Transformers for Orbit Determination Anomaly Detection and Classification

Nathan Parrish Ré
nathan.re@advancedspace.com

Matthew Popplewell
matthew.popplewell@advancedspace.com

Michael Caudill
michael.caudill@advancedspace.com

Timothy Sullivan
tim.sullivan@advanced-space.com

Tyler Hanf
tyler.hanf@advancedspace.com

Benjamin Tatman
ben.tatman@advancedspace.com

Kanak Parmar
kanak.parmar@advancedspace.com

Tyler Presser
tyler.presser@advancedspace.com

Sai Chikine
sai.chikine@advancedspace.com

Michael Grant
michael.grant@advancedspace.com

Richard Poulson
richard.poulson@advancedspace.com

Advanced Space, LLC
1400 W 122nd Ave, Westminster, CO 80234

Abstract

This paper presents the judicious application of machine learning algorithms to solve two fundamental challenges in spacecraft orbit determination (OD): identification of systematic anomalies (nominal vs anomalous behavior) and the classification of these anomalies to explain probable causes (such as unexpected spacecraft maneuvers or mis-modeled small forces). Traditional OD is based on well-tested iterative linearization methods (variations of the Kalman filter). These are commonly understood in the astrodynamics community to require manual tuning for a given set of mission assumptions and re-tuning when those assumptions are invalidated. OD data are typically long, sparse, multi-variate sequences consisting of multiple observation phenomenologies. These characteristics make the OD problem fundamentally well-suited to the same machine learning architectures (namely, transformers) that have found success with language modeling and other sequence-based data modeling. The approach taken here is to simulate various failure modes in the traditional OD process using NASA's Monte software, then process the simulated data in three different transformer-based architectures. The three transformer architectures all act as classifiers and can be described at a high level as: 1) treat each epoch with data as a feature vector, with the input data comprising a single, long sequence; 2) similar to the first, but with nested self-attention to efficiently handle longer sequences; and 3) plot the data, then classify the plot with a vision transformer (ViT) model. Model performance is studied as a function of the hyperparameter trade space.

1. Introduction

Tracking spacecraft through maneuvers, predicting maneuvers, and tracking trajectories with limited ground-based observability are some of the key challenges facing the in-space intelligence community. Current approaches to spacecraft tracking rely on human experts and regular spacecraft communications to guide standard algorithms toward the subjectively most reasonable solution given imperfect and incomplete information. Even with over 60 years of spacecraft orbit determination practice, hard-coded logic is insufficient to achieve complete autonomy. The hard-to-quantify "engineering intuition" remains an irreplaceable component of current practice. Machine learning is a natural fit for the core orbit determination problem: conditional probability estimation of nonlinear systems.

The goal of this investigation is to enhance efficiency in spacecraft orbit determination by reducing human-in-the-loop effort. To achieve this goal, this research employs supervised learning based on over one hundred thousand high fidelity simulated spacecraft trajectories, navigation measurements, and corresponding navigation filter results. Hundreds of years' worth of simulated data provide a broad basis for machine learning models to mimic the kinds of choices human operators make every day, distilling the data into operational algorithms and flight software. Machine learning may offer an advantage over human decision making because the training process for these algorithms can easily scale to include more example data than any individual engineer could see in a lifetime of practice. In addition, today's computer processors make it possible to evaluate these machine learning algorithms with increasing efficiency, enabling them to operate directly onboard spacecraft.

This paper presents the application of machine learning to solve two fundamental challenges in spacecraft tracking: identification of systematic anomalies (such as mismodeled forces or unexpected maneuvers), and classification of anomalies to explain probable causes of deviations from the expected motion. This technology aims to simplify spacecraft operations by freeing operators from the task of determining what happened when an anomaly occurs, and instead focus on why an anomaly happened. Providing insights into the source of a problem helps operators make educated decisions in real-time.

An example stressing case for traditional navigation methods is the near rectilinear halo orbit (NRHO) that is used by the CAPSTONE (Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment) mission and will be used by NASA's planned lunar Gateway. CAPSTONE, designed and operated by Advanced Space, launched on June 28, 2022. CAPSTONE is demonstrating the novel inter-spacecraft navigation technology CAPS (Cislunar Autonomous Positioning System) via cross-link with NASA's LRO (Lunar Reconnaissance Orbiter). CAPS uses relative range and range-rate between participating spacecraft and the time-varying, asymmetrical gravity field of the Earth-Moon system to generate absolute state estimates of all participating spacecraft without ground intervention. Robust automation is necessary for such a system to work at scale as intended, especially in chaotic N-body motion. The increasing traffic in cislunar space by commercial, scientific, and national security organizations points to the need for new approaches to spacecraft position-navigation-timing (PNT) and spacecraft autonomy.

Autonomous navigation is even more challenging onboard spacecraft where computation is constrained. Traditional algorithms are computationally intensive and quickly become infeasible in an onboard and operational environment. Machine learning shifts the computational heavy lifting to ground-based assets at training time and minimizes the requirements for onboard memory and compute capability. Anomaly detection and classification aims to extend superhuman domain knowledge to real-time space-based space domain awareness. The resulting increase in domain awareness for a spacecraft of its own and other spacecraft's states creates a substantial capacity for growth in autonomous decision-making.

Since artificial neural networks (ANNs) are primarily composed of simple linear algebra, current chips such as CPUs and GPUs can evaluate them very efficiently. Future computing hardware (ground-based or space-based, and whether GPU, ANN-optimized Application Specific Integrated Circuits (ASIC's), optical computing chip, or neuromorphic processor) will be able to use these algorithms even more effectively at orders of magnitude better performance per watt. The same innovations that currently allow consumer devices to process large data streams in real-time will, in the future, enable spacecraft to do the same, autonomously make intelligent decisions and achieve mission objectives that are impossible with current ground-in-the-loop systems. However, new algorithms must be developed to reformulate PNT mathematical problems into a form that can take advantage of these computer hardware advances. This research is intended to address that need.

2. Background

2.1. Spacecraft orbit determination

Traditional spacecraft navigation uses a series of limited observations of a spacecraft (for instance, instantaneous range and Doppler relative to a fixed ground station) to build up an estimate of the spacecraft state and/or other parameters of interest over time. This relies on careful use of linearization, for example, iteratively linearizing the spacecraft motion relative to an assumed trajectory and minimizing the sum of squared observation residuals. In the typical orbit determination parlance, "prefit residuals" refer to the difference between observed measurements and the expected measurements according to the *a priori* state estimate and dynamical model. "Postfit residuals" refer to the this difference after updating the state estimate and/or dynamical model according to the measurements. Postfit residuals in general correspond to a solution that locally minimizes the sum of squared residuals.

Measurement residuals are essentially the measurements detrended by the current best estimate of the spacecraft state and the forces acting on the spacecraft. Human operators typically use prefit and postfit residuals as one of the strongest indicators of filter success. If the residuals are normally distributed and centered at zero, that is evidence that the state estimate and dynamical model explain the motion of the spacecraft. If the residuals have some trend or signal present, that is evidence that either the force model or the state estimate have some error.

When the predicted behavior is well known, the filter is kept within its linearizable region. When the various uncertainties in the dynamical model are balanced correctly, these methods (i.e. variations of the batch filter and Kalman filter) work well. However, experienced space navigators know that navigation is as much an art as it is a science. Much of the tuning of a spacecraft navigation filter comes down to navigator experience and intuition. When anomalies occur (either on the ground side or the spacecraft side), experience is required to identify the unexpected behavior, and creativity is required to develop and test reasonable hypotheses to explain the observed discrepancies.

The proposed ML models primarily use the postfit residuals to infer what small force(s) are mismodeled in the filter dynamics. Additional data about the orbit and observer(s) are also provided to the models for context.

2.2. Anomaly detection

For most real-world anomaly detection datasets, a traditional linear method would misclassify a portion of each group of data. An ANN can be trained to manipulate the data in a higher-dimensional latent space, making it easier to separate classes. The ANN's utility grows as the dimensionality of the problem increases, given nonlinear and potentially hidden relationships in the state space.

Two broad approaches to training ANN models are supervised learning and unsupervised learning. In supervised learning the training algorithm is given labeled data and adjusts the model weights to get the model to classify inputs correctly. In unsupervised learning, the ANN learns with unlabeled inputs. ANN classifiers and outlier detectors have been demonstrated in applications such as medical diagnosis [1], battery monitoring systems [3], radar activity classification [21], and other areas [5], [31]. In this work, we primarily use supervised learning, where the training algorithm is provided true anomaly class labels for each input data sequence.

2.3. Transformer models

In this research, we use a recently developed deep learning architecture known as the Transformer [28]. Transformers have been found to be highly effective at modeling sequence data and were originally developed to solve problems within the Natural Language Processing (NLP) domain. Transformers were able to achieve state-of-the-art performance in machine translation, question answering, and language inference [10]. Following this success, the Transformer framework was extended to sequence problems with image [11] and audio data in the form of sequences of bits of a digitized stream [26], [8].

The original paper on Transformers, "Attention is All You Need" [28], describes the goal of the Transformer architecture as being to simplify the process for creating sequence-to-sequence (seq2seq) model architectures by relying solely on the attention mechanisms and removing recurrence and convolution operators entirely. This also removes the sequential processing constraints, allowing for highly efficient parallel architectures. These breakthroughs are made possible due to the sole reliance on an attention mechanism. The attention mechanism was first introduced by Bahdanau, Cho, and Bengio in their paper on neural machine translation [2]. Attention allows models to draw dependencies between inputs and outputs without regard to the distance between elements in a sequence. In simpler terms, attention mechanisms allow a model to look at an input sequence and decide which other parts of that sequence are important. When utilized within the Transformer architecture, the attention mechanism is referred to as self-attention, which relates different positions of a single sequence to compute a representation of that sequence [28].

Transformers have also been extended to complex data types including multivariate time series data [29], which is also the sequence data type that this research focuses on. Compelling examples in the literature have focused on both time series forecasting for physical systems, and time series classification tasks such as anomaly detection and classification [14],[6]. Zerveas et. al. present a novel Transformer-based framework for multivariate time series representation learning and use that framework to push the limits of state-of-the-art performance on both regression and classification tasks [30]. Work has also been done to improve the Transformer architecture to accommodate longer data sequences; for example, Long-Range Transformers [13] also achieved state of the art results on benchmarks such as traffic forecasting and weather prediction.

The Recurrent-Neural-Network (RNN) [9] structure and Long Short-Term Memory (LSTM) [15] networks were long established as state of the art for seq2seq modeling problems. However, these architectures were fundamentally constrained to process inputs one at a time, while Transformers process a batch of inputs simultaneously. As a result, Transformer models allow seq2seq modeling to scale up by orders of magnitude both in model size and sequence length. Transformers can effectively work with sequences that are thousands or even millions of elements long without the problem of extended gradients which must be traced back through a recurrent model. Thus, even large sets of time-ordered navigation measurements can be used effectively – constrained primarily by the memory and computation speed of the computers used for training and inference. Many of the tasks humans perform daily (perception, reasoning, predicting, decision making) can be naturally represented as seq2seq modeling problems.

Transformers are applied here to understand sequences of navigation observations. This research was inspired by considerable success with Transformer-based language models such as the GPT series [4], [24], [23], BERT [10], and variants thereof [16, 17, 19, 22, 25]. Natural language processing typically represents language as a sequence of vectors, where each vector represents a character, word part, or phrase. This data structure is similar to spacecraft navigation data, which, like many engineering data sources, is a multivariate, sparse sequence with uneven time steps. Observations of spacecraft can consist of multiple data types (such as range, Doppler, and angle in the sky relative to a ground station) each of which might be present at different times and cadences.

Transformers may be well-suited for time series data, but they have only recently been applied to time-series modeling problems. The reason that Transformers are particularly suited for time series is that they can represent each element in an input sequence by considering its context within the entire sequence. This ability to consider sequence context is the direct result of the attention mechanism. Concurrently, the addition of Multi-Head attention allows the model to consider different representations of the same element which can represent multiple aspects of relevance like periodicity [30]. There have been several research projects which have been able to exemplify the Transformers ability to work with time series data [29]. Ma et al. showed that an encoder-decoder Transformer could impute missing values in a multivariate time series and achieve superior results to state-of-the-art RNN models [20].

Building on the work of Ma et al., Zerveas et al. develop a unified framework for unsupervised representation learning with Transformer for multivariate time series data [30]. The framework built by Zerveas can achieve superior performance on both regression and classification tasks compared to traditional methods and RNN/LSTM methods. The datatypes included a wide range of multivariate series from the Monash University, UEA, and UCR Time Series Regression Archive [27]. The data includes regression time series with up to 24 dimensions and classification series with up to 963 dimensions that span different areas across science and engineering. Grigsby et al. also developed a separate Transformer architecture that can incorporate spatial relationships between variables as well as temporal ones with the Spacetimeformer model [13]. Finally Li et al. [18] incorporate a sparse-attention mechanism to work with longer time series sequences and prevent models from extracting irrelevant information.

3. Training data

The methods and results described in this paper used supervised learning with a large set of simulated trajectories. Approximately 20,000 examples were generated for each of five anomaly classes.

Orbit states were randomly chosen from a range of orbital elements: altitude between 400-4,000 km, eccentricity between 0-0.5, inclination between 0-180°, and all values of longitude of ascending node, argument of perigee, and true anomaly. Spacecraft mass was randomly sampled from 50-500 kg, drag coefficient between 1.6 and 3.0, and thrust for finite thrust maneuvers between 1-50 N.

Five anomaly classes are considered in this study:

- Drag – erroneous estimate of the spacecraft's coefficient of drag.
- Gravity – reduction in spherical harmonics degree and order in the estimation filter's dynamical model.
- Maneuver – erroneous finite thrust maneuver direction and magnitude estimate.
- Nominal – no dynamical or measurement mismodel present.
- SRP – erroneous estimate of the solar radiation pressure scale factor.

To build a set of samples, an initial orbit and corresponding set of spacecraft properties are randomly sampled, then propagated with a high fidelity force model for 24 hours. Each of the observation platforms records simulated measurements whenever the spacecraft is in view. Simulated measurements are collected from the following observers:

- 3 arbitrarily-placed ground stations collecting coherent (low noise) range/Doppler measurements.
- 5 arbitrarily-placed ground stations collecting noncoherent (high noise) range/Doppler measurements and Azimuth/Elevation optical measurements.
- 8 arbitrarily-placed spacecraft in MEO collecting optical Right Ascension/Declination measurements.

Realistic noise and biases are added to the simulated measurements, with different noise levels chosen for each measurement type. Finally, the *a priori* state estimate and set of measurements are iteratively processed through a Kalman filter (Monte's UD-factorized sequential batch filter-smoother [12]).

For each anomaly class, the filter is forced to fail in different ways. For example, the dynamical model in the filter is purposefully altered, or an estimated term is forced to an incorrect value by choosing an incorrect *a priori* estimate and a very small *a priori* variance on that term.

The following outputs are generated for each filter case:

- Epoch
- Measurement type
- Observing station name and state
- Estimated spacecraft state
- Pre-fit and post-fit residuals

Visual examples for two typical ANN model inputs are provided in Figures 1 and 2. Note that the magnitude of the signal varies, with some samples having strong signals and some having imperceptibly-small signals.

Classification models were set up to take in these outputs and return a probability distribution function across the five anomaly classes.

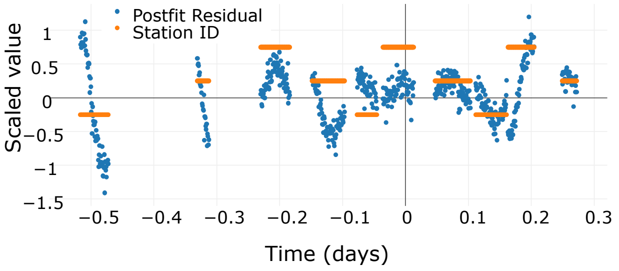


Figure 1. Example input data with SRP mismodel.

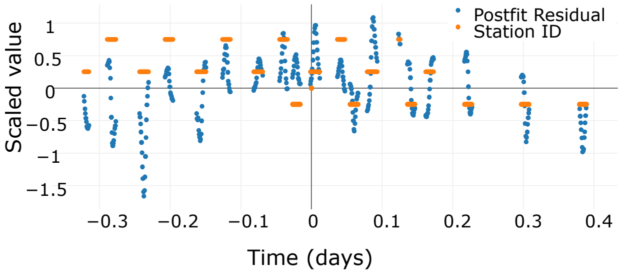


Figure 2. Example input data with gravity field mismodel.

4. Model architectures

All three architectures studied here use the "class token" idea first implemented by the Vision Transformer (ViT) architecture [11]. Similar to the ViT model, the input data sequence is prepended with a "class token" – a vector of zeros. The class token contains no information on the input, but when transformed by the model, the class token represents the anomaly category of the sequence of data. This was found to be an effective mechanism to get the models to compress a 2-dimensional input (time-ordered sequence of feature vectors) to a 1-dimensional output.

All three architectures take the same inputs and return the same outputs. Preprocessing and the model forward pass vary between architectures.

Unless stated otherwise, all inputs are scaled to have zero mean and unit variance. For all models, the Transformer encoder implementation is the unedited, vanilla torch.nn.TransformerEncoder block from PyTorch. All models use the Adam training algorithm. The output head of all models is a linear layer with a softmax applied. Model training uses the cross-entropy loss function, which interprets the model outputs as a probability distribution function (PDF) across the five output classes.

4.1. Measurement Transformer (MT)

The Measurement Transformer builds off the BERT architecture (Transformer encoder only, no Transformer decoder block) [10], adapted for time series data by using a time encoding instead of position encoding. The data flow of this model is illustrated in Figure 3 and described below:

1. Pre-processing:
   (a) Most of the data are scaled on a per-input-file basis. However, the ground station ID numbers are scaled globally (so the same ID number always refers to the same ground station across all training samples). Time scale is also global, so the time scale is consistent in every input file.
   (b) Input sets are padded or trimmed to equal length.
2. Model forward pass:
   (a) Use the Time2Vec algorithm [7] to expand the time element of the feature vector. This time encoding takes the place of position encoding typical in most other applications of Transformers.
   (b) Pass each feature vector through a linear layer to expand the length of the feature vector.
   (c) Prepend the time series with a "class token" vector of zeros.
   (d) Pass the time series through a Transformer encoder layer, which uses multi-head self attention to draw relationships between different elements of the sequence.
   (e) Pass the transformed sequence through another linear layer to reduce the length of the feature vector to 5 (the number of anomaly classes).
   (f) Finally, the 0^{th} vector of the sequence (the class token) is interpreted as the class of the anomaly. The rest of the sequence (elements 1 through N) are not used.

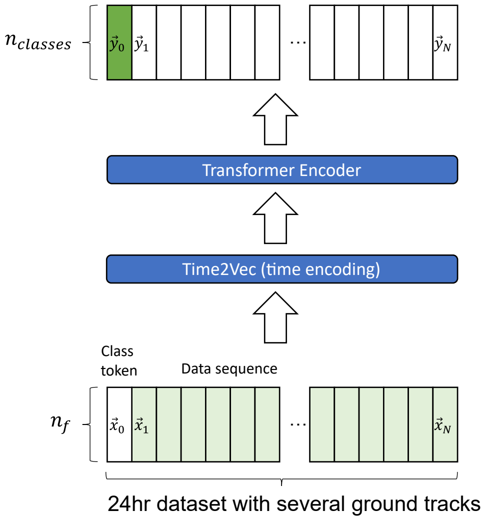


Figure 3. Conceptual diagram of the Measurement Transformer architecture.

4.2. Tracking Pass Transformer (TPT)

The TPT model is similar to the MT but uses two nested Transformer encoders instead of a single Transformer encoder. The motivation for this difference is that each ground track may have biases (in time tags, or in the measurements themselves) independent from the others. It is also common for spacecraft tracking to be very sparse – short periods (on the order of minutes or hours, depending on the orbital regime) with dense observations interspersed with long periods (on the order of hours or days, depending on the orbital regime) with no observations. The first encoder extracts information from a short period of measurements and condenses it to a single vector. The second encoder combines the vector representation of each short track and extracts information from the whole set of tracks. In this way, the Transformer's attention mechanism can take advantage of the natural sparsity of the problem. This model architecture is shown conceptually in Figure 4 and described below:

1. Pre-processing:
   (a) Each 24-hr navigation measurement file is separated out into individual ground tracks. For the data used here, a measurement file typically contains 10-22 ground tracks with about 200 measurements each.
   (b) Most of the data are scaled on a per-track basis. As before, station ID and time are scaled globally.
   (c) Each track is padded or trimmed to have an equal number of measurements per track.
   (d) Each list of tracks is padded or trimmed to have an equal number of tracks per input sequence.
   (e) The sample is reshaped from a simple time series of dimensions $(n_f \times N)$ into a 3-dimensional sample of size $(s \times p \times n_f)$, where s is the number of passes in each arc, p is the number of points in each pass, and n_f is the number of features.
2. Model forward pass:
   (a) The Time2Vec algorithm [7] is used to expand the time element of the feature vector, for each tracking pass ("track").
   (b) All feature vectors pass through a linear layer to expand the length of the feature vector.
   (c) Prepend each tracking pass with a "class token" vector of zeros.
   (d) The data for each tracking passes through the first Transformer encoder layer. to draw relationships between data within a single track.
   (e) The 0^{th} vector of each transformed track is kept and concatenated into a condensed sequence. The rest of each sequence (elements 1 through N) is not used.
   (f) Prepend the condensed sequence with a "class token" vector of zeros.
   (g) The condensed sequence passes through the second Transformer encoder layer.
   (h) The condensed sequence passes through another linear layer to further reduce the length of the feature vector to 5 (the number of anomaly classes).
   (i) Finally, the 0^{th} vector of the condensed sequence (the class token) is interpreted as the class of the anomaly. The rest of the sequence (elements 1 through N) is not used.

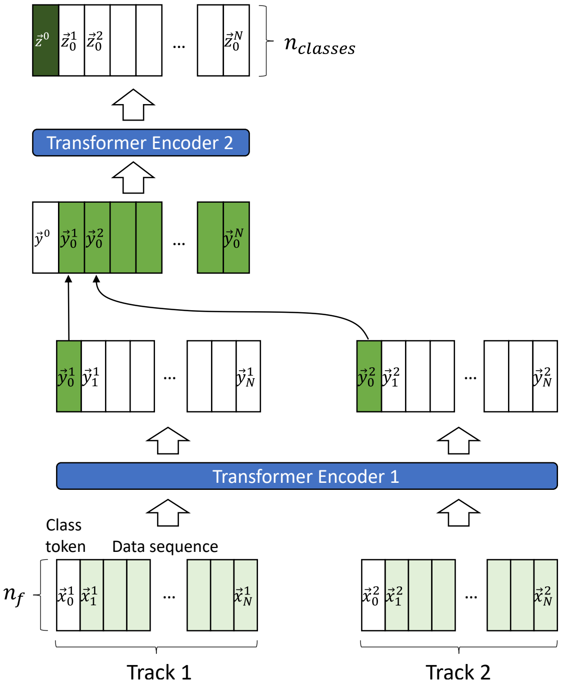


Figure 4. Conceptual diagram of the TPT architecture.

4.3. Vision Transformer (ViT)

Human operators typically plot parameters of interest over time for visual inspection and identification of anomalies. Given the great recent advances in computer vision, we consider the question: can a computer vision model function like a human and learn to visually identify anomalous data?

The Vision Transformer (ViT), invented by Google Research [11], divides an image into a set of 2-dimensional patches, then treats this set of image patches as a sequence that can be processed with a Transformer encoder model. Attention-based models such as the ViT and its numerous derivatives are the best performing architecture on a wide range of computer vision benchmarks.

An image to a computer is simply a 3D array or tensor with dimensions (channels × height × width); an RGB image has dimensions (3 × height × width). Any tensor which has this shape can be used by the ViT model, and there is no limit on the number of channels. The navigation inputs are constructed by taking estimated parameters from the filter such as pre-fit and post-fit residuals and plotting an image of each over time. This results in a tensor of data for each feature with size $(1 \times 300 \times 300)$ (grayscale images have only one channel). Each of an arbitrary number of features is then concatenated along the channels dimension, creating a new tensor of dimensions $(features \times value \times time)$. This essentially a one-hot encoded, scaled, and down sampled version of all features over time.

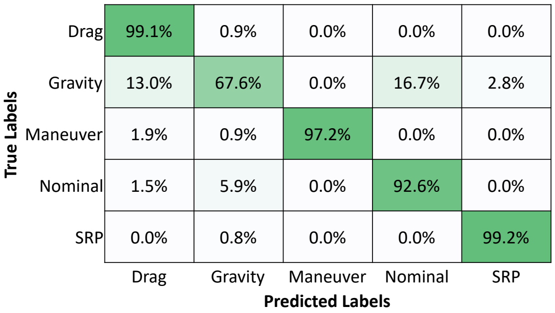
The ViT preprocessing and forward pass are described below. For more detail on the forward pass, we refer the reader to [11].

1. Pre-processing:
   (a) Plot each data feature over time (e.g., with matplotlib or an equivalent library), generating a set of several grayscale images.
   (b) Concatenate the images along the channel dimension, creating a "hyper-image."
2. Model forward pass:
   (a) Divide the input image into a set of patches.
   (b) Flatten each patch, add a positional encoding (to allow the Transformer encoder to learn spatial dependencies between patch locations), and embed in a higher dimensional space.
   (c) Prepend the sequence of embedded patches with a "class token."
   (d) Pass the sequence through a Transformer encoder.
   (e) Pass the sequence through another linear layer to reduce the length of the feature vector to 5 (the number of anomaly classes).
   (f) Finally, the 0^{th} vector of the sequence (the class token) is interpreted as the class of the anomaly. The rest of the sequence (elements 1 through N) are not used.

5. Results

The confusion matrix for a top-performing Tracking Pass Transformer model is shown in Fig 5. The confusion matrices for the Measurement Transformer and Vision Transformer architectures show the same trends, and they are not included here to save space. The largest source of confusion for all the models is between the "gravity", "drag", and "nominal" classes. The reason these classes are easily confused is that the drag mismodel and gravity mismodel can be very subtle in the simulated data. For example, error in high-order gravity field terms can only be observed for spacecraft in low altitudes, and the line between "nominal" and "mismodel" is subjective. Drag effects are similarly only apparent on the time scale of the input data (24 hours) in lower altitudes. For all four anomalous classes, the subjective size of the anomaly in the simulated data ranges from near-zero (indistinguishable from nominal, even to a human expert) to large and obvious. Most confusion takes place with nearly-zero anomalies, where trends are below the noise level.

Across all subsets of the simulated data generated in this research, the "maneuver mismodel" class was most accurate and most easily generalized between distributions (e.g., a model trained to identify maneuvers in low-altitude spacecraft could reliably identify maneuvers in high-altitude spacecraft).



True Labels \ Predicted Labels	Drag	Gravity	Maneuver	Nominal	SRP
Drag	99.1%	0.9%	0.0%	0.0%	0.0%
Gravity	13.0%	67.6%	0.0%	16.7%	2.8%
Maneuver	1.9%	0.9%	97.2%	0.0%	0.0%
Nominal	1.5%	5.9%	0.0%	92.6%	0.0%
SRP	0.0%	0.8%	0.0%	0.0%	99.2%

Figure 5. Confusion matrix for Tracking Pass Transformer, tested with in-distribution validation data withheld from the training dataset.

All three of the model architectures described above achieved over 80% validation accuracy, with the best performing Tracking Pass Transformer models achieving 93% validation accuracy. However, accuracy alone is an insufficient metric. One objective of this research is to develop the capability for fully autonomous spacecraft. Typically, the compute capability available on spacecraft is orders of magnitude lower than that of a modern laptop or smartphone. As a result, it is important to develop models that are both accurate and small. Initial models were 10's to 100's of MB in size, which can be hard to fit on traditional spacecraft computers. Hyperparameter tuning allowed us to reduce model sizes to be on the order of 1-10 MB disk space. The trade space of model size and accuracy is shown in 6.

Interestingly, the Measurement Transformer and Vision Transformer architectures were found to have similar performance in terms of model accuracy as a function of model size, despite taking fundamentally different approaches to data preprocessing and model inference. The Tracking Pass Transformer architecture was most accurate overall and also showed the best performance when tested on out-of-distribution data. A detailed analysis of performance on in-distribution vs out-of-distribution data will be reserved for a future publication.

6. Conclusion

This paper presents some of the key findings from an effort to apply AI/ML techniques to spacecraft autonomy. The work presented here focused on detecting anomalies in spacecraft navigation and classifying them by type of

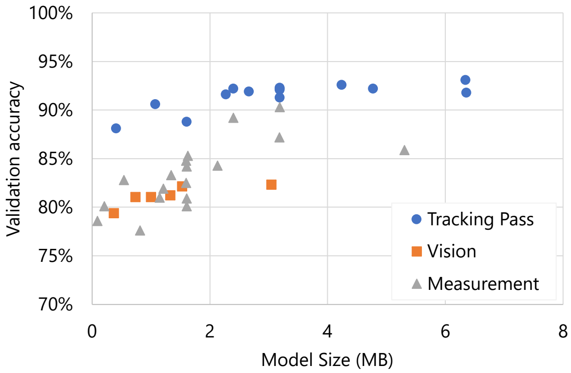


Figure 6. Summary of model trades displaying results for validation accuracy vs. model size (MB)

mismodeled acceleration. Spacecraft navigation is the process of mapping a sparse, irregular time series of geometric measurements to estimates of spacecraft state (position and velocity) at epochs of interest. We find that Transformers, specifically the Transformer encoder block with self-attention, are highly effective for data with these characteristics and are a natural fit for at least some aspects of spacecraft navigation. Three model architectures are presented, each of which may be preferable in different circumstances.

In February and March of 2024, a miniaturized version of the Measurement Transformer was successfully uplinked and tested onboard the CAPSTONE spacecraft near the Moon. This test demonstrated successful flight software implementation of the model preprocessing and model inference, numerical precision agreement between ground and flight implementations, and end-to-end connection of the various ground and space elements. Detailed description of that onboard test will be presented in a separate publication. We mention it here as evidence that cutting-edge ML algorithms can be realistically implemented in space and as further encouragement to others to continue investigating AI for space.

7. Acknowledgements

The authors wish to thank the MicroE4AI (Microelectronics for Artificial Intelligence) program team for their support.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2022-2011100600002]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] Atheer Algherairy, Wadha Almattar, Eman Bakri, and Salma Albelali. The impact of feature selection on different machine learning models for breast cancer classification. In *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, pages 91–96, 2022. 3

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 3

[3] Srikar Beechu. *Development of Lithium Ion Battery Dynamic Model*. PhD thesis, Technische Universitat Chemnitz, 2016. 3

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 4

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41, 2009. 3

[6] Zekai Chen, Dingshuo Chen, Zixuan Yuan, Xiuzhen Cheng, and Xiao Zhang. Learning graph structures with transformer for multivariate time series anomaly detection in iot. *CoRR*, abs/2104.03466, 2021. 3

[7] Zhongwu Chen, Chengjin Xu, Fenglong Su, Zhen Huang, and Yong Dou. Incorporating structured sentences with time-enhanced bert for fully-inductive temporal relation prediction, 2023. 5, 6

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019. 3

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. 3

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 3, 4, 5

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3, 5, 6, 7

[12] Scott Evans, William Taber, Theodore Drain, Jonathon Smith, Hsi-Cheng Wu, Michelle Guevara, Richard Sunseri, and James Evans. Monte: the next generation of mission design and navigation software. *CEAS Space Journal*, 10: 79–86, 2018. 4

[13] Jake Grigsby, Zhe Wang, Nam Nguyen, and Yanjun Qi. Long-range transformers for dynamic spatiotemporal forecasting, 2023. 3, 4

[14] Haixuan Guo, Shuhan Yuan, and Xintao Wu. Logbert: Log anomaly detection via BERT. *CoRR*, abs/2103.04475, 2021. 3

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997. 3

[16] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020. 4

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. 4

[18] Yang Li, Chunhua Tian, Yuyan Lan, Chentao Yu, and Keqiang Xie. Transformer with sparse attention mechanism for industrial time series forecasting. *Journal of Physics: Conference Series*, 2026(1):012036, 2021. 4

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 4

[20] Jiawei Ma, Zheng Shou, Alireza Zareian, Hassan Mansour, Anthony Vetro, and Shih-Fu Chang. Cdsa: Cross-dimensional self-attention for multivariate, geo-tagged time series imputation, 2019. 4

[21] Farzan M. Noori, Md. Zia Uddin, and Jim Torresen. Ultra-wideband radar-based activity recognition using deep learning. *IEEE Access*, 9:138132–138143, 2021. 3

[22] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. 4

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. 4

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 4

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. 4

[26] Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. Self-attentional acoustic models. *CoRR*, abs/1803.09519, 2018. 3

[27] Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21, 2020. 4

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 3

[29] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey, 2023. 3, 4

[30] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 2114–2124, New York, NY, USA, 2021. Association for Computing Machinery. 3, 4

[31] G.P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000. 3