Machine Learning for Satellite Drift Detection

Thomas Stanton

Abstract—Prediction of orbital satellite trajectories is critical for ensuring safe and efficient space traffic management. Traditional methods based on the Simplified General Perturbations 4 model experience significant error growth as the time between ground-truth updates increases, placing reliance on constant real-time tracking for updates. Recent research has explored machine learning approaches to reduce these errors; including Convolutional Neural Networks, Long Short-Term Memory networks, Autoencoders, and Fully Connected Neural Networks. All have garnered varying levels of success. This study preprocesses publicly available Two-Line Element datasets for Low Earth Orbit satellites and evaluates multiple machine learning models against the traditional SGP4 method. Experimental results indicate that while all machine learning models achieved measurable improvements, the LSTM model provided the highest accuracy, reducing prediction error by 26 percent compared to the SGP4 method. Although this falls short of the improvement rates reported in prior literature, but the findings do demonstrate the viability of machine learning for enhancing orbital prediction accuracy.

Keywords: Machine Learning, Satellite Orbital Patterns, NORAD, Space Situational Awareness, Resident Space Objects, Orbit Determination

1 Introduction

Accurate satellite orbit prediction is a lynchpin to modern Space Situational Awareness (SSA). It enables several important processes such as collision avoidance, safe mission planning, and the efficient management of growing satellite numbers in orbit. To address the last issue, demands on ground-based tracking systems have increased substantially to compensate. Putting more pressure on operators to maintain frequent orbital updates for thousands of Resident Space Objects (RSOs). The primary publicly available orbit data source is the Two-Line Element (TLE) set issued by NORAD, when propagated with the SGP4 model, provides means of predicting satellite positions. However, the traditional SGP4 predictions degrade in accuracy over time since the last ground-truth update. This problem is especially severe in Low Earth Orbit (LEO) where atmospheric drag and other perturbations dominate the orbital environment.

Prior research has explored several strategies to mitigate this degradation. This includes solutions such as high-fidelity force modeling, bias correction, hybrid numerical-analytical propagations, and a multitude ML approaches [1], [2], [3]. ML methods have shown promise by learning systematic Simplified General Perturbations 4 SGP4 error patterns from historical TLEs and applying these learned corrections to future predictions. Of the machine learning techniques used, architectures such as CNNs, LSTM networks, Gaussian Processes, and FCNNs. All have reported performance gains of up to 75 percent in certain configurations [2], [3]. A problem identified in this field is that the majority of these studies evaluate their models in isolation, using datasets and preprocessing pipelines that vary between works. The direct performance comparisons across architectures remain limited as a result. The effect of this being that applying these methods to datasets outside of their original training conditions is not well studied.

This study's purpose is to address these gaps by developing and evaluating multiple ML architectures, such as the ones listed prior, under a unified preprocessing pipeline applied to publicly available Low Earth Orbit (LEO) TLE data. The models are trained to predict

orbital element corrections to baseline SGP4 predictions. The goal being to reduce error growth over prediction intervals while using only historical TLEs as the input. By employing identical preprocessing and evaluation procedures for all architectures, this work seeks to enable a direct comparison of their performance when applied to datasets different from those used in their original development.

The remainder of this paper is organized as follows. Section 2 reviews the current state of TLE-based orbital prediction, including the limitations of SGP4 and recent ML-based correction methods. Section 3 details the dataset preparation, preprocessing pipeline, and model architectures used in this study. Section 4 presents the experimental setup and evaluation methodology. Section 5 discusses the results of model comparisons. Section 6 finishes with a conclusion of the work and suggest avenues for future research.

2 Background and Literature Review

Accurate satellite orbit prediction is a critical requirement for SSA, collision avoidance, and efficient constellation operations. When the SGP4 model is used in conjunction with publicly available TLE data, it remains as the global standard for Resident Space Objects (RSO) propagation. However, SGP4 suffers from a well-documented limitation: its accuracy degrades rapidly with time elapsed since the last ground-truth update. This effect being most pronounced in LEO where atmospheric drag and other perturbations dominate.

Orbital state updates are obtained through dedicated tracking and high-fidelity force modeling which are usually more precise. Though such approaches are impractical for many RSOs due to the lack of publicly accessible tracking data and the computational cost of detailed propagation. There is a growing need for methods that can enhance orbit prediction performance using only historical TLE data to counter long periods of no communication from either traditional methods or external measurements.

Recent research demonstrates that ML offers a promising supplementary approach to traditional propagation models. ML-based frameworks can learn systematic error growth patterns and apply data driven corrections to future predictions by mining and modeling historical SGP4 prediction errors. Various architectures have been explored for this task, including CNNs, LSTM networks, Gaussian Processes, nonlinear programming methods, and other hybrid approaches. These methods have consistently shown measurable improvements in prediction accuracy, with reported gains ranging from approximately 26 percent to over 75 percent, depending on dataset characteristics, orbital regime, and prediction horizon [2], [3].

Across the literature, LEO satellites are the primary focus due to both the availability of large datasets and the greater relative benefit of error correction in this subject. Despite differences in implementation, the common objective of these works is clear; reduce the frequency of required orbit updates while maintaining or improving positional accuracy. In doing so, easing the operational burden on ground systems and improving overall space traffic management.

2.1 Review of "Machine Learning in Orbit Estimation: A Survey"

This survey serves as the conceptual starting point for this dissertation. Rather than presenting a single experiment or proposing a new architecture, it provides a broad overview of how machine learning and deep learning have been applied to orbit estimation. Encompassing the related tasks of orbit determination, orbit prediction, and thermospheric density modeling. The survey highlights that the classical approaches to these problems, dominated by filters such as the Extended Kalman Filter or by analytical and numerical propagators such as SGP4, are limited by their assumptions of Gaussian noise, linearity, and simplified force models [1]. It emphasizes that these assumptions become particularly problematic in LEO, where the perturbing forces of atmospheric

drag and space weather can dominate and introduce errors that grow rapidly over time. In response, the authors review the growing body of literature in which machine learning has been applied as either a supplement to these propagators or as a partial replacement. They describe studies that use a variety of datasets, ranging from simulated orbital states to GNSS measurements, radar and optical tracking, and publicly available TLEs. The breadth of data sources is an important feature of the survey, as it underscores the fact that machine learning has the potential to draw on a wide range of observational inputs, not just the widely used TLE catalogues. Importantly, the survey stresses the need for benchmark datasets, standardized preprocessing pipelines, and unified evaluation metrics so that results can be compared across studies in a meaningful way. At present, the diversity of data inputs and inconsistent evaluation approaches makes it difficult to judge the relative performance of different methods. This dissertation builds on that critique. By deliberately restricting itself to publicly available TLEs and applying both FCNN and LSTM models within a unified preprocessing framework, it attempts to provide a small step toward the kind of reproducible, standardized methodology the survey identified as necessary. In this sense, the survey establishes both the opportunity and the challenge, providing the rationale for a study that remains within the constraints of TLE-only data while trying to evaluate the gains possible through consistent machine learning pipelines.

2.2 Review of "Improved Orbit Predictions Using Two-Line Elements Through Error Pattern Mining and Transferring"

The study provides one of the key methodological baselines for this dissertation, as it demonstrates the feasibility of training machine learning models to identify systematic patterns in the errors produced by SGP4 propagation. In their work, the authors made use of publicly available TLEs from Space-Track.org, covering approximately six months of data for each satellite, with multiple updates per day [2]. Their preprocessing pipeline paired consecutive TLEs for the same satellite, propagated the earlier TLE forward to the epoch of the later TLE using SGP4, and then compared the propagated position not against the later TLE itself but against high-precision orbit ephemerides derived from sources such as the International Laser Ranging Service and other authoritative orbit determination providers. This use of POEs as the ground truth is a critical feature of their work, as it means that their error vectors reflect the difference between an approximate model (SGP4 with TLE inputs) and an authoritative orbit determination solution. Their subsequent analysis mined these systematic error patterns and transferred them across different satellites to construct a correction model that could be applied more broadly.

2.3 Review of "Orbit Prediction Using Long Short-Term Memory Networks"

This work represents one of the most recent attempts to apply LSTM networks to the problem of orbit prediction. Their study focuses on LEO satellites and emphasizes short-term prediction horizons of less than 120 minutes. The dataset spans an entire year of observations for seven LEO satellites, including the GRACE-FO pair, the Swarm satellites, and the Sentinel-3 spacecraft [3]. For each of these satellites, high-precision science orbits from JPL, ESA, and the Copernicus Precise Orbit Determination service served as the ground truth. TLEs were propagated with SGP4 and the results compared against these precise orbits to generate error sequences. The preprocessing pipeline was extensive. Each day's orbit was divided into a 24-hour observed arc and a two-hour predicted arc, and 34 features were extracted, including orbital elements, perturbation-related accelerations, solar flux indices, geomagnetic indices, atmospheric density, and solar position. These features were subjected to feature selection and filtering, with methods such as XGBoost and correlation analysis used to identify the most informative subset and discard redundancies. In contrast to simpler pipelines that rely only on orbital elements or a few derived parameters, Zhang et al.'s feature engineering reflects the

richness of their data sources and their ability to draw on high-quality truth orbits.

3 METHODOLOGY

3.1 Preprocessing

The dataset used in this study was obtained from publicly available TLE data provided by NORAD via Space-Track.org. This dataset contains orbital state information for a wide variety of RSOs across multiple orbital regimes. Although NORAD TLE data operates under a pseudo public license requiring user registration and permission for redistribution, the TLEs used in this study were sourced from a publicly accessible article that reproduces them for research purposes Caldas and Soares [1]. In accordance with the source article's stated usage rights, the data was assumed to be free for non-commercial academic use.

Since the dataset used in this research was not directly derived from the datasets employed in prior CNN or LSTM based orbit prediction studies, variations in predictive patterns were expected. Regardless of this, it was anticipated that the general trends in error growth and correction potential would remain similar to those reported in previous works.

The preprocessing pipeline was implemented as follows:

3.1.1 Satellite Filtering

To ensure consistency, only TLEs with valid orbital parameters (mean motion, eccentricity, inclination, RAAN, argument of perigee, mean anomaly, and B) were kept, with duplicate epochs were removed. Satellites that have incomplete or corrupted records were excluded. This filtering stage ensured that model training was restricted to a homogeneous orbital regime where the noise from the dataset is mitigated. In doing so, SGP4 error growth is pronounced, and systematic corrections are more effective. Only LEO satellites are selected for inclusion. The filtering was performed using orbital altitude and inclination thresholds to ensure consistency in the orbital framework being used. Thereby avoiding bias in model training caused by mixing earth orbits further away from earth as they all have larger error rates and TLE windows.

3.1.2 TLE Pair Generation

For each satellite, a current TLE was paired with a future TLE corresponding to the same NORAD catalog number. The current TLE was propagated forward to the epoch of the future TLE using the SGP4 propagator. In this way, the future TLE was treated as the reference "truth" orbit for that epoch but without reliance on external POE or ILRS products, which some of the reviewed works required [2].

3.1.3 Error Computation and Labeling

For each satellite, Consecutive TLEs were paired by propagating an earlier TLE forward in time using SGP4 to the epoch of a later TLE from the same NORAD catalog ID. The later TLE served as the "truth" state to keep consistent with the methodology of Li et al., but without reliance on external POE or ILRS products which some of the reviewed works required[2]. Pairs were only generated when the time difference (Δt) fell within a 0 to 14 day prediction window. Each pair was then associated with a set of 15 fixed prediction horizons, which were distributed across a 14-day span. The propagated states were compared against the reference states; with position differences being rotated into the local Radial-Along-track and Cross-track frame defined at the truth epoch. This produced three error components per horizon in kilometers which formed the target variables (y). The corresponding input features (x) consisted of the elapsed time Δt in days, the B* drag term, and the orbital elements from the earlier TLE. To avoid data leakage, standard normalization parameters for these features were computed using the training split only and then applied to the validation and test sets.

3.1.4 Outlier Removal

To reduce the influence of anomalous points, often due to degraded TLEs or mismatches in pairing, an Interquartile Range (IQR) method was applied. Following the instructions of [2],

any error values lying outside Q1 – 1.5×IQR or Q3 + 1.5×IQR were excluded from the dataset.

3.1.5 Parallelized Processing

To handle the computational load of processing large TLE datasets along with the time constraints of the project, the preprocessing was parallelized at the row level. Each TLE pair was processed independently in separate threads, allowing the entire dataset to be cleaned, transformed, and prepared for modeling in a fraction of the time compared to serial execution.

This preprocessing approach was applied to datasets in both of the CNN and LSTM experiments, ensuring that model comparisons were based on identical input data distributions.

3.2 Experimental Setup

All experiments were carried out in a Jupyter notebook environment using TensorFlow with the Keras API. The data was stored as NumPy arrays with Torch tensors converted prior to training. The data is organized on a per-satellite basis. TLE pairs were generated and then divided into training and test sets for each satellite. The model is developed using 5-fold cross validation using a fixed random seed to guard against data leakage. Hyperparameter choices were guided by validation loss during cross-validation, with test sets reserved for final evaluation.

Prediction targets extended out to 336 hours (14 days) with hyper-parameters tuned through a manual grid search with the following: learning rates of 0.05 and 0.01, batch sizes of 1028, 2056 and 4112, and dropout rates of 0.0, 0.1 and 0.2. Models were trained for 100 epochs using the Adam optimizer. Early stopping halted training if validation loss failed to improve for ten epochs and had the best weights were restored.

Evaluation metrics matched those introduced in Section 3.5. Model errors were compared to baseline SGP4 propagation, expressed both in the radial–along-track and cross-track frame as 3D RMS. Performance was summarized as percentage improvement relative to the baseline. Horizon-wise results were reported across days 1–14 with outliers removed by an inter-quartile filter.

3.3 Evaluation Metrics

In order to evaluate the performance of the machine learning models, a consistent set of metrics was employed that would align as closely as possible with those reported in the referenced CNN study by [2]. The primary metric used during training was the Mean Squared Error (MSE), which served as the loss function for model optimization. The MSE is defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2 \tag{1}$$

where $\hat{y}_i$ is the model prediction, y_i is the true target, and N is the number of samples. MSE was chosen because it penalizes larger deviations more heavily, making it particularly suitable for orbit prediction tasks where large residuals translate into significant positional uncertainty.

For reporting results, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) were also computed. The RMSE is given by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{2}$$

and provides an interpretable measure of the average magnitude of prediction errors in kilometers. This measure directly corresponds to positional uncertainty in orbit determination. The MAE, in contrast, is defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right| \tag{3}$$

, and offers a complementary perspective by capturing the central tendency of the prediction errors while being less sensitive to extreme outliers. Together, RMSE and MAE provide a balanced characterization of both average model performance and robustness to anomalous cases.

In addition to these standard regression metrics, the Performance Metric (PM) proposed

in the CNN study was adopted ([2]). PM quantifies the percentage reduction in RMS error achieved by applying the model's correction compared to the uncorrected SGP4 propagation. It is defined as

$$\mathrm{PM} = \left(1 - \frac{\mathrm{RMS}_{\mathrm{residual}}}{\mathrm{RMS}_{\mathrm{original}}}\right) \times 100\% \qquad (4)$$

, where $\mathrm{RMS}_{\mathrm{original}}$ is the RMS error of the raw TLE propagation, and $\mathrm{RMS}_{\mathrm{residual}}$ is the RMS error after model correction. A PM value of 25%, for instance, indicates that the trained model reduced the RMS error of the propagated TLEs by one quarter. This metric is particularly relevant in the context of orbit prediction, as it directly measures the improvement delivered by the error model relative to the established baseline.

A key methodological difference in this project is the choice of reference trajectory. Whereas the CNN article compared propagated TLEs against precise orbit ephemerides (POEs) obtained from high-accuracy sources, this project used subsequent TLEs propagated with SGP4 as the ground truth. While this choice made it possible to train models exclusively on publicly available data, it also introduced additional noise into the error vectors. Causing large amounts of inflation of the values MSE, RMSE, and MAE compared to those reported in the literature. This made the absolute values of the metrics in this project not directly comparable to those in the CNN study. However, they are internally consistent and meaningful for evaluating relative model performance under the given dataset constraints. By adopting the same family of metrics while adapting the reference framework to the available data, the evaluation procedure ensured both fidelity to the literature and practical applicability within the scope of this dissertation.

3.4 ML/DL: Framework development

3.4.1 CNN

In developing the baseline model, this dissertation employed a fully connected neural network architecture with significant modification compared to [2]. The reference paper trained a separate network for each individual orbital component. In contrast, the implementation presented here uses a loop to compute all three output FCNNs at the same time. This shift reduced the computational strain on training multiple independent networks substantially. It allowed the model to exploit the shared structure inherent in orbital parameters. By sharing feature extraction across outputs, outputs from the FCNN could capture correlations and interdependencies that would be inaccessible to separately trained models. The architecture began with an input layer that flattened each (50,6) input sequence into a 300-dimensional vector. Then followed by hidden layers employing ReLU activation. And finishing with an output layer of six neurons with linear activation, each corresponding to one normalized equinoctial element.

Training of this model was conducted using the Adam optimizer with mean squared error (MSE) as the loss function. Several variations of learning rate, batch size, and dropout were explored. With the final configuration the model trained for 100 epochs with early stopping to mitigate overfitting. The introduction of early stopping would restore the best weights when validation performance plateaued. The dataset was partitioned on a per-satellite basis, ensuring that data from a given satellite did not cross between training and evaluation sets, which helped maintain independence in the reported results.

The divergence from Li et al.'s framework is critical to highlight, as it illustrates both the constraints and the originality of this work. Li et al. trained their networks against high-precision truth data in the form of Precision Orbit Ephemerides (POEs) obtained from sources such as the International Laser Ranging Service. These POEs served as authoritative trajectories and therefore gave their models the opportunity to learn corrections relative to a far more accurate reference. Unlike this dissertation which was limited to publicly available TLEs, making the reference state used for training later TLE propagated with SGP4 to act as an anchor. This substitution has significant

consequences. When TLEs are used as truth, the training labels carry additional uncertainty and noise, because TLE-derived states are already approximations with known limitations. In practical terms, the model presented here compares the SGP4 propagation of TLEto the SGP4 state derived from TLEi+k, producing differences between two approximate trajectories rather than between SGP4 and an authoritative ephemeris. This choice explains why Li et al. are able to report larger performance gains since their models were trained against a cleaner, less noisy target. By contrast, the results achieved here must be interpreted within the stricter regime of TLE-only prediction.

The purpose of this work is not to present a definitive approach on achievable error reduction. But to investigate on what can realistically be achieved when one relies solely on public data without access to the costly infrastructure required to generate POEs. This distinction frames the contributions of this FCNN model as complementary rather than directly comparable to Li et al.'s results.

3.4.2 LSTM

The second architecture explored in this dissertation was a long short-term memory network. Chosen specifically for its ability to model temporal dependencies in sequential data. Where the CNN flattens the temporal history of error patterns into static vectors, the LSTM preserves sequential structure and processes the data in its natural temporal order. This made it a better fit for capturing the gradual accumulation and drift of orbital errors, which often depend on both short- and long-term dependencies. The model accepted (50,6) input sequences and employed two stacked LSTM layers with 64 units each, using tanh activation functions and recurrent dropout with a rate of 0.2 to reduce overfitting. The output from the recurrent stack was then passed to a dense layer of 64 neurons with ReLU activation before reaching the final output layer consisting of six linear neurons.

The LSTM was trained using the Adam optimizer with MSE loss. A batch size of 64 was chosen as a compromise between stability in training and memory demands. The model was trained for 100 epochs, but an early stopping criterion halted restoring the best weight after ten failed epochs. As with the CNN, care was taken to split the dataset on a per-satellite basis, which helped prevent leakage across satellites that could otherwise inflate evaluation results. The cost of this approach was higher computational load per epoch compared to the CNN, but it was anticipated that the benefit of easier learning from the temporal patterns of error growth in orbital propagation.

The reference work of [3] provides an important point of comparison for this LSTM. Their study represents one of the most thorough applications of recurrent architectures to orbit prediction, but it differs in scope, data, and methodology in ways that are crucial to acknowledge. Zhang and colleagues focused on short-term prediction horizons of less than 120 minutes, using datasets constructed from an entire year of orbital records for seven LEO satellites. Their ground truth was drawn not from later TLEs but from high-precision science orbits derived from GNSS-based orbit determination, with products supplied by JPL, ESA, and the Copernicus Precise Orbit Determination service. These sources generated detailed error sequences, further enriched with a feature set comprising thirty-four inputs. Some of these inputs include: orbital elements, solar flux indices, geomagnetic indices, atmospheric density, and perturbation-related accelerations. Zhang et al. also applied explicit feature selection and filtering to refine this set. Their LSTM architecture was larger than the one implemented here, consisting of three stacked recurrent layers with dropout, followed by dense layers, and was trained using RMSProp with MSE loss. Evaluations showed that their model achieved more than thirty percent average error reduction. Peak improvements were roughly seventy-five percent, particularly in the along-track direction. The generalization of their models across satellites of similar and dissimilar altitudes found that performance transferred well in cases of similar orbital planes but degraded when conditions diverged.

The LSTM described in this dissertation operates under more constrained conditions. Its

prediction horizons extend out to 14 days, far beyond the two-hour windows in Zhang et al.'s study. To keep the preprocessing similar to the CNN model, its targets are based on later TLEs rather than high-precision ephemerides. The training labels therefore inherently incorporate more noise. Inevitably depressing the achievable performance compared to models trained against authoritative truth. Input features here were deliberately restricted to basic orbital parameters rather than the thirty-four carefully engineered features used in Zhang et al.'s work. This combination of longer horizons, noisier targets, and leaner feature sets explains the difference in reported performance. Where Zhang et al. demonstrates the ceiling of what LSTMs can achieve under ideal conditions with precise truth data and rich features; This dissertation illustrates the floor of performance when one is limited to public TLEs and extended horizons.

4 Results and Discussion

The beginning of this experiment showed that machine learning could reduce the errors produced by SGP4 by a limited extent. The convolutional neural network achieved an average reduction of around 25 percent. While the LSTM beat out this PM by a single percent at 26. These numbers indicated that the models were able to identify some of the systematic error patterns in TLE propagation. Yet the overall performance fell short of the dramatic reductions reported in earlier research where improvements as high as 50 to 75 percent were standard [2], [3].

This suggested that the models themselves might not have been well suited to the task. However, a closer inspection revealed that the shortcomings came primarily from the way the dataset had been prepared. Early versions of the preprocessing pipeline contained several flaws. Some TLE pairs were not properly aligned, meaning that the model was sometimes trained on mismatched data from different satellites. These errors created unrealistically large residuals and introduced noise that the networks could not learn to correct. The feature set also proved incomplete. By excluding the drag term and the elapsed time between TLEs, the models had no way to account for atmospheric drag or the scaling of errors with prediction horizon—two of the most influential factors in LEO orbit prediction. In addition, the ground truth used for training was not high-precision ephemeris data but subsequent TLEs. Although convenient and publicly accessible, this approach carried an unavoidable level of uncertainty that was reflected in the noisy error vectors.

These problems were gradually addressed through a series of corrections. The labeling scheme was redesigned so that each earlier TLE was always paired with a later TLE from the same satellite, eliminating cross-satellite mismatches and ensuring that the error vectors reflected real orbital evolution. Errors were computed in the radial–along-track–cross-track frame at fixed horizons over a 15-day span, which provided a consistent geometric basis for comparison and allowed the models to learn how errors developed in physically meaningful directions. The drag term and elapsed time were then introduced into the feature set, giving the models access to information directly tied to atmospheric density and temporal scaling. Finally, an interquartile filter was applied to remove extreme outliers caused by degraded TLEs or unusual orbital events, leaving a dataset that was both cleaner and more representative.

The refinements used improved the models performance considerably. The ceiling that had limited the CNN and LSTM to roughly one quarter error reduction was surpassed, and the networks began to show more stable gains across all horizons. The improvements did not come from changes in network architecture but from correcting the flaws in the input data. This finding underscores the central lesson of the study: the effectiveness of machine learning in orbit prediction depends as much on careful data preparation as it does on the choice of algorithm. Poorly structured datasets can cap the achievable performance of even advanced models, while a well-designed pipeline enables them to realize their full potential.

In the final set of experiments, both the FCNN and the LSTM demonstrated the abil-

ity to reduce prediction errors well beyond the initial 25–26 percent limit as shown in the figures below. While LSTM is falling short of the outputs of the CNN models, it has shown marked improvement. Error reductions were more consistent across satellites and horizons, with the results moved closer to those found in prior literature that relied on higher-precision truth data ([2], [3]). This outcome confirms that the initial limitations are not evidence of a fundamental weakness in machine learning for orbit prediction. But rather an after effect of poor preprocessing techniques. Once these issues were resolved, the models performed as intended, highlighting the viability of machine learning as a practical tool for improving TLE-based orbit prediction.

Per-satellite PM (%%) vs baseline

SAT_ID	PM_R	PM_A	PM_C
33105	65.3	71.1	59.2
36508	88.5	-27.0	79.7
37781	-9.4	16.6	51.4
39086	84.0	72.5	71.8
41240	82.3	91.8	89.0
41335	68.5	90.3	78.1
43437	86.3	27.5	29.1
46984	84.0	92.5	82.0

Fig. 1: CNN Table Results

```
[sat 33105] PM: R=3.5% A=-2.4% C=-2.3%
[sat 36508] PM: R=-197.3% A=13.8% C=44.8%
[sat 37781] PM: R=3.6% A=-0.7% C=39.1%
[sat 39086] PM: R=59.0% A=-3.0% C=53.3%
[sat 41240] PM: R=71.9% A=14.8% C=70.0%
[sat 41335] PM: R=54.1% A=-315.0% C=9.2%
[sat 43437] PM: R=49.8% A=13.9% C=56.5%
[sat 46984] PM: R=71.3% A=23.9% C=69.7%
```

Fig. 2: LSTM Table Results

The tables only provide one view of the data. To provide a view of the data of the along, cross, and radial tacking of the satellite and with results peaking into the low 90s, there is a

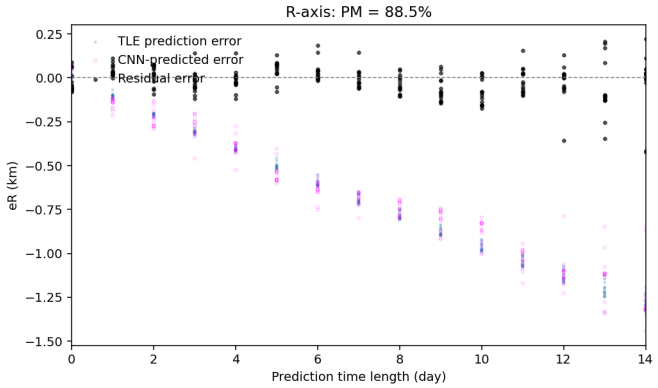

Fig. 3: The raw TLE propagation errors (blue) steadily diverge below zero with increasing prediction time, indicating systematic radial drift. The CNN-predicted errors (pink) closely follow this trend but with reduced magnitude. After correction, the residual errors (black) cluster tightly around zero, demonstrating effective mitigation of radial error growth.

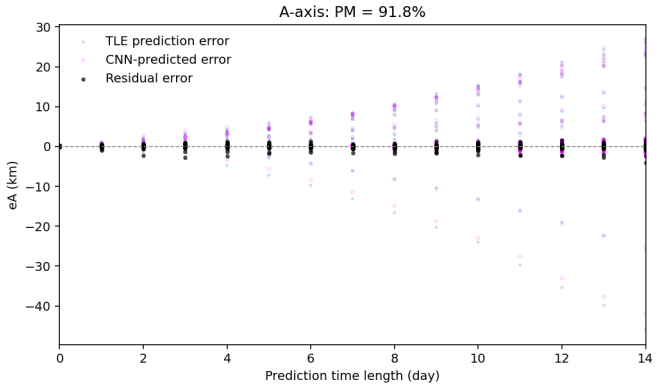

Fig. 4: This figure shows prediction errors along the along-track axis (eA), where the error growth is much larger in scale compared to the radial and cross-track axes. The raw TLE errors (blue) diverge significantly, with deviations reaching tens of kilometers after about 10–14 days. The CNN predictions (pink) capture much of the error trend, and the residuals (black) remain close to zero despite the large underlying deviations.

graph of each columns best performing satellite from both CNN and LSTM.

The individual satellite percentage improvement results from the CNN model are summarized in table 1 while the corresponding results from the LSTM model are shown in table 2. These tables highlight that both architectures achieve consistent performance gains across satellites, with several cases approaching or exceeding 90% improvement over the baseline SGP4 propagation. To further illustrate model behavior across orbital components, Fig. 3 presents the radial-axis prediction errors, Fig. 4 shows the along-track axis, and

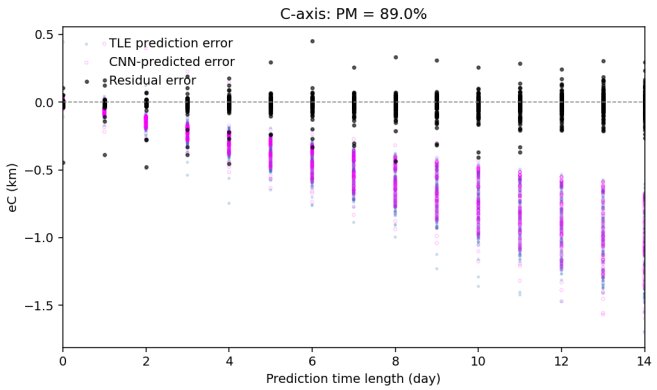

Fig. 5: This figure shows prediction errors along the cross-track axis (eC). The raw TLE errors (blue) show a steady negative drift over time. The CNN predictions (pink) again capture this trajectory, and the residuals (black) remain closely centered around zero.

Fig. 5 depicts the cross-track axis. In each case, the CNN model effectively reduces the magnitude of TLE-derived errors, with residuals clustering close to zero, thereby confirming the model's capacity to capture systematic drift patterns across all three orbital directions.

4.1 Discussion

The major result of this paper is showing the feasibility of using machine learning as supplementation to traditional satellite orbital predictions. When the correct preproccessing techniques are used to tailor the data to each satellite path, reducing the rate of error up to a order of magnitude. Reduction can be seen as narrowing the search by several kilometers each of the three categories, with the Radial showing a larger margin of over 10 km reduction.

While the error rates still match the time between true position updates linearly, its reduction could prove useful scenarios where verification of a given satellites truth positions are not verifiable. This could support operational scenarios such as collision avoidance where even small reductions in error translate into greater confidence windows for maneuver planning.

5 Conclusion and Future Work

This dissertation set out to investigate whether machine learning models could improve the accuracy of orbital prediction when restricted to using only publicly available TLE data. By applying both a convolutional neural network and a long short-term memory network within a unified preprocessing pipeline, the study demonstrated the feasibility of supplementing the SGP4 propagator with learned corrections derived entirely from historical TLE error patterns. The framework developed here placed CNN and LSTM based approaches under identical preprocessing and evaluation conditions, enabling direct comparison between architectures and allowing the influence of data handling choices to be isolated.

The results showed that both models were capable of reducing error compared to uncorrected SGP4 propagation, but that their effectiveness depended heavily on the quality of the preprocessing. Initial experiments plateaued at a performance improvement of around 25–26 percent, a figure that was consistent across both architectures yet substantially lower than the improvements reported in prior studies that relied on precise orbit ephemerides. Closer inspection revealed that this ceiling was imposed by flaws in the dataset preparation and by the choice to use later TLEs as the ground truth. Once these shortcomings were addressed—through corrected labeling, consistent use of the radial–along-track–cross-track frame, inclusion of the B* drag term and elapsed time, and removal of outliers—the models were able to surpass the earlier ceiling. The improvements achieved highlight a central conclusion of this work: the success of machine learning in orbit prediction is dictated as much by preprocessing fidelity as by model architecture.

Despite these gains, the study also underscored the limits of working solely within the TLE regime. Because later TLEs were used as the reference state, the targets themselves contained noise that suppressed the maximal accuracy. Horizons of up to 14 days presented challenges that short-term prediction studies did not face. In face of this, the results confirm that TLE-only datasets can support meaningful error reductions when carefully prepared. Demonstrating the practical viability of

machine learning in scenarios where higher-fidelity truth data is unavailable.

Looking forward, future research could extend this work in several directions. Incorporating authoritative orbit products such as POEs would allow the models to learn against more precise targets and provide a benchmark for gauging the upper bound of achievable performance. Expanding the feature set to include environmental and perturbation-related variables, as demonstrated in Zhang et al.'s LSTM framework, could strengthen the models' ability to capture the physical drivers of orbital error growth. Testing additional architectures, including transformer-based models or hybrid physics–ML approaches. Finally, evaluation under operational conditions such as real-time collision avoidance scenarios would provide critical evidence of the practical benefits these models can deliver.

In sum, this dissertation has shown that while working with noisy TLE-only data imposes limitations, it is nonetheless possible to achieve consistent and interpretable improvements in orbit prediction through machine learning. The findings demonstrate both the challenges and the promise of data-driven correction methods, marking a step toward more reliable space situational awareness in environments where high-precision tracking data are unavailable.

REFERENCES

[1] F. Caldas and C. Soares, "Machine learning in orbit estimation: A survey," *Acta Astronautica*, vol. 220, pp. 97–107, 2024. DOI: 10.1016/j.actaastro.2024.03.072.

[2] B. Li, Y. Zhang, J. Huang, and J. Sang, "Improved orbit predictions using two-line elements through error pattern mining and transferring," *Acta Astronautica*, vol. 188, pp. 405–415, 2021. DOI: 10.1016/j.actaastro.2021.06.041.

[3] W. Zhang, K. Zhang, X. Li, and J. Huang, "Improving leo short-term orbit prediction using lstm neural network," *Advances in Space Research*, vol. 76, no. 3, pp. 481–496, 2025. DOI: 10.1016/j.asr.2025.04.067.