Artificial Intelligence Research and Development
I. Sanz et al. (Eds.)

doi:10.3233/FAIA230654

Inter-Satellite Link Prediction with Supervised Learning Based on Kepler and SGP4 Orbits

Estel FERRER [a,1], Joan A. RUIZ-DE-AZUA [a], Francesc BETORZ [a] and Josep ESCRIG ESCRIG [a]

[a] *i2CAT Foundation*

ORCiD ID: Estel Ferrer https://orcid.org/0000-0001-9999-2844, Joan A. Ruiz-de-Azua https://orcid.org/0000-0001-5892-3404, Francesc Betorz https://orcid.org/0009-0004-4022-6626, Josep Escrig Escrig https://orcid.org/0000-0002-0918-8148

Abstract.
Distributed Space Systems (DSS) are gaining prominence in the space industry due to their ability to increase mission performance by allowing cooperation and resource sharing between multiple satellites. In DSS where communication between heterogeneous satellites is necessary, achieving autonomous cooperation while minimizing energy consumption is a critical requirement, particularly in sparse constellations with nano-satellites. In order to minimize the functioning time and energy consumed by the Inter-Satellite Links (ISL) established for satellite-to-satellite communication, their temporal encounters must be anticipated. This work proposes an autonomous solution based on Supervised Learning that allows heterogeneous satellites in circular polar Low-Earth Orbits to predict their encounters, given the Orbital Elements (OE) and assuming isotropic antenna patterns. The model performance is evaluated and compared in two different scenarios: 1) a simplified scenario assuming that satellites follow Kepler orbits and 2) a realistic scenario assuming that satellites follow Simplified General Perturbations 4 (SGP4) orbits. This work could be considered the first stage of a promising and alternative approach in the field of DSS.

Keywords. Distributed Space Systems, Non-Terrestrial Networks, Inter-Satellite Links, Low-Earth Orbits, Kepler, Simplified General Perturbations, SGP4, Simplified Perturbations Models, 6G, Supervised Learning

1. Introduction

In recent years, the space industry has undergone a significant shift toward the implementation of Distributed Space Systems (DSS) [1]. As opposed to traditional monolithic systems, DSS consists of multiple satellites that communicate and coordinate their activities to increase the overall mission performance and provide global and continuous

[1] Corresponding Author: Estel Ferrer, estel.ferrer@i2cat.net

coverage for applications such as telecommunications, navigation, and remote sensing.

DSS requiring satellite-to-satellite communications gain importance with the definition of Non-Terrestrial Networks (NTN) and sixth generation (6G). Due to the satellites' dynamics, the connections established for satellite-to-satellite communications, known as Inter-Satellite Links (ISL), are characterized by a limited lifetime. In order to minimize the functioning time and energy consumed by the ISL, these temporal encounters must be anticipated, especially in sparse constellations where the connections can be sporadic.

The traditional solution for satellite encounter prediction is based on deterministic and centralized on-ground orbit propagation [2], [3]. This approach utilizes an orbital model to propagate the satellite's initial state, which is encoded in the Two-line Elements (TLE). Given the satellite trajectories and assuming isotropic antennas, the encounters can be scheduled defining a maximum threshold distance for communication. While on-ground propagation solutions can generate precise contact plans, the centralized nature of these solutions makes them unsuitable for DSS with heterogeneous satellites since it would require substantial cooperation efforts between different stakeholders. To overcome these limitations, new investigations are moving towards deterministic and decentralized solutions, providing autonomy to the satellites [4]. However, these techniques are still based on orbital propagation, which can result in a high processing cost for satellites with limited resources, such as nano-satellites. To avoid orbital propagation while keeping a decentralized and autonomous solution, authors in [5] have designed predictive algorithms to self-learn and self-construct the contact plan. However, in order to mathematically formulate the solution, they apply some linearizations only valid for circular and same-high satellites.

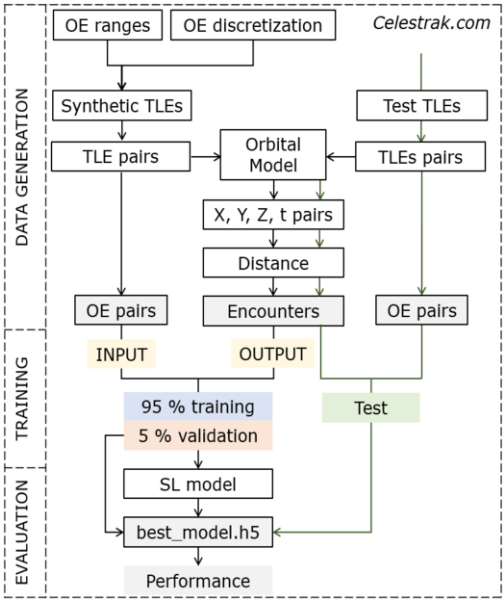
Recent studies have investigated the use of Graph Neural Networks (GNNs) together with Recurrent Neural Networks (RNNs) to model the temporal evolution of dynamic systems and predict future behavior based on past data [6–12]. The feasibility of treating a constellation as a time-evolving graph has been studied in [13] but discarded due to its complexity, poor performance, and strong dependency on past ground-truth data. However, new under-revision studies [14] are further investigating in this promising direction.

Building on a previous publication [15], the current work presents some advances in the proof of concept of using supervised learning to predict the contact plans between heterogeneous satellites. Concretely, this work presents and compares the results of predicting the encounters between Low Earth Orbit (LEO) satellites with isotropic antenna radiation patterns in two different scenarios: 1) ideal scenario assuming that satellites follow Kepler orbits, and 2) realistic scenario assuming that satellites follow Simplified General Perturbations 4 (SGP4) model, which accounts for perturbations caused by Earth's shape and atmospheric drag. In both scenarios, the same model architecture and input data are used. Even though the computation cost of the approach is a crucial point, this work focuses on obtaining good prediction performances. The analysis of the inference time and resources is left for future work.

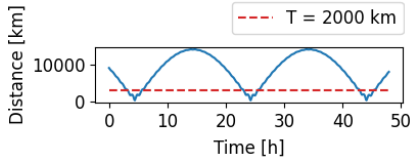
The remainder of the article is structured as follows. Section 2 defines the problem statement, introducing the datasets used for training, validation, and testing. Section 3 details the SL model architecture used to fit the input and output data. Section 4 presents the model performance in the validation and test set and compares the results obtained with the two different orbital models. Finally, Section 5 concludes the paper with a summary of keyfindings and suggestions for future research in this area.

2. Problem statement

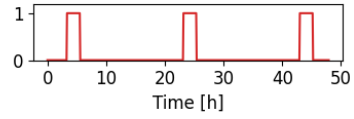
The encounter prediction problem consists in predicting the communication opportunities between any pair of satellites at any time. This problem is directly related to the communication features of the satellites and their trajectories. In this case, an isotropic antenna pattern is assumed for all the satellites. Therefore, having an encounter means having a satellite-to-satellite distance lower than a given threshold. Concerning satellite trajectories, both Kepler and SGP4 models are used and compared to propagate the initial state of the satellites.



(a) Block diagram showing the most important blocks used for data generation, model training and model evaluation



(b) Satellite-to-satellite distance and threshold distance.



(c) Output binary sequence of encounters with 576 samples corresponding to 48 hours with 5 minutes discretization.

Figure 1. Encounter prediction diagram, together with an example of the satellie-to-satellie distance and the derived binary sequence of encounters

The solution proposed in this work consists in training a SL model with a large set of input-output pairs. Figure 1a illustrates a simplified diagram of the three main steps that define the solution: data generation, model training, and model evaluation. The

role of the data generation part is to create the input and output data used for training, validation, and testing. On the one hand, the input data contains the Orbital Elements (OE) for different satellite pairs, a set of six parameters that locate a satellite by uniquely identifying its orbit shape, size, and orientation. Both training and validation input data are synthetically generated from 1000 different OE random values within the range for circular, polar, and low orbits defined in Table 1. Notice that, since all orbits are circular, both the eccentricity (e) and the argument of periapsis (ω) are always zero and can be excluded from the input set. As an example, Table 2 presents the first six rows of the input training data, featuring the satellite pairs 0-1, 0-2, 1-0, 1-2, 2-0, and 2-1. On the other hand, the output ground-truth data is the binary time sequence that determines when the satellite-to-satellite distance is below a given upper bound threshold set to 2000 kilometers (see Figs. 1b and 1c). To determine the satellite-to-satellite distance, it is assumed that the OE values are defined at the exact same moment in time, which coincides with the start of the encounter signal. Each encounter signal is characterized by a set of 576 binary output features, representing a total duration of 48 hours discretized into 5-minute intervals. It is important to note that these output data points are directly linked to both the input data and the orbital model used to propagate the satellites' initial positions.

Table 1. OE ranges for circular, polar, and low orbits. e: eccentricity, M: mean anomaly, i: inclination, ω: argument of periapsis, Ω: right ascending node, a: semi-major axis.

OE	e [-]	M [°]	i [°]	ω [°]	Ω [°]	a [km]
From	0	0	90	0	0	6800
To	0	360	100	0	360	7300

Table 2. First six samples of the input training set, containing the orbital necessary information for satellites 0, 1 and 2

Pair ID	$M1$ [rad]	$i1$ [°]	$\Omega1$ [rad]	$a1$ [km]	$M2$ [rad]	$i2$ [°]	$\Omega2$ [rad]	$a2$ [km]
0-1	4.40	95.2	6.05	7,335	3.04	97.7	2.88	6,906
0-2	4.40	95.2	6.05	7,335	3.11	96.0	4.50	6,746
1-0	3.04	97.7	2.88	6,906	4.40	95.2	6.05	7,335
1-2	3.04	97.7	2.88	6,906	3.11	96.0	4.50	6,746
2-0	3.11	96.0	4.50	6,746	4.40	95.2	6.05	7,335
2-1	3.11	96.0	4.50	6,746	3.04	97.7	2.88	6,906

In the second part of the diagram in Fig. 1a, the SL model is fitted with the training input-output sequences: 95 % of synthetic samples are used for training, while the remaining 5 % is used for validation to avoid overfitting. Details about the model architecture are presented in section 3. Finally, the model parameters learned during training are saved and used to predict both validation and test sets. The test set is a collection of 48 realistic satellites obtained from Celestrak [16] by gathering all active satellite TLE data and filtering those in circular, polar, and low orbits. The model performance, presented in section 4, is obtained by comparing the predicted encounters with the ground truth.

Notice that, to anticipate the encounter, this approach assumes that the satellite not only needs the trained model and its own OE values but also the OE values of the other satellite. This information is assumed to be known with enough anticipation.

3. Supervised Learning Model

Supervised Learning (SL) is a well-known machine-learning method used to solve problems where a large set of input-output pairs can be obtained yet the mathematical relationship between them is highly complex, costly, or unknown. While the training stage is usually costly, the simplicity of the operations in the inference part renders it a cost-effective approach, which could be suitable for resource-constrained systems like nanosatellites. However, it is important to clarify that this work focuses on finding the SL model architecture that best fits the input data. Future research will address tasks concerning resource optimization and trade-off analysis between performance and computational costs.

Figure 2 illustrates the Fully-Connected Neural Network (FCNN) architecture proposed to map the input OE pairs to the output sequence of encounters. The optimal number of layers and neurons per layer depends on the number of input and output features. As stated in section 2, the input set consists of 8 values, while the output describes the next 48 hours with 576 binary features. Since the number of input features is much lower than the number of output features, increasing the number of neurons while progressing deeper into the model is a good approach. Based on this reasoning, multiple architectures with varying numbers of hidden layers and neurons were evaluated. Among all the evaluated models, the one depicted in Fig. 2, with 8 layers and 270,048 trainable parameters, presents one of the best training curves with high generalization.

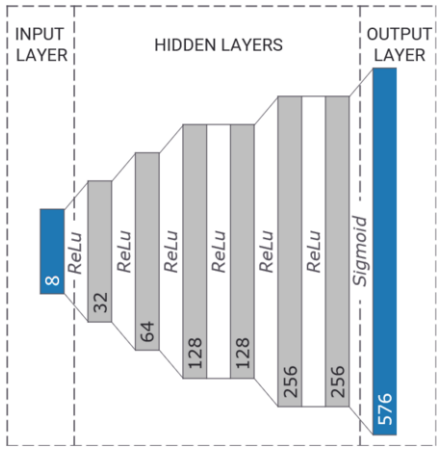


Figure 2. SL model architecture consisting of 8 layers. The values within each layer represent the number of neurons while the word in-between each layer is the applied activation function

Rectified Linear Unit is used as the activation function for the first seven layers. Contrarily, Sigmoid activation function is used at the output layer to ensure each neuron value ranges from 0 to 1. Notice that, since the ground truth is a binary signal, a decision threshold should be applied at the end to decide if the predicted values should be considered as no encounter (values close to 0) or encounter (values close to 1).

Other training settings include 500 epochs of training, a batch size of 3000 samples, the use of a binary cross-entropy loss function, the Adam optimizer with a learning rate of 0.01, and early stopping based on a patience of 200 epochs.

4. Results

This section presents the performance of the SL model presented in section 3. Two different scenarios are evaluated and compared: the ideal scenario and the realistic scenario. In the ideal scenarios, the output sequence of encounters used for training, validating, and testing are computed using Keplerian orbits, an ideal orbital model where external perturbations are not considered. Contrarily, in the realistic scenario, SGP4 orbital model, which includes Earth's shape and atmospheric drag, is assumed. Poorer results are expected with SGP4, since output-dependent information, concerning epoch time, satellite mass, and drag coefficient, is missing in the input data.

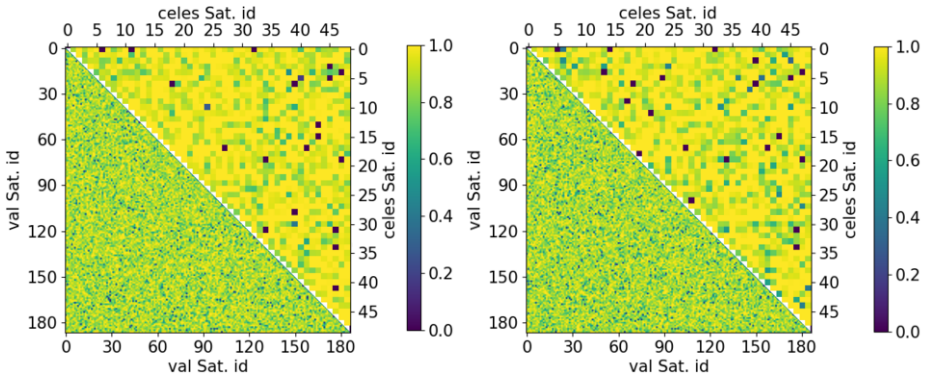
In both scenarios, the $F1$ score is used to quantify the model's performance for all satellite pairs. Equation (1) defines $F1$ as the harmonic mean of Precision (P) and Recall (R). In our scenario, P is the probability of estimating a contact inside the real timeslot (viewed as a measure of wasted energy in the prediction), while R assesses the proportion of successful encounters relative to the total available communication time and serves as an indicator of missed communication opportunities.

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{1}$$

where TP, FP, and FN stand for True Positive, False Positive, and False Negative, respectively. Note that, to compute TP, FP, and FN, a decision threshold needs to be applied at the output of the model. For the following results, a threshold of 0.5 is used, meaning that all model-output values higher or equal to 0.5 are set to one, while values below this threshold are set to zero.

Figures 3a and 3b show the $F1$ score obtained for both validation satellite pairs (triangle below diagonal) and test satellite pairs (triangle above diagonal) assuming Kepler and SGP4 orbital models, respectively. Notice that some concrete pairs present zero $F1$. These pairs correspond to those with a limited number of encounters, in which having zero TP is almost unavoidable, and a single FP or FN leads to an overall $F1$ of zero. Given these observations, it is crucial to also consider other metrics, such as Balance Accuracy (BA) to gain a more comprehensive understanding of the model's performance. As observed in equation 2, BA can also be used for unbalanced datasets, and unlike $F1$, it is especially useful for critical situations where $TP = 0$.

$$BA = \frac{TPR + TNR}{2} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{2}$$



(a) Ideal scenario assuming Kepler orbits to compute the encounters

(b) Realistic scenario assuming SGP4 orbits to compute the encounters

Figure 3. F1-score for all validation and test satellite pairs. The triangle below the diagonal corresponds to the validation results while the triangle above the diagonal contains the test results

where TPR and TNR stand for True Positive Rate (also known as Recall or Sensitivity) and True Negative Rate (also known as Specificity), respectively.

Table 3 shows the mean BA, as well as the mean $F1$, mean P, and mean R for both scenarios. Unexpectedly, despite output-dependent information is missing in the realistic case with SGP4, the results for both scenarios are very similar, with a mean BA above 90 % and a mean $F1$ above 80 %. Based on this similarity, it can be inferred that, despite the stark difference between Keplerian and SGP4 trajectories, the satellite-to-satellite distance, and therefore, the output sequence of encounters. Consequently, only knowing the semi-major axis a, the mean anomaly M, and the right-ascending node Ω for two satellites in circular, polar, and low orbits, the future encounters under realistic SGP4 trajectories can be predicted with a BA of around 95 %. However, this is only true for small propagation times of less than a week: as time progresses, the influence of missing initial state information related to Earth's shape and drag (epoch, satellite mass, and drag coefficient) gains exponential importance.

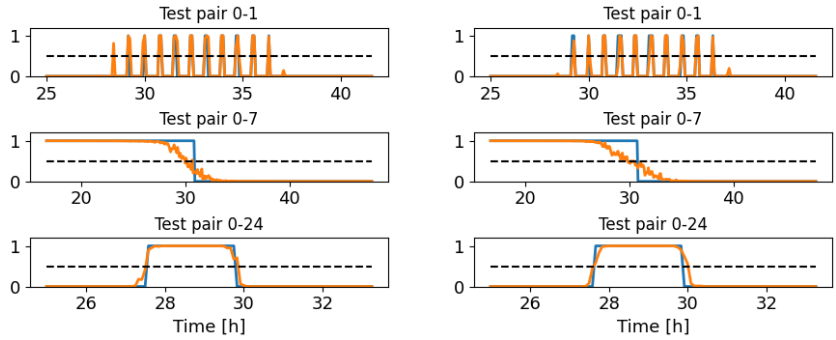
Although the results presented in Table 3 are notably good, they also indicate a decrease in the validation performance compared to the test performance. This behavior can be attributed to the fact that the validation dataset has an inclination range between 90 ° and 100 °, whereas, in a typical satellite constellation, such as the test Celestrak dataset, the polar satellites are confined to a much smaller inclination range between 96 ° and 100 °, corresponding to the Sun-synchronous orbits.

Finally, Figure 4 presents a comparison between the encounter ground truth and the model prediction for three different satellite pairs in the Celestrak test set during a specific time range of interest. The results show that the model is able to correctly predict the encounters for unseen and realistic input data, even in complex scenarios where contacts are really short in time.

Notice that the decision threshold plays an important role in the performance of the model and should be used depending on the user interests: if the maximum number of

Table 3. Mean model performance obtained for both validation and test sets and for both Kepler and SGP4 scenarios

	Kepler		SGP4	
	Val.	Test	Val.	Test
BA [%]	92.34	96.54	91.91	95.69
*F*1 [%]	85.62	92.76	84.98	91.76
P [%]	89.26	94.58	89.33	94.96
R [%]	85.06	93.47	84.19	91.63



(a) Ideal scenario assuming Kepler orbits to compute the encounters

(b) Realistic scenario assuming SGP4 orbits to compute the encounters

Figure 4. Comparison of the true encounters (in blue) with the model prediction (in orange). A decision threshold of 0.5 (dashed back line) is used to post-process the results and obtain the predicted encounters

encounters want to be detected, the threshold should be reduced; otherwise if the false encounters want to be minimized, the threshold should be increased. In future works, the impact of different decision thresholds will be studied.

5. Conclusions

This work presents a supervised machine-learning approach for forecasting encounters between satellites in low, circular, and polar orbits. The model uses 8-dimensional synthetic input training data obtained by combining various pairs of orbital elements within the region of interest for low, circular, and polar orbits. The output data is a set of 48-hour binary time series that describe the encounters of different satellite pairs assuming isotropic antenna patterns with a maximum threshold distance of 2000 km. Two different orbital models are used to compute the output data used for training, validating, and testing: ideal Kepler model and realistic SGP4, which accounts for the most important perturbations in LEO orbits. A Fully-Connected Neural Network architecture is used for training and predicting the output given the input information. The model performance is analyzed for the two scenarios.

Results show that both scenarios present high and similar performance, with $F1$ and *BA* of around 80 % and 90 %, respectively. On the one hand, these results demonstrate that a supervised learning model can be used to predict the encounters in polar, circular,

low, and realistic SGP4 orbits only knowing the relative inclination, the semi-major axis a, the mean anomaly M, and the right-ascending node Ω. On the other hand, it can be inferred that, despite the highly notable variation between Keplerian and SGP4 trajectories, the satellite-to-satellite distance, and therefore, the output sequence of encounters, remains almost invariant for a propagation time of only two days.

Future work must encompass more challenging frameworks, moving from low, circular, and polar orbits to low orbits with different eccentricities and a broader range of inclinations. We expect that larger training sets and more complex Machine Learning models will be required, providing novel insights for a future onboard AI-assisted device. In this sense, the computational cost of the model inference should be analyzed and optimized in the near future to be used in a real application.

References

[1] C. A. L opez. In pursuit of autonomous distributed satellite systems. Ph.D. dissertation, Polytechnic University of Catalonia, 2019

[2] D. Fischer, D. Basin, K. Eckstein, and T. Engel. Predictable mobile routing for spacecraft networks. IEEE Transactions on Mobile Computing, vol. 12, no. 6, pp. 1174–1187, 2012.

[3] J. A. Fraire and J. M. Finochietto. Design challenges in contact plans for disruption-tolerant satellite networks. IEEE Communications Magazine, vol. 53, no. 5, pp. 163–169, 2015

[4] O. C. FSu, J. B. Malaer and K. Suh. Using mobility prediction to enhance network routing in leo crosslink network. Proc. Int. Astron. Congr. (IAC), p. 1, 2019.

[5] J. A. Ruiz-De-Azua, V. Ramırez, H. Park, A. Calveras, and A. Camps. Assessment of satellite contacts using predictive algorithms for autonomous satellite networks. IEEE access, vol. 8, pp. 100 732–100 748, 2020.

[6] F. Manessi, A. Rozza, and M. Manzo. Dynamic graph convolutional networks. Pattern Recognition, vol. 97, p. 107000, 2020.

[7] J. Chen, X. Wang, and X. Xu. Gc-lstm: Graph convolution embedded lstm for dynamic link prediction. arXiv preprint arXiv:1812.04206, 2018

[8] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan. E-lstm-d: A deep learning framework for dynamic network link prediction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 6, pp. 3699–3712, 2019.

[9] A. Taheri, K. Gimpel, and T. Berger-Wolf. Learning to represent the evolution of dynamic graphs with recurrent models. Companion proceedings of the 2019 world wide web conference, 2019, pp. 301–307.

[10] L. Qu, H. Zhu, Q. Duan, and Y. Shi. Continuous-time link prediction via temporal dependent graph neural network. Proceedings of The Web Conference 2020, 2020, pp. 3026–3032.

[11] P. Goyal, S. R. Chhetri, and A. Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowledge-Based Systems, vol. 187, p. 104816, 2020.

[12] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019, pp. 388–396

[13] E. Ferrer. Graph neural networks for optimization in telecommunications satellite constellations. Master's thesis, Polytechnic University of Catalonia, 2022.

[14] G. Casadesus and E. Alarcon. Toward autonomous cooperation in heterogeneous nanosatellite constellations using dynamic graph neural networks. International Astronautical Congress, 2023.

[15] E. Ferrer, J. Escrig and J. A. Ruiz-De-Azua. Inter-Satellite Link Prediction for Non-Terrestrial Networks Using Supervised Learning. European Conference on Networks and Communications (EuCNC) Congress, 2023.

[16] D. T. Kelso. (1985) Celestrak. [Online]. Available: https://celestrak.org/