Neural Network Enhanced Numerical Propagation (NEP) to Improve SSA/SDA

Duane DeSieno, CTO
Data Fusion & Neural Networks, LLC
Carlsbad, CA

ABSTRACT

The goal of this research was to demonstrate a novel approach that uses a neural network, trained on recent history, to substantially improve the performance of the base propagator (either numerical or semi-analytical). The neural enhanced propagation (NEP) described in this paper can be run on a fast platform and used by customers such as US Space Force's Joint Commercial Operations (JCO) cell. Automated neural network training and retraining can take place as often as necessary to ensure the neural network enhanced propagation is as accurate as possible. This capability should contribute to more accurate detection of maneuvers, proximity analyses, and computation of collision probabilities.

In astrodynamics, propagation refers to the process of predicting the future position and velocity (state) of an object in space, such as a satellite, spacecraft, or celestial body, based on its current state and the physical forces acting upon it. The future state can be estimated using two basic methodologies, numerical propagation and semi-analytical propagation.

A numerical propagator integrates the forces acting on the spacecraft over time. The accuracy of such a propagator depends heavily on the modeled physical forces it considers. The primary forces modeled in a numerical propagator are gravitational forces. The primary force model is the central body gravity of the Earth. This force is modeled using spherical harmonics including high order terms to account for irregularities in earth gravitational field. Third body gravity force models have also been produced for the influences of other celestial bodies, such as the Moon, the Sun, and planets, which perturb the satellite's orbit.

There are other factors that influence the motion of the spacecraft, including the atmospheric drag, solar radiation pressure (space weather), Earth's albedo and Earth's infrared radiation, the Earth's magnetic field, tidal forces and relativistic corrections. There may be additional factors not covered in this list. While estimates can be made for some of these factors, many change with time. Modeling these forces for a numerical propagator would require a model for all points in space which can be extremely difficult to produce. Also, modeling these forces can be dependent on the characteristics of the spacecraft that may not be known such as its shape, orientation, mass, and magnetic properties. As one attempts to improve the performance of numerical propagation, the computational load continues to increase.

The second methodology is semi-analytical propagation. These use a combination of analytical techniques and some numerical methods to provide a balance between accuracy and computational efficiency. Simplified General Perturbations 4 (SGP4) falls into this category because it uses analytical expressions derived from perturbation theory, with simplifications that make it computationally efficient. SGP4 is widely used for orbit determination of Earth-orbiting objects, particularly with Two-Line Element sets (TLEs). It provides a practical and computationally efficient way to predict satellite positions with reasonable accuracy.

The goal of this research was to demonstrate a novel approach that uses a neural network, trained on recent history, to substantially improve the performance of the base propagator (either numerical or semi-analytical). Regardless of the force models used in the base propagator, recent history for a specific satellite can be used to produce a "local" model of the forces on that specific satellite that were not available to the base propagator. Since the recent history of observations available for a specific satellite is limited, the machine learning methodology applied requires making the neural network as small as possible to achieve the desired result. This learning is specific to the object from which the observations are being made. The goal of the learning is to correct the output of the base propagator in order to improve the accuracy of the propagated position and velocity.

The Neural Enhanced Propagation (NEP) described in this paper can be run on a fast platform, such as DF&NN's Alert Management System (AMS), used by customers such as the Air Force Research Laboratory's DRAGON Army, automated neural network training and retraining can take place as often as necessary to ensure the neural network enhanced propagation is as accurate as possible. A custom spacecraft-specific propagator approach simplifies what the neural network must learn. DF&NN has implemented this approach on the whole catalog of satellites it monitors in AMS.

The outcome expected from using the neural enhanced propagation is to lower the uncertainty of where the spacecraft is at a future point in time. Because the neural enhanced propagation is applied as a learned correction to the base propagator, its performance should always be better than the base propagator it is correcting. The importance of this capability for the SSA and SDA communities should be more accurate detection of maneuvers, proximity analyses, and computation of collision probabilities.

1. Background

Many papers have been written about using neural networks / machine learning for orbital prediction. The goals of various papers vary from learning a constellation of satellites using differentiable programming [1] to using LSTM and Deep Learning for long term prediction [2]. For the most part, papers are discussing large nets that require substantial numbers of training examples to achieve the targeted performance. In the differentiable dSGP4 paper, the net had 3,454 learnable parameters and the training set contained 6,563,599 examples from 1,519 Starlink satellites. For the number of learnable parameters, using a constellation gives a high quantity of real observations to support the training.

The deep learning paper used 100 days of data from a single satellite sampled at a 1-minute interval. This suggests that the data may have been generated using a propagator and not from real observations.

A great deal of experience has been gained from the development of the currently used autoencoder neural networks in DF&NN's AMS system. This experience shows that using a trained neural network specific to a single space object is a viable approach that can be run for a large number of space objects and a variety of data sources on a single server.

The goal of this paper is to suggest a methodology that can be implemented on a large catalog of real observations. It assumes that each object will be trained independently from available real observations. Since this limits the amount of data available for training, the net used must contain a small number of learnable parameters.

2. Methodology

There are several key points to neural enhanced propagation. The design and training are tailored to the way the trained nets will be used in a real-world system. In the case considered for this paper, the goal is to improve the accuracy of the predicted parameters at the time of the next TLE. This is typically how these nets would be used for a maneuver detection use case. For the purposes of this paper the term "residual" refers to the difference between either the NEP output or the SGP4 Propagated output, and the value based of the current observation. In neural network terms, this is the difference between the net output and the desired output, often called error or loss. Here, residual will be plotted in the CDF plots as an absolute value and in other plots as a signed value. Distance residuals are always absolute value. A large residual between the target parameter and the propagated parameter would be the indicator of a maneuver. Improving the prediction at the next TLE makes the residuals lower for non-maneuvers and therefore improves the performance of the maneuver detector.

Below are the top-level steps in the training and use of the NEP nets. Remember that a separate neural network is created for each space object and target output. Fig. 1 below shows the flow of information from a prior and current TLE to a training/test propagated record. A key point is that the NEP is not trying to learn the complex orbital mechanics of each space object. In this case the SGP4 propagator provides the initial estimate of the orbital parameters at the current observation based on the prior observation. With that information available, the NEP uses the information from the propagator and additional derived variables to improve the estimate from the propagator. If

derived variables are found that correlate with the residuals from the SGP4 then NEP will be able to reduce the magnitude of the residuals.

1. Collect TLE data for the Variable Selection process. This can be up to a year of data. Build the training set using the SGP4 propagator to produce the derived variables to be used (sections 2.3-2.5)
2. Ignore training points that are likely maneuvers or data errors (high SGP4 residual values).
3. Perform variable selection to find the best combination of derived variables to use in the training on a reasonable size set of data (section 2.6).
4. Using the selected variables from step 3, run the NEP processing on a small recent sample for the current estimate (section 2.7).
5. Monitor the performance of each object's neural network and retrain when needed.

A key point is a change of mindset from traditional orbit propagation. Think from the perspective of a neural network, how it works, and what it is good at. In the end, it is the inputs to training a network that determine its performance. The rest of this section will focus on why a small number of inputs used for training that have been derived from the available variables is the optimum choice. Note that the derived variables may vary for different space objects. Finally, for production, the process of variable selection must be automated.

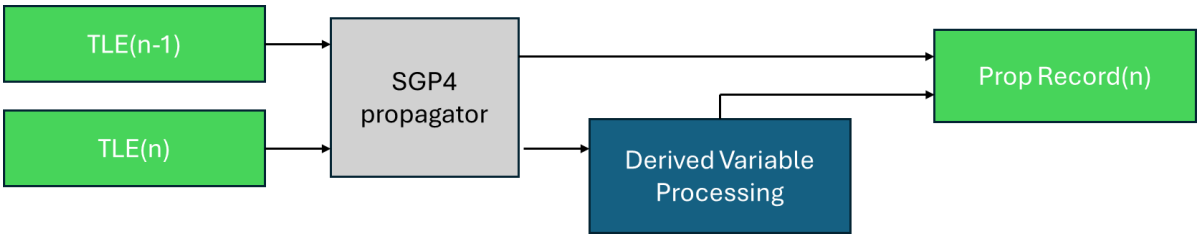


Fig. 1. Block diagram of information flow for propagated training or testing records.

2.1 Training Size Required for Generalization

In the neural network world, the basis for a processing element is a weighted sum of the inputs plus a bias factor. The output of a processing element is some function of the weighted sum. Neural networks typically have multiple layers of processing elements. Their claim to fame is that they represent a universal approximator of any mapping from inputs to desired outputs.

In this discussion we will start by looking at the Taylor series and the Fourier series in terms of their ability to approximate a function of a single variable. A cubic equation is an example of a simple Taylor series represented by equation 1:

$$Y = a + b \cdot X + c \cdot X^2 + d \cdot X^3 \qquad (1).$$

This equation could be rewritten as the equation of a single processing element in a neural network with three inputs. Those inputs would be X, the original variable, and two additional (derived inputs) X^2 and X^3. The parameters a, b, c, and d are the weights of that processing element that are learnable from the data.

The Taylor series is a good example for illustrating overfitting. There is an exact solution to the parameters for our cubic equation if we have 4 data samples (X, Y pairs). As the order n of the Taylor series increases, it takes (n + 1) examples to get an exact (0.0 residual) fit. If the data had no noise, this exact fit would be the perfect solution. However, with any noise, this becomes an easily understood example of overfitting. The more noise that is present, the more examples are needed to produce accurate values for the fit parameters. This is as true for neural networks as it is for the Taylor series expansion of a function.

The better series to consider, since we are looking at orbital mechanics, is the Fourier series. The sin and cos function can be used as a perfect fit for the X and Y position of an object in a perfect circular orbit around a planet. Assumptions include no drag, even gravity, no other accelerations from other sources. The equation for the Fourier series in one variable (t) is represented by equation 2:

$$Y = a + b\cdot\sin(t) + c\cdot\cos(t) + d\cdot\sin(2\cdot t) + e\cdot\cos(2\cdot t) + \ldots \quad (2).$$

This equation can be implemented as a neural network provided that the inputs to the net become the derived functions sin(n·t) and cos(n·t) [3]. Again, a perfect fit can be achieved if the number of examples in the data set equal the number of free parameters in the function. For the neural network representation, that is the number of weights in the network. Noise, as seen in the Taylor series example, has the same effect on the number of examples to achieve a good fit.

Tests on real data for space objects in GEO orbits (low eccentricity) show very good fits for a Fourier estimation of X and Y position data using just the first and second harmonics of the orbital period. This sounds like it has promise to represent the orbit of a satellite to a high degree of accuracy. However, the number of terms needed depends on the cycles of the forces acting on the satellite. There are different periods for the effects of the Sun, Moon, tidal effects, and nonuniform gravity. The number of samples needed to get good generalization exceeds the number of available observations.

For either of the above examples, it is the number of weights (learnable parameters) in the neural network that determine the lower limit on the number of examples needed to train a network that generalizes well. Added noise in the observations increases the number of samples needed.

An important aspect of the Taylor series and the Fourier series above is that the output "Y" in both cases did not use a transfer function that is normally seen in a neural network. Transfer functions that are often used are Sigmoid, RELU, Tanh, Gaussian, etc. Because the derived inputs, X^i for the Taylor series and sin(i·t) cos(i·t) for the Fourier series are already non-linear in nature, a linear combination of these terms can approximate any function Y=f(x).

This leads to the question of what a transfer function really does in a neural network? The answer is that transfer functions provide the means to smoothly transition between the features in a layer of processing elements. Take the example of the Sigmoid function. It has a clear "active" region near 0.0 and asymptotically goes to 0.0 or 1.0 as the input to the Sigmoid moves away from 0.0. In the active region the magnitude of the derivative of the Sigmoid is significant but quickly goes to 0.0 as you move away from 0.0 input. The KAN net [4] actually trains a custom spline transfer function for each processing element in the net. The KAN net significantly reduces the number of processing elements needed to learn a mapping but does so at the expense of many free parameters (weights) needed to define the learned transfer function.

When we look at the derived variables, we are using for the NEP net (2.3 below), they are all already cyclic in nature. Thus, these derived variables act in the net as predefined features that do not contain learnable parameters. Yet many derived variables still lead to a large number of learnable parameters for the trained net. To reduce the size of the net, variable selection (2.6 below) is used to minimize the number of derived features used in the trained net.

2.2 Variables Available

The analysis for this paper was focused on the use of the Astrodynamics Standards Libraries that provide a validated method to utilize the TLEs to obtain a satellite position and velocity at a particular point in time. The routines used take TLE data available from SpaceTrack and perform the SGP4 propagations. Thes records will be referred to as "Prop" records and are used in the variable selection training and in the NEP TLE processing.

At a specified point in time, there are four groups of information that are available either direct from the library or through standard transformation (Keplerian to Equinoctial). The groups are:

1. Osculating Elements, "o"
   a. X(KM), Y(KM), Z(KM) position variables
   b. XDOT(KM/S), YDOT(KM/S), ZDOT(KM/S) velocity variables
   c. VelNorm(KM/S) derived from velocities (VelNorm = sqrt($XDOT^2 + YDOT^2 + ZDOT^2$))
   d. LAT(DEG), LON(DEG), HT(KM) position variable

    e. SMAo(KM), ECCo(-), INCo(DEG), RAANo(DEG), AOPo(DEG), TRUEANOMo(DEG) Keplerian variables
2. Mean Elements, "m"
    a. N(REVS/DAY), ECCm(-), INCm(DEG), RAANm(DEG), AOPm(DEG), MAm(DEG) Keplerian variables
3. Equinoctial Elements (derived from Osculating Elements)
    a. h, k, p, q, L
4. Nodal Variables
    a. NODAL PER(MIN), NODAL(REVS/DAY), ANOM PER(MIN), APOGEE(KM), PERIGEE(KM)
5. Sin/Cos transforms from degrees to avoid discontinuity at 360 degrees
    a. RAANo(sin), RAANo(cos), AOPo(sin), AOPo(cos), TRUEANOMo(sin), TRUEANOMo(cos) from Osculating Elements
    b. RAANm(sin), RAANm(cos), AOPm(sin), AOPm(cos), MAm(sin), MAm(cos) from Mean Elements
6. Time Related variables
    a. TIMEA time of the current TLE record
    b. TIMEP time of the prior TLE record
    c. TSINCE time from the prior TLE to the current TLE record

The above variables were determined for three cases.

1. Variables associated with the prior record in a sequence of TLE records.
2. Variables associated with the prior record propagated to the time of the current record "P".
3. Variables associated with the current record in the sequence of TLE records "A".

2.3 Input Variables

From the above available variables, only the variable associated with the propagated prior record were allowed as inputs to the neural networks. Also allowed as inputs to the NEP were the TIMEA and the TSINCE variables. When referred to in the paper these variables will be prepended with a "P".

2.4 Desired Output Variables

Network output variables were selected only from the variables associated with the current TLE record. For this paper the following nets were trained for these desired outputs (prepended with "A").

1. AX(KM),
2. AY(KM),
3. AZ(KM),
4. AINCo(DEG),
5. ASMAo(KM),
6. ALON(DEG),
7. AVelNorm(MK/S)

2.5 Variable Selection Methodology for Neural Network Training

The variable selection process assumes it is working on data that does not contain maneuvers. The residual encountered, because there was a maneuver between the prior record and the current record, is likely to be substantially higher than if no maneuver occurred. With mean squared error loss function, the residuals related to maneuvers should be eliminated from the training data. The simplest way to accomplish this is to look at the propagated residual values for the desired output and eliminate records where the absolute value of the residual is above a percentile threshold. The threshold used as in training for this paper was 98%. Records with SPG4 propagated residual value above this "maneuver threshold" will be considered maneuvers and will not be used in training or in the results statistics.

The most common way to do variable selection is to use correlation to pick the variables. In the case of doing a single variable linear regression, the highest correlating variable would be the reasonable choice. However, if

picking two variables, using the highest two correlating variables will likely not yield the best selection of two variables. This is due to the fact that the cross correlations between variables will affect the outcome. To find the best two variables to be used in a regression (linear or neural net), one would have to run all pairs of variables.

The method used here uses the loss from a linear regression over the variable selection training data. A sorted list of single variables is generated for variables taken one at a time. Then the top n items on this list are evaluated further. The top n selections are then evaluated by each of the remaining variables, one at a time. These results are sorted, and the process is repeated for the top n selections.

This process stops based on one of the following criteria:

1. The percentage of improvement in the loss is below a threshold for the next added variable.
2. The performance of the current selection has fallen below a threshold and there are at least m variables.
3. The process has reached the max variables limit.

The variables selection process was run for the seven desired output variable for all 20 satellites. The variables selected for each NEP net are listed in Appendix 1.

2.6 NEP TLE Processing

Once the variable selection has been performed for a given object and the selected output variable, training on data recent to the current and prior TLEs must be performed. This involves selecting recent TLEs prior to the current record for the training. The size of the training set chosen should be at least twice the numbers of variables selected in the variable selection process. The selections of variable used in the NEP nets trained ranged from 3 to 7 (Appendix 1). For this analysis the size used was 16 samples. If any propagated residual was larger than the threshold used in variable selection training, then that training record should be excluded. The NEP net is then trained on this small training set to determine the current estimated learnable parameters. Fig. 2 shows the processing flow. Prop Records are generated from the TLEs using the SGP4 propagation. Current estimates of outputs are generated by the NEP net. NEP net learnable parameters are updated based on the Prop records prior to the current Prop Record.

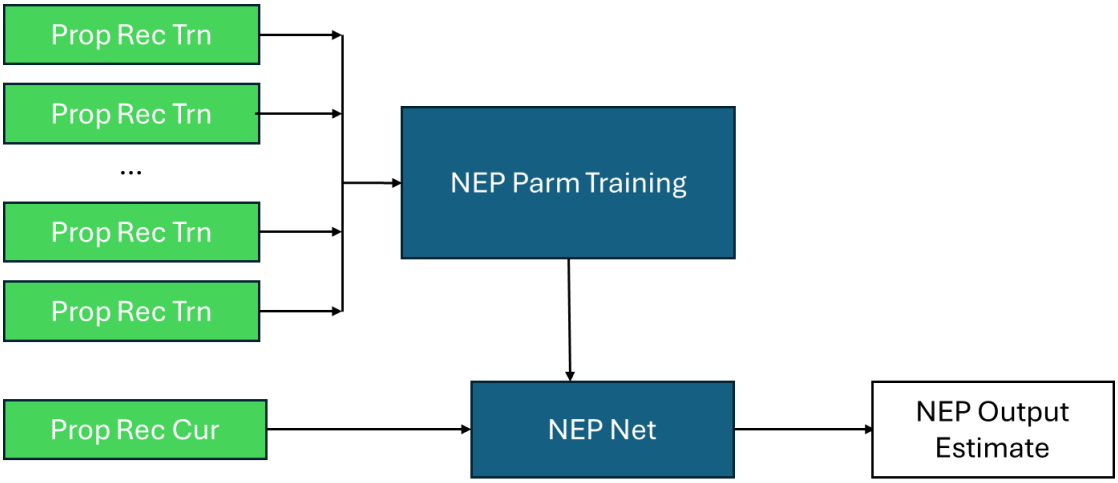


Fig 2. Training process for the NEP estimation in forward mode on test data.

The whole process for generating a current NEP output estimate for a parameter is approximately 1.9ms on the i9 laptop in Python in a single thread.

Once trained the NEP net is run in forward mode to generate the NEP Residual (the current observation value minus the NEP net output. The NEP net output is the estimate of the value of the target variable at the time of the current record based on the SGP4 propagated information from the prior record.

2.7 Satellite/Data Selection

Data used for this analysis are TLEs, for the space objects in Table 1, downloaded from SpaceTrack.org. Data for this analysis was taken from 2022-01-01 until 2024-07-31. In the case of object 58573 there was no data available prior to 2023-12-14.

Table 1 shows the selection of space objects used for this paper. The goal of the selection process was to show a good mix of space objects from the standpoint of orbital type (LEO, MEO, GEO) and ability to maneuver (ROCKET BODY vs PAYLOAD). Several were selected because they had been the subject of QC analysis for the DF&NN AMS system.

Records	NORAD ID	StartDate	EndDate	SatName	Type	Country	Period	Decay
1683	14114	22001	24212	SL-12 R/B(2)	ROCKET BODY	CIS	1436.63	
582	23645	22005	24210	SL-6 R/B(2)	ROCKET BODY	CIS	722.77	
2349	29236	22001	24212	GALAXY 16	PAYLOAD	US	1436.12	
4807	35951	22001	24212	DMSP 5D-3 F18	PAYLOAD	US	101.83	
825	37140	22001	24209	SL-12 R/B(2)	ROCKET BODY	CIS	674.62	
2380	37834	22001	24212	INTELSAT 18	PAYLOAD	ITSO	1436.09	
2062	41021	22001	24212	CHINASAT 2C	PAYLOAD	PRC	1436.07	
2287	41911	22001	24212	TJS-2	PAYLOAD	PRC	1436.13	
1636	43602	22001	24212	BEIDOU 3M11	PAYLOAD	PRC	773.23	
1916	43867	22001	24212	COSMOS 2533	PAYLOAD	CIS	1436.06	
3325	45250	22001	24156	XJS D	PAYLOAD	PRC	87.30	6/4/2024
1434	45254	22001	24212	MERIDIAN 9	PAYLOAD	CIS	717.86	
2275	45344	22001	24212	BEIDOU 3 G2	PAYLOAD	PRC	1436.09	
3689	45460	22001	24212	YAOGAN-30 R	PAYLOAD	PRC	96.28	
3581	45611	22001	24212	XJS G	PAYLOAD	PRC	93.69	
1923	45863	22001	24212	APSTAR 6D	PAYLOAD	PRC	1436.09	
2438	48444	22001	24212	STARLINK-2645	PAYLOAD	US	95.59	
2437	49330	22001	24212	SJ-21	PAYLOAD	PRC	1435.70	
2203	49818	22001	24212	LDPE-1	PAYLOAD	US	1441.57	
660	58573	23348	24213	PRC TEST	PAYLOAD	PRC	91.36	

Table 1. Space objects selected for this analysis. Dates are formatted “YYdoy”

2.8 Training

Training for each space object shown in Table 1 was performed on the data from 2022-01-01 through 2023-01-01 except object 58573 where all available was used for training. The training process used the variables selection methodology to a subset of the available variables for each of the NEP nets to be produced for the desired output variables in section 2.5. With 20 selected space objects and 7 desired output variables, a total of 140 variables selection runs were performed.

Appendix 1 contains the table of the space object, the desired output variable for the NEP output, the SGP4 propagated variable used for the performance evaluation and the selection of inputs variables found in the variable selection process. It was expected that the SGP4 propagated variable should have the highest correlation with the desired output and therefore would always be one of the selected variables in the variable selection process. However, in 17 of the 140 runs, the SGP4 propagated variables were not used as one of the NEP net inputs. The number of variables selected ranged from the 3 to 7 based on the criteria used in the processing.

The average training time for the variable selection process was 11 seconds in a single thread in Python on an i9 laptop. Since this training process can be run infrequently, this represents a reasonable processing load for a server running a large catalog of space objects.

Appendix 2 contains the histograms of the time between observations (TSINCE) for all available data used in the training and testing process. This is of interest as it shows that the observation rates for space object can vary substantially. An interesting observation is that the NEP net variable selection process chose TSINCE as an input in 59 of the 140 nets trained (42%).

Histograms of TSINCE are of interest because they can be used to QC the detections. This can be done by comparing the TSINCE for the current detection to the histogram to determine a "trust" score. If there is an unusual gap in the observations, then a detection by the NEP net should not be trusted. This will be discussed further in section 3.2 Maneuver Results.

3. Testing of NEP and Comparison with Traditional Propagation

Testing was performed on all data after 2023-01-01. The one exception is 58573 where all available was used for testing. Results are provided for both position estimation and maneuver detection. Evaluating test data is performed as described in the NEP TLE Processing section 2.6 above. In this report for each test TLE, the prior TLEs SGP4 propagated values will be the inputs to the NEP net to generate the NEP estimate for the current test TLE.

The question of how to visualize the relative performance of the NEP net vs just the SGP4 Prop results is an issue. As explained above, in the variable selection training and in the NEP TLE Processing, it is necessary to segregate maneuvers (and data errors) from the training data. Therefore, in reporting results, on test data, we must also do the same partitioning of the data.

Fig. 3 shows a plot of the NEP residuals and the SGP4 Prop residuals. The plot is generating by taking the absolute value of the residual values (even above the training threshold) and sorting them independently. Then the points are plotted low to high in log scale. This type of plot is commonly referred to as a Cumulative Distribution Function (CDF) Plot. Note that at the low values of residual on the left side of the plot the difference between the two curves is about 1.5 orders of magnitude less than at the higher values. This doesn't mean that NEP test examples had a lower residual than SPG4 on every example, but just that NEP is more likely to produce a lower residual. The CDF plot is also useful in visualizing the effect of setting a threshold on the residual value for the purpose of maneuver detection. A horizontal line drawn on the plot at a selected residual value will always have fewer examples above the line for the NEP residuals then for the SGP4 Prop residuals. This suggests that the NEP residuals are less likely to contain false positives as the SGP4 Prop residuals.

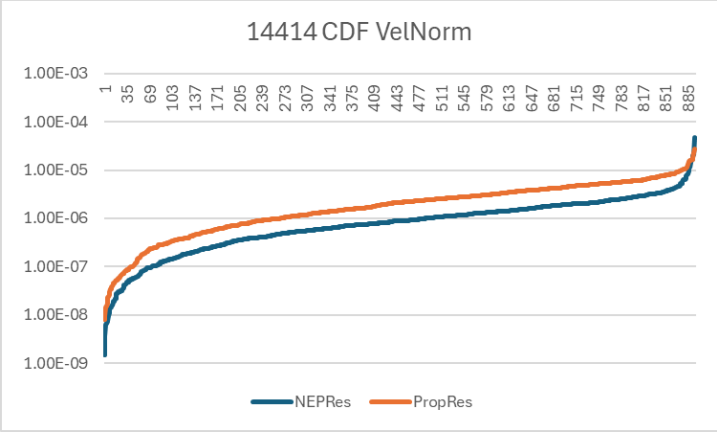


Fig 3. CDF plot given clear visual representation of performance improvement using NEP.

3.1 Position Results

Table 2 gives the summary of the results for the position analysis using Prop (SGP4) and NEP. Statistics are related to the estimates of X(KM), Y(KM) and Z(KM) for the SGP4 propagator (Prop) and the NEP. Position residuals are calculated as the vector difference between the Prop or NEP position and the current record position.

PosResAvg is the average improvement of NEP vs Prop in km. for records with Prop residuals below the maneuver threshold. **NEPAvg** is the average position residual between the NEP calculated position vector and the current record position vector. **PropAvg** is the average position residual between the SGP4 propagated position and the current record position. **%Imp** is the (PropAvg – NEPAvg)/PropAvg as a percentage.

These plots for position residuals for all trained space objects are included in Appendix 3. It should be noted that 5 out of the 20 CDF plots showed portions of the plots, specifically at the lower residual values where the SGP4 propagated values had lower residuals than the NEP net values. Since the plots are log scaled, the differences between the two curves at the lower values (left side of the CDF) are less significant than the difference at the higher values (right side of the curve. Because of this, the average improvement for the NEP was still positive.

NORAD ID	PosResAvg	NEPAvg	PropAvg	% Imp	TestCnt
14114	0.130211	0.118523	0.248733	52.349478	894
23645	1.384227	0.147334	1.531561	90.380172	210
29236	0.167136	2.386580	2.553715	6.544805	1206
35951	0.035599	0.005222	0.040821	87.207276	2266
37140	0.052723	0.090220	0.142943	36.883883	353
37834	0.478958	2.856357	3.335316	14.360204	1236
41021	0.280057	2.709035	2.989093	9.369302	1214
41911	0.175079	0.099012	0.274091	63.876221	1269
43602	0.083787	0.095512	0.179299	46.730361	644
43867	0.963513	2.603442	3.566955	27.012205	931
45250	0.525158	0.150394	0.675552	77.737570	1435
45254	0.775884	0.295071	1.070955	72.447863	593
45344	0.343727	0.220011	0.563738	60.972880	1237
45460	0.103089	0.077887	0.180976	56.962943	1889
45611	0.222047	0.116786	0.338834	65.532860	1652
45863	1.189307	2.227331	3.416638	34.809275	904
48444	4.652600	0.153432	4.806031	96.807519	1123
49330	0.076335	0.120159	0.196495	38.848640	1299
49818	0.092199	0.135329	0.227529	40.522048	1169
58573	0.240838	0.037678	0.278516	86.471922	578

Table 2. Summary test statistics comparing NEP to SGP4 propagated position.

3.2 Maneuver Results

Performance assessment for maneuvers is more difficult than for position estimation. For maneuvers one must have ground truth as to where the maneuvers occurred. Then based on residuals on the parameters that NEP nets were trained, thresholds must be set to determine when a residual should be considered a maneuver.

Looking at a rocket body, a space object that should not be capable of maneuvers, can illustrate certain aspects of using propagated residual estimated to determine a maneuver. Fig. 5 shows the inclination residuals for MEO space object 23645 which is classified as a rocket body. Plots for both the NEP and SGP4 propagated residuals are shown for comparison. Both show elevated residual values at the same points in time for osculating inclination. The time between observations (TSINCE) is larger where the magnitude of the residual is high. The size of the residual is likely to increase as the time one propagates to increases, but that does not mean the object maneuvered.

Under normal baseline conditions, the detection threshold for the NEP could be set around 0.0025 and the Prop threshold could be set at around 0.0065. Lower non-maneuver residuals indicate that, NEP did a better job estimating inclination for baseline conditions.

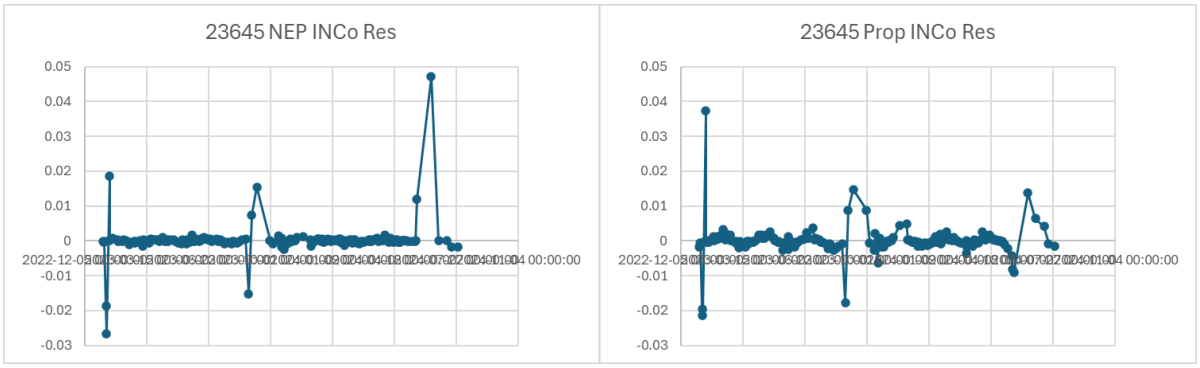


Fig. 5. Residual plots for osculating inclination "INCo" residuals.

Fig. 6 shows the VelNorm residuals for the same space object. It also shows large residual values at the right of the plot for both NEP and SGP4 Prop which are possibly due to large TSINCE values. However, the Fig. 7 plot of the observed values shows multiple observations during that period that appear abnormal. Both plots are scaled to show velocity residuals from -2.0 m/sec to 2.0 m/sec. A reasonable maneuver detection threshold for NEP would be at 1 m/sec magnitude. SGP4 Prop residuals would have had many false positive detections at that same threshold.

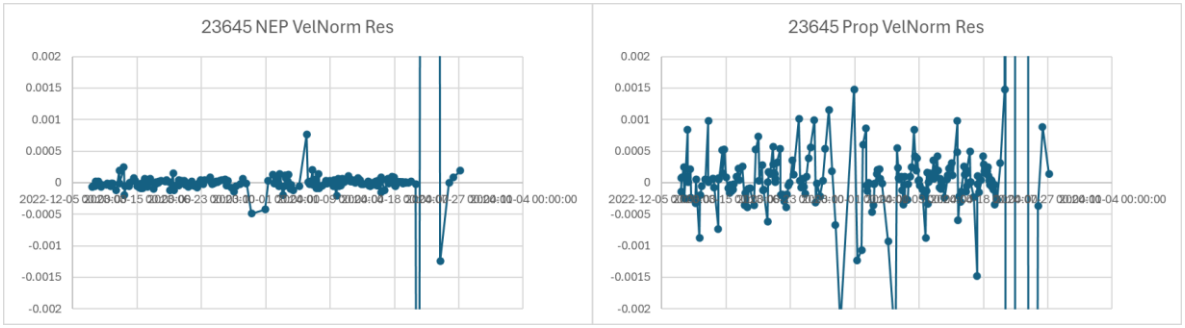


Fig. 6. 23645 VelNorm residual for both NEP and SGP4 Prop.

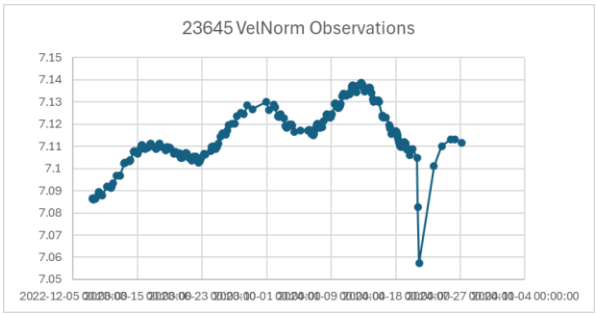


Fig. 7. 23645 VelNorm observed values.

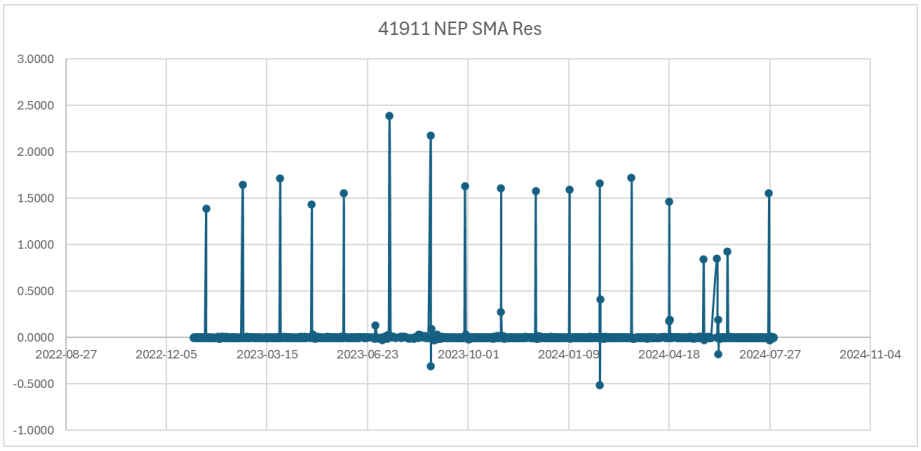


Fig. 8 NEP residuals for 41911 over the test data.

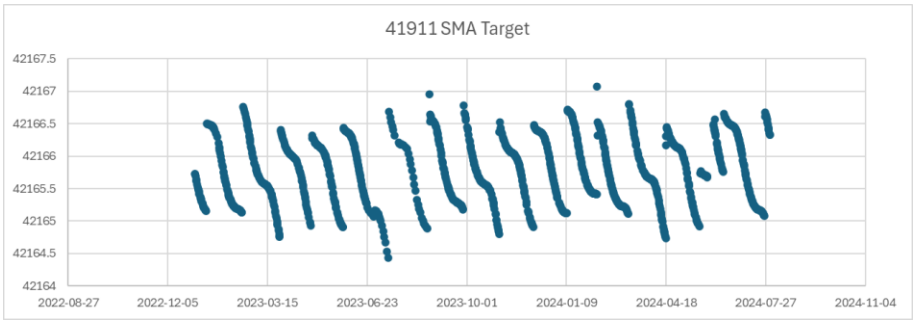


Fig. 9 SMA values form the observations for 41911 over the test data

The next example is that of a clear sequence of maneuvers over the test period. This is the 41911 GEO Satellite showing the semi-major axis (SMA) residual as calculated from the NEP net. This represents East-West Station Keeping pattern of life. Fig. 8 shows the residuals and Fig. 9 shows the corresponding SMA values from the observations over the test data. The steps in SMA in Fig. 9 clearly match the detected SMA maneuvers assuming a 0.5km magnitude threshold on the SMA residual.

3.3 Potential Data Error

An important consideration in NEP is identification of errors in the propagated data. What would a data error look like? The example in Fig.10 at the center of the plot is a likely inclination data error. A step followed immediately by a step in the opposite direction suggests a potential data error. The corresponding plot of osculating inclination (Fig. 11) shows no visible step at the time of the detection.

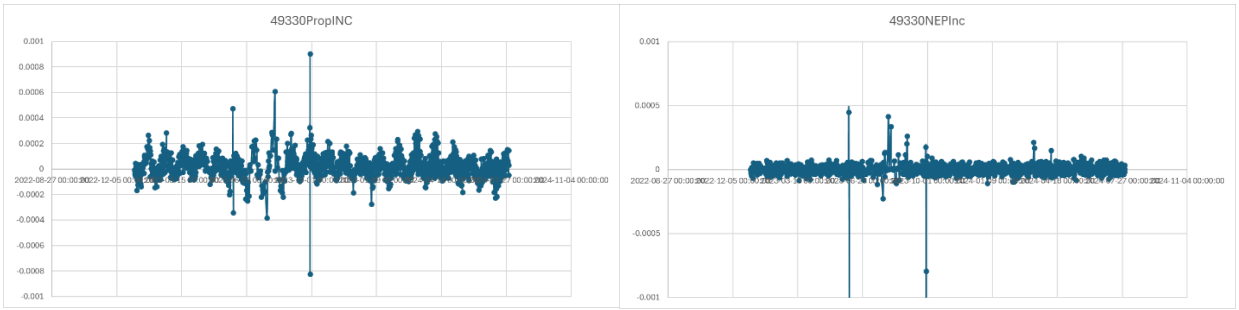


Fig. 10. Possible Data Error seen by symmetric steps in inclination.

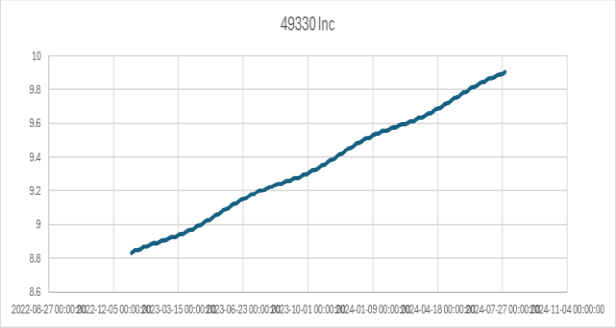


Fig. 11. Actual Inclination plot. No visible change.

It is important to note that Fig. 10 plots were scaled the same on the y axis to show that NEP nets produce less variation in the baseline residuals. This allows for lower detection thresholds and greater sensitivity and lower chance of false positives.

4. Potential Improvements

Training thresholds could be optimized on a per satellite basis to better eliminate maneuvers and data errors from effecting the training process. The current implementation is using a fixed percentile threshold on the SGP4 propagated residual value to eliminate records from the training data.

In this paper the number of non-maneuver samples for the NEP secondary training was set to 16 for all space objects. Because observation rates vary between space objects, the number of observations used could be all samples within a time window prior to the current TLE epoch. Optimization of the thresholds used will improve the overall performance of the methodology.

In this paper NEP nets were trained on seven target parameters. The number of parameters for NEP training could be increased to do all PV Coordinates and all Keplerian parameters. Data Fusion of the NEP outputs could be used to increase the confidence of the detected maneuvers and aid in the classification of the type of pattern of life maneuver that was detected.

This methodology is dependent on the quality of the derived features used in the training process. While the derived features used here were readily available from the outputs that could be generated from SGP4 and some simple features derived from the SGP4 output, other cyclic features that could be synced to the epoch of the observation could improve the NEP net performance. Fig. 11 shows the ASMAo(KM) variable over the test data for space object 23645. This variable shows an oscillation at a period of about 72.5 days at an amplitude of 2.55km. Adding a derived feature that correlates with this oscillation could improve the performance of the NEP net where longer propagation times are of interest.

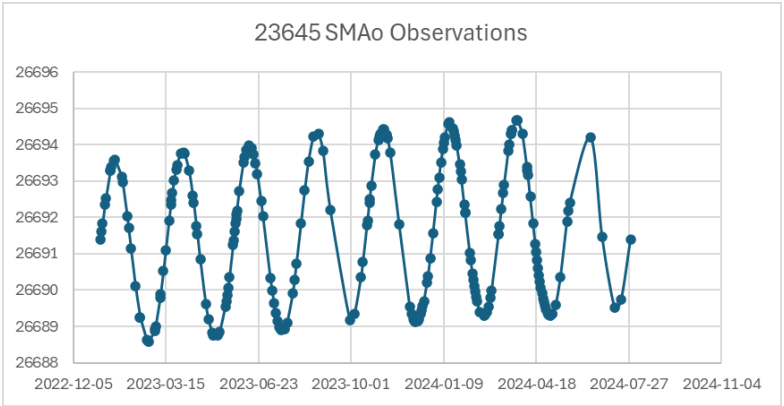


Fig 11. Plot of semi-major axis for 23645 potential for an additional derived feature.

5. Lessons Learned

Training in this paper consisted of propagating only to the next observation. The training data preprocessing can be modified to provide training data for propagation times greater than seen in the TSINCE histograms. For this or any methodology, the characteristics of the training set should match the way the nets will be run in testing or production.

While this paper focused on the use of SGP4 as the propagator to be enhanced, the methodology can be applied with SGP4 being replaced by any propagator.

6. Summary

This paper reviews recent research aimed toward evaluating the value of neural network enhanced orbit propagation (NEP). The results show that NEP is practical. NEP runs faster than numerical propagation and there is improved predicted positional accuracy and velocity versus actual observed space object position and velocity.

Based on indications summarized in this paper, DF&NN will continue investigating this capability on a broader set of objects and will likely result in real-time implementation of NEP across the unclassified catalog. Over time we will be able to quantify where NEP provides the most value over traditional propagators.

REFERENCES

[1] Giacomo Acciarini, Atılım Günes Baydin, and Dario Izzo. Closing the Gap between SGP4 and High-Precision Propagation via Differentiable Programming. *arXiv:2402.04830v3 [cs.LG] 27 Feb 2024*

[2] Hai-tao Yang, Jun-peng Zhu, and Jian Zhang. The Research of Low Earth Orbit Prediction of Satellite Based on Deep Neural Network. *2017 2nd International Conference on Computer, Mechatronics and Electronic Engineering (CMEE 2017) ISBN: 978-1-60595-532-2*

[3] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *arXiv:2006.10739v1 [cs.CV] 18 Jun 2020*

[4] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Solja, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–Arnold Networks. *arXiv:2404.19756v2 [cs.LG] 2 May 2024*

Appendix 1- Variable Selection Results

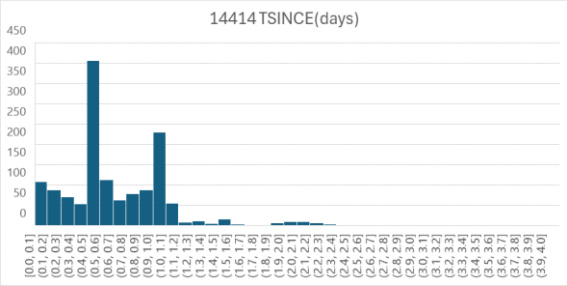
Below is the list of the most significant variables (fourth column) for each output variable selected for neural networks to learn the behavior of the seven output variables for the 20 space objects in this work. There are 17 out of 140 cases where the Prop output from SGP4 was not selected as an input to the NEP neural network (shown in red). One would might expect that the Prop output would always be used, but that was not always the case.

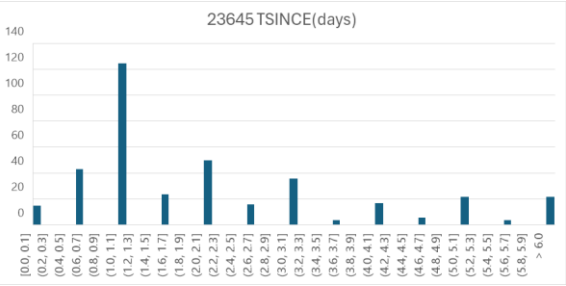
SAT ID	NEP Output	Prop Output	Variables Selection Used
14114	AX(KM)	PX(KM)	('PAOPm(sin)', 'PINCm(DEG)', 'PPERIGEE(KM)', 'PRAANm(cos)', 'PX(KM)', 'PY(KM)', 'Ph')
14114	AY(KM)	PY(KM)	('PHT(KM)', 'PMAm(cos)', 'PMAm(sin)', 'PRAANm(sin)', 'PY(KM)', 'Ph', 'Pp')
14114	AZ(KM)	PZ(KM)	('PANOM_PER(MIN)', 'PAOPo(sin)', 'PAPOGEE(KM)', 'PPERIGEE(KM)', 'PSMAo(KM)', 'PZ(KM)', 'TIMEA')
14114	ASMAo(KM)	PSMAo(KM)	('PAOPo(DEG)', 'PAOPo(cos)', 'PECCo(-)', 'PSMAo(KM)', 'TSINCE(MIN)')
14114	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PLON(DEG)', 'PRAANm(DEG)', 'TIMEA', 'TSINCE(MIN)')
14114	ALON(DEG)	PLON(DEG)	('PLON(DEG)', 'PMAm(sin)', 'PVelNorm(KM/S)', 'PX(KM)', 'TSINCE(MIN)')
14114	AVelNorm(KM	PVelNorm(KM	('PL', 'PTRUEANOMo(sin)', 'PVelNorm(KM/S)', 'PX(KM)', 'TSINCE(MIN)')
23645	AX(KM)	PX(KM)	('PAPOGEE(KM)', 'PHT(KM)', 'PINCm(DEG)', 'PLAT(DEG)', 'PX(KM)', 'PXDOT(KM/S)', 'PYDOT(KM/S)')
23645	AY(KM)	PY(KM)	('PANOM_PER(MIN)', 'PINCo(DEG)', 'PLAT(DEG)', 'PNODAL_PER(MIN)', 'PY(KM)', 'PZ(KM)', 'TSINCE(MIN)')
23645	AZ(KM)	PZ(KM)	('PAOPm(DEG)', 'PAOPo(cos)', 'PINCo(DEG)', 'PMAm(sin)', 'PNODAL(REVS/DAY)')
23645	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PAOPm(cos)', 'PLAT(DEG)', 'PSMAo(KM)', 'TIMEA')
23645	AINCo(DEG)	PINCo(DEG)	('PAOPm(DEG)', 'PECCo(-)', 'PINCo(DEG)', 'PRAANo(DEG)', 'PX(KM)')
23645	ALON(DEG)	PLON(DEG)	('PINCm(DEG)', 'PINCo(DEG)', 'PLAT(DEG)', 'PLON(DEG)', 'TSINCE(MIN)')
23645	AVelNorm(KM	PVelNorm(KM	('PAOPm(cos)', 'PHT(KM)', 'PLAT(DEG)', 'PRAANo(DEG)', 'TSINCE(MIN)')
29236	AX(KM)	PX(KM)	('PANOM_PER(MIN)', 'PINCm(DEG)', 'PN(REVS/DAY)', 'PSMAo(KM)', 'PX(KM)', 'Pk')
29236	AY(KM)	PY(KM)	('PANOM_PER(MIN)', 'PAOPm(sin)', 'PL', 'PSMAo(KM)', 'PX(KM)', 'PY(KM)')
29236	AZ(KM)	PZ(KM)	('PLAT(DEG)', 'PTRUEANOMo(cos)', 'PX(KM)')
29236	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PECCo(-)', 'PLON(DEG)', 'PNODAL_PER(MIN)', 'PTRUEANOMo(sin)')
29236	AINCo(DEG)	PINCo(DEG)	('PAOPo(sin)', 'PECCo(-)', 'PINCo(DEG)', 'PRAANm(sin)', 'TIMEA')
29236	ALON(DEG)	PLON(DEG)	('PLON(DEG)', 'PMAm(sin)', 'PN(REVS/DAY)', 'PNODAL_PER(MIN)', 'PRAANm(cos)')
29236	AVelNorm(KM	PVelNorm(KM	('PANOM_PER(MIN)', 'PHT(KM)', 'PLON(DEG)', 'PSMAo(KM)', 'PTRUEANOMo(sin)')
35951	AX(KM)	PX(KM)	('PRAANm(cos)', 'PRAANo(cos)', 'PX(KM)', 'PXDOT(KM/S)', 'Ph')
35951	AY(KM)	PY(KM)	('PX(KM)', 'PY(KM)', 'PZ(KM)', 'Pk', 'TSINCE(MIN)')
35951	AZ(KM)	PZ(KM)	('PX(KM)', 'Ph', 'Pq')
35951	ASMAo(KM)	PSMAo(KM)	('PLAT(DEG)', 'PLON(DEG)', 'PSMAo(KM)', 'TSINCE(MIN)')
35951	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PRAANm(sin)', 'PRAANo(sin)', 'Ph', 'TSINCE(MIN)')
35951	ALON(DEG)	PLON(DEG)	('PLAT(DEG)', 'PLON(DEG)', 'PRAANo(sin)', 'PYDOT(KM/S)', 'TSINCE(MIN)')
35951	AVelNorm(KM	PVelNorm(KM	('PLON(DEG)', 'PRAANm(sin)', 'PVelNorm(KM/S)', 'PYDOT(KM/S)', 'TSINCE(MIN)')
37140	AX(KM)	PX(KM)	('PL', 'PRAANo(DEG)', 'PRAANo(sin)', 'PTRUEANOMo(sin)', 'PX(KM)', 'Pq', 'TSINCE(MIN)')
37140	AY(KM)	PY(KM)	('PMAm(DEG)', 'PNODAL_PER(MIN)', 'PY(KM)', 'PZ(KM)', 'Ph', 'Pp', 'TIMEA')
37140	AZ(KM)	PZ(KM)	('PAOPo(sin)', 'PL', 'PLON(DEG)', 'PYDOT(KM/S)', 'PZ(KM)', 'Pp', 'TIMEA')
37140	ASMAo(KM)	PSMAo(KM)	('PAOPo(cos)', 'PN(REVS/DAY)', 'PSMAo(KM)', 'PTRUEANOMo(DEG)', 'TSINCE(MIN)')
37140	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PN(REVS/DAY)', 'PRAANo(sin)', 'Pp', 'TSINCE(MIN)')
37140	ALON(DEG)	PLON(DEG)	('PINCm(DEG)', 'PLON(DEG)', 'PNODAL(REVS/DAY)', 'Pq', 'TSINCE(MIN)')
37140	AVelNorm(KM	PVelNorm(KM	('PVelNorm(KM/S)', 'PYDOT(KM/S)', 'PZ(KM)', 'Ph', 'TSINCE(MIN)')
37834	AX(KM)	PX(KM)	('PLON(DEG)', 'PRAANo(sin)', 'PSMAo(KM)', 'PX(KM)', 'PXDOT(KM/S)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
37834	AY(KM)	PY(KM)	('PANOM_PER(MIN)', 'PMAm(cos)', 'PRAANm(cos)', 'PSMAo(KM)', 'PVelNorm(KM/S)', 'PX(KM)', 'PY(KM)')
37834	AZ(KM)	PZ(KM)	('PAOPo(cos)', 'PINCm(DEG)', 'PMAm(DEG)', 'PTRUEANOMo(DEG)', 'PZ(KM)', 'Pq', 'TSINCE(MIN)')
37834	ASMAo(KM)	PSMAo(KM)	('PAOPm(DEG)', 'PECCm(-)', 'PLON(DEG)', 'PMAm(sin)', 'PN(REVS/DAY)', 'Ph', 'TSINCE(MIN)')
37834	AINCo(DEG)	PINCo(DEG)	('PECCo(-)', 'PINCo(DEG)', 'PRAANm(sin)', 'PRAANo(cos)', 'Pq')
37834	ALON(DEG)	PLON(DEG)	('PAOPm(cos)', 'PLON(DEG)', 'PSMAo(KM)', 'PTRUEANOMo(sin)', 'Pk')
37834	AVelNorm(KM	PVelNorm(KM	('PHT(KM)', 'PRAANm(cos)', 'PRAANm(sin)', 'PTRUEANOMo(sin)', 'PVelNorm(KM/S)')
41021	AX(KM)	PX(KM)	('PLAT(DEG)', 'PX(KM)', 'PXDOT(KM/S)', 'Pk', 'Pq')
41021	AY(KM)	PY(KM)	('PAOPm(DEG)', 'PRAANm(sin)', 'PRAANo(DEG)', 'PX(KM)', 'PY(KM)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
41021	AZ(KM)	PZ(KM)	('PAOPo(DEG)', 'PAOPo(sin)', 'PRAANm(cos)', 'PRAANo(cos)', 'PZ(KM)', 'PZDOT(KM/S)', 'Pq')
41021	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PECCo(-)', 'PLON(DEG)', 'PSMAo(KM)', 'PTRUEANOMo(sin)', 'Ph', 'TSINCE(MIN)')
41021	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PRAANm(sin)', 'PRAANo(cos)', 'PRAANo(sin)', 'Pp')
41021	ALON(DEG)	PLON(DEG)	('PAPOGEE(KM)', 'PECCo(-)', 'PLON(DEG)', 'PTRUEANOMo(sin)', 'TSINCE(MIN)')
41021	AVelNorm(KM	PVelNorm(KM	('PANOM_PER(MIN)', 'PHT(KM)', 'PMAm(sin)', 'Pp', 'TSINCE(MIN)')
41911	AX(KM)	PX(KM)	('PANOM_PER(MIN)', 'PL', 'PMAm(DEG)', 'PRAANm(DEG)', 'PX(KM)', 'PXDOT(KM/S)')
41911	AY(KM)	PY(KM)	('PECCo(-)', 'PL', 'PLON(DEG)', 'PPERIGEE(KM)', 'PRAANm(DEG)', 'PY(KM)')
41911	AZ(KM)	PZ(KM)	('PHT(KM)', 'PMAm(sin)', 'PRAANm(cos)', 'PXDOT(KM/S)', 'PYDOT(KM/S)', 'PZ(KM)', 'TSINCE(MIN)')
41911	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PLON(DEG)', 'PTRUEANOMo(DEG)', 'Pq', 'TSINCE(MIN)')
41911	AINCo(DEG)	PINCo(DEG)	('PECCm(-)', 'PHT(KM)', 'PINCo(DEG)', 'Pk', 'TIMEA')
41911	ALON(DEG)	PLON(DEG)	('PANOM_PER(MIN)', 'PLON(DEG)', 'PSMAo(KM)', 'PTRUEANOMo(DEG)', 'TSINCE(MIN)')
41911	AVelNorm(KM	PVelNorm(KM	('PTRUEANOMo(sin)', 'PVelNorm(KM/S)', 'PXDOT(KM/S)', 'PZ(KM)', 'PZDOT(KM/S)')
43602	AX(KM)	PX(KM)	('PAPOGEE(KM)', 'PINCm(DEG)', 'PPERIGEE(KM)', 'PTRUEANOMo(sin)', 'PX(KM)', 'Pp', 'TSINCE(MIN)')
43602	AY(KM)	PY(KM)	('PANOM_PER(MIN)', 'PAOPm(sin)', 'PMAm(cos)', 'PY(KM)')
43602	AZ(KM)	PZ(KM)	('PMAm(cos)', 'PPERIGEE(KM)', 'PY(KM)', 'PZ(KM)', 'Ph')
43602	ASMAo(KM)	PSMAo(KM)	('PN(REVS/DAY)', 'PSMAo(KM)', 'Pp', 'Pq', 'TIMEA')
43602	AINCo(DEG)	PINCo(DEG)	('PINCm(DEG)', 'PINCo(DEG)', 'PRAANm(DEG)', 'PRAANo(DEG)', 'Pp')
43602	ALON(DEG)	PLON(DEG)	('PANOM_PER(MIN)', 'PAOPo(cos)', 'PINCm(DEG)', 'PLON(DEG)', 'PRAANo(sin)')
43602	AVelNorm(KM	PVelNorm(KM	('PINCm(DEG)', 'PTRUEANOMo(sin)', 'PVelNorm(KM/S)', 'PYDOT(KM/S)', 'TIMEA')

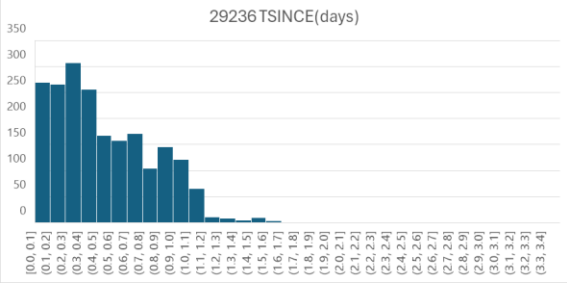
43867	AX(KM)	PX(KM)	('PINCo(DEG)', 'PRAANo(DEG)', 'PX(KM)', 'PXDOT(KM/S)', 'PY(KM)', 'Ph', 'Pk')
43867	AY(KM)	PY(KM)	('PLON(DEG)', 'PX(KM)', 'PXDOT(KM/S)', 'PY(KM)', 'PYDOT(KM/S)', 'Pk', 'TSINCE(MIN)')
43867	AZ(KM)	PZ(KM)	('PAOPo(cos)', 'PL', 'PMAm(DEG)', 'PYDOT(KM/S)', 'PZ(KM)', 'TIMEA')
43867	ASMAo(KM)	PSMAo(KM)	('PAPOGEE(KM)', 'PLON(DEG)', 'PN(REVS/DAY)', 'PRAANm(DEG)')
43867	AINCo(DEG)	PINCo(DEG)	('PAOPo(cos)', 'PINCm(DEG)', 'PINCo(DEG)', 'Pq', 'TSINCE(MIN)')
43867	ALON(DEG)	PLON(DEG)	('PAOPo(cos)', 'PLON(DEG)', 'PN(REVS/DAY)', 'PRAANm(DEG)', 'PTRUEANOMo(sin)')
43867	AVelNorm(KM	PVelNorm(K	('PINCo(DEG)', 'PMAm(DEG)', 'PNODAL(REVS/DAY)', 'PVelNorm(KM/S)')
45250	AX(KM)	PX(KM)	('PAOPo(DEG)', 'PHT(KM)', 'PINCm(DEG)', 'PRAANm(sin)', 'PRAANo(sin)', 'PX(KM)', 'PYDOT(KM/S)')
45250	AY(KM)	PY(KM)	('PAOPm(sin)', 'PECCo(-)', 'PMAm(sin)', 'PRAANo(sin)', 'PY(KM)', 'PZDOT(KM/S)', 'Pp')
45250	AZ(KM)	PZ(KM)	('PECCo(-)', 'PINCm(DEG)', 'PPERIGEE(KM)', 'PZ(KM)', 'PZDOT(KM/S)', 'Pk', 'Pq')
45250	ASMAo(KM)	PSMAo(KM)	('PAOPm(DEG)', 'PINCo(DEG)', 'PMAm(DEG)', 'PSMAo(KM)', 'TSINCE(MIN)')
45250	AINCo(DEG)	PINCo(DEG)	('PHT(KM)', 'PINCm(DEG)', 'PINCo(DEG)', 'PYDOT(KM/S)', 'PZDOT(KM/S)')
45250	ALON(DEG)	PLON(DEG)	('PAOPm(DEG)', 'PLON(DEG)', 'PMAm(DEG)', 'TIMEA', 'TSINCE(MIN)')
45250	AVelNorm(KM	PVelNorm(K	('PRAANm(DEG)', 'PRAANo(DEG)', 'PVelNorm(KM/S)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
45254	AX(KM)	PX(KM)	('PAOPm(DEG)', 'PINCm(DEG)', 'PMAm(DEG)', 'PTRUEANOMo(sin)', 'PX(KM)', 'PY(KM)', 'TSINCE(MIN)')
45254	AY(KM)	PY(KM)	('PMAm(DEG)', 'PMAm(cos)', 'PSMAo(KM)', 'PTRUEANOMo(DEG)', 'PY(KM)', 'PYDOT(KM/S)', 'TSINCE(MIN)')
45254	AZ(KM)	PZ(KM)	('PANOM_PER(MIN)', 'PN(REVS/DAY)', 'PTRUEANOMo(cos)', 'PTRUEANOMo(sin)', 'PXDOT(KM/S)', 'PZ(KM)', 'PZDOT(KM/S)')
45254	ASMAo(KM)	PSMAo(KM)	('PRAANo(DEG)', 'PSMAo(KM)', 'Pp', 'Pq', 'TSINCE(MIN)')
45254	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PLON(DEG)', 'PMAm(sin)', 'PVelNorm(KM/S)', 'PY(KM)')
45254	ALON(DEG)	PLON(DEG)	('PLON(DEG)', 'PPERIGEE(KM)', 'PTRUEANOMo(DEG)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
45254	AVelNorm(KM	PVelNorm(K	('PAOPo(sin)', 'PTRUEANOMo(DEG)', 'PVelNorm(KM/S)', 'PXDOT(KM/S)', 'TIMEA')
45344	AX(KM)	PX(KM)	('PINCo(DEG)', 'PRAANm(sin)', 'PX(KM)', 'TIMEA', 'TSINCE(MIN)')
45344	AY(KM)	PY(KM)	('PAOPm(sin)', 'PN(REVS/DAY)', 'PX(KM)', 'PY(KM)', 'PYDOT(KM/S)')
45344	AZ(KM)	PZ(KM)	('PECCo(-)', 'PL', 'PMAm(DEG)', 'PRAANo(DEG)', 'PZ(KM)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
45344	ASMAo(KM)	PSMAo(KM)	('PAOPm(sin)', 'PAOPo(cos)', 'PECCo(-)', 'PLON(DEG)', 'PRAANo(DEG)', 'PSMAo(KM)', 'TSINCE(MIN)')
45344	AINCo(DEG)	PINCo(DEG)	('PANOM_PER(MIN)', 'PECCm(-)', 'PINCo(DEG)', 'PRAANo(sin)', 'PSMAo(KM)')
45344	ALON(DEG)	PLON(DEG)	('PHT(KM)', 'PLON(DEG)', 'PNODAL_PER(MIN)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
45344	AVelNorm(KM	PVelNorm(K	('PMAm(sin)', 'PVelNorm(KM/S)', 'PXDOT(KM/S)', 'PZ(KM)', 'TSINCE(MIN)')
45460	AX(KM)	PX(KM)	('PECCm(-)', 'PECCo(-)', 'PMAm(cos)', 'PRAANo(sin)', 'PTRUEANOMo(cos)', 'PX(KM)', 'Pp')
45460	AY(KM)	PY(KM)	('PECCm(-)', 'PINCm(DEG)', 'PTRUEANOMo(DEG)', 'PY(KM)', 'Ph')
45460	AZ(KM)	PZ(KM)	('PAOPo(cos)', 'PNODAL_PER(MIN)', 'PPERIGEE(KM)', 'PSMAo(KM)', 'PTRUEANOMo(cos)', 'TIMEA', 'TSINCE(MIN)')
45460	ASMAo(KM)	PSMAo(KM)	('PLAT(DEG)', 'PRAANm(sin)', 'PRAANo(sin)', 'PSMAo(KM)', 'PZ(KM)')
45460	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PX(KM)', 'PYDOT(KM/S)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
45460	ALON(DEG)	PLON(DEG)	('PINCm(DEG)', 'PLAT(DEG)', 'PLON(DEG)', 'PZ(KM)', 'TSINCE(MIN)')
45460	AVelNorm(KM	PVelNorm(K	('PLON(DEG)', 'PVelNorm(KM/S)', 'TSINCE(MIN)')
45611	AX(KM)	PX(KM)	('PANOM_PER(MIN)', 'PINCm(DEG)', 'PNODAL_PER(MIN)', 'PRAANo(cos)', 'PX(KM)', 'Pq', 'TSINCE(MIN)')
45611	AY(KM)	PY(KM)	('PECCm(-)', 'PINCm(DEG)', 'PMAm(DEG)', 'PMAm(cos)', 'PY(KM)', 'TSINCE(MIN)')
45611	AZ(KM)	PZ(KM)	('PANOM_PER(MIN)', 'PNODAL_PER(MIN)', 'PRAANm(DEG)', 'PRAANo(DEG)', 'PYDOT(KM/S)', 'PZ(KM)', 'PZDOT(KM/S)')
45611	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PINCm(DEG)', 'PNODAL_PER(MIN)', 'PSMAo(KM)', 'PX(KM)')
45611	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PLAT(DEG)', 'PN(REVS/DAY)', 'PX(KM)', 'PZ(KM)')
45611	ALON(DEG)	PLON(DEG)	('PINCo(DEG)', 'PLON(DEG)', 'PN(REVS/DAY)', 'PNODAL(REVS/DAY)', 'TSINCE(MIN)')
45611	AVelNorm(KM	PVelNorm(K	('PAOPm(cos)', 'PLAT(DEG)', 'PMAm(cos)', 'PVelNorm(KM/S)', 'TSINCE(MIN)')
45863	AX(KM)	PX(KM)	('PLON(DEG)', 'PRAANm(DEG)', 'PRAANm(sin)', 'PRAANo(DEG)', 'PX(KM)', 'PXDOT(KM/S)', 'Pq')
45863	AY(KM)	PY(KM)	('PAOPm(cos)', 'PRAANm(sin)', 'PRAANo(DEG)', 'PRAANo(cos)', 'PX(KM)', 'PY(KM)', 'TSINCE(MIN)')
45863	AZ(KM)	PZ(KM)	('PAOPo(sin)', 'PECCo(-)', 'PRAANo(DEG)', 'PRAANo(cos)', 'PRAANo(sin)', 'PZ(KM)', 'Pk')
45863	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PHT(KM)', 'PLON(DEG)', 'PRAANo(cos)', 'PSMAo(KM)', 'PX(KM)', 'PYDOT(KM/S)')
45863	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PRAANo(sin)', 'Pp', 'Pq', 'TSINCE(MIN)')
45863	ALON(DEG)	PLON(DEG)	('PLON(DEG)', 'PN(REVS/DAY)', 'PYDOT(KM/S)', 'Pp', 'TSINCE(MIN)')
45863	AVelNorm(KM	PVelNorm(K	('PHT(KM)', 'PRAANm(cos)', 'PRAANm(sin)', 'PRAANo(cos)', 'PSMAo(KM)')
48444	AX(KM)	PX(KM)	('PANOM_PER(MIN)', 'PRAANm(cos)', 'PRAANo(cos)', 'PX(KM)', 'PXDOT(KM/S)', 'PY(KM)', 'PZ(KM)')
48444	AY(KM)	PY(KM)	('PMAm(sin)', 'PRAANm(sin)', 'PRAANo(sin)', 'PVelNorm(KM/S)', 'PX(KM)', 'PY(KM)', 'PYDOT(KM/S)')
48444	AZ(KM)	PZ(KM)	('PNODAL(REVS/DAY)', 'PRAANm(DEG)', 'TIMEA')
48444	ASMAo(KM)	PSMAo(KM)	('PINCm(DEG)', 'PINCo(DEG)', 'PLAT(DEG)', 'PSMAo(KM)', 'PZ(KM)')
48444	AINCo(DEG)	PINCo(DEG)	('PINCm(DEG)', 'PINCo(DEG)', 'PMAm(cos)', 'Ph', 'TIMEA')
48444	ALON(DEG)	PLON(DEG)	('PAOPm(cos)', 'PLAT(DEG)', 'PLON(DEG)', 'PMAm(DEG)', 'TSINCE(MIN)')
48444	AVelNorm(KM	PVelNorm(K	('PHT(KM)', 'PINCo(DEG)', 'PMAm(DEG)', 'PZDOT(KM/S)', 'TIMEA')
49330	AX(KM)	PX(KM)	('PINCm(DEG)', 'PX(KM)', 'TIMEA')
49330	AY(KM)	PY(KM)	('PLAT(DEG)', 'PY(KM)', 'PZDOT(KM/S)')
49330	AZ(KM)	PZ(KM)	('PAOPo(sin)', 'PAPOGEE(KM)', 'PINCm(DEG)', 'PRAANo(sin)', 'PZ(KM)', 'Ph', 'TIMEA')
49330	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PAOPm(DEG)', 'PAOPm(cos)', 'PAOPo(cos)', 'PECCm(-)', 'PPERIGEE(KM)', 'Pp')
49330	AINCo(DEG)	PINCo(DEG)	('PAPOGEE(KM)', 'PINCo(DEG)', 'PLON(DEG)', 'PRAANm(DEG)', 'TSINCE(MIN)')
49330	ALON(DEG)	PLON(DEG)	('PINCm(DEG)', 'PLON(DEG)', 'PRAANm(DEG)', 'PRAANm(cos)', 'Pq')
49330	AVelNorm(KM	PVelNorm(K	('PMAm(sin)', 'PPERIGEE(KM)', 'PVelNorm(KM/S)', 'PZDOT(KM/S)', 'TSINCE(MIN)')
49818	AX(KM)	PX(KM)	('PAOPo(DEG)', 'PLON(DEG)', 'PN(REVS/DAY)', 'PNODAL_PER(MIN)', 'PX(KM)', 'TSINCE(MIN)')
49818	AY(KM)	PY(KM)	('PVelNorm(KM/S)', 'PY(KM)', 'Pq')
49818	AZ(KM)	PZ(KM)	('PHT(KM)', 'PMAm(sin)', 'PRAANo(sin)', 'PSMAo(KM)', 'PVelNorm(KM/S)', 'PXDOT(KM/S)', 'PZ(KM)')
49818	ASMAo(KM)	PSMAo(KM)	('PANOM_PER(MIN)', 'PAOPm(DEG)', 'PAOPo(DEG)', 'PAOPo(cos)', 'PAOPo(sin)', 'PSMAo(KM)', 'TSINCE(MIN)')
49818	AINCo(DEG)	PINCo(DEG)	('PECCo(-)', 'PINCo(DEG)', 'PRAANo(DEG)', 'PRAANo(cos)', 'PRAANo(sin)')
49818	ALON(DEG)	PLON(DEG)	('PLON(DEG)', 'PMAm(sin)', 'Ph', 'TSINCE(MIN)')
49818	AVelNorm(KM	PVelNorm(K	('PMAm(sin)', 'PVelNorm(KM/S)', 'PXDOT(KM/S)', 'PYDOT(KM/S)', 'PZDOT(KM/S)')
58573	AX(KM)	PX(KM)	('PLAT(DEG)', 'PRAANm(cos)', 'PRAANo(cos)', 'PVelNorm(KM/S)', 'PX(KM)', 'PZ(KM)', 'PZDOT(KM/S)')
58573	AY(KM)	PY(KM)	('PAOPo(cos)', 'PLAT(DEG)', 'PRAANm(sin)', 'PRAANo(sin)', 'PTRUEANOMo(cos)', 'PY(KM)', 'PZ(KM)')
58573	AZ(KM)	PZ(KM)	('PLAT(DEG)', 'PRAANm(cos)', 'PRAANm(sin)', 'PRAANo(sin)', 'PYDOT(KM/S)', 'PZ(KM)', 'PZDOT(KM/S)')
58573	ASMAo(KM)	PSMAo(KM)	('PECCo(-)', 'PLAT(DEG)', 'PN(REVS/DAY)', 'PRAANm(sin)', 'PRAANo(sin)', 'PZ(KM)', 'PZDOT(KM/S)')
58573	AINCo(DEG)	PINCo(DEG)	('PINCo(DEG)', 'PRAANm(cos)', 'PRAANo(cos)', 'PVelNorm(KM/S)', 'PX(KM)')
58573	ALON(DEG)	PLON(DEG)	('PAOPo(sin)', 'PLAT(DEG)', 'PLON(DEG)', 'PTRUEANOMo(sin)', 'PZ(KM)', 'Ph')
58573	AVelNorm(KM	PVelNorm(K	('PAOPm(sin)', 'PLAT(DEG)', 'PMAm(sin)', 'PNODAL(REVS/DAY)', 'PVelNorm(KM/S)')

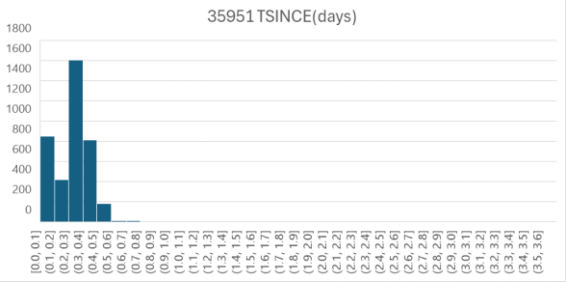
Appendix 2 - Histograms of Time Between Observations

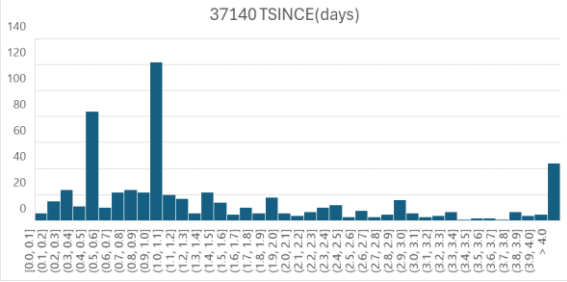
The x-axis shows the bin size in fractions of a day. The bin size for 14414 is 0.1 of a day and extends from 0 to 4.0 days. The y axis is the number of observations in each bin. The x and y axis are not normalized for each space object. The plots illustrate the variability between the rates of observations for the space objects analyzed.

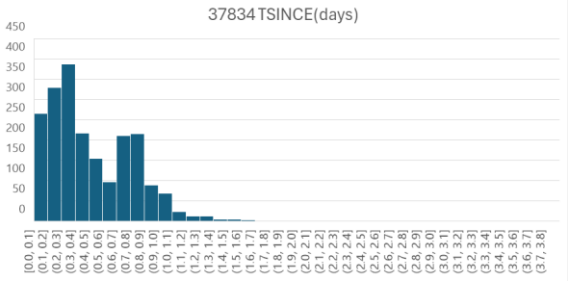


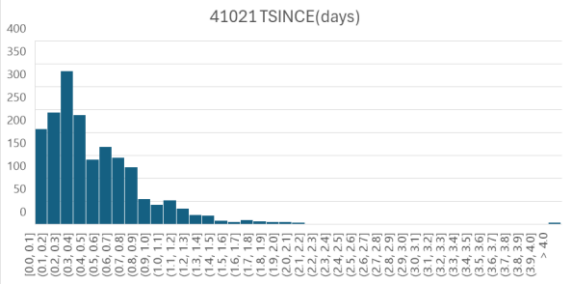


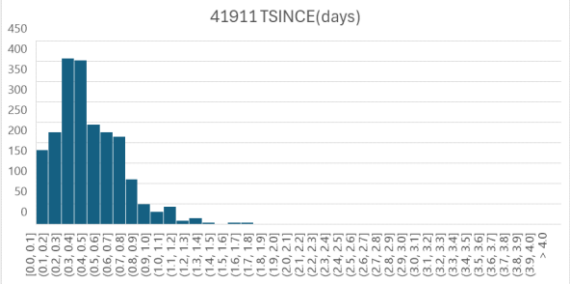


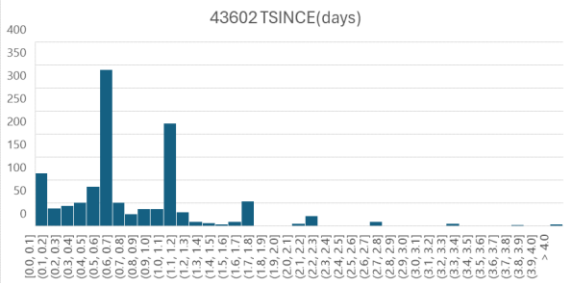


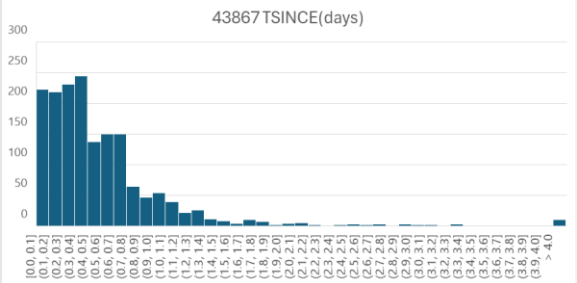


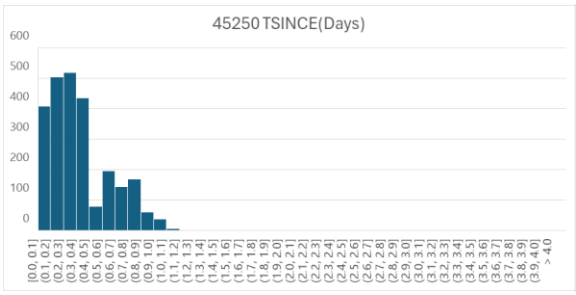


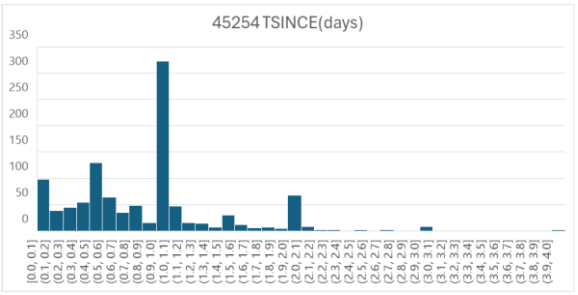


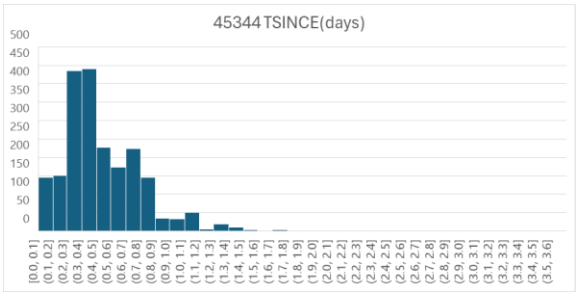


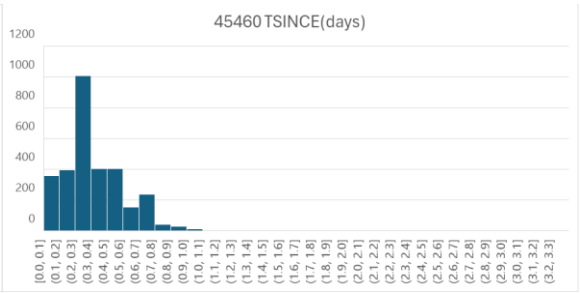


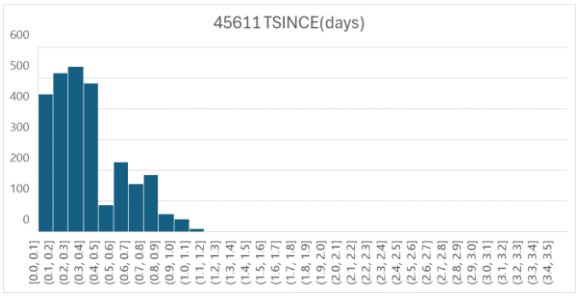


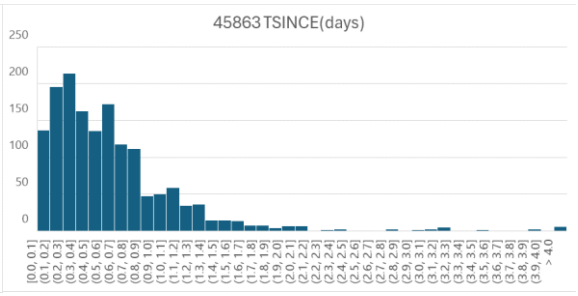


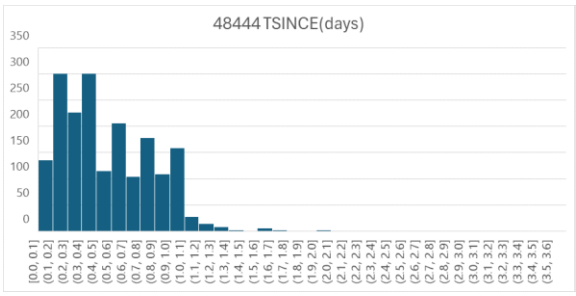


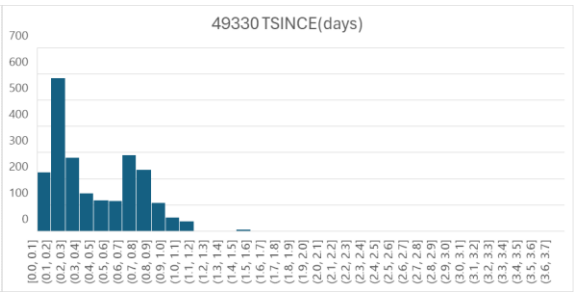


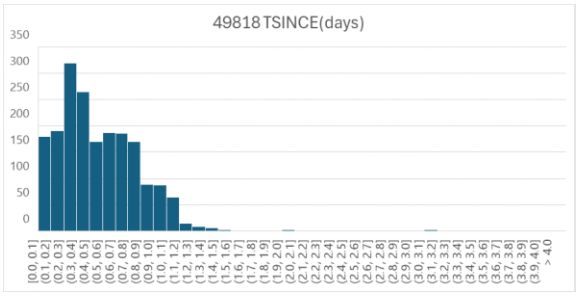


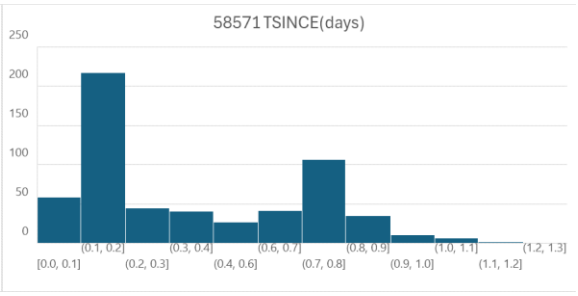












Appendix 3 - Cumulative Distribution Function (CDF) of Residuals

The Cumulative Distribution Function Plots for all the 20 selected space objects analyzed in this study. The x and y axis are not normalized for each space object. The y axis is a log scale to help illustrate the differences between residuals from NEP and traditional propagation.