SCRAM Final Report

Gabriella Armijo, Adelina Baca, and Emlee Taylor-Bowlin

Monte Del Sol Charter School

1 Executive Summary

There are about 25,800 man-made objects in space, with 5,400 of these objects being functional satellites, and more being added every year. With so many artificial items up in space, how do we keep track of it all and make sure there are no collisions? Even the slightest collision can damage the fragile equipment on board these satellites and lead to the satellite becoming useless. We have programs, such as the SGP4 and SDP4, which predict the orbital paths of satellites but they have their flaws. In this model, we address these issues and create a machine learning that can more accurately predict the orbital paths of payloads.

SCRAM (Satellite Collision Reduction and Avoidance Model) is an AI-based on the K-Nearest Neighbors (Peterson) model. It is trained off of Two-Line Element data sourced from spacetrack.org.

Current accepted physics-based models have issues that make them problematic to use. The main issue is that none of these models use more than one past data point in its prediction. To improve the prediction technique and provide a more accurate orbital path, we created a model that considers multiple past data points and uses them in the prediction generation.

We found that SCRAM can predict the orbital path of a Low Earth Satellite (LEO) with an average accuracy of 92

2 Motivation and Plan of the Work

The goal of this project is to more accurately predict the orbital patterns of Low Earth satellites using past points of data in comparison to the SGP4 model. We hope this will help better avoid collisions in space.

3 Background Research

3.1 Orbits

Space collisions are likely to occur at any time and can happen in any of the orbit paths that are located in the Earth's atmosphere. There are different orbits that occur around Earth that vary on how close or far away they are. Low Earth orbit or LEO is in the lower end of possible orbits around the Earth, which can be around 1,200 miles. Then there is Middle Earth Orbits and the difference between all of these orbits is how close they are to the earth and how long it takes an object like a satellite to complete its orbit. Objects in LEO orbits go around the Earth in roughly 90 minutes. An object in MEO can take about . . . , while an object in GEO takes 24 hours.

3.2 Collisions

Most satellites are in a LEO orbit path causing this area of space travel to be highly congested. Which in itself poses a significant threat of debris-generating collisions that can have high impacts on surrounding satellites in LEO orbits. Any collision that occurs has the potential to create thousands of shards of debris that create an uncontrollable mass of floating debris and make the already congested orbits more congested ultimately leading to more congestions.

3.3 Kessler Syndrome

Kessler syndrome(Adilov) is the theoretical scenario that predicts that a single satellite collision would cause a chain reaction that would cause all satellites in orbit to collide with each other. This mass destruction would lead to the downfall of human technology. Not only this but, it would cause thousands of pounds of space debris to either be launched into the universe or come crashing down to Earth's surface.

3.4 Current Solutions

None of this would be great, which is why it is regulated that any satellite that is launched into orbit must have a booster system to correct its orbit if there is a chance of there being a collision. This is great however, many older satellites that are still in commission don';t have these technological systems posing more problems which is why we have created this model to aid in this process.

4 Models and Methods

4.1 Introduction to Recurrent Neural Networks and Convolutional Neural Networks

Both Recurrent Neural Networks (RNN) and Convolutional neural networks (CNN) models have the capabilities to return many inputs to many outputs. They are commonly used models and often considered fundamental models in machine learning. This leads them to both be flexible and powerful models with different strengths and weaknesses.

Convolutional Neural Networks are most commonly used for language processing. They are typically faster than other model types and use a feed forward network. This means that information never flows backwards and impacts previous nodes. CNN models excel at processing data that is grid like. This refers to data like audio and images that have to be broken up into matrices to account for color and pixel size or wavelength and frequency.

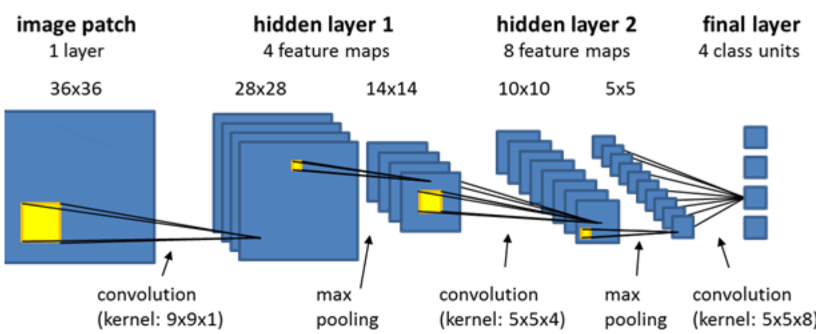

Fig. 1. An example of the data flow in a CNN model
Image borrowed from:

Recurrent Neural Networks however use a bi-directional network so information flows forward and backwards. RNN models also have long short term memory so they can learn and identify trends in large data sets. This makes RNN'S better for pattern recognition and numerical predictions. RNN models are most commonly used with large packets of data and through data processing. The most well known application of RNN is its use in handwriting recognition.

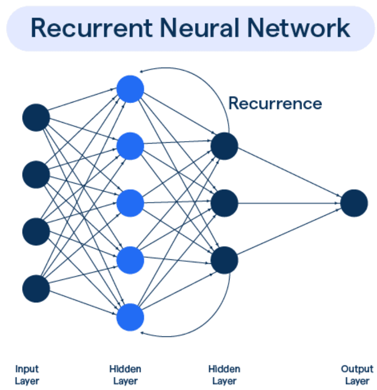

Fig. 2. An example of the data flow in a RNN model
Image borrowed from:

4.2 Implementation of our Model

We decided to use an RNN model because of the sheer size of our satellite data. We feed in different columns with information about the location, speed and orbit of the satellite. SCRAM takes in a total of 13 columns and 181,936 rows. This information is all fed into the model from Peewee queries and then turned into numpy arrays with a 32 float value so they can be turned into tensors and processed.

The data is queried out by odd and even rows. The odd rows are the current location of the satellite and the even rows are where SCRAM should predict them ending. Once this data is sorted into the input sequences and output sequences they are fed into the model class.

The model class establishes the hidden layer of zeros then starts to process the input sequences and output sequences. It sees the input row and then outputs what it guesses it should have been and compares it to the true output and then learns from there. After iterating through all the rows it reports its final loss as a percentage before beginning another iteration. It does a total of ten iterations.

5 Results

The results change drastically with different sizes of data. This is due to the fact that the model applies its own weights accordingly and the hyperparameters work better at different levels of data. We tried running the model at different levels of data, we tried this on different TLE files, and then the full 181,936 rows.

To analzyze the accuracy of our model, we found the mean of error and compared it to the SGP4 model. We calculate the mean of error value using Predicted values - the Actual values / by the Actual Values. This was done using a numpy array so that every predicted value correctly aligns with the actual value. When the data is run on, AGENA-TARGET_1966-065A, EXPLORER-4_1958-005A, and LEMUR-2-ONREFLECTION_2023-054E CSV files we get an average error of 31%. This means that almost 70% of the time SCRAM predicts accurately. Considering this is the results ran on only around 100 - 200 hundred lines this is an impressive feat.

When iterated over the full data set we collected SCRAM has an accuracy of 92%. This makes us confident that SCRAM not only works but also isn't over trained because it still has a bit of room for error.

Data Type	Mean of Error
1	42. 21
2	16.50
3	24.02
4	28.56
5	30.83
6	32.59
7	34.01
8	35.28
9	36.46
10	37.57

6 Conclusions

6.1 Takeaways

In conclusion, we were able to create an AI Model that is able to predict satellite positioning with an average accuracy of 70%. This project once we increase it's accuracy has the potential to replace current SGP4 models in predicting satellite positions and detecting the likelihood of collisions.

6.2 Limitations and Errors

With our current model due to the size of the data sets that we are using our model takes four and a half hours to process all the data. Given that our data set is 181,936 rows it is understandable why it would tak ea long time to process. However in satellite orbits in this same amount of time any given satellite in a LEO could have completed three orbits, which dramatically changes things. This is something that has held us back and we would like to improve the performance time so we can have our project predictions running on real time.

6.3 Future Work

Further refinements that we would like to make to our model include making it more user friendly, making it closer to a SGP4, and making our model more efficient. We would like to include to our program a user interface close to an SGP4 where users can input a few lines of data and our model is able to give the user quick results on the positioning of the satellites they inputed.

7 References

Adilov, Nodir, et al. "An Economic "Kessler Syndrome": A Dynamic Model of Earth Orbit Debris." Economics Letters, vol. 166, May 2018, pp. 79–82, www.sciencedirect.com/science/article/pii/S0165176518300818, https://doi.org/10.1016/j.econlet.2018.02.025. Accessed 8 Nov. 2019.

"An Economic "Kessler Syndrome": A Dynamic Model of Earth Orbit Debris." Economics Letters, vol. 166, May 2018, pp. 79–82, www.sciencedirect.com/science/article/pii/S0165176518300818, https://doi.org/10.1016/j.econlet.2018.02.025. Accessed 8 Nov. 2019.
Hoots, Felix, and Ronald Roehrich. SPACETRACK REPORT NO. 3 Models for Propagation of NORAD Element Sets. 1980.
SPACETRACK REPORT NO. 3 Models for Propagation of NORAD Element Sets. 1980. "Space-Track.org." Www.space-Track.org, www.space-track.org/documentation. "Space-Track.org." Www.space-Track.org, www.space-track.org/documentation.
Wikipedia Contributors. "Convolutional Neural Network." Wikipedia, Wikimedia Foundation, 27 Feb. 2019, en.wikipedia.org/wiki/Convolutional$_neural_network.$
\ConvolutionalNeuralNetwork."Wikipedia, WikimediaFoundation, 27Feb.2019, en.wikipedia.org/wiki/Co
\RecurrentNeuralNetwork."Wikipedia, WikimediaFoundation, 3Dec.2018, en.wikipedia.org/wiki/Recurre
− − .
\RecurrentNeuralNetwork."Wikipedia, WikimediaFoundation, 3Dec.2018, en.wikipedia.org/wiki/Recurre
\SimplifiedPerturbationsModels."Wikipedia, 17Oct.2020, en.wikipedia.org/wiki/Simplified$_p$erturbations
\SimplifiedPerturbationsModels."Wikipedia, 17Oct.2020, en.wikipedia.org/wiki/Simplified$_p$erturbations