Decision tree Classifier. The results of the classifier are as shown below[Fig-7,8,9]:-

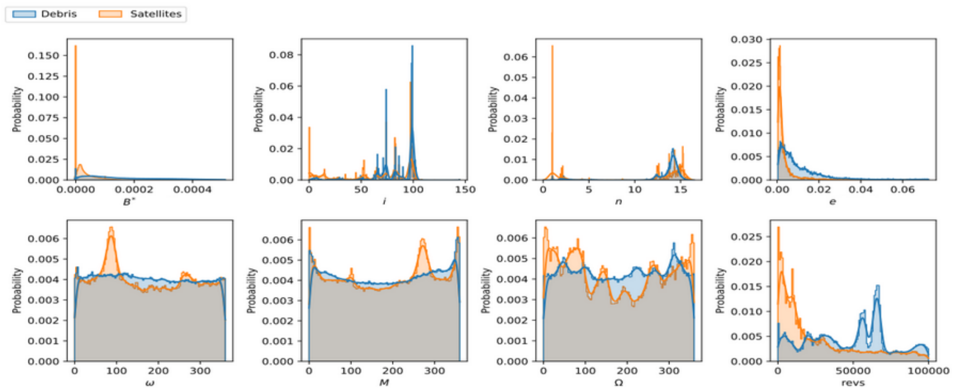

Figure 7: Distribution of orbital elements by object type

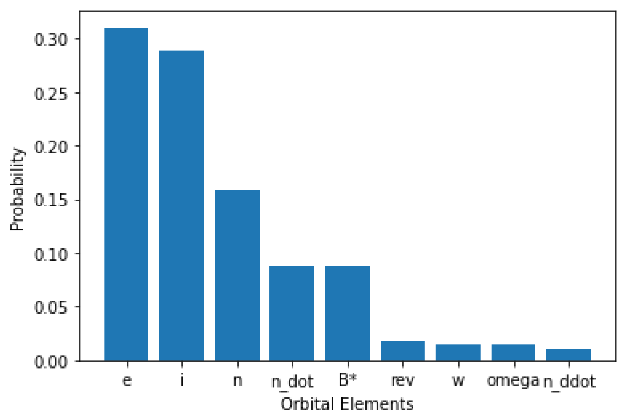

Figure 8: Random Forest Feature Importances

```
              precision    recall  f1-score   support

         Sat       0.96      0.88      0.92     45337
         Deb       0.84      0.95      0.89     30116

    accuracy                           0.91     75453
   macro avg       0.90      0.91      0.90     75453
weighted avg       0.91      0.91      0.91     75453

Accuracy =  0.9062197659470134
```

Figure 9: Results of Random Forest Classifier

3.5 Maneuver Detection

This module serves the purpose to detect the orbital maneuvers of satellites at different scales. In this module, the unsupervised classification method of **K-means** clustering is used to handle the TLE data. The K-means-based contour map method is applied to the characteristic variable selection and cluster number determination. By data mining, the orbital maneuvers of the remote sensing Chinese satellite - 'YAOGAN-9' is detected.

First, data pre-processing steps are performed on the historical data of 'YAOGAN-9'. These include **0-1 Standardization**, **Wavelet Filtering**. For detecting maneuvers, we observe the changes in orbital elements, as most maneuvers are done to either change a satellite's orbital path or restore to its original(if deviated). Therefore, we take differences between adjacent/consecutive values of orbital elements.

Kmeans clustering requires optimal number of clusters and optimal choice of characteristic variables for deciding the cluster similarity. To find both, we measured average contour values for number of clusters(2-15) as well as consecutive change in features from TLE data. The resulting values are as shown in Fig-10,11.

Number of clusters	2	3	4	5	6
Average contour value	0.9804	0.9757	0.9172	0.9569	0.9378
Number of clusters	7	8	9	10	11
Average contour value	0.8568	0.8044	0.8588	0.8706	0.8717
Number of clusters	12	13	14	15	
Average contour value	0.8719	0.8135	0.8607	0.8081	

Figure 10: Average contour value of different numbers of clusters.

From the values in Fig-10,11-12, one can conclude that optimal number of cluster should be 3 and most prominent characteristic variables should be $[\delta a, \delta, \delta r_z, \delta v_x]$. Therefore KMeans result from above chosen parameters in shown in Fig-13,14.

Characteristic variable	δa	δi	$\delta \Omega$	$\delta \omega$	δM
Average contour value	0.9981	0.9483	0.8173	0.9996	0.9852
Characteristic variable	δe	δr_x	δr_y	δr_z	$\delta \lvert \boldsymbol{r} \rvert$
Average contour value	0.9700	0.9982	0.9897	0.9984	0.9803
Characteristic variable	δv_x	δv_y	δv_z	$\delta \lvert \boldsymbol{v} \rvert$	
Average contour value	0.9932	0.9973	0.9918	0.9924	

Figure 11: Average contour value of different characteristic variables when the number of clusters is 3

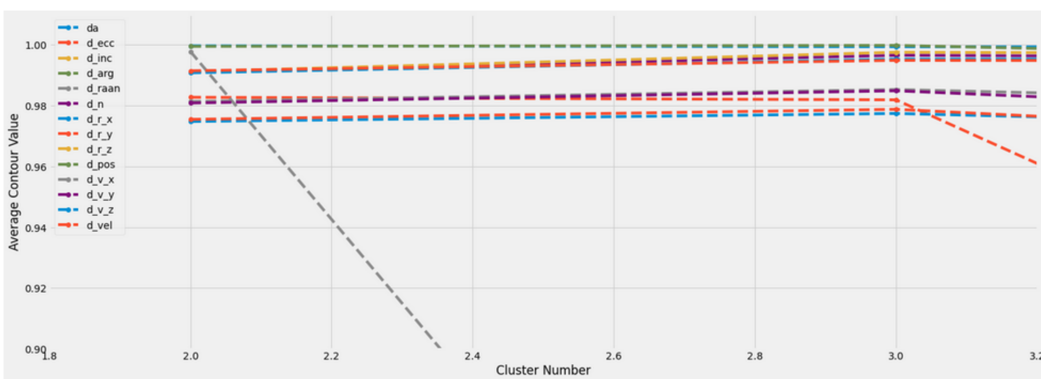

Figure 12: Plot of average contour value corresponding to each characteristic variable when the number of clusters is 2 and 3.

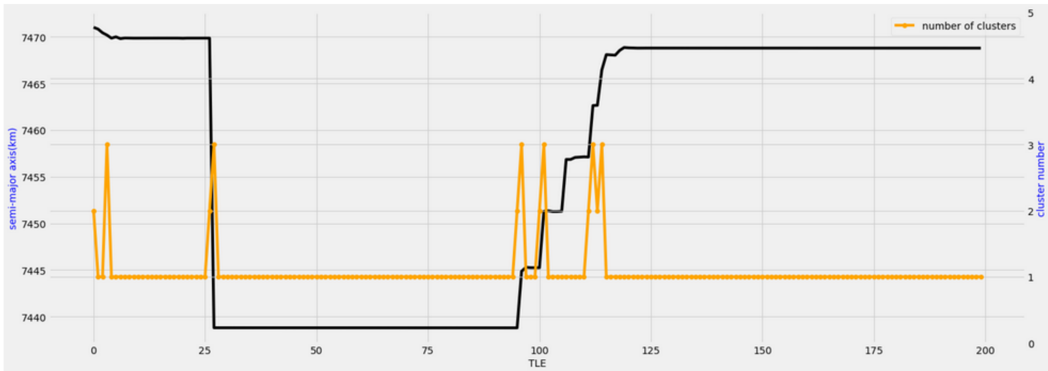

Figure 13: Semi-major axis variation curve and the K-means clustering result: the black line indicates the semi-major axis variation with time and the red line indicates the clustering result.

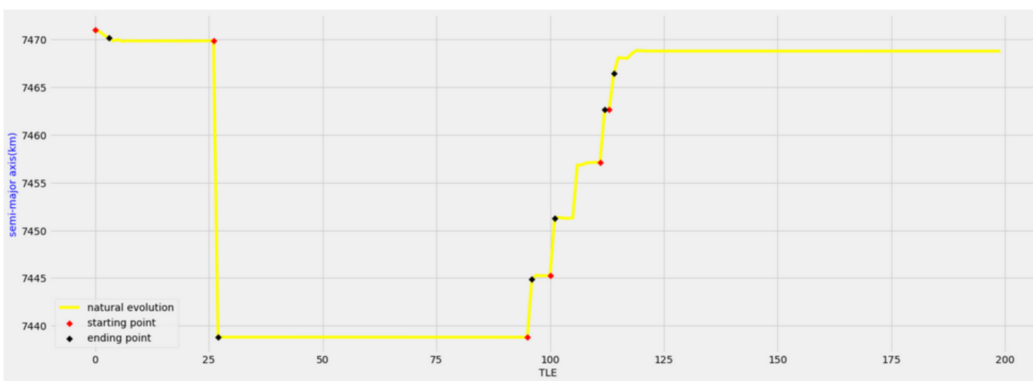

Figure 14: Semi-major axis curve marked with orbit control points by the K-means clustering method: the yellow line indicates the natural evolution process; the red asterisk indicates the starting point of the orbital maneuver; and the black asterisk indicates the end point of the orbital maneuver.

Note:-Thus we finally conclude the details of all the modules our Data-Mining from TLEs. However there some additional features that were created along with these main modules. These features did not give meaningful results or were significant enough. We would briefly discuss about them here:-

- **Principal Component Analysis(PCA)**: This dimensionality reduction technique was tried to reduce the number of features in TLE data(originally 9) but the eigen values for all the eigen vectors were significant enough, so dimensionality reduction was not good enough.

- **Data Filtering:** In clustering module, the time computation for Kmeans and DBSCAN over large amount of TLE data was very high. Therefore there was need to eliminate the unwanted data. As per the domain knowledge, objects with suffix-"Fuel Body","R/B" etc. are defunct satellites or parts of satellites/launch vehicles that have been left in outer space. Thus, we manually removed such objects from the TLE DATA.

- **Fuzzy-C-Means Clustering:** In Maneuver Detection module, we did the same analysis with Fuzzy C means algorithm instead of Kmeans. However since the results with Kmeans were better, we only considered Kmeans.

4 Conclusion

In this project we saw the possibilities of data-mining in the domain of TLE data. We saw that despite not having information about the nature(Sat/Deb) of object , we can detect it by supervised learning. Clustering algorithms can find geometrical patterns in orbits of satellites and this can help in identifying the purpose of these satellites(as most satellites have functions specific to their orbits and altitudes) along with identification of satellites that are similar to each other(in terms of launch company, location etc.). Maneuvers of active satellites can be detected by both statistical and clustering methods which can help in collision avoidance and prediction of future path of the concerned object.

Overall there are many interesting possibilities that have been explored and remain to be explored. Future work of this project would focus on improving existing results and discovering more tasks for which TLEs can be useful.

References

[1] Tom Kelecy, Boeing LTS, Colorado Springs, CO Kihei, HI , Satellite Maneuver Detection Using Two-line Element (TLE) Data.

[2] Xue Bai, Chuan Liao, Mining Two-Line Element Data to Detect Orbital Maneuver for Satellite, IEEE.

[3] Creon Levit, Improved orbit predictions using two-line elements, NASA Ames Research Center, Moffett Field, MS202-3, CA 94035, USA.

[4] David A. Vallado, Benjamin Bastida Virgili, IMPROVED SSA THROUGH ORBIT DETERMINATION OF TWO-LINE ELEMENT SETS, Centre for Space Standards and Innovations.