Approximating Orbit Uncertainties of Catalog Objects using Neural Networks

Matthew D. Popplewell[1], Samuel Kwon[2]

[1] Advanced Space, LLC, 1400 W 122nd Ave, Westminster, CO 80234, USA, Email: matthew.popplewell@advancedspace.com
[2] Advanced Space, LLC, 1400 W 122nd Ave, Westminster, CO 80234, USA, Email: samuel.kwon@advancedspace.com

ABSTRACT

The rapid growth of the space catalog over the past few decades, and the anticipated trends towards proliferated constellations in the future, demand automation in collision avoidance procedures. However, most processes rely on state uncertainty information for both primary and secondary objects, which is not included in public space catalogs. This paper introduces a machine learning-based framework for generating synthetic covariances for such catalogs. Unsupervised learning techniques are used to discover natural data relationships within a database of historical Conjunction Data Messages. Neural networks are then trained to map the relationship between physical parameters and covariance trends. The result is a framework that enables improved collision risk assessment for spacecraft operators.

1 INTRODUCTION

The recent surge in space activity has exacerbated the orbital debris problem – the European Space Agency currently approximates the Earth debris population to be greater than 130 million. Of the millions of resident space objects (RSO) in orbit, only about 10,000 are operational satellites. Industry trends toward proliferated constellations for commercial, civil, and defense applications will accelerate the growth in the number of space objects, both functional and non-functional. Many future missions are planned for operation in Low Earth Orbit (LEO), a valuable yet already congested region. Already the effects of the growing debris population are apparent as the number of collision avoidance maneuvers (CAM) performed by the International Space Station within the past several years has increased drastically.

To avoid the dire consequences posed by a potential collision, satellite operators rely on designated collision avoidance (COLA) groups to monitor the debris environment, predict close approaches, and provide the information necessary for planning avoidance maneuvers. The 18th Space Defense Squadron (SDS) operates the U.S. Space Surveillance Network (SSN) to track RSOs and the associated tracking information is published in the Special Perturbations (SP) catalog. Typically, once a potential conjunction is identified, information concerning the event will be provided to spacecraft owners in the form of Conjunction Data Messages (CDM) at regular intervals. COLA operators monitor the evolution of the event via these CDMs to determine if a CAM is necessary. In situations when a CAM is required, the operators will design a maneuver in-house but must deliver the post-maneuver ephemeris to the 19th SDS for additional conjunction screening. The screening process is human-in-the-loop intensive and relies on antiquated technologies, and consequently can be time-consuming. There is also no guarantee the first designed CAM will pass further screening, potentially requiring multiple iterations between the operator and 19th SDS.

As the space catalog continues to grow, it is infeasible to rely entirely on humans-in-the-loop for COLA processes and automation is necessary. However, while the SP catalog is available to COLA operators and is suitable for some basic analyses, the catalog does not include orbit determination uncertainty on any objects. This covariance information is necessary for more thorough risk assessment, such as calculating the probability of collision (Pc) between the primary satellite and any secondary object. This limitation restricts satellite operators to be reliant on the increasingly obsolescent technologies in-place today.

This paper introduces a technique to approximate the covariances of objects in the space debris catalog using machine learning (ML). Unsupervised techniques are applied to a large dataset of historical NASA CDMs to systematically discover relationships between features like orbital state and object size, and the orbital uncertainties of secondary objects. Natural clusters in the dataset are extracted and a neural network (NN) learns the high-dimensional mapping between the input features and object covariances.

2 RELATED WORK

The use of historical CDMs to derive state uncertainties was first introduced as part of Assessment of Risk Event Statistics (ARES), a module in the European Space Agency's (ESA) Debris Risk Assessment and Mitigation Analysis (DRAMA) toolkit [1]. ARES provides spacecraft operators with statistics concerning the

Proc. 9th European Conference on Space Debris, Bonn, Germany, 1–4 April 2025, published by the ESA Space Debris Office
Editors: S. Lemmens, T. Flohrer & F. Schmitz, (http://conference.sdo.esoc.esa.int, April 2025)

frequency of conjunction events and required CAMs between a given spacecraft and the Earth debris environment. To determine if an avoidance maneuver is necessary, operators use a combination of metrics such as miss distance, Pc, and Mahalanobis distance (MD), with Pc commonly being the primary decision factor. Calculating Pc and MD, however, requires the orbital state uncertainties of the primary and secondary objects involved in the conjunction. While the primary object covariance is often known by the spacecraft operator, public space catalogs, such as the SP Ephemeris catalog provided by SpaceTrack, do not contain covariance information on secondary objects. Therefore, in order for ARES to provide statistics on expected CAMs, the tool must be able to estimate expected covariance values for different classes of debris impact flux.

Originally, in DRAMA 1, a Two-Line Element (TLE) analysis provided these uncertainties by comparing states propagated using the Simplified General Perturbations (SGP4) model to those propagated in a high-fidelity numerical model [2]. The state differences are compared in a radial, along-track, cross-track (UVW) frame and the associated covariances used in a look-up table. With the introduction of Conjunction Summary Messages (CSM), the DRAMA 2 covariance estimation model was augmented using the newly added data in the CSMs, with the original TLE analysis providing context in cases where CSMs were not available. DRAMA 3, the most recent iteration of the toolkit, currently leverages covariance data from a historical database of millions of CDMs. The DRAMA 3 approach divides the covariance lookup table into several classes based on physical parameters such as object size and orbit eccentricity, perigee, and inclination [3]. For each physical class, the UVW uncertainties are represented by a linear regression model. The regression model is fit using the covariance values contained in CDMs belonging to the given class.

Recently, the German Space Operations Center (GSOC) has extended this approach to their database of historical CDMs for use with SpaceTrack's SP Ephemeris catalog [4]. Similar to DRAMA 3, the CDM database is categorized into distinct domains based on combinations of various physical characteristics. In addition to those used by the DRAMA 3 team, the GSOC authors introduce an additional classification parameter, the 10.7 cm solar radio flux (F10.7). The authors also add two additional classes to both the perigee altitude and inclination domains to account for objects in high-altitude and retrograde orbits. After categorizing the CDMs to their respective classes, the UVW standard deviations of secondary objects are modeled using a non-linear least square curve fit. A Trust Region Reflective algorithm fits a quadratic function to relate the time difference to Time of Closest Approach (TCA) and the one-sigma position error of each UVW component. A modified Powell's algorithm is also used to generate 95th percentile upper and low bounds for each class.

While effective, the covariance domain discretization technique has several noticeable limitations, primarily due to the manual selection of characterization class bounds. The selection of the characterization parameters is justified by qualitatively examining the factors that influence orbit state uncertainty:

- Lower perigee altitudes lead to a higher drag perturbation but also make collecting longer ground-based tracking arcs difficult.
- Lower inclinations may present greater challenges for collecting measurements, as most sensor networks are located at higher latitudes.
- Smaller objects may be more difficult to observe and are more heavily influenced by drag and solar radiation pressure perturbations. Smaller objects also are usually debris particles, which rely on less accurate, passive observations.
- F10.7 influences both the drag and solar radiation pressure environment.

However, the process of defining class boundaries for these parameters largely relies on operator intuition, introducing several drawbacks.

The most significant limitation, as described by the DRAMA and GSOC teams, is the occurrence of classes with no or limited CDM data. In these cases, the class must be populated either by the original TLE analysis, as used in DRAMA 1 and 2, or by borrowing the curve fit coefficients of a neighboring class, as demonstrated by DRAMA 3 and GSOC. In the latter case, the preferred approach is to borrow from classes along the inclination dimension. However, using covariance models from higher inclination classes may underrepresent the true covariances of lower inclination classes due to the poor observability of low inclination orbits. The challenge of empty or limited classes becomes more pronounced if one naively applies the predefined boundaries to a different CDM dataset, which may have a different distribution across the characterization parameters.

Additionally, manually specifying the characterization class bounds neglects the exploration of underlying structures within the CDM dataset. While the prior look-up table technique effectively encapsulates the range of possible values for each class parameter, it relies on rigor alone to do so. As a result, the method risks overlooking more effective patterns inherent in the data. Natural groupings or correlations between parameters may not directly align with the predefined class boundaries. Furthermore, the approach lacks flexibility to robustly generate additional classes when new data becomes available. Overall, the look-up table approach limits the ability to uncover more nuanced, data-driven insights that could improve the overall process.

Prior work also relies on low-dimensional curve fits to model the relationship between CDM creation time to TCA and the UVW component standard deviations for each class. In doing so, one must maintain many, albeit simple, models and introduce a process to access the correct models for a given secondary object. Furthermore, the UVW component standard deviations alone are modeled by the curve fits. The covariance terms of the uncertainty are neglected, despite being critical for accurate Pc calculations. Since the covariances are synthetic, one could argue that the slight improvement from modeling the off-diagonal components of the covariance matrix is negligible given the inherent modeling error. However, this is primarily limited by the choice of a low-dimensional curve fit. The overall performance of the technique can be improved by using a higher-dimensional model that also incorporates covariance terms.

This study introduces a new approach for synthetic covariance generation using ML techniques. Unsupervised learning methods are used to uncover natural clusters and relationships between key parameters in the CDM dataset, offering data-centric insights into the training set and improving model performance. Furthermore, as new CDM data becomes available, unsupervised learning provides a rapid yet robust method for reevaluating the training dataset and identifying new patterns. After clustering, each cluster from the training dataset is extracted and used to train a specialized "expert" NN for covariance approximation. A "general" NN is also trained on the entire CDM dataset to generate covariance values for secondary objects that fall outside of the training distributions of the expert models. The NN model learns to approximate the high-dimensional relationships between orbit state, object size, and solar flux and the full UVW covariance matrix, providing a robust, accurate framework for synthetic covariance generation.

3 METHODS

3.1 Model Architecture

The overall architecture to approximate the covariance of objects in the space debris catalog consists of a Feed Forward Neural Network (FNN), a supervised learning method, along with an unsupervised learning algorithm, K-Means clustering. Given the dataset of historical CDMs, the K-Means algorithm is utilized to uncover natural groupings in the dataset. After determining the appropriate K value, the cluster labels are appended to the historical CDM dataset and exported for modeling. An FNN is initiated and trained on each cluster to evaluate the fit and prediction performance on each grouping. The following sections further highlight the model architectures and specific parameters utilized to execute the goal of approximating secondary object uncertainties.

K-Means, an unsupervised learning method, seeks to identify natural clusters in the dataset using a distance metric. The K in the algorithm's name represents the desired number of clusters for the algorithm to create. Given a cluster, the distance between the centroid and the incoming datapoint is measured to identify the closest cluster to which the datapoint belongs. This method allows the user to discover and characterize natural patterns in the dataset. Given the historical CDM dataset, the K-Means clustering algorithm is utilized to identify different groupings among various conjunction messages. Since the K-Means algorithm requires a predefined K value, the inertia value at various K values are extracted to generate an elbow plot. The ideal K value occurs when the change in inertia between subsequent K values begins to plateau. Note that the default distance metric, the Euclidean distance, is utilized in the K-Means algorithm. After identification of the ideal K value and executing the algorithm, the cluster labels are extracted and appended to the historical CDM dataset for FNN training and testing.

FNNs are a type of Artificial Neural Network that seek to mimic the mechanisms of a human neuron. The models consist of a network of "neurons" that receive input, perform a simple mathematical operation, and pass the resulting output to the next neuron. As the name implies, FNNs consist of several fully connected layers of neurons that pass information in one direction, from inputs to outputs. The model is trained by performing successive forward and backward passes of the neural network. During a forward pass, for each neuron, a linear operation is performed on a given input based on an associated weight and bias, then a nonlinear activation function is applied before the output is passed to the next neuron. Without an activation function, the mathematical formulation is similar to a linear regression model which is only suitable for linearly separable data. The ability to apply an activation function allows the network to identify and model nonlinearly separable data. During the backward pass, also known as backpropagation, the partial derivatives of a defined cost function are computed with respect to each weight and bias. The weights and biases are then updated based on a gradient descent algorithm such that the final error between the predicted and desired model outputs is minimized. Figure 1 is an example FNN. In this example architecture, the input layer consists of three nodes, a single hidden layer with four nodes, and a final output layer with a single node.

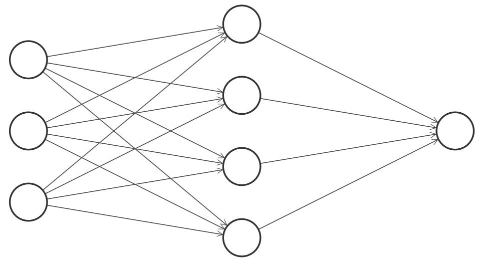

Figure 1. Simple Feed Forward Neural Network.

The FNN utilized to approximate the covariances of objects in the space debris catalog consists of 24 nodes in the input layer, four hidden layers with 4096 nodes in each, and a final output layer with six nodes. Note that the six nodes represent the second body covariance predictions in the UVW frame. After each layer, the FNN utilizes the Leaky ReLU activation function along with a dropout layer set to 0.1, which randomly sets 10% of the nodes to zero during model training. Both techniques are implemented to prevent overfitting and to better bound the covariance predictions. During the training process, the model learns to map the relationship between the secondary object's orbit state, size, ballistic coefficient, solar radiation pressure coefficient, and creation time to TCA and the associated state uncertainties in the UVW frame. Mean squared error (MSE) is used as a loss function alongside the Adam optimizer to find the optimum model parameter values.

3.2 Uncertainty Quantification Using Adaptive Monte Carlo Dropout

Adaptive Monte Carlo Dropout is utilized as a method for uncertainty quantification. When the user interprets the predictions of the NN, the goal is to generate a corresponding confidence interval (CI) bound, which provides insight into the model prediction variability. Monte Carlo Dropout is a popular method for performing uncertainty quantification on NNs. During the training phase, the NN goes through a dropout layer which deactivates a set percentage of neurons in the network. While the dropout layer is activated during the training phase, the layer is normally turned off during the evaluation phase so that the full architecture is utilized to generate predictions. In Monte Carlo Dropout, the dropout layer is still activated during the evaluation phase. Using the mechanisms of a dropout layer, at each forward pass, a set percentage of neurons are switched off, ideally allowing the user to generate and make predictions using various NN architectures. Given this property, n number of predictions are made, and the user can then generate a confidence interval from the prediction set. A downside of Monte Carlo Dropout is the difficulty in finding the ideal prediction set size where a from which a CI can be computed. A larger prediction set size allows for a more accurate representation of the CI. However, in some cases, it is easy to perform more computations than necessary when generating the prediction set, decreasing computational efficiency. Daniel Bethell et al., [5] proposes an alternative method called Adaptive Monte Carlo Dropout to decrease computational demand. The fundamental process of the Adaptive Monte Carlo Dropout method is like the Monte Carlo Dropout method, but the authors introduce three extra parameters: δ, K, and P. In Adaptive Monte Carlo Dropout, the NN will generate multiple forward pass predictions until the absolute difference in prediction variance reaches the δ threshold P times or the number of forward passes is equal to the maximum number of passes, K. During each forward pass, the predicted value is appended to the prediction set and the absolute difference between the prediction variance at the $(i-1)^{th}$ and i^{th} step is computed and compared with the δ threshold. If the absolute difference in variance from the prediction set reaches the δ threshold P times, the model has converged, and the confidence interval is computed. Compared to a generic Monte Carlo Dropout method, the Adaptive Monte Carlo dropout method integrates a stopping mechanism using the absolute difference in variances to mitigate excessive amounts of forward pass predictions.

4 TRAINING DATA

The NNs presented in prior sections are trained using a large database of NASA Conjunction Assessment Risk Analysis (CARA) CDMs. The database contains over 575,000 CDMs covering nine primary satellites from October 1, 2016, to October 31, 2022. The primary objects all occupy similar Sun-synchronous orbits and are Earth science missions. The first five, for example, are some of the first satellites deployed as part of the Earth Observing System. The primary objects are detailed in Table 1.

Table 1. Primary objects in NASA CDM training dataset.

Primary ID	Perigee (km)	Inclination (°)	Eccentricity
25994	6992.4	98.20	0.015
27424	6993.3	98.22	0.015
28376	6994.1	98.21	0.015
29107	6972.0	98.23	0.017
29108	6980.4	98.24	0.016
29479	6914.6	98.16	0.016
38337	6993.1	98.21	0.015
40059	6994.6	98.21	0.015
40376	6968.8	98.13	0.016

Over this period, the nine primary objects encounter 4,928 unique secondaries. Statistics on the secondary objects are given in Table 2.

Table 2. Secondary object orbital element statistics.

Parameter	Minimum	Maximum
Perigee (km)	120.1	817.15
Inclination (°)	0.28	144.23
Eccentricity	3.09E-4	0.991
Radar cross section (m^2)	1.00E-3	26.06

The CDMs include all parameters recommended in the 2013 Consultative Committee for Space Data Systems (CCSDS) Conjunction Data Message Blue Book, plus some additional quantities computed by the NASA CARA team. Before training the ML models, the dataset goes through several preprocessing steps:

1. **Corrupt CDMs are removed from the dataset.** Corrupt CDMs are defined as CDMs containing non-positive definite primary or secondary covariance matrices, negative "days to TCA" values, or faulty entries in any column (e.g., an invalid CDM creation year).
2. **Outlier CDMs are removed from the dataset.** Outlier CDMs are defined as CDMs created more than seven days from TCA, CDMs reporting no observations or tracks used for either object's orbit determination, and CDMs with extreme parameter values – such as ballistic coefficients larger than ±100, radar cross sections of zero, or covariance entries larger than 1E12 km^2.
3. **Feature engineering introduces additional features to the dataset.** The new features include the percentage of observations/tracks used, eccentricities, semi-major axes, and periods for both objects, the F10.7 solar flux at the CDM creation time, and correlation coefficients for the UVW covariances. The correlation coefficients are naturally bounded from [-1,1], making them more stable and predictable during the training process.
4. **Unused features are removed from each CDM.** The inputs and outputs used to train the NNs are given below.
5. **Each feature is individually scaled to approximately [-1,1].** Keeping the inputs/outputs of the NN model close to an order of magnitude of one makes the model training process more stable, as large input/output values can lead to gradient issues during the backpropagation step.

After preprocessing, the training dataset contains over 550,000 unique CDMs. The input to the NN models consists of the following 24 features:

- Miss distance – the relative distance between the primary and secondary objects.
- Relative speed – the relative speed between the primary and secondary objects.
- Apogee – The apogee of the secondary object's orbit.
- Perigee – The perigee of the secondary object's orbit.
- Semi-major axis – The semi-major axis of the secondary object's orbit.
- Eccentricity – The eccentricity of the secondary object's orbit.
- Inclination – The inclination of the secondary object's orbit.
- Period – The period of the secondary object's orbit.
- Radar cross section (RCS) – The radar cross section of the secondary object as reported in the CDM. The RCS serves as a proxy to the object's size.
- Ballistic coefficient – The ballistic coefficient of the secondary object, which is a function of the object's coefficient of drag, frontal area, and mass.
- Solar radiation pressure (SRP) coefficient – The solar radiation pressure coefficient of the secondary object, which is a function of the object's coefficient of reflectivity, frontal area, and mass.
- UVW orbital state – The six-dimensional position and velocity of the secondary object in the UVW frame centered on the primary.
- Inertial orbital state – The six-dimensional position and velocity of the secondary object in an Earth-centered inertial frame.
- F10.7 – The 10.7 cm solar radio flux at the CDM creation time.
- Creation time to TCA – The time difference, in days, between the CDM creation time and TCA.

The output of the NN models consists of the following six features:

- The three UVW positional variances.
- The correlation coefficients for the UV, UW, and VW covariances.

5 RESULTS

Six separate NN models are trained to map the relationship between the secondary's physical parameters, such as orbit state and object size, and the UVW covariances of the object. The framework consists of one "general" NN, trained on the entirety of the CDM dataset, and five "expert" models, each trained on a natural cluster identified by the K-Means algorithm. Similar to the look-up table approach used by DRAMA and GSOC, the expert models provide more accurate covariance approximations for secondary objects that fall within their training cluster. For objects that fall outside the distributions of the expert models, the general model provides a less accurate, but more robust prediction.

5.1 K-Means

To explore underlying patterns in the CDM dataset, only more static parameters are used in the K-Means algorithm. Although the instantaneous orbital state of the secondary object serves as an input feature to the NNs, it is excluded from the feature list when applying K-Means due to the large range of possible values. Instead, the K-Means algorithms focuses on parameters like orbital elements and object size. The algorithm has access to the following features: apogee, perigee, semi-major axis, eccentricity, inclination, orbit period, RCS, ballistic coefficient, solar radiation pressure coefficient, and F10.7.

To determine the best number of clusters K for the dataset, the inertia values of various K values are plotted. Following the elbow method, the ideal K value occurs when the change in inertia between subsequent values of K begins to plateau. Figure 2 is the elbow plot of the K-Means algorithm fitted at K values ranging from one to ten.

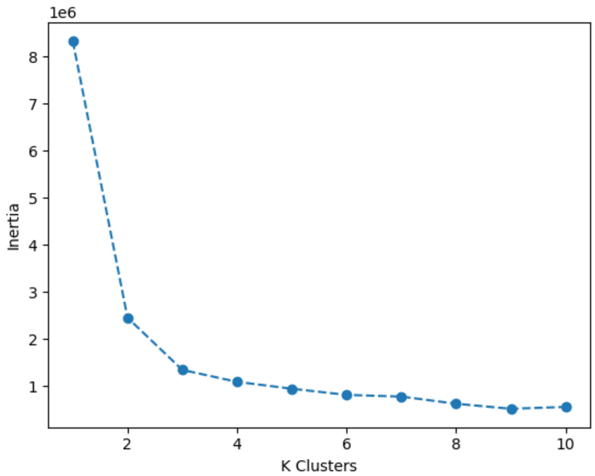

Figure 2. K-Means elbow plot comparing inertia for different values of K *clusters.*

Based on the elbow plot, the ideal K value occurs around $K = 5$. After determining the optimal K value, the classic K-Means algorithm is applied to the dataset to generate five clusters. The mean of each input feature to the K-Means algorithm is shown in

Table 3, along with the number of samples in each cluster. There is little variance across each cluster for all features except RCS, suggesting that RCS may be the most distinguishing feature for this CDM dataset. This is further supported by Figure 3, which plots the RCS statistics for each cluster group and shows that the RCS values of each cluster do not overlap.

Table 3. Mean of K-Means features for each cluster

Parameter	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**
Number of samples	14,000	6,031	4,560	464,607	13,546
Apogee	794.7	1,073.7	1,092.6	828.3	766.1
Perigee	678.6	660.7	681.0	665.7	675.3
Semi-major axis	6,904.0	6,826.2	7,006.4	6,708.0	6,817.1
Inclination	79.8	93.5	92.4	84.2	91.9
Eccentricity	0.038	0.082	0.050	0.076	0.052
RCS	2.21	8.21	13.93	0.045	5.16
Ballistic coeff.	0.024	0.023	0.025	0.409	0.023
SRP coeff.	0.015	0.013	0.018	0.179	0.012
F10.7	82.3	90.4	115.6	81.3	80.4

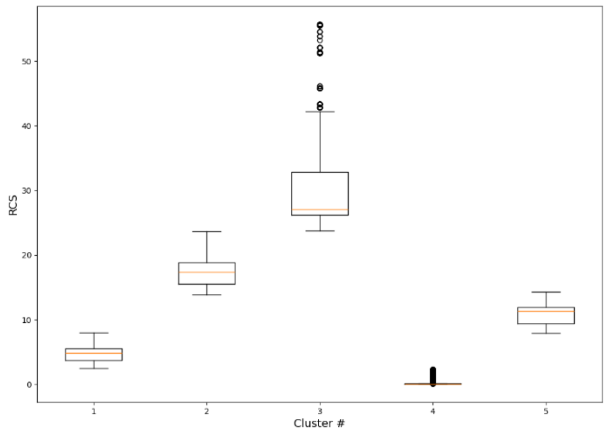

Figure 3. Box plot of RCS values for each cluster.

5.2 NN Models

To train each NN model, the relevant training data is randomly split into three datasets: training, validation, and test. This dataset splitting approach is a standard practice in ML to ensure the model is not overfitting. During the training, the NN updates its internal weights and biases by comparing its predictions to the true covariance values in the training set. The model processes the training set in batches, performing a parameter update via gradient descent after each batch. At the end of a training epoch, once the model has seen the entire training dataset once, the NN is evaluated on the validation dataset to determine if it is overfitting. A well-performing model will deliver similar training and validation losses at the end of each epoch. This process is repeated for some number of epochs (500 in the case of the models presented in this study).

The test dataset serves to evaluate the model on a completely unseen set of data at the end of the training process. The NN's performance on the test data is often indicative of its true performance, i.e., good performance on the test set often points to a model that is generalizable and did not overfit the training set.

Most NNs are essentially black boxes, providing point predictions through a convoluted series of mathematical transformations that are difficult to interpret. For an application as sensitive as COLA, model explainability is critical so operators can trust the predicted information. Therefore, when evaluating the performance of the NNs in this study the Adaptive Monte Carlo Dropout technique is utilized, which evaluates the NN several times for each test case. The Pc for each prediction is then calculated, providing a mean prediction as well as upper and lower confidence interval bounds.

For each model in this section, two metrics are calculated. The first compares the Pc values derived from the point predictions of the NNs to the true Pc values from the CDM dataset, providing an assessment of the NN's accuracy. The second evaluates how often the true Pc falls within the confidence intervals of the model prediction, providing insight into the reliability of the NN. Utilizing these metrics, the authors aim to demonstrate the applicability of an ML approach for synthetic covariance generation.

5.2.1 Expert Models

Each cluster identified by the K-Means algorithm is used to train an "expert" NN model. The expert models are then evaluated on the test dataset relevant to the respective cluster. The secondary UVW covariance predictions are combined with the true primary UVW covariances and miss distances to calculate the predicted Pc. To analyze the accuracy of the NN predicted Pc values, their impact on a COLA operator's decision-making is assessed. An actionable risk of 1E-4 is chosen as a CAM threshold, that is for events with a Pc greater than 1E-4 it is assumed a COLA operator would perform a CAM. It is assumed no CAM is performed for events with a Pc less than 1E-4. Then, the percentage of test cases in which the model prediction would change an operator's decision relative to the truth is calculated. Table 4 reports this metric for each cluster group. A false positive is defined as when the true Pc would not result in a CAM, but the model prediction dictates a maneuver. A false negative is the reverse situation, when the true Pc would require a CAM but the model predicted Pc does not. The final category occurs when the NN predicted Pc and true Pc would result in the same CAM decision.

Table 4. Percentage of test cases in which the expert NN predictions result in a different CAM decision

Cluster #	No Change in Action %	False Positive %	False Negative %
1	99.66	0.20	0.14
2	99.95	0.03	0.02
3	99.87	0.06	0.06
4	99.68	0.00	0.32
5	99.95	0.04	0.01

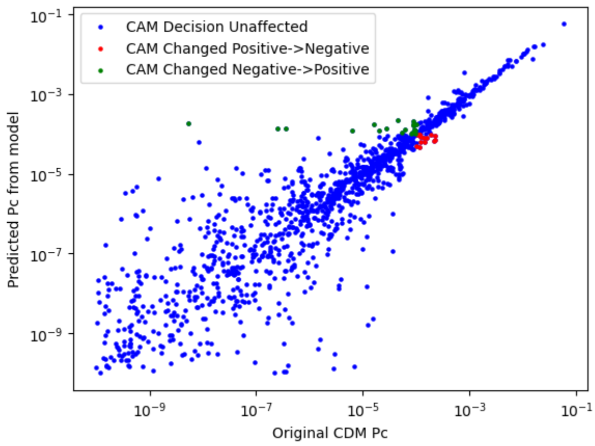

Figure 4. Cluster 1 NN-predicted Pc versus true Pc from CDM for events with Pc > 1e-10.

To assess the reliability of the NNs, the Adaptive Monte Carlo Dropout technique is applied on each cluster's test set. Table 5 reports how often the true Pc label falls within the model's confidence interval. An example of the NN confidence intervals for Cluster 1 is shown in Figure 5.

Table 5. Percentage of test cases in which the NN confidence interval captures the true Pc

Cluster #	True Pc lies within CI (%)
1	92.2
2	100.0
3	94.1
4	100.0
5	98.0

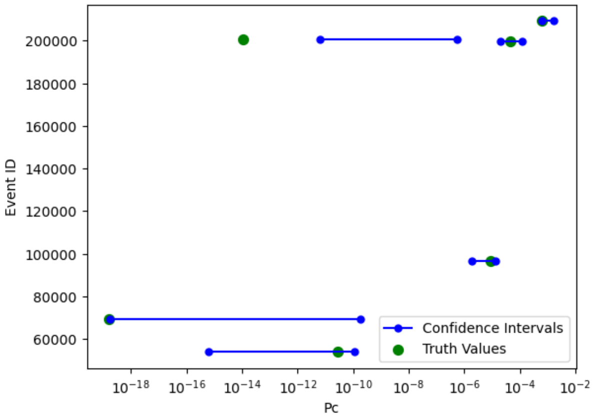

Figure 5. Examples of NN confidence interval predictions for Cluster 1.

As demonstrated in the results, the expert NN models generate accurate and reliable synthetic UVW covariance matrices that often do not impact an operator's decision to perform a CAM. For every clustered NN, the CAM decision is changed less than one percent of the time, suggesting that the NN predicted covariances, while not truly representative of reality, are sufficient for non-critical COLA processes. Additionally, the NNs provide confidence interval bounds that often capture the true Pc, offering COLA operators additional context. This allows them to make more informed decisions by examining the upper and lower bounds in conjunction with the mean prediction.

5.2.2 General Model

The general model is assessed in the same manner as the expert NNs, with the exception that the entire CDM test dataset is used. For each element in the test set, the NN predicts the full secondary object UVW covariance matrix, which is combined with the primary uncertainties from the CDM to calculate Pc. The CAM decision metric is then evaluated and reported in Table 6.

Table 6. Percentage of test cases in which the general NN predictions result in a different CAM decision

No Change in Action %	False Positive %	False Negative %
99.57	0.19	0.24

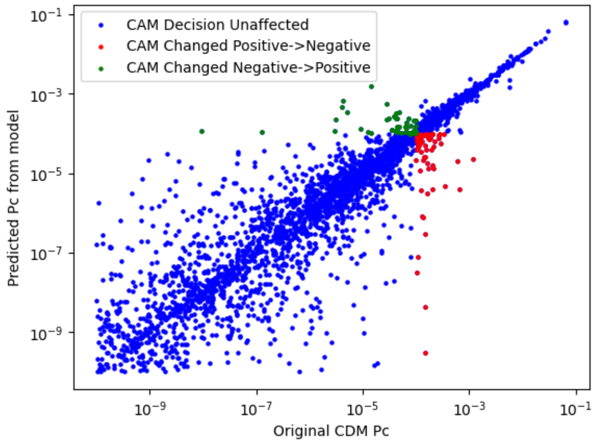

Figure 6. General NN-predicted Pc versus true Pc from CDM for events with Pc > 1e-10

Similarly, the reliability of the general NN is evaluated using the Adaptive Monte Carlo Dropout technique, revealing that the model's predicted CI captures the true Pc in 86.3% of test cases. An example of the NN confidence intervals for the general model is shown in Figure 7.

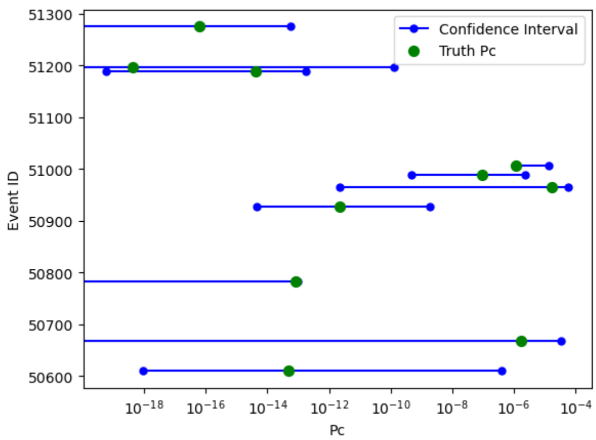

Figure 7. Examples of NN confidence interval predictions.

As anticipated, the general NN performs slightly worse than the expert NNs, as it must generalize to a much broader parameter distribution. However, the model is still shown to be highly accurate, changing an operator's decision to perform a CAM in less than one percent of test cases. A notable flaw in the current model is the poor CI performance; ideally, the NN's CI would capture the true Pc in nearly 100% of the test cases. This poor performance is most likely a result of improper tuning of the Adaptive Monte Carlo Dropout algorithm. Future work will refine the algorithm's δ, K, and P input parameters to deliver improved CI scores.

6 CONCLUSIONS AND FUTURE WORK

This study introduces an ML-based framework for generating synthetic covariance approximations for space catalogs. A K-Means algorithm is first used to uncover natural clusters in the historical CDM dataset. These clusters represent CDMs with high-dimensional similarities, which aids subsequent NN training. Then, expert NNs are trained on each cluster group to provide tailored covariance approximations. A general NN is also trained on the full dataset to provide coverage for objects that fall outside of the expert distributions. This ML-based approach improves upon prior methods by providing means to dynamically adapt to new data, reduce reliance on manually defined class boundaries, and utilize higher-dimensional relationships between physical parameters and historical covariances trends.

A clear weakness of the presented results is the lack of orbital variance in the training and testing data. The NASA CARA CDM dataset used to train the NN models consists of nine spacecraft in nearly identical orbits, which severely limits the distribution of variables seen by the NNs during training. Likewise, the test sets on which the NNs were evaluated is taken from the same historical database as the training data. Therefore, the generalizability of the presented framework to different orbital regimes has not been thoroughly explored. Recently, the authors have acquired access to Maxar Intelligence's DigitalGlobe constellation CDMs on SpaceTrack. Future work will use these CDMs to explore the robustness of the ML approach to different orbital regimes. Improvements in the training data and architectures of the NNs will also be investigated to improve accuracy and generalizability.

Additionally, the framework presented in this paper currently serves as a component in a larger conjunction screening system designed to automate COLA processes. The system enables the real-time screening of CAMs against the space debris catalog immediately after their design, reducing iteration with groups such as the 19th SDS. The tool is currently being deployed to a cloud-based environment, where external users will access it via a simple application programming interface (API). The authors intend to provide the full system as a free service for COLA operators in the near future, enhancing automation, accessibility, and efficiency in space traffic management.

7 ACKNOWLEDGMENTS

This research was sponsored by the NASA Small Business Innovation Research (SBIR) program, contract number 80NSSC24CA155.

The authors extend their gratitude to NASA's CARA office, particularly Ms. Lauri Newman and Dr. Matt Hejduk, for generously supplying the CDM dataset used in this study. The authors would also like to thank the Maxar Intelligence team for graciously giving access to the DigitalGlobe constellation CDMs.

8 REFERENCES

[1] V. Braun, A. Horstmann, B. Reihs, S. Lemmens, K. Merz, and H. Krag, "Exploiting Orbital Data and Observation Campaigns to Improve Space Debris Models," *J. Astronaut. Sci.*, vol. 66, no. 2, pp. 192–209, Jun. 2019, doi: 10.1007/s40295-019-00155-6.

[2] T. Flohrer, H. Krag, and H. Klinkrad, "Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue".

[3] V. Braun, "Expected collision avoidance manoeuvre rates in DRAMA-ARES based on a history of conjunction data messages".

[4] A. Zollo, "Usage of historical Conjunction Data Messages to reverse-engineer orbit errors of Special Perturbation catalogued objects.".

[5] D. Bethell, S. Gerasimou, and R. Calinescu, "Robust Uncertainty Quantification Using Conformalised Monte Carlo Prediction," Jan. 22, 2024, *arXiv*: arXiv:2308.09647. doi: 10.48550/arXiv.2308.09647.