AAS 20-734

NEURAL NETWORK BASED TLE ERROR MODELING AND QUANTIFICATION

Nathan Reiland*, Simon Jaxy†, Di Wu‡ and Aaron J. Rosengren§

This paper proposes a neural network based method for TLE (Two-Line Element) data error modeling. The complicated mapping between TLE error and a satellite's position and velocity is largely unknown due to the lack of data driven techniques that are effective when applied to complex, nonlinear dynamical systems. For this study, the passive MEO (medium Earth orbit) satellites, Lageos 1 and Lageos 2 are considered. Their high accuracy satellite laser ranging (SLR) based ephemeris from the ILRS (international laser ranging service) is compared with corresponding TLE data to create a dataset for the neural network's training and validation. The results of this work illustrate the possibility of utilizing neural networks to estimate TLE error information, thereby improving the usability of TLEs for SSA applications.

INTRODUCTION

In February 2017 the number of Resident Space Objects (RSOs) larger than 10 cm occupying circumterrestrial space was estimated to be less than 21,000.[1] As of January of 2018, that estimate had grown to be 29,000[2] and by May 30th 2020[3] it is predicted that a shocking 34,000 RSOs, largely comprised of debris, will orbit our planet. The exponential growth of space debris is considered to be a threat not only to active satellites, representing billions of dollars of investments, but also to the general sustainability of the space environment, a resource which Kessler theorized could be put rendered unusable if the growth in man made debris is not kept in check.[4]

The United States Space Surveillance Network (SSN), operated by JSPOC, is one of the few sources that provides satellite ephemeris to the public. Using a global network of radar and optical sensors, the SSN tracks and maintains a large catalogue of about 22,300 RSOs[3] in the form of Two-Line Elements (TLE). These TLEs provide vital information for commercial satellite operators as well as the Space Situation Awareness (SSA) community as a whole. This information is used on a daily basis to predict conjunctions and inform operators when a collision avoidance maneuver is necessary. Unfortunately, the efficacy of TLEs for such applications is greatly limited by the inaccuracy of each TLE state, as well as the lack of information regarding the uncertainty of that state. The inaccuracy of the TLE states can be attributed largely to measurement error and error resulting from orbit determination, the process of fitting a physical orbit to observations. However, because the specific mechanisms through which TLEs are produced is unknown[5] and likely not consistent, most estimations of the uncertainty of TLEs states is the result of empirical evidence.

*Systems Engineer, Raytheon Missiles and Defense, Tucson, Arizona.
†BSc., Artificial Intelligence Department, Vrije Universiteit Brussels.
‡Graduate Student, Aerospace and Mechanical Engineering Department, University of Arizona, 85719.
§Assistant Professor, Department of Mechanical and Aerospace Engineering, University of California San Diego, 92093.

The danger of relying too heavily on TLEs was demonstrated in the inability of SSA systems to predict and prevent the famed Iridium-Cosmos collision of 2009.[6] Furthermore, if the uncertainty of the TLEs themselves had been better known and less faith had been placed in systems relying upon them, perhaps other techniques could have been utilized and the disaster could have been avoided. Of course this is rather hyperbolic, but the point stands, that reliable knowledge about the uncertainty of state estimates is crucial for the successful functioning of any real world system. Therefore, the study of TLEs, and specifically the study of their uncertainties and error sources has been central to a great deal of research. For example, Danielson et al. applied batch least squares differential correction to TLE data,[7] while Levit and Marshall used similar techniques to prove that TLE data accuracy could be improved.[8] While classical orbit parameter-based differential correction methods illustrate the possibility of reducing TLE error in orbit determination, the resulting exponential growth in the TLE's error also showcases its limitations.

Data driven techniques have proven to be successful in many fields and SSA, which involves both complex dynamical systems and huge amounts of data, offers many scenarios where such techniques could be employed. Hartikainen et als. introduced a way of building non-parametric data-driven quasi-periodic latent force models,[9] Sharma and Cutler demonstrated applying distribution regression and transfer learning in orbit determination for the LEO cubsat GRIFEX,[10] and Peng and Bai proposed a general supervised learning framework for orbit error estimation.[1] Furthermore, Rasit et al., accepting the dynamical limitations of TLEs, applied data science inspired techniques for computing initial condition corrections leading to improved orbit determination.[11]

While researchers focusing on orbit determination explored the utilization of data driven techniques for improving both dynamical models and initial conditions for RSOs, statistical studies of TLE data without the involvement of dynamical processes also yielded profound results. In order to provide a reasonable covariance value for TLE data, Flohrer et al. estimated TLE uncertainties and created covariance look-up tables by analyzing eight separate, complete, 60-day TLE catalogs and comparing their states with precise ephemeris from both TIRA (Tracking and Imaging Radar), operated by FGAN (Forschungsgesellschaft fuÌˆr Angewandte Naturwissenschaften), and SLR-based predictions published by the ILRS (International Laser Ranging Service).[12] Furthermore, Racelis and Joerger, inspired by the use of overbounding theory in aviation navigation,[13–15] chose to apply the same theory to the problem of quantifying positional errors in TLEs for a set of MEO and GEO (geosynchronous equatorial orbit) satellites over the past 20 years.[2] Their findings indicated that TLE error distributions for both GEO and MEO objects were non-unique, and furthermore that the general trend of distributions between GEO and MEO were different. These findings could indicate the involvement of dynamical process in generating TLE. Because the dynamical process will vary with unique initial conditions, different errors will be introduced into the TLE.

The goal of this research is to investigate the ability of a neural network based machine learning method to connect TLE uncertainty with the Cartesian state of the satellite. The mapping between orbital information and TLE error, as illustrated by Racelis and Joerger's analysis over MEO and GEO TLE error distribution,[2] appears to be very complex and traditional bayesian techniques have so far been unable to reliably discern their relationship. Neural networks are known for their success in mapping complicated domains such as image recognition (connecting pixels with objects) and natural language processing (connecting digitalized information with other digitalized information or abstract meaning). Conversely, neural networks often struggle to predict future states of complex, dynamic systems and as such their success in the field of astrodynamics has been quite limited. Despite this, it is possible that TLE error is not strictly speaking, a time-dependent process, thus the prediction of this error may prove nevertheless to be a good application for machine learning based methods.

The focus of this study lies on MEO objects with publicly available high accuracy laser ranging ephemeris from the ILRS. In order to simplify the problem, two passive satellites were selected as test objects, Lageos 1 and Lageos 2. Because these are a passive satellites (non-maneuvering) with no power, communications, or moving parts, their motion is determined only by the gravity of the Earth and the perturbing forces of the Earth-Moon-Sun system. The obatined SLR-based ephemeris have uncertainties on the order of centimeters and therefore they serve as the "truth" data, while the TLEs serve as the "predicted" data. Training and testing data in the form of residual errors between the Cartesian position and velocity of the truth and predicted data is generated in an Earth Centered Inertial (ECI) frame. Several combined structures of neural networks are then employed to find the mapping between Cartesian state and TLE error. The paper is organized as follows. In section 2, we describe the process for transforming TLE and SLR states into Cartesian position and velocity vectors in an ECI frame. In section 3, we present our design of the neural networks and the Gaussian baseline. Finally, in section 4, we provide an analysis of our experiments and in section 5, we state our conclusion.

COORDINATE SYSTEMS AND DATASET GENERATION

Both the training and testing data for the study were generated using publicly available SLR and TLE ephemeris for the Lageos 1 and Lageos 2 satellites. The SLR data for Lageos 1 and Lageos 2 is collected and archived by the Crustal Dynamics Data Information System (CDDIS). The precise orbits can be found on the CDDIS website*. The TLE data on the other hand is obtained from Space-Track†, an online data base housing TLE time histories for a large number of RSOs. The common frame selected for the calculation of residual error is an ECI (Earth-Centered-Inertial) frame, specifically the GCRF (Geocentric Celestial Reference Frame). An ECI frame was selected because future work will involve the use of residuals expressed in Keplerian orbital elements, which are typically computed using position and velocities in an ECI frame. Because neither TLE nor SLR ephemeris is provided in the GCRF, both sets of data must be converted.

First, TLE states of the satellites of interest are converted to Cartesian position and velocity vectors using SGP4 or SDP4 depending on the altitude of the satellite. The version of the SGP4/SDP4 used is described in Vallado et al.[16] and the source code was obtained from Celestrak‡. Following Vallado and Crawford,[17] the TEME (True Equator Mean Equinox) frame is assumed for all SGP4/SDP4 operations and as such the Cartesian position and velocity vectors resulting from the propagation of a TLE with SGP4 using an integration time span of zero can be transformed to equivalent coordinates in the GCRF following a fairly straightforward procedure. First, the position and velocity vectors in the TEME frame are transfomred to the Perifocal Frame (PEF) with a rotation about the $\hat{z}$ axis equal to the Greenwich Mean Sidereal Time, θ_{GMST82}, yielding:

$$
\begin{aligned}
\boldsymbol{r}_{PEF} &= R_3(\theta_{GMST82})\boldsymbol{r}_{TEME} \\
\boldsymbol{v}_{PEF} &= R_3(\theta_{GMST82})\boldsymbol{v}_{TEME},
\end{aligned}
\tag{1}
$$

where R_3 is the rotation matrix corresponding to a rotation solely about the $\hat{z}$ axis. The position and velocity vectors in the PEF are then converted to the ITRF (International Terrestrial Reference

*URL: https://cddis.nasa.gov/Data_and_Derived_Products/SLR/Precise_orbits.html
†URL:www.space-track.org
‡URL:https://celestrak.com

Frame), a rotating, Earth-Centered, Earth-fixed frame. Because this transformation is between an inertial and rotating coordinate frame, the angular velocity of Earth must be considered in the velocity transformation,[18] resulting in:

$$\begin{aligned} \boldsymbol{r}_{ITRF} &= W(t)\boldsymbol{r}_{PEF} \\ \boldsymbol{v}_{ITRF} &= W(t)^T\left(\boldsymbol{v}_{PEF} - \boldsymbol{\omega}_{Earth} \times \boldsymbol{r}_{PEF}\right), \end{aligned} \tag{2}$$

where $W(t)$ is the terrestrial polar motion matrix and $\boldsymbol{\omega_{Earth}}$ is the Earth's angular velocity vector, given by:

$$\boldsymbol{\omega}_{Earth} = (0, 0, 7.292115146706979E-5\left(1 - \frac{LOD}{86400}\right))^T. \tag{3}$$

Here, LOD, which can be found with the other EOPs, is the instantaneous rate of change in seconds of UT1 with respect to UTC.

The terrestrial polar motion captured by $W(t)$ is dependent on the parameters x_p, y_p, and s'. These parameters account for nutations with periods of less than two days in the GCRF as well as contributions from ocean tidal and libration effects. Values for x_p and y_p at any given epoch are based on observations, and as with the other Earth Orientation Parameters (EOPs), can be obtained from the IERS (International Earth Rotation and Reference Systems Service) website*. The parameter s' can be computed by:[19]

$$s' = -\frac{1}{2}\int_{t0}^{t} \dot{x_p} y_p - x_p \dot{y_p} dt, \tag{4}$$

which has a rate of change on the order of 10^{-4} micro arc seconds per century and has a current value of $s' = -47$ micro arc seconds.[20] The final matrix, $W(t)$, can then be computed from:

$$W(t) = R_2(-x_p)R_1(-y_p)R_3(s' - \frac{x_p y_p}{2}), \tag{5}$$

where R_1, R_2, and R_3 are the rotation matrices about the 1, 2, and 3 axes respectively.

For the computation of residual errors, the transformation between the GCRS and ITRS is also required. This transformation can be written as:

$$\boldsymbol{r}_{GCRS} = Q(t)R(t)W(t)\boldsymbol{r}_{ITRF}, \tag{6}$$

where $Q(t)$, $R(t)$ are the transformation matrices corresponding to the motion of the celestial pole in the ICRS (International Celestial Reference System) and the rotation of the Earth around its polar axis. The familiar $W(t)$ corresponds to the motion of the celestial pole in the ITRS as before. The matrix $Q(t)$ is composed of the B, P, and N (bias, precession, and nutation) matrices. For a given TT (terrestrial time) date, the X and Y coordinates of the Celestial Intermediate Pole and the CIO (Celestial Intermediate Origin) locator, s, using the IAU 2000A precession-nutation model are computed.[20] Next, corrections from the EOPs are added to X and Y. Then, the corrected X, Y, and s are used to form the celestial to intermediate-frame-of-date matrix, $Q(t) = BPN$, which has the form:

$$Q(t) = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} R_3(s), \tag{7}$$

*URL:https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html where $a = \frac{1}{2} + \frac{1}{8}(X^2 + Y^2)$. The Earth rotation angle is then computed as a function of the date in UT1 and is given by:

$$\theta_{ERA}(T_u) = 2\pi(0.7790572732640 + 1.00273781191135448T_u), \tag{8}$$

where T_u is the difference between the UT1 Julian date and 2451545.0. The Earth rotation angle is then used to compute $R(t) = R_3(\theta_{ERA})^T$, accounting for the rotation of the Earth about its polar axis. Finally, the velocity in the GCRF accounting for the Earth's rotation is given by:

$$\boldsymbol{v}_{GCRF} = Q(t)R(t)\left(W(t)\boldsymbol{v}_{ITRF} + \boldsymbol{\omega}_{Earth} \times R(t)^T Q(t)^T \boldsymbol{r}_{GCRF}\right). \tag{9}$$

In order to generate sufficient training data, all available SLR data for Lageos 1 was utilized. Additionally, every available TLE within the range of dates covered by the SLR data was downloaded and converted to GCRF position and velocity coordinates. For each TLE observation, the SLR "truth" state closest to the TLE observation epoch was likewise converted to GCRF position and velocity coordinates. This nearest truth state was then propagated to the TLE epoch using the `THALASSA`* orbit propagator,[21] which when modified to incorporate the effects of the Earth's polar nutation and precession, has been shown to yield extremely small errors even over multiple-month propagations (Figure 1, Table 1). Because the SLR-based trajectories used in this work have an extremely high temporal resolution, the `THALASSA` propagations used to align the epochs of the SLR and TLE observations of Lageos 1 have maximum propagations time spans of two seconds. Therefore, the miniscule error introduced by the numerical integration process will have virtually no effect on the accuracy of the generated residuals. Between the Modified Julian Dates of 57453.5 and 59033.7, this procedure was used to generate 5014 residual data points for Lageos 1 and 1576 for Lageos 2.

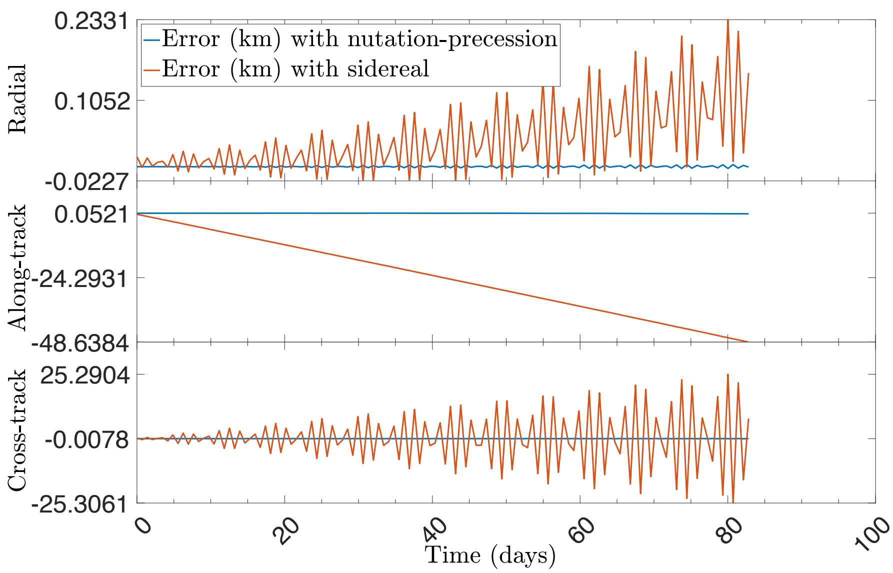

Figure 1. A comparison of the radial, cross-track, and along-track residuals in the predicted position of Lageos 1 after twelve weeks of propagation with THALASSA including nutation-precession in the calculation of drag and gravitational potential and only considering sidereal rotation.

*URL: `https://gitlab.com/souvlaki/thalassa`

Table 1. RMS errors after twelve weeks of propagation of Lageos 1 using `THALASSA` with and without accounting for polar nutation-precession.

	RMS r (km)	RMS s (km)	RMS w (km)
Nutation-precession unmodeled	0.0739	28.1057	10.8734
Nutation-precession modeled	0.0015	0.06931	0.0194

NEURAL NETWORK STRUCTURE AND GAUSSIAN BASELINE

Neural Networks

Artificial Neural Networks (ANN) are machine learning techniques that can approximate any Borel measurable function given at least one hidden layer with a sufficient amount of hidden units.[22] Input to the model is passed through a chain of weighted connections that are followed by nonlinear activation functions. During the training process, the loss calculated between the ground-truths and the estimations of the neural network is minimized by iteratively updating the network's weights and thus, refining its predictions.

A specific type of ANN is the Recurrent Neural Network (RNN). RNNs carry temporal information of the data in their hidden states which are updated and passed on through time. Long Short-Term Memory (LSTM) networks[23,24] are further advancements of vanilla RNN that are composed of an inner memory cell storing information over time and several so-called gates managing the flow of information running through the network by determining the values to be forgotten, the ones to be updated or replaced as well as the ones to become part of the output.

For our purposes, we employ two models, a fully connected neural network (FCNN) and an LSTM-based network that take each a vector of 3×1 of TLE state, (x, y, z) for the former (v_x, v_y, v_z) for the latter, as input and predict the corresponding TLE error of the input state. The FCNN consists of seven hidden layers (3×1024, 1024×512, 512×256, 256×128, 128×64, 64×32, 32×3) where every layer except the output layer is activated by tanh functions. For the LSTM model, we combine three LSTM-layers (256, 64, 32) after which the information is split into three separate paths, one for each velocity direction, composed of three linear connections.

Gaussian Baseline

It is of great importance to have a baseline to compare to the results from the neural network. Because TLE error is the result of random measurement error as well as deterministic methods of orbit determination, no purely stochastic model could perfectly predict the error of all TLEs. Therefore we employ a Gaussian model as the baseline with the expectation that it will only partially capture the TLE error. Despite this, it is of interest to note, that the Gaussian model performs relatively well when compared to the distribution of true TLE error as seen in Figure 2. Here, the error in the TLE's Cartesian x-coordinate is calculated using the SLR truth data. This "true error" is then used to determine the Gaussian, baseline model for the x-coordinate error based on MSE (mean squared error) estimation. The two parameters (μ, σ) of the Gaussian model are calculated by:

$$\mu = \sum_{i=1}^{n} \frac{x_i}{n}, \tag{10}$$

and

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n}. \tag{11}$$

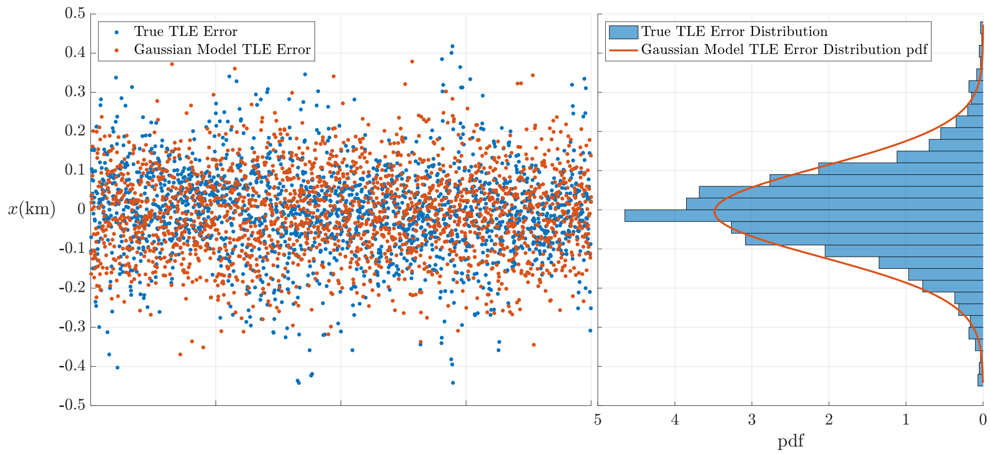

Figure 2. Modeling Lageos 1 (NORAD 8820) TLE errors as a Gaussian stochastic process. Left: true TLE errors and errors generated from Gaussian Model. Right: true TLE errors distribution and Gaussian model based on MSE (mean squared error) estimation.

The TLE error of the other Cartesian coordinates (y, z, v_x, v_y, v_z) in the case of Lageos 1 case, have distributions similar to that of Figure 2, where the Gaussian modeled error distribution is close to the true error distribution. Therefore, we are expand this Gaussian modeling method into a general baseline creation model for all three Cartesian coordinates and their derivatives.

ANALYSIS

Training Process

The data for Lageos 1 is split into 80% of training data, used to train the networks and calculate the parameters of the Gaussian model, 20% of validation data that is operated on to find suitable hyperparameters and 20% of test data. A ReLU layer is added to the output of the LSTM-based model at validation and testing time to enforce non-negative output. During training, we use MSE as a loss function where one batch of batch size 1 is fed into the network for 100 epochs. The TLE state parameters are normalized by:

$$\frac{x - \mu}{\sigma}, \tag{12}$$

using their respective mean and standard deviation before being fed into the models. Both network model's use SGD as their optimizing procedure at a learning rate of $3e - 4$ and a momentum of 0.1. Our experiments are fully implemented in python and specifically, PyTorch[25] is used to implement the ANNs. For our baseline comparison, the parameters shaping the Gaussian model are calculated using the training data set.

Results

Two testing procedures are conducted. The first one is based on the remaining test data of Lageos 1 and the second one on the entire data of Lageos 2. As for the Lageos 2 data, three outliers are removed, namely, the two states with the highest TLE error and the one with the lowest. To measure the success of our proposed models, we compare their predictions to the ground-truth by calculating their MSE. Similarly, we do this for the Gaussian model to see whether our models manage to beat the baseline.

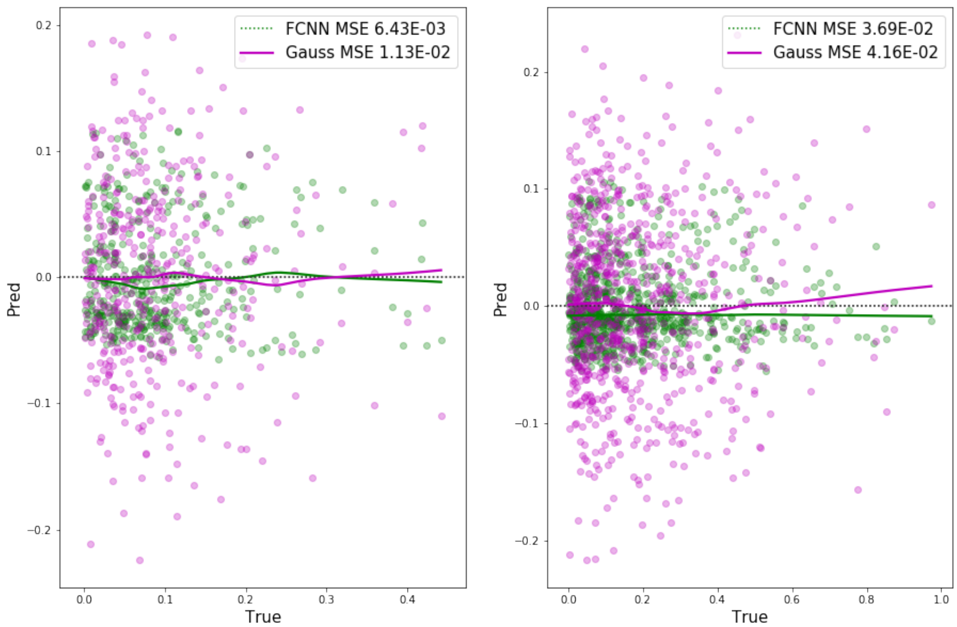

Figure 3. The predicted residuals of the FCNN model and Gauss process for the x-coordinate are plotted against the true TLE error. The left plot shows the test results for Lageos 1 data and the right one for Lageos 2.

Data	Model	x	y	z	v_x	v_y	v_z
Lageos 1	ANN	$6.43E-03$	$6.65E-03$	$9.65E-03$	$3.58E-08$	$2.12E-08$	$1.39E-07$
	Gauss	$1.13E-02$	$1.17E-02$	$1.84E-02$	$2.93E-08$	$2.51E-08$	$3.02E-08$
Lageos 2	ANN	$3.69E-02$	$3.32E-02$	$4.09E-02$	$2.02E-08$	$3.20E-08$	$5.11E-08$
	Gauss	$4.16E-02$	$3.83E-02$	$5.12E-02$	$3.12E-08$	$3.32E-08$	$2.94E-08$

Table 2. MSE results for all tested models on Lageos 1 and 2 data, where ANN stands for FCNN in the case of (x, y, z) and LSTM in the case of (v_x, v_y, v_z).

The results of the testing procedure of x and v_y is depicted in Figure 3 and 4 for both Lageos 1 and Lageos 2 and further summarized in Table 2.

Our FCNN manages to beat the baseline in both testing procedures. Especially for Lageos 1 data, the neural network is superior to the Gaussian. As for the LSTM model, it manages to yield better results as compared to our baseline for v_y predictions in both testing cases. The model does have a worse performance for v_x than the Gaussian tested on Lageos 1 data while achieving a lower

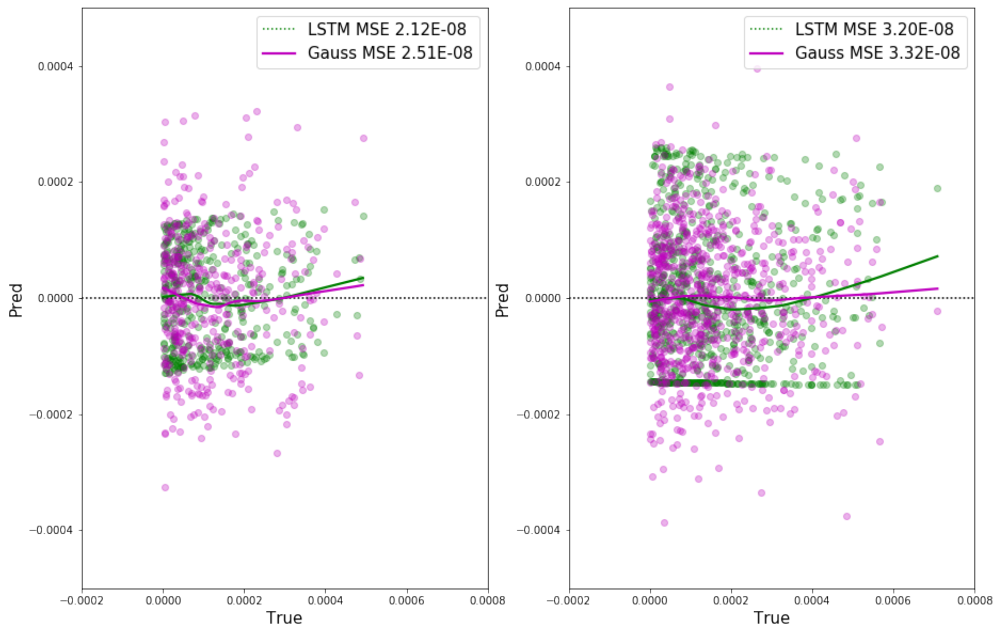

Figure 4. The predicted residuals of the LSTM model for v_y plotted against the true TLE error. The left plot shows the test results for Lageos 1 data and the right one for Lageos 2.

MSE in the case of Lageos 2 data. The LSTM-based model does fail to beat the baseline for v_z predictions.

Discussion

The results of the FCNN successfully show that we managed to construct a neural network capable of constructing a mapping from TLE states to TLE error. The FCNN yielded a better performance compared to the Gaussian baseline in all cases of (x, y, z) estimation.

Our proposed LSTM-based model manages to perform better estimating the v_y parameter but not for the others. By observing Figure 4 we can see that many of the predictions made by the model clustered around a specific value for both testing cases while being more present in Lageos 2 testing. This may be caused due to an incapability to recreate the true data distribution and thus, relying on close-to-mean predictions. Since the LSTM has to deal with many values that are coming extremely close to zero, the difference between estimation and ground-truth may have been too small to provide sufficient weight updates during training. An improvement would likely be reached when transforming the TLE error of the training set into a similar range as (x, y, z). We, however, rejected this approach for our experiments to investigate the performance of ANNs with raw TLE error data. In case of the LSTM model, further analysis is needed to construct a model capable of estimating (v_x, v_y, v_z) reliably.

With our models we successfully show that it is possible to employ neural networks for TLE state to TLE error mapping, especially for the coordinates of (x, y, z). Being the firsts to do so, we establish a baseline for future experiments and present a proof of concept. In upcoming research, we will investigate several techniques of pre-processing the data that could result in an improvement to better fit the neural networks' learning mechanisms. Moreover, we will probe the possibility of transforming the mapping from TLE state to TLE error into a classification problem, enabling a realm of well-researched strategies and metrics other than plain predictions.

CONCLUSION

In this paper, we propose a novel framework for TLE error modeling and quantification. This framework takes advantage of the coexistence of publicly available high accuracy orbital data (i.e., CDDIS SLR ephemeris) and the corresponding TLE data (i.e., Space-track TLE ephemeris) to incorporate deep neural networks into the prediction process of the error between these two sources. First, we revisited and clarified the coordinate frames and transformations between truth (SLR) and predicted (TLE) satellite ephemeris. Then, we introduced our method for generating training and testing data and presented our machine learning models as well as a baseline Gaussian model for comparison. Our results indicate that neural networks can be successfully used in a novel approach for estimating the error in TLE states and furthermore that the TLE error quantification can be converted from a dynamical systems problem to a data problem.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Davide Amato, David Vallado, and T.S. Kelso for their guidance as well as past work, which has made this study possible.

REFERENCES

[1] H. Peng and X. Bai, "Improving orbit prediction accuracy through supervised machine learning," *Advances in Space Research*, Vol. 61, May 2018, pp. 2628–2646, 10.1016/j.asr.2018.03.001.

[2] D. Racelis and M. Joerger, *High-Integrity TLE Error Models for MEO and GEO Satellites*, 10.2514/6.2018-5241.

[3] "Space debris by the numbers," `https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers`. Accessed: 2020-05-30.

[4] D. J. Kessler and B. G. Cour-Palais, "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt," *Journal of Geophysical Research*, Vol. 83, 1978, pp. 2637–2646.

[5] E. Delande, J. Houssineaub, and M. Jah, "Physics and human-based information fusion for improved resident space object tracking," *Advances in Space Research*, Vol. 62-7, 2018, pp. 1800–1812.

[6] T. Kelso, *Analysis of the Iridium 33-Cosmos 2251 Collision*.

[7] D. A. Danielson, D. Canright, D. N. Perini, and P. W. Schumacher, "Paper AAS 99-366 THE NAVAL SPACE COMMAND AUTOMATIC DIFFERENTIAL CORRECTION PROCESS,"

[8] C. Levit and W. Marshall, "Improved orbit predictions using two-line elements," *Advances in Space Research*, Vol. 47, Apr. 2011, pp. 1107–1115, 10.1016/j.asr.2010.10.017.

[9] J. Hartikainen, M. Seppanen, and S. Sarkka, "State-Space Inference for Non-Linear Latent Force Models with Application to Satellite Orbit Prediction," *arXiv e-prints*, June 2012, p. arXiv:1206.4670.

[10] S. Sharma and J. W. Cutler, "Robust Orbit Determination and Classification: A Learning Theoretic Approach," *Interplanetary Network Progress Report*, Vol. 42-203, Nov. 2015, pp. 1–20.

[11] R. Abay, R. Boyce, M. Brown, and S. Balage, "Two-Line Element Estimation From Space Catalog Using Deep Neural Network," *42nd COSPAR Scientific Assembly*, Vol. 42, July 2018, pp. PSD.1–23–18.

[12] T. Flohrer, H. Krag, H. Klinkrad, B. Virgili, and C. Frueh, "Improving ESA's Collision Risk Estimates by an Assessment of the TLE orbit Errors of the US SSN Catalogue," 01 2009.

[13] WGC-ARAIM, "ARAIM Technical Subgroup Milestone 2 Report," 2015.

[14] WGC-ARAIM, "ARAIM Technical Subgroup Milestone 3 Report," 2017.

[15] R. T. C. f. Aeronautics (RTCA) Special Committee 159, "Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment," 2001.

[16] D. A. V. e. al., *Revisiting Spacetrack Report 3*.

[17] D. A. Vallado and P. Crawford, *SGP4 Orbit Determination*.

[18] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Hawthorne, CA: Microcosm Press, 2013.

[19] N. Capitaine, "The Celestial Pole Coordinates," *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, 1990, p. 127–143.

[20] G. Petit, Brian Luzum, "IERS Conventions (2010)," 2012. IERS Technical Note. No. 36).

[21] D. Amato, C. Bombardelli, G. Baú, V. Morand, and A. J. Rosengren, "Non-averaged regularized formulations as an alternative to semi-analytical orbit propagation methods," *Celestial Mechanics and Dynamical Astronomy*, Vol. 131, 2019, p. 21 (38 pp.).

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359 – 366, https://doi.org/10.1016/0893-6080(89)90020-8.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, Vol. 9, 12 1997, pp. 1735–80, 10.1162/neco.1997.9.8.1735.

[24] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, Vol. 2, Sep. 1999, pp. 850–855 vol.2, 10.1049/cp:19991218.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.