



Research Report

JONATHAN TRAN, PRATEEK PURI, JORDAN LOGUE, ANTHONY JACQUES, LI ANG ZHANG,  
KRISTA LANGELAND, GEORGE NACOUZI, GARY J. BRIGGS

# Artificial Intelligence and Machine Learning for Space Domain Awareness

---

The Development of Two Artificial Intelligence Case Studies



For more information on this publication, visit [www.rand.org/t/RRA2318-2](http://www.rand.org/t/RRA2318-2).

#### About RAND

RAND is a research organization that develops solutions to public policy challenges to help make communities throughout the world safer and more secure, healthier and more prosperous. RAND is nonprofit, nonpartisan, and committed to the public interest. To learn more about RAND, visit [www.rand.org](http://www.rand.org).

#### Research Integrity

Our mission to help improve policy and decisionmaking through research and analysis is enabled through our core values of quality and objectivity and our unwavering commitment to the highest level of integrity and ethical behavior. To help ensure our research and analysis are rigorous, objective, and nonpartisan, we subject our research publications to a robust and exacting quality-assurance process; avoid both the appearance and reality of financial and other conflicts of interest through staff training, project screening, and a policy of mandatory disclosure; and pursue transparency in our research engagements through our commitment to the open publication of our research findings and recommendations, disclosure of the source of funding of published research, and policies to ensure intellectual independence. For more information, visit [www.rand.org/about/research-integrity](http://www.rand.org/about/research-integrity).

RAND's publications do not necessarily reflect the opinions of its research clients and sponsors.

Published by the RAND Corporation, Santa Monica, Calif.

© 2024 RAND Corporation

RAND® is a registered trademark.

Library of Congress Cataloging-in-Publication Data is available for this publication.

ISBN: 978-1-9774-1414-4

Cover: Getty Images/24K-Production, Vitalii Pasichnyk, da-kuk.

#### Limited Print and Electronic Distribution Rights

This publication and trademark(s) contained herein are protected by law. This representation of RAND intellectual property is provided for noncommercial use only. Unauthorized posting of this publication online is prohibited; linking directly to its webpage on rand.org is encouraged. Permission is required from RAND to reproduce, or reuse in another form, any of its research products for commercial purposes. For information on reprint and reuse permissions, please visit [www.rand.org/pubs/permissions](http://www.rand.org/pubs/permissions).

# About This Report

This report is the second of two companion reports and is intended to document the artificial intelligence and machine learning case study development conducted as part of this research. Consequently, this report is tailored to a more technical audience and should be of interest to space domain awareness practitioners (18th and 19th Space Defense Squadrons, National Reconnaissance Office, National Aeronautics and Space Administration [NASA]); supporting organizations within the U.S. Department of Defense (Air Force Research Laboratory, National Geospatial-Intelligence Agency), including the civil and commercial space traffic management community (NASA, Federal Communications Commission, U.S. Department of Commerce); and the broader artificial intelligence industry and community. For the policy-focused key findings and recommendations, please refer to the companion report, *Artificial Intelligence and Machine Learning for Space Domain Awareness: Characterizing the Impact on Mission Effectiveness* by Li Ang Zhang, Krista Langeland, Jonathan Tran, Jordan Logue, Prateek Puri, George Nacouzi, Anthony Jacques, and Gary J. Briggs (RR-A2318-1). Note that these closely related reports share some descriptive material. The research was conducted between October 2022 and September 2023, and this report does not reflect any additional analysis or changing circumstances beyond those dates.

The research reported here was commissioned by the Office of the Chief Scientist of the U.S. Air Force (AF/ST) and conducted within the Force Modernization and Employment Program of RAND Project AIR FORCE as part of a fiscal year 2023 project, “Artificial Intelligence for Space Domain Awareness.”

## RAND Project AIR FORCE

RAND Project AIR FORCE (PAF), a division of RAND, is the Department of the Air Force’s (DAF’s) federally funded research and development center for studies and analyses, supporting both the United States Air Force and the United States Space Force. PAF provides the DAF with independent analyses of policy alternatives affecting the development, employment, combat readiness, and support of current and future air, space, and cyber forces. Research is conducted in four programs: Strategy and Doctrine; Force Modernization and Employment; Resource Management; and Workforce, Development, and Health. The research reported here was prepared under contract FA7014-22-D-0001.

Additional information about PAF is available on our website:  
[www.rand.org/paf/](http://www.rand.org/paf/)

This report documents work originally shared with the DAF on September 25, 2023. The draft report, dated September 2023, was reviewed by formal peer reviewers and DAF subject-matter experts.

## Acknowledgments

The RAND team would like to thank Victoria Coleman, Joel Mozer, Col David Montminy, Daniel Eckhardt, and the Office of the Chief Scientist of the U.S. Air Force (AF/ST) for the opportunity to conduct this research and the guidance we received from them as the sponsors of this research. We would also like to thank the leadership, analysts, and operators from the following organizations for providing us with their time and assisting in this effort: the 18th and 19th Space Defense Squadrons, the Space Command and Control Program Office (Kobayashi Maru) team, and the U.S. Space Force Chief Technology and Innovation Office, as well as Justin Fletcher. We also appreciate the mentorship and guidance from several RAND colleagues: Peter Whitehead, Bonnie Triezenberg, Éder Sousa, and Brien Alkire. Special thanks go to the reviewers of this report, Kanna Rajan and Krista Romita Grocholski. Lastly, we thank our program leadership, Sherrill Lingel and Alex Hou, for their guidance and oversight, as well as the director of PAF, James Chow.

# Summary

## Issue

To address the growing demands of operating in the space domain, space domain awareness (SDA) operators must determine how to prioritize sensor observations more effectively, scale up to meet the sheer volume of resident space objects, and develop analytic capabilities that reflect the complexity of orbital mechanics and space operations, all while maintaining the responsiveness necessitated by operations in a warfighting domain. These factors present significant challenges to those tasked with the SDA mission and point to this mission as a prime candidate for support from artificial intelligence (AI) and machine learning (ML) tools, as such tools have the potential to increase the analysis tempo, expand the amount of usable data for this analysis, and free up operator time for more-complex tasks. AI/ML tools may help SDA operators meet these increasing challenges.

Although AI/ML tools have the potential to help meet these SDA challenges, the impact of these tools on the overall success of the SDA mission is not well understood, and this lack of understanding is a barrier to plan for and optimize the integration of these tools. The research documented in this report seeks to develop insight into the following questions through the development of AI/ML case studies:<sup>1</sup>

- How can existing AI/ML and automation technologies benefit the SDA mission?
- Which processes within the SDA mission area can be improved from increased automation and AI/ML?
- What are the prerequisites and enabling technologies that are necessary for AI/ML implementation?
- What limitations (inherent or built in) should be considered when employing these technologies?

## Approach

Our approach outlined in this report focuses on the development of two AI/ML case studies that explore the value of AI/ML for SDA mission processes. These two case studies offer a way to investigate in detail how AI/ML could be leveraged and to emphasize potential challenges to implementing an operational AI/ML tool. Accomplishing this first requires an understanding of the processes that make up this mission, the requirements and metrics for these processes, where there are opportunities for impact through leveraging AI/ML tools, and the nature of what these tools would change in the process and the outcomes. We then document the AI/ML tool development process

---

<sup>1</sup> For a policy-focused discussion of the challenges facing the SDA mission, we refer the reader to the companion report (Zhang et al., *Artificial Intelligence and Machine Learning for Space Domain Awareness*).

and methodology, characterizing the algorithm design space, and discuss algorithm performance and the downstream implications for SDA operators and practitioners. Lastly, we provide recommendations and opportunities for further research and analysis to improve our AI/ML tools.

The work presented here focuses on the highly resource-intensive conjunction assessment mission under the broader SDA mission set. The case studies focus on prediction and classification capabilities of neural networks and use of the capabilities to improve the conjunction assessment. We also address the need for uncertainty quantification in AI/ML tools for both improving the prediction and classification performance and mitigating some of the potential AI/ML policy barriers, including facilitating operator-machine teaming and operator trust.

## Key Findings

- Neural networks are capable approximators for complex nonlinear functions. We demonstrate that prediction and classification capabilities can be applied to SDA mission processes with standard and Bayesian neural networks.
- AI/ML tool developers should focus on tools that are compatible with the operational SDA architecture, data infrastructure, and processes.
- Quantification of risk and uncertainty tolerance provides an operator or analyst with the capability to provide feedback to an AI/ML model by setting thresholds based on acceptable risk.
- Quantification of risk and uncertainty tolerance can support improved performance of AI/ML tools focused on prediction and classification when compared with standard neural network approaches.
- Active learning may be an attractive AI/ML feature when paired with SDA mission processes.

## Recommendations

- The development of AI/ML tools requires significant investment in ensuring high-quality training data. The Office of the Chief Scientist of the U.S. Air Force, with input from SDA operators, should consider the availability of such data for AI/ML investment.
- AI/ML tool developers should consider SDA processes and limitations that have implications for tool design, including the selection of algorithms, metrics of performance, and benchmark requirements.
- The U.S. Department of Defense should continue to consider the value of uncertainty quantification methodologies when developing and deploying operational AI/ML tools. SDA operators should be trained to use the uncertainty quantification provided by these methodologies.

# Contents

About This Report.....	iii
Summary .....	v
Figures and Tables .....	viii
<b>CHAPTER 1.....</b>	<b>1</b>
Space Domain Awareness Mission Overview.....	1
Space Command and Control .....	2
CA Process .....	11
Future Data Architecture Uncertainties .....	15
<b>CHAPTER 2.....</b>	<b>17</b>
Artificial Intelligence and Machine Learning Overview .....	17
Current AI/ML Efforts in SDA .....	17
Selection of Two Case Studies .....	19
Neural Network Architecture.....	20
Managing Uncertainty .....	22
Model Evaluation .....	22
Summary .....	24
<b>CHAPTER 3.....</b>	<b>25</b>
Elliptical Screening Process Tool .....	25
Elliptical Screening Workflow Description.....	25
Data Generation .....	26
Model Architecture.....	27
Model Output and Postprocessing .....	31
Implementation Challenges and Future Work .....	34
Summary .....	35
<b>CHAPTER 4.....</b>	<b>36</b>
Orbital State Propagator Tool .....	36
Covariance Propagation Workflow Description.....	36
Data Generation .....	37
Model Architecture.....	40
Model Performance.....	42
Additional BNN Features.....	45
Operational Frameworks.....	49
Implementation Challenges and Future Work .....	51
Summary .....	52
<b>CHAPTER 5.....</b>	<b>53</b>
Conclusion.....	53
Abbreviations .....	55
References.....	57

# Figures and Tables

## Figures

Figure 1.1. Space C2 Process .....	2
Figure 1.2. SSN .....	3
Figure 1.3. Tracked Space Objects, as of April 2022.....	4
Figure 1.4. Higher-Order Perturbation Forces .....	10
Figure 1.5. Relative Positions, Uncertainties, and Velocities of Two Objects .....	14
Figure 1.6. Decomposition of Collision Geometry .....	15
Figure 2.1. SNN (Left) and BNN (Right) .....	21
Figure 3.1. Case Study 1: Elliptical Screening Tool Workflow .....	25
Figure 3.2. Notional Combination Pair Selection .....	27
Figure 3.3. BNN Distance of Closest-Approach Optimization .....	29
Figure 3.4. MAPE Versus Distance of Closest Approach.....	30
Figure 3.5. Root MSE Versus Distance of Closest Approach .....	30
Figure 3.6. Categorization of BNN Output for Group A (Left) and Group B (Right) .....	33
Figure 3.7. Performance Optimization.....	34
Figure 4.1. OSP Tool Workflow .....	37
Figure 4.2. Increase in Covariance Versus Time of Flight .....	38
Figure 4.3. Monte Carlo Trial Convergence Versus Trial Number.....	40
Figure 4.4. Notional Neural Network Model Architecture .....	41
Figure 4.5. BNN Sample Output.....	42
Figure 4.6. BNN and SNN Proof-of-Concept Performance.....	43
Figure 4.7. BNN Error Flagging, Relative Frequency (Left) and MAPE (Right) .....	46
Figure 4.8. Active-Learning Training Curve.....	48
Figure 4.9. Active-Learning Spatial Distribution of KL Divergences Versus Training Epoch .....	48
Figure 4.10. Notional Recursive Propagation .....	50
Figure 4.11. Flexible Time-Step BNN Performance .....	51

## Tables

Table 1.1. CA Screening Volumes.....	12
Table 1.2. Basic CA Reporting Criteria .....	13
Table 2.1. AI Capabilities .....	18
Table 3.1. Classification Matrix .....	32
Table 4.1. BNN and SNN Comparative Performance.....	43



## Chapter 1

# Space Domain Awareness Mission Overview

The third space age began around 2015, and since then we have seen significant changes to the space domain: (1) Space is becoming increasingly congested and contested with the proliferation of low-earth-orbit (LEO) objects and burgeoning interest from both commercial and national actors; (2) the United States is increasingly reliant on space as a major component of military, commercial, and civilian operations; and (3) there is a real risk of space becoming a semipermanently degraded and operationally limited warfighting domain due to intentional or unintentional fragmentation events.<sup>1</sup> The U.S. government's space domain awareness (SDA) mission "encompasses the effective identification, characterization and understanding of any factor associated with the space domain that could affect space operations and thereby impact the security, safety, economy, or environment of our Nation."<sup>2</sup> SDA is vital to maintaining safe and secure access to space, deterring adversary actions in space, and achieving space superiority should a crisis or conflict occur. As a result, SDA plays a pivotal role within the space mission assurance framework.<sup>3</sup>

The changing SDA landscape introduces new requirements and challenges for operators and warfighters. Present-day U.S. Department of Defense (DoD) SDA infrastructure is a complex system of systems from the Cold War era, and the underlying design assumptions were not optimized for today's congested and contested space environment. However, making improvements to the underlying infrastructure has proven to be a challenging task; Space Systems Command has numerous efforts underway to update the software and hardware that facilitate SDA. In parallel with improvements to data infrastructure, artificial intelligence (AI) and machine learning (ML) technologies present one opportunity to improve SDA mission processes by providing operators with methods to quickly analyze large volumes of data. We would like to demonstrate that understanding how to optimally leverage these technologies and planning for their expected impact on the SDA mission could potentially help improve mission effectiveness, optimize resource management, and inform investment decisions. In this chapter, we provide an overview of the technologies, processes, and organizations that compose SDA operations at U.S. Space Command (USSPACECOM). We focus the discussion on select current mission processes that provide context to the AI/ML case studies presented later in the report. Many of the insights are gathered from discussions that team members had while embedded with the 18th Space Defense Squadron (SDS). For a broader

---

<sup>1</sup> SES Space and Defense, "U.S. Space Command Deputy Commander"; U.S. Department of Defense (DoD), *Defense Space Strategy Summary*.

<sup>2</sup> U.S. Space Force, *Spacepower*.

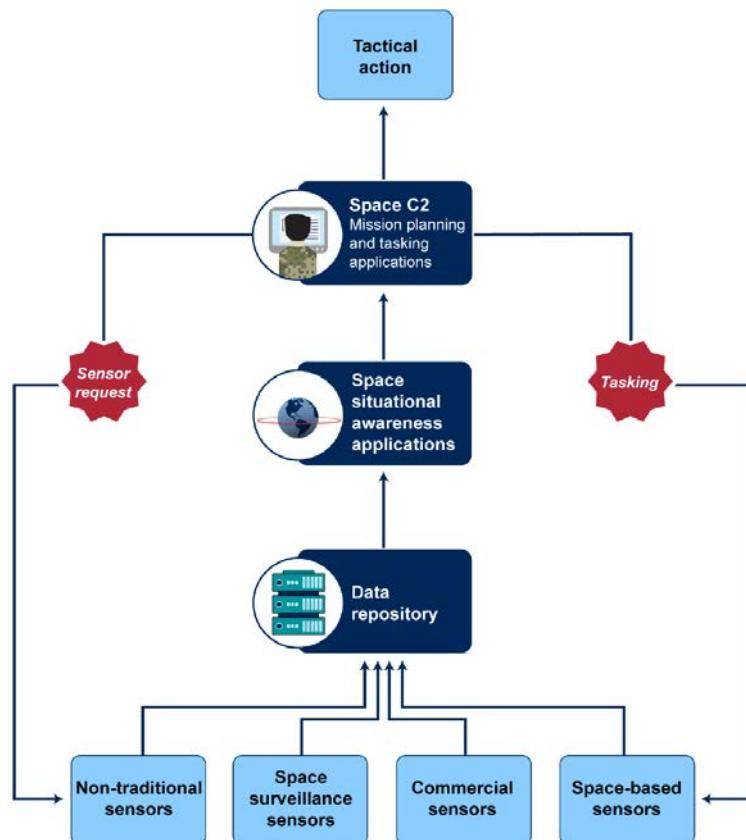
<sup>3</sup> Joint Publication 3-14, *Space Operations*.

discussion of the SDA mission, including future improvements underway, we refer the reader to our companion report.<sup>4</sup>

## Space Command and Control

The space command and control (C2) program is the backbone of SDA operations at USSPACECOM. Depicted by the U.S. Government Accountability Office in Figure 1.1, the program ingests data from a variety of sensor sources into a data repository. The data are then processed using numerous SDA applications. The processed data are then passed along to decisionmakers for both mission planning and execution of tactical operations, as well as for tasking the sensor network for additional observations to refine SDA analyses. This process is iterative and continuously ongoing at the 18th and 19th SDSs, components of Space Delta 2 tasked with performing USPACECOM's SDA mission. Each node in this process is a complex set of subprocesses, which we describe in subsequent sections.

Figure 1.1. Space C2 Process



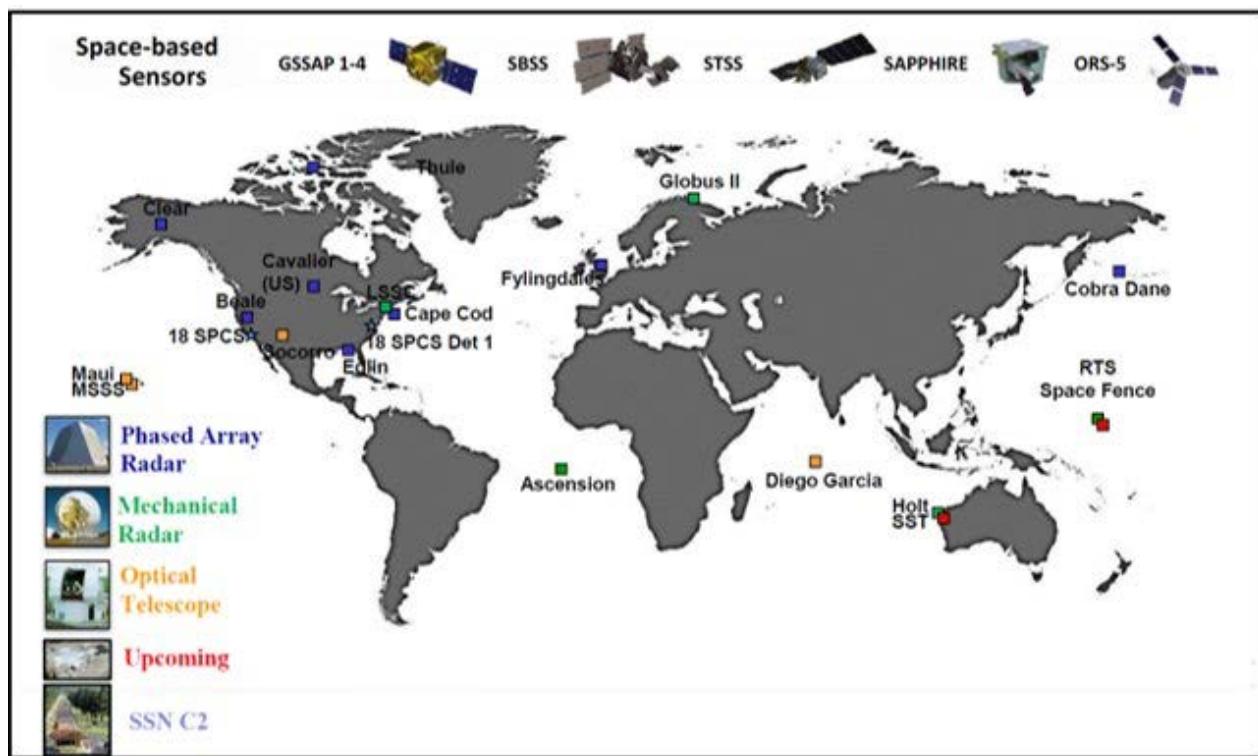
SOURCE: Reprinted from U.S. Government Accountability Office, *Space Command and Control*.

<sup>4</sup> Zhang et al., *Artificial Intelligence and Machine Learning for Space Domain Awareness*.

## Historical Context of SDA

The U.S. government has a long history of executing something akin to SDA, culminating in its prioritization as one of the U.S. Space Force's (USSF's) five core competencies.<sup>5</sup> The system architecture known as the Space Defense Operations Center (SPADOC) and the attendant Space Surveillance Network (SSN), shown in Figure 1.2, trace their histories back to the early 1960s, when they primarily supported ballistic missile defense. In the early days of space operations, operators held that space was so voluminous that the odds of collision were remote—this is sometimes referred to as the *big sky theory*. However, there remained a national security imperative for tracking domestic and foreign objects in orbit. Once someone realized that man-made satellites were merely objects on noncolliding ballistic trajectories, many missile defense sensors became dual-hatted with supporting the new space situational awareness (SSA) mission in their spare time.

Figure 1.2. SSN



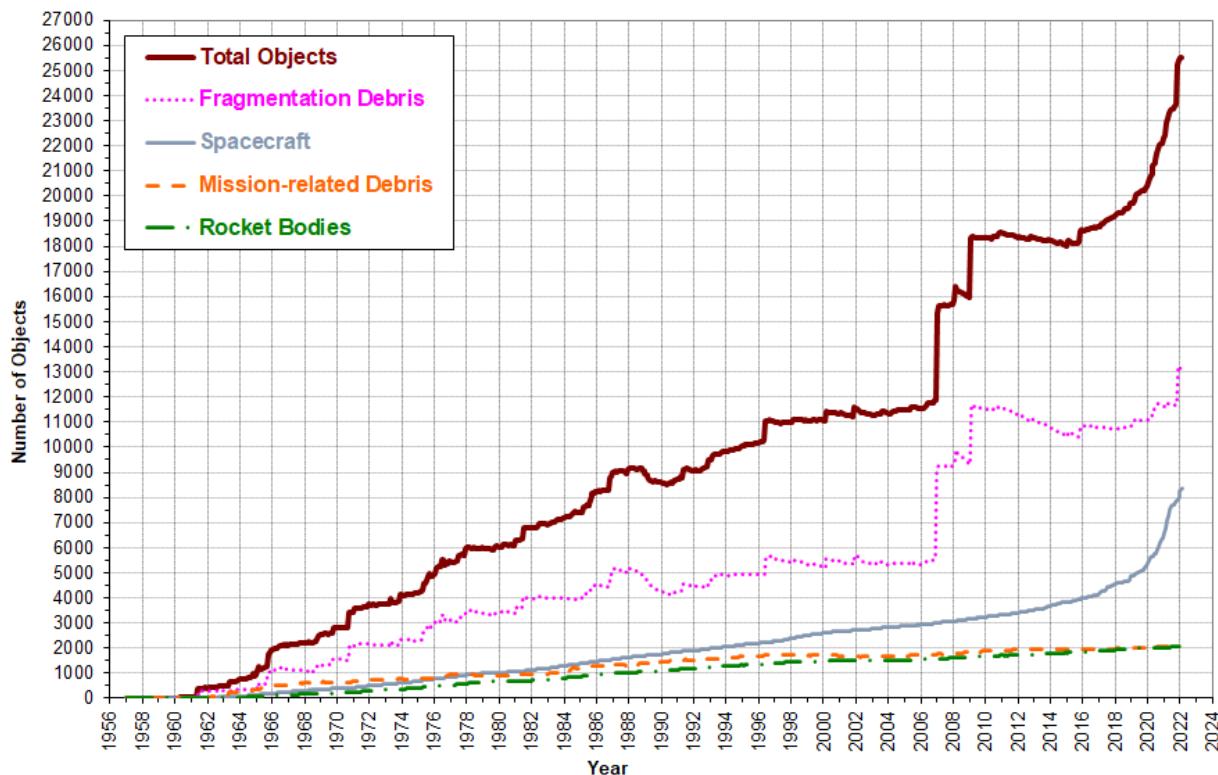
SOURCE: Reprinted from National Aeronautics and Space Administration (NASA), *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*.

NOTE: Det = detachment; GSSAP = Geosynchronous Space Situational Awareness Program; LSSC = Lincoln Space Surveillance Complex; MSSS = Maui Space Surveillance System; ORS = Operationally Responsive Space; RTS = Reagan Test Site; SBSS = Space Based Space Surveillance; SPCS = space control squadron; SST = Space Surveillance Telescope; STSS = Space Tracking and Surveillance System.

<sup>5</sup> For information on the five core competencies, see U.S. Space Force, *Spacepower*.

Over the course of the next few decades following the 1960s, the SSA mission began to grow, alongside the population of space objects, as shown by Figure 1.3. The 18th SDS and its past forms became known for coordinating the detection, identification, and tracking of all earth-orbiting resident space objects (RSOs) above 10 cm. These actions compose the critical task of space catalog maintenance: the bookkeeping of RSO custody over time by way of position and velocity state variables for the purpose of attribution.

**Figure 1.3. Tracked Space Objects, as of April 2022**



SOURCE: Reprinted from Astromaterials Research and Exploration Science, NASA, "Legend."

In October 2019, Air Force Space Command (AFSPC) coined the phrase *space domain awareness* to reflect the acknowledgment of space as a domain of warfare—as stated by then-Maj Gen John Shaw: “[T]he implication of space as a warfighting domain demands we shift our focus beyond the Space Situational Awareness mindset of a benign environment to achieve a more effective and comprehensive SDA.” That environment includes the “identification, characterization and understanding of any factor, passive or active, associated with the space domain that could affect space operations and thereby impact the security, safety, economy or environment of our nation.”<sup>6</sup> Recent doctrinal guidance now identifies SSA as the subset of SDA that focuses on the orbital segment—in particular, the provisional knowledge and characterization of space objects for space safety and

<sup>6</sup> Shaw, “Space Domain Awareness (SDA), Memorandum.”

sustainability.<sup>7</sup> Characterization adds the requirement for some modicum of understanding regarding an RSO's intended usage and capabilities.

## Key SDA Organizations and Systems

The 18th SDS is the component of the USSF Space Delta 2 tasked with executing USSPACECOM's SDA mission out of Vandenberg Space Force Base (SFB). In particular, the 18th SDS coordinates the acquisition and postprocessing of SSN data to monitor activity in space and maintain custody of RSOs. The 18th SDS does so through the aggregation of observations from the SSN: calculating current and predicted orbital states for RSOs, conducting launch detection, tracking, and identifying potential collisions for NASA manned and unmanned assets. The squadron's comprehensive catalog of satellites had more than 46,000 objects, including 6,000 active spacecraft, as of 2022.<sup>8</sup> Activated in 2022, the 19th SDS is also a component of Space Delta 2 and has assumed the mission of conjunction assessment (CA) from the 18th SDS. The 19th SDS serves as a primary backup to the mission functions at the 18th SDS. Together with the 18th SDS, the 19th SDS maintains and populates the publicly available space catalog Space-Track.org.<sup>9</sup> The squadrons' primary mission functions are noted in the box below.

### Primary Mission Functions for 18th and 19th SDSs

USSF Space Delta 2: Prepares and presents combat-ready SDA operations to assigned and attached forces

18th SDS: Squadron within Space Delta 2 tasked with tracking and monitoring space objects, collision prediction, maneuver detection

19th SDS: Squadron within Space Delta 2 tasked with backup to 18th SDS functions, CA, beyond geostationary earth orbit (xGEO) SSA

Combined Force Space Component Command: USSPACECOM subordinate command responsible for the management of sensors used to conduct SSA

Joint Task Force—Space Defense: Protects and defends U.S. space assets, working with the intelligence community through the National Space Defense Center SDA units

SOURCES: USSF, "18th Space Defense Squadron"; Peterson Schriever Space Force Base, USSF, "19th Space Defense Squadron"; U.S. Government Accountability Office, *Space Situational Awareness*.

The two primary systems used by the 18th SDS as part of the SDA mission are the SPADOC and the Command, Analysis, Verification, and Ephemeris Network (CAVENet). SPADOC was brought online in 1979 in the Cheyenne Mountain Complex as a cataloging system, and its last significant upgrade was in 1989.<sup>10</sup> Gen Jay Raymond, then head of AFSPC, said in a speech in 2017:

<sup>7</sup> Joint Publication 3-14, *Space Operations*.

<sup>8</sup> Wilson and McKissock, "18 SDS Small Satellite Support"; U.S. Space Force, "18th Space Defense Squadron."

<sup>9</sup> 18th and 19th SDSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*; Space-Track.org, homepage.

<sup>10</sup> Clark, "What About JMS?"

“I can’t wait until we can take a hammer to SPADOC and just blow it to bits. It’s an old clunker and it’s a catalog system: it’s not a warfighting command and control system.”<sup>11</sup> By the 2000s, the challenge of a growing satellite catalog was apparent, as was the opportunity to take advantage of advances in computing to help address it. This prompted the USSF to develop CAVENet to augment SPADOC’s capabilities and provide data analysis and sharing capabilities on a local network. CAVENet workstations run the Astrodynamics Support Workstation (ASW) software, which provides analytic tools that were originally intended to conduct analysis of historical element sets and observations. CAVENet is an offline system for conducting classified analysis; therefore, this system runs the ASW software for collision analysis and threat assessment at the classified level. As the growing SDA missions continue to outpace SPADOC computing infrastructure, CAVENet has taken on expanding tasks. The ASW software on CAVENet now keeps a separate space catalog for analysis, and CAVENet is responsible for developing the daily tasking plan for the SSN. ASW currently hosts several typically custom-built software tools that maintain the High Accuracy Catalog (HAC) and facilitate analysis to support different SDA tasks, among them Automatic Differential Corrections (ADC), Dynamic Calibration of the Atmosphere (DCA), and Automatic Special Perturbations Ephemeris Generation (ASP).<sup>12</sup>

## Sensor Tasking

Contemporary SSA data originate from various sources, including the SSN, commercial networks, and allies and partners that contribute data from both ground- and space-based sensors. However, the legacy systems underlying USSPACECOM’s SDA mission currently support only observations from the SSN network, an overview of which is presented in Figure 1.2. Data flow from dedicated and nondedicated SDA sensors, the latter of which remain dual-hatted with early warning and missile detection missions. Each sensor in the SSN exhibits different phenomenologies, reliabilities, percentage downtime for maintenance, and precisions; all of these must be accounted for as part of the sensor tasking process. For example, phased-array sites use narrowband radar with electronic steering to cover wide swathes of the sky—the trade-off is that these waves do not provide adequate resolution at higher altitudes and so are mostly limited in use to LEO. Additionally, narrowband radar is more greatly affected by scintillation, the disruption of electromagnetic waves by water molecules suspended in the atmosphere. There are also both ground- and space-based optical sensors that are more suitable for deep space observation, but there are numerous factors that can affect the sensor performance, including weather for ground-based sensors and the orientation of the sun for space-based sensors.<sup>13</sup>

Even under normal operating conditions, there is far more demand on the SSN than availability. SDA tasking must account for the relative priority of observation, so analysts must coordinate observation collection according to sometimes conflicting data needs, priorities, and sensor suitability and availability. For example, high-interest objects (HIOs) may require a higher revisit rate for sensor

---

<sup>11</sup> Quoted in Clark, “What About JMS?”

<sup>12</sup> Omitron, “Astrodynamics Support Workstation (ASW).”

<sup>13</sup> Weeden, “Space Situational Awareness Fact Sheet.”

collection than an object characterized as space debris. In addition to these varying priorities, the sheer data volume, the complexities of orbital mechanics and space operations, and the accelerated operations tempo required to conduct the SDA mission in a crisis or conflict present significant challenges to the orbital analyst (OA).

The daily sensor tasking process begins with the swing-shift OA from the 18th SDS.<sup>14</sup> One of the OA's responsibilities is building the daily composite (COM) file, which includes taskings for all the sensor sites and may also be referred to as the *consolidated tasking list* (CTL).<sup>15</sup> The sensor pointing is based on locally stored sets of two-line elements (TLEs), called *field elements sets*. The sensor sites record their raw observations as time series of range, elevation angle, and azimuth angle, which are then reported back to the 18th SDS to be merged with data across all sensor sites. The cyclical process is then completed with an update of the HAC using the orbit determination process described below.<sup>16</sup>

## Orbit Determination

*Orbit determination* describes the process employed by the 18th SDS for determining a space object ephemeris from raw sensor observations. These raw observations are collected from a multitude of sensor types and are composed of such data as range, range rate, azimuth angle, and elevation angle—all with respect to the sensor site's local frame of reference. The raw data collected by the sensor sites accumulate on SPADOC's Space Data Server. Each observation is then transformed into a common reference frame and reformatted into cartesian position and velocity vectors for compatibility with the CAVENet's ASW software suite.

An ephemeris represents an average state calculated from numerous individual observations according to a linear algebra regimen known as *differential corrections* (DC).<sup>17</sup> Strings of ephemerides are then referred to as *tracks* if they were calculated from past observations in the space catalog or *trajectories* if they were predicted into the future using orbit propagation methods. The ADC tool is the nominal, automated tool in charge of calculating new HAC ephemerides as new observations come into CAVENet from SPADOC. ADC applies a weighted root-mean-squared threshold to screen the combined error carried forward into each new ephemeris from its constituent observations while rigorously accounting for the underlying uncertainties; those ephemerides failing this test get flagged for manual inspection by human analysts.<sup>18</sup> ADC then produces both a mean estimate of the state and a covariance for the RSO.

The NASA human spaceflight orbital safety analyst (OSA) initiates the process of updating the 18th SDS's predictions for all future RSO trajectories from the most-recent ephemerides produced by ADC. The resulting trajectory updates go on to inform the CA process performed by the 19th SDS

---

<sup>14</sup> A *swing shift* is the eight-hour shift between 1600 and 0000 military time.

<sup>15</sup> 18th and 19th SDSs, *Combined Force Space Component Command, Spaceflight Safety Handbook for Satellite Operators*, p. 6.

<sup>16</sup> NASA, *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*.

<sup>17</sup> For more information on differential corrections, we recommend that the reader reference Vallado, *Fundamentals of Astrodynamics and Applications*, Chapter 10.

<sup>18</sup> NASA, *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*.

and NASA, facilitate SSN sensor pointing, and support on-orbit operations by the commercial owner-operator (O/O), among other tasks.

A process known as *extended general perturbations* (EGP) converts the RSO state vectors contained within the HAC into the canonical TLE format long used by SPADOC and the broader space community to represent RSO states. The TLEs that are transferred and cataloged in SPADOC for internal use are known as *operational element sets* and are updated as frequently as new values arrive.

## Measurement Uncertainty

Measurement uncertainty begins at the observation level and is carried forward through each downstream SDA mission process. Remote sensor observation is DoD's primary method for measuring RSO state, so it is important to characterize the numerous sources and impacts of uncertainty—this is the primary reason for performing DC. Note that the satellite O/O oftentimes has access to more-precise methods of orbit determination, such as GPS navigation or transponder pings. USSPACECOM can integrate these data into the SDA mission process but has no authority to require the O/O to provide them. This situation is especially true for adversarial satellites, noncooperative space objects, and inert debris. Sensor uncertainty is quantified using covariance, the matrix of values containing the standard deviation of each state variable with respect to itself and the other state variables. The standard deviation of a given observation state ultimately depends on such sources of uncertainty as structural modeling uncertainties, dynamic parameter uncertainties, measurement noise and sensor bias, compounding numerical errors, and hardware and software faults.<sup>19</sup>

To characterize these sources of uncertainty, the sensor sites regularly calibrate against more-accurate independent observations. For example, there exists a network of laser calibration sites concurrent with the SSN that provide highly accurate observations of specialized calibration satellites.<sup>20</sup> Additionally, certain satellite constellations—such as GPS or the Tracking and Data Relay Satellite system—accurately account for their states as part of their core functions and therefore can also provide reference for calibration.<sup>21</sup>

The state of an RSO can never be definitively established because of these and other sources of uncertainty. However, it is possible to reduce the relative impact of the combined uncertainty on the final-state estimate using such methods as nonlinear batched least squares. Numerous analysis methodologies leverage the standardized behavior characteristic of otherwise random normal distributions of observations to facilitate the SDA mission.<sup>22</sup>

---

<sup>19</sup> Poore, Aristoff, and Horwood, *Covariance and Uncertainty Realism in Space Surveillance and Tracking*.

<sup>20</sup> NASA, *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*.

<sup>21</sup> National Coordination Office for Space-Based Positioning, Navigation, and Timing, "Control Segment."

<sup>22</sup> Vallado, *Fundamentals of Astrodynamics and Applications*.

## Simplified General Perturbation Propagation

Simplified general perturbation (SGP) is a group of standard mathematical algorithms used to calculate future orbital state vectors and trajectories of an RSO. The SGP algorithm was originally developed in 1959 by Yoshihide Kozai, updated by C. G. Hilton and J. R. Kuhlman in 1966, and later made publicly available in 1980.<sup>23</sup> Since its release, it has become the most widely used orbit propagator in the world. Over time, there have been numerous updates, enhancements, and refactoring efforts, but the fundamental algorithm has remained largely unchanged. More recently, the USSF has continually released “an official implementation of the SGP4 algorithm that is used in operations as part of the AstroStds software library.”<sup>24</sup> For a mathematical description of the different algorithms, we refer the reader to *Spacetrack Report No. 3*, compiled by CelesTrak.<sup>25</sup>

The SGP algorithms use TLE sets and employ either an analytic or semianalytic approach for orbit propagation. Although SGP4 propagation is now widely adopted in various applications, it was originally designed to offer fast and computationally efficient trajectory predictions, where a certain level of error was deemed acceptable. One limitation to the general perturbation methods is it does not produce a covariance matrix requiring new techniques, such as special perturbation, to be developed.<sup>26</sup>

SGP4-XP was an algorithm developed by the USSF that was released in December 2020 and was designed for the propagation of TLEs featuring extended perturbations. The release notes associated with this release claim that SGP4-XP is appropriate for applications that require HAC-level accuracy.<sup>27</sup> Although the SGP4-XP algorithm expands on the capabilities of its predecessor, SGP4, with minimal additional requirement in terms of input TLE format and computational cost, SGP4-XP is still an analytic method that does not produce a covariance matrix. At the time of writing, we are uncertain whether SGP4-XP is being used in an operational context by either the 18th or the 19th SDS.

## Special Perturbation Propagation

The 18th SDS leverages a more accurate special perturbation (SP) methodology for maintaining its internal HAC and propagating future trajectories for conjunction screening. SP algorithms are necessary for precise orbit determination because they facilitate bookkeeping of RSO state uncertainty by taking a numerical, sequential approach to orbit propagation. This approach involves dividing the satellite’s motion into discrete time steps and calculating each following position and velocity based on the effect of higher-order perturbation forces modeled across the time step. Such higher-order forces extend beyond the idealized two-dimensional gravitational pull from earth to include the temporal difference of gravity caused by earth’s oblateness (equatorial bulge is referred to by the nomenclature  $J_{l,m}$ , representing the different-order zonal, sectoral, and tesseral harmonics), other nearby celestial

<sup>23</sup> Kozai, “The Motion of a Close Earth Satellite”; Hilton and Kuhlman, “Mathematical Models for the Space Defense Center”; Der and Danchick, “Conversion of Osculating Orbital Elements to Mean Orbital Elements.”

<sup>24</sup> Conkey and Zielinski, *Assessing Performance Characteristics of the SGP4-XP Propagation Algorithm*.

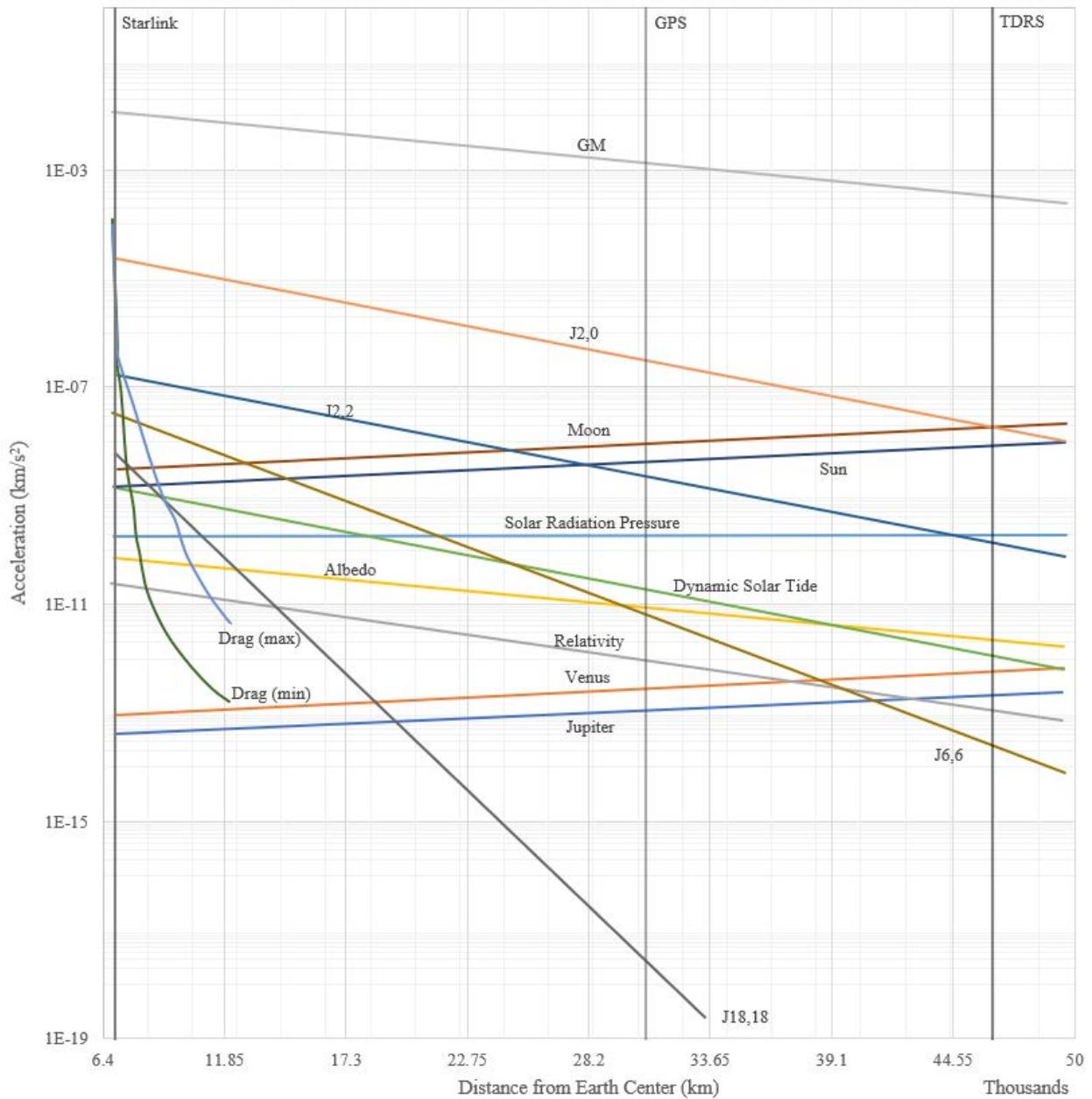
<sup>25</sup> Hoots and Roehrich, *Spacetrack Report No. 3*.

<sup>26</sup> Vallado and Crawford, “SGP4 Orbit Determination.”

<sup>27</sup> Conkey and Zielinski, *Assessing Performance Characteristics of the SGP4-XP Propagation Algorithm*.

bodies, and other irregularities, such as atmospheric drag and solar radiation pressure.<sup>28</sup> These higher-order perturbations are presented in Figure 1.4.

**Figure 1.4. Higher-Order Perturbation Forces**



SOURCE: Adapted from Montenbruck and Gill, *Satellite Orbits*.

NOTE: TDRS = Tracking and Data Relay Satellite.

<sup>28</sup> Vallado shared this nonexhaustive list of higher-order force terms: two-body gravity, nonsphericity of central body, atmospheric drag and lift, third-body perturbations, solar radiation pressure, thrust, magnetic fields, solid earth and ocean tides, and earth reradiation (albedo) (Vallado, *Fundamentals of Astrodynamics and Applications*).

The specific SP implementation in use operationally has not been publicly released, but we can speculate based on methodologies described by NASA to guide our AI algorithm development.<sup>29</sup> A commonly cited algorithm for SP is Cowell's method, which simply performs a linear summation of the perturbing forces.<sup>30</sup> Cowell's method assumes that the perturbing forces are small relative to the primary Keplerian forces, requires a high-order numerical integration method to evolve the equations of state, and relies on the user to account for many significant digits.

Previous sections discussed the sensor measurement uncertainty captured at the time of observation and accounted for by the DC process to produce HAC ephemerides. There is also propagation error present in numerical methods: errors or uncertainties in the input data or initial conditions of a computational problem that can affect the accuracy of the computed results. Numerical methods involve approximations and computations, and even small errors in the input data or numerical procedures can accumulate and lead to significant discrepancies in the final output. Forms of numerical errors include roundoff error, truncation error, and algorithmic error, all of which can exacerbate the initial measurement uncertainty. These numerical errors are essential for the CA process and are output by the SP algorithms in the form of a covariance matrix.

## CA Process

CA is the probabilistic process by which the 18th and 19th SDSs quantify risk of collision in space. The process is probabilistic rather than deterministic because of the numerous and chaotic orbital forces that progressively perturb an object's trajectory away from its theoretical path over time. Because the closing velocities involved in space collisions necessitate avoidance maneuvering far in advance of the predicted time of closest approach (TCA), it is (currently) impossible to know exactly where the two colliding objects will be at the TCA. The CA process flags pairs of RSOs that are at high risk of collision.

As mentioned above, in the early days of space operations, operators held that the space domain was so voluminous that the odds of collision were remote—this was sometimes referred to as the *big sky theory*. Subsequent collisions demonstrated the invalidity of this assumption,<sup>31</sup> as well as the outsize impact of even a single collision on the health of the space environment, leading to broad agreement on the need to assess the risk of collision between space objects; specifically, the risk of collision requires a dynamic assessment of risk that could support continuous ingestion of new information to improve prediction confidence and avoid unnecessary maneuvering.<sup>32</sup> Probability of collision ( $P_c$ ) serves as this measure of confidence in a predicted collision actually occurring. Increasing  $P_c$  accuracy is thus a key objective for CA; to achieve it, the 18th and 19th SDSs leverage their orbit determination and sensor tasking processes to aggregate raw sensor observations in sufficient quantities and qualities.

---

<sup>29</sup> NASA, *Application and Analysis of Satellite Orbit Prediction Techniques*.

<sup>30</sup> Moulton, *Introduction to Celestial Mechanics*.

<sup>31</sup> Weeden, “2007 Chinese Anti-Satellite Test Fact Sheet.”

<sup>32</sup> Alfriend et al., “Probability of Collision Error Analysis.”

The 19th SDS is currently responsible for conducting CA for all space objects. Using the latest RSO epoch states and trajectories from the orbit determination process, the 19th SDS continuously conducts CAs using the ASW tools Automatic Conjunction Assessment (AutoCA) and Super Computation of Miss Between Orbits (SuperCOMBO). AutoCA performs the *all versus all* initial screenings that check each HAC object against every other HAC object for five to ten days into the future, depending on the orbital regime (see Table 1.1).<sup>33</sup> For those satellites whose O/Os provide screening ephemerides, the 19th SDS performs additional all versus O/O and O/O versus O/O screenings.

**Table 1.1. CA Screening Volumes**

Orbital Regime	Orbital Regime Definition	Propagation (days)	Radial (km)	In-Track (km)	Cross-Track (km)
HAC screening	Deep space— ellipsoid	1,300 min. < period < 1,800 min. Eccentricity < 0.25 Inclinations < 35 degrees	10	10	10
LEO 4— covariance	1,200 km < perigee < 2,000 km Eccentricity < 0.25	5	0.4	2	2
LEO 3— covariance	750 km < perigee < 1,200 km Eccentricity < 0.25	5	0.4	12	2
LEO 2— covariance	500 km < perigee ≤ 750 km Eccentricity < 0.25	5	0.4	25	25
LEO 1— covariance	Perigee ≤ 500 km Eccentricity < 0.25	5	0.4	44	51
O/O screening	Deep space— Ellipsoid Near earth— covariance	Period > 225 min. Period ≤ 225 min.	10 7	20 25	20 25

SOURCE: Features data from 18th and 19th SDSSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*.

<sup>33</sup> 18th and 19th SDSSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*, p. 10.

The 19th SDS applies the same underlying screening mathematics across all orbital regimes. Potential collisions are flagged by AutoCA when a secondary RSO's trajectory pierces the perimeter of an ellipsoid centered on the primary object being tested. For deep space objects, an elliptical screening method is employed in lieu of calculating  $P_c$ . All other RSOs are analyzed using a covariance-based screening approach that employs "a joint uncertainty volume that is based on time-varying covariance values for the primary and secondary objects" at TCA.<sup>34</sup> Depending on the results of this test, the 19th SDS may deem it necessary to refine the orbits of the two objects to obtain a more confident assessment of collision risk—additional SSN taskings and updated orbit determinations may be requested from the 18th SDS. Whether or not the collision possibility is reported to the O/O depends on the reporting criteria found in Table 1.2.

SuperCOMBO handles this iterative refinement and rescreening process, freeing AutoCA to continue screening the vast majority of RSOs.

**Table 1.2. Basic CA Reporting Criteria**

<b>Notification Method</b>	<b>Space-Track.org Criteria: Conjunction Data Message</b>	<b>Emergency Criteria: Conjunction Data Message and Close Approach Notification Email</b>
Deep space HAC	TCA ≤ 10 days & overall miss ≤ 5km	TCA ≤ 3 days & overall miss ≤ 5km
Deep space O/O ephemeris	TCA ≤ 10 days & overall miss ≤ 5km	TCA ≤ 3 days & overall miss ≤ 5km
Near earth HAC	TCA ≤ 3 days & overall miss ≤ 1 km & $P_c \geq e^{-7}$	TCA ≤ 3 days, overall miss ≤ 1 km & $P_c \geq e^{-4}$
Near earth O/O ephemeris	TCA ≤ 3 days & overall miss ≤ 1 km & $P_c \geq e^{-7}$	TCA ≤ 3 days & overall miss ≤ 1 km & $P_c \geq e^{-4}$

SOURCE: Features data from 18th and 19th SDSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*.

NOTE: Different reporting criteria may be used for O/Os with advanced reporting agreements with USSPACECOM.

The exact calculation of  $P_c$  may proceed from several different formulations. One such formulation is the long-practiced two-dimensional-projection methodology. This method applies a number of assumptions to enable decomposition of the problem from three to two dimensions: (1) near-instantaneous rectilinear motion, (2) negligible velocity uncertainty, (3) constant position uncertainty, and (4) linearly independent Gaussian uncertainty distributions. Additionally, the objects' geometries are greatly simplified down to circumscribing spheres; these are combined to form a single combined object sphere—a *zone of exclusion*—whose violation is taken to imply collision (Figure 1.5, panel a).

This zone of exclusion is collocated with the estimated location of the primary object at TCA. The assumption of linearly independent Gaussian uncertainty distributions allows the superposition

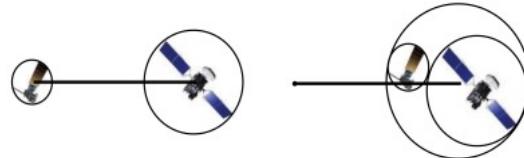
---

<sup>34</sup> 18th and 19th SDSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*.

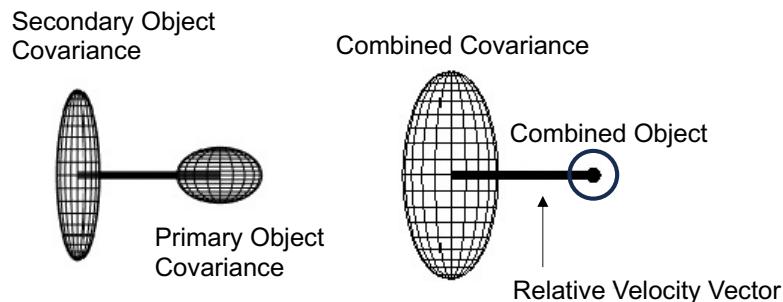
of both objects' uncertainty ellipsoids into a single distribution (Figure 1.5, panel b), by convention collocated with the secondary object. Finally, the velocities for the two objects are combined into a single relative velocity vector, which is the velocity bringing them closer together or farther apart relative to the primary object.

**Figure 1.5. Relative Positions, Uncertainties, and Velocities of Two Objects**

**(a) Combined Object Size (Exclusion Zone)**



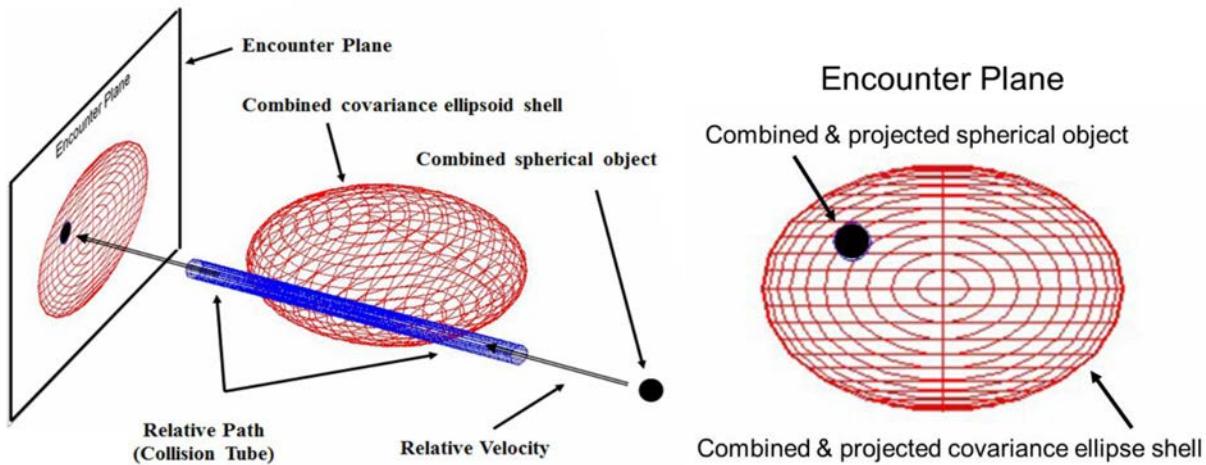
**(b) Combined Covariance**



SOURCE: Adapted from NASA, *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*, pp. 109–110.

Figure 1.6 applies these two concepts, combined object and combined covariance, into three dimensions. By associating all positional uncertainty around the secondary object's estimated location and all object sizes around the primary object's estimated location (Figure 1.5, panel b), the path of the combined exclusion zone may be taken to be exact. Thus, an *encounter plane* drawn orthogonally to the relative velocity vector will have a 100 percent  $P_c$  with the combined exclusion zone in the direction defined by this vector, leaving the two dimensions defined by the encounter plane as the sole determinants of overall  $P_c$ .

**Figure 1.6. Decomposition of Collision Geometry**



SOURCE: Reproduced from Alfano and Oltrogge, “Probability of Collision,” Figures 2 and 3 (CC BY 4.0); see also Alfano, “Relating Position Uncertainty to Maximum Conjunction Probability.”

The right side of Figure 1.6 represents the final two-dimensional geometry of the conjunction at TCA. This geometry is by definition static at TCA, so the  $P_c$  at TCA may be taken as the area integral of the combined uncertainty distribution over the location of the zone of exclusion.<sup>35</sup>

$$P_c = \frac{1}{2\pi|Det(C)|^{1/2}} \iint_{x^2+y^2 \leq d^2} e^{-\frac{1}{2}(\vec{r}-\vec{r}_{S/P})^T C^{-1} (\vec{r}-\vec{r}_{S/P})} dx dy,$$

where

- $C$  is the combined position covariance matrix at TCA
- $d$  is the combined exclusion zone’s radius
- $\vec{r} = (x, y)^T$  is any given point in the collision plane
- $\vec{r}_{S/P} = (r_{s/p}, 0)^T$  is the position of the secondary object with respect to the primary (being along the x-axis in the encounter plane).

## Future Data Architecture Uncertainties

To meet the emerging SDA challenges, there is a need to decommission and replace legacy systems used for this mission. The recognition of this need is not new, and finding a way to modernize or replace these systems has been a goal for DoD since the 1980s.<sup>36</sup> One particular challenge emerges

<sup>35</sup> 18th and 19th SDSs, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*.

<sup>36</sup> See, for example, Albon, “US Space Force Preparing to Decommission Legacy Command and Control System.” Attempts to modernize SPADOC began in the 1980s. Notably, in 2009, the Joint Space Operations Center Mission System was yet another attempt to replace SPADOC, but only one of three proposed increments was fielded, and the program was well behind schedule and over budget.

because sensor systems in the SSN were designed during the Cold War to address threats from nuclear weapons. Replacing computing and software to better address the complexities of today's SDA must be done without disrupting the ability to continue to monitor, detect, and assess these nuclear threats. Another challenge is turning the SSN sensor data into actionable information, requiring computers and software to integrate, manage, and analyze these data.

An effort to modernize the computing and data infrastructure of the SDA is currently underway at such organizations as L3Harris with the Advanced Tracking and Launch Analysis System and other products managed by Kobayashi Maru and Space Systems Command, Special Programs Directorate.<sup>37</sup> Although recommendations on additional computing power, improvements to data infrastructure, and workforce training are important, they are beyond the scope of this study.

The need for accurate SSA has never been more widely distributed across actors in space as it is today. DoD finds itself supporting not only the national security space architecture but also scores of multinational commercial O/Os. Government-subsidized CA has artificially warped the economic calculus for potential commercial SSA providers, although more than a few currently exist. Research by these and other academics has called into question the validity of USSPACECOM's legacy CA methodology. As the orbital environment, and in particular LEO, grows increasingly crowded, conjunctions pose a growing threat. The transfer of non-DoD CA to the Department of Commerce may free up USSPACECOM personnel to interrogate their internal assumptions, and the department will be unencumbered by the same national security concerns that constrain innovation at the 18th and 19th SDSs.

Algorithms that support the SDA mission, whether AI or otherwise, should be developed and implemented with future operating conditions in mind. Additionally, SDA is likely to be augmented by non-SSN data in the future as the commercial and civil SSA sectors build out capabilities. AI/ML applications should be compatible with such an environment to ensure that future operations are supported effectively.

---

<sup>37</sup> For more information on the Kobayashi Maru product portfolio, we refer the reader to Zhang et al., *Artificial Intelligence and Machine Learning for Space Domain Awareness*.

## Chapter 2

# Artificial Intelligence and Machine Learning Overview

In this chapter, we introduce the two case studies we developed to better understand the application of AI and ML for the SDA mission. We also provide a few examples of current efforts to leverage different AI capabilities and the technical background required to understand our methodology. For a broad discussion of the application and impact of AI for the SDA mission and implications for policymakers and decisionmakers, we refer the reader to the companion report.<sup>38</sup> We also note that *AI*, *ML*, and *deep learning* are phrases that are often perceived by the general public as interchangeable but, though related to one another, have distinct definitions.<sup>39</sup> A comprehensive discussion of AI taxonomy is not within the scope of this report; thus, for the remainder of this work, we will use the term *AI/ML* inclusively when referring to concepts within this taxonomy. However, we acknowledge that AI/ML is a multifaceted domain with many subcomponents, some of which may more directly encompass the case study models discussed in later sections of this report.

## Current AI/ML Efforts in SDA

AI/ML technologies, which offer many capabilities with different degrees of complexity, are being leveraged to build scalable, flexible solutions to challenges in a wide variety of sectors. Table 2.1 presents a subset of core AI/ML capabilities and brief descriptions about the utility of each, although this subset is certainly not exhaustive or mutually exclusive. Given the breadth of these capabilities and their overlap with SDA processes, many organizations have recently explored how these tools can be adapted to address SDA needs.<sup>40</sup> Because of the high-stakes nature of many SDA processes, many decisions are currently too complex, high stakes, and nuanced to automate fully, even with advanced AI/ML systems. Therefore, many efforts to integrate AI/ML into SDA processes have focused on decision support (providing suggestions to operators) or automation of lower-stakes, repetitive tasks to free up analyst bandwidth for more-complex decisionmaking.

---

<sup>38</sup> Zhang et al., *Artificial Intelligence and Machine Learning for Space Domain Awareness*.

<sup>39</sup> We refer to U.S. Code, Title 15, Section 9401, which provides policy-oriented definitions for AI and ML. *Artificial intelligence* is defined as “a machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations, or decisions influencing real or virtual environments.” *Machine learning* is “an application of artificial intelligence that is characterized by providing systems the ability to automatically learn and improve on the basis of data or experience.” Therefore, we consider AI/ML as one broad category that provides capabilities. These capabilities follow the standard text of Russell and Norvig, *Artificial Intelligence*, with additional insight from Wehbe and Ramdas, “Introduction to Machine Learning.”

<sup>40</sup> Aptima, Inc., “Space Domain Awareness”; Booz Allen Hamilton, “Artificial Intelligence for Space Missions.”

**Table 2.1. AI Capabilities**

AI Capability	Description
Computer vision	Models that ingest visual imagery and perform such tasks as object identification, classification, and detection
Automated planning and scheduling, as well as execution	Systems optimized to identify a sequence of actions that leads to a defined goal in a complex, multidimensional decision space
Prediction and classification	Discriminative models that perform categorizations of inputs or predict future elements of a sequence, often by drawing on historical data
Generative learning	Systems that synthetically produce language, images, or decisionmaking plans by leveraging past data
Decision support systems	Rule-based systems created from expert and heuristic knowledge used to either support operators or automate tasks
Natural language processing	Systems designed to ingest, process, or generate human language for the purposes of topic modeling, text classification, and the synthesis of human-quality text

To provide additional context on existing AI for SDA integrations, we also provide a subset of core SDA operational needs and describe a subset of ongoing efforts in the academic, private, and government sectors to address each need using AI capabilities. This list is not meant to be exhaustive, and items within it are not necessarily mutually exclusive; however, we hope that this overview paints a partial picture for readers of the current AI for SDA landscape:

- Increased sensor detection and accuracy: Space Systems Command, Space Domain Awareness (SZG), within Space Delta 2, 15th Space Surveillance Squadron, is developing computer vision algorithms to improve the detection rate and precision for satellite observations in the Machine Learning for Space Superiority (MISS) program. The SensorCapsule program connects nontraditional sensors to the unified data library, enabling increased use of commercial sensing data. Finally, the mission-driven, autonomous, collaborative, heterogeneous, intelligent, network architecture (MACHINA) is an ongoing effort to optimize sensor management through sensor orchestration.<sup>41</sup>
- AI sandbox: Space Systems Command/SZG has the Pivot SDA portfolio and aims to provide a digital sandbox for AI and automation tools. This tool interfaces with unified data library data and enables individual providers to develop analytic tools that suit their specific

---

<sup>41</sup> Fletcher, “AI and Autonomy for SDA.”

needs on a government-owned platform, removing barriers to third-party development of tools and more-rapid deployment of models.<sup>42</sup>

- Anomaly detection: The Aerospace Corporation published research on calibrating uncertainties in AI anomaly detection algorithms.<sup>43</sup> Aptima, a private firm that was awarded a SpaceWERX contract, is developing unsupervised AI algorithms to detect anomalous maneuvers.<sup>44</sup> The Massachusetts Institute of Technology is researching the performance of tree-based algorithms and neural networks in identifying object maneuvers through simulations.<sup>45</sup> Commercial organizations, such as ExoAnalytic Solutions, are also researching a variety of ML techniques and methods to perform anomaly detection and classification.<sup>46</sup>
- Data integration: Palantir won a \$32.5 million contract to develop an AI-powered software platform to merge data from the USSF, the U.S. Air Force, and North American Aerospace Defense Command.<sup>47</sup> SCOUT Space Inc. earned a small-business innovation research contract to develop its AI-centric software platform, Vision, for merging sensor data from multiple satellites to track space objects.<sup>48</sup>
- CA: Booz Allen Hamilton is developing a platform for mapping space-object trajectories and assessing the downstream outcomes of potential conjunction-avoiding maneuvers.<sup>49</sup> The University of Colorado, Boulder, won a \$5.5 million contract to automate collision-avoidance maneuvers.<sup>50</sup> Other universities and private organizations are launching efforts to aid the USSF in this area.

## Selection of Two Case Studies

In this section, we explore how the CA process may be improved through AI/ML tools. The goal is to use AI/ML as a function approximator to provide fast, albeit less accurate, trajectory estimations, thereby speeding up bottlenecks in existing processes. The two case studies were used to inform whether AI/ML tools could feasibly meet this goal in two aspects of the CA process.

The first case study imitates the elliptical screening process (ESP), and the second attempts to predict the future state and covariance of a given RSO ephemeris. As stand-alone applications, these two case studies may assist operators in key but limited aspects of SDA. Cooperatively, the case

---

<sup>42</sup> Space Systems Command, “Space Systems Command’s New TAP Lab Kicks Off with Project Apollo to Accelerate Innovation in Space Domain Awareness.”

<sup>43</sup> Woodward et al., “Uncertainty Quantification for Trusted Machine Learning in Space System Cyber Security.”

<sup>44</sup> Aptima, Inc., “Space Domain Awareness.”

<sup>45</sup> Perovich, Folcik, and Jaimes, “Satellite Maneuver Detection Using Machine Learning and Neural Network Methods Behaviors.”

<sup>46</sup> Ingram et al., “ExoALERT”; Lane et al., “Using Machine Learning for Advanced Anomaly Detection and Classification”; Temple, “Synthetic Heterogeneous Anomaly and Maneuver—Neural Network Event Winnowing System (SHAM NNEWS).”

<sup>47</sup> Erwin, “Space Force to Increase Spending on Technologies to Turn Data into Knowledge.”

<sup>48</sup> Erwin, “Scout Wins Defense Contract to Demonstrate Utility of Commercial Data from Sensors in Space.”

<sup>49</sup> Woodie, “Why AI Is a Critical Capability for the US Space Force.”

<sup>50</sup> Zehnder, “CU Boulder Lands \$5.5 Million Air Force Project to Advance Orbital and AI Research.”

studies may alleviate the computational and manpower demands of CA analysis at a more significant level.

These case studies are selected for four primary reasons: (1) relevance to SDA operational needs, (2) the availability of training data representative of what is used by SDA operators, (3) a desire to deconflict with use cases being studied at other organizations, and (4) a desire to evaluate the extent to which SDA ML solutions can be prototyped on short time frames using open-source code.

With these considerations in mind, we observed that covariance propagation and elliptical screening are focus areas in which case study training sets can be built through internal RAND systems and are spaces that are relatively unsaturated by other existing AI/ML in SDA efforts, marking them as well-suited case study candidates for our report.

Lastly, the case studies were developed with consideration of current SDA processes. Although they may still have applicability if SDA infrastructure undergoes significant changes, as anticipated in the future, many of the design considerations and performance benchmarks would need to be reevaluated.

## Neural Network Architecture

In this section, we provide an overview of the main AI/ML architectures leveraged within our case studies, as well as the performance metrics used to evaluate them.

### Standard Neural Networks

Standard neural networks (SNNs) are ubiquitous AI/ML architectures that serve as the foundation for large language models, computer vision platforms, and audio processing systems. Although more-thorough backgrounds on neural networks are provided elsewhere,<sup>51</sup> we provide a summary here. SNNs generally consist of a series of input, hidden, and output network layers that themselves are populated with network neurons, each of which is connected to neurons within other layers through linear combination operations and nonlinear activation functions. The weights and biases within each linear combination operation can be tuned to minimize a network objective function through a process known as *backpropagation*. The combination of tunable model parameters and nonlinear activation functions enables an SNN to capture highly nuanced, sophisticated relationships present within its underlying training sets.

### Bayesian Neural Networks

A Bayesian neural network (BNN) is a probabilistic approach to neural networks that extends standard feedforward neural networks by incorporating Bayesian principles.<sup>52</sup> BNNs provide a framework for modeling uncertainty in neural network predictions, making them particularly useful in

---

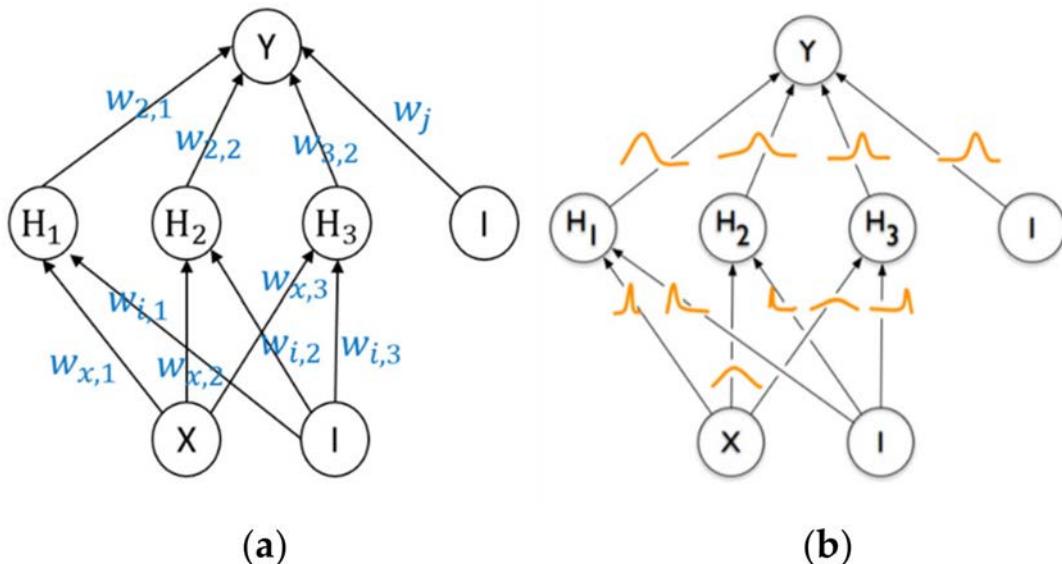
<sup>51</sup> Alzubaidi et al., “Review of Deep Learning.”

<sup>52</sup> Jospin et al., “Hands-On Bayesian Neural Networks—a Tutorial for Deep Learning Users.”

tasks in which uncertainty quantification is essential, such as in medical diagnosis,<sup>53</sup> autonomous driving,<sup>54</sup> and financial forecasting.<sup>55</sup>

Like SNNs, BNNs consist of interconnected layers of neurons, including input, hidden, and output layers. These neurons apply activation functions to their inputs and collectively learn to map input data to output predictions. In BNNs, instead of using fixed weights, as in SNNs, the parameters (weights and biases) of each neuron are treated as probability distributions. These distributions capture the uncertainty in the parameters. Common choices for these distributions include Gaussian distributions, but other distributions,<sup>56</sup> such as scale mixtures or Cauchy distributions, can also be used, depending on the specific problem. An illustrative difference between standard and Bayesian neural networks is presented in Figure 2.1.

**Figure 2.1. SNN (Left) and BNN (Right)**



SOURCE: Reprinted from Yu et al., "Structural Health Monitoring Impact Classification Method Based on Bayesian Neural Network," Figure 1 (CC BY 4.0).

NOTE:  $W$  represents a scalar weight that conveys the importance of that corresponding feature.

To make predictions for a new input data point, the BNN performs a prediction by sampling from the posterior distribution of the underlying model parameters. This process yields multiple predictions, each corresponding to a different set of sampled weight and bias values. The uncertainty in the predictions arises from the uncertainty in the parameter samples. One of the primary advantages of BNNs is their ability to provide uncertainty estimates along with their predictions. This uncertainty quantification can be used to assess the model's confidence in its predictions. For instance,

<sup>53</sup> McLachlan et al., "Bayesian Networks in Healthcare."

<sup>54</sup> Michelmore et al., "Uncertainty Quantification with Statistical Guarantees in End-to-End Autonomous Driving Control."

<sup>55</sup> Chandra and He, "Bayesian Neural Networks for Stock Price Forecasting Before and During COVID-19 Pandemic."

<sup>56</sup> Fortuin, "Priors in Bayesian Deep Learning."

a wider posterior distribution indicates higher uncertainty, whereas a narrower distribution indicates lower uncertainty.

BNNs can be computationally intensive and require specialized inference techniques. Additionally, given their modeling of uncertainties, BNNs generally require a larger number of parameters than SNNs of the same neuron number, leading to increased training times and potential hyperparameter optimization challenges. Finally, choosing appropriate prior distributions and inference methods is also crucial to application success and therefore is an active area of research.<sup>57</sup>

## Managing Uncertainty

In modeling problems, there are two sources of uncertainty that are typically discussed: aleatoric and epistemic.<sup>58</sup> Both combine to form the aggregate uncertainty of a model.

*Epistemic uncertainty* refers to uncertainty related to a model's inability to fully capture the patterns within its training data. Generally speaking, no model is perfect at reproducing the statistical process it is attempting to represent, and epistemic uncertainty characterizes the degree by which a model fails to do so. Epistemic uncertainty can often be reduced by increasing the amount of data a model is trained in, leveraging more-sophisticated model architectures or modifying optimization algorithms.

*Aleatoric uncertainty*, on the other hand, refers to uncertainty inherently present within the training dataset used to construct a model. A training set obtained from a real-world phenomenon may contain two rows of vectors with identical inputs yet different output values. A model trained on these data will be generally incapable of recapturing this pattern, since it is a result of irreducible *label noise* that is not detectable from the model inputs. For example, if you are predicting the outcome of a dice roll, the aleatoric uncertainty would account for the randomness in that outcome. This form of uncertainty cannot be reduced by gathering more data. In the context of SDA, the uncertainties and errors from sensor measurements and limitations in numerical orbit propagation algorithms contribute to the aleatoric uncertainty.

BNNs are particularly well positioned to characterize epistemic uncertainty. As mentioned above, a single input can be passed through the model multiple times to generate a distribution of outcomes that reflects the model's uncertainty of how the inputs map onto the output response variable. In the following sections, we discuss how these pathways to evaluating uncertainty may be leveraged for SDA-relevant operational cases.

## Model Evaluation

In the subsequent two chapters, we will make use of four common metrics to evaluate the performance of our AI/ML case studies. The first is mean absolute percentage error (MAPE), which is a commonly used metric in the field of statistics and data analysis to assess the accuracy of a predictive model, particularly in the context of regression analysis. It measures the percentage

---

<sup>57</sup> Fortuin et al., "Bayesian Neural Network Priors Revisited."

<sup>58</sup> Hüllermeier and Waegeman, "Aleatoric and Epistemic Uncertainty in Machine Learning."

difference between the predicted and actual values of a variable, making it a useful tool for evaluating the relative accuracy of a model's predictions. MAPE is calculated as

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - P_i}{A_i} \right| \times 100,$$

where  $n$  is the total number of data points or observations,  $A_i$  represents the actual or observed value for a specific data point  $i$ , and  $P_i$  represents the predicted value for the same data point  $i$ .

Mean squared error (MSE) is a widely used metric in the field of statistics, ML, and data analysis. It quantifies the average squared difference between predicted values and actual (observed) values in a dataset. MSE is a measure of the quality of a predictive model or estimator, and it is particularly useful in regression analysis, where the goal is to predict continuous numerical values. MSE is calculated as

$$MSE = \frac{1}{n} \sum_{i=1}^n (A_i - P_i)^2,$$

where all variables are defined as above.

In the use cases we will explore, the output of an AI/ML model will be a three-dimensional (3D) probability distribution intended to represent knowledge of a satellite object's spatial position. The quality of the produced distributions will be assessed by measuring their similarity with a set of target probability distributions generated from traditional numerical distances. Therefore, additional metrics to assess the difference between probability distributions are needed.

The first metric we use for this purpose is Kullback-Leibler (KL) divergence, denoted as  $D_{KL}$ , a measure of the difference between two probability distributions.<sup>59</sup> It is used to quantify how one probability distribution diverges from another. The KL divergence between two probability distributions  $P(x)$  and  $Q(x)$  is defined as follows:

$$D_{KL}(P||Q) = \int P(x) \log \left( \frac{P(x)}{Q(x)} \right) dx.$$

In the special case in which both of these distributions are 3D Gaussians with diagonal covariance matrices, this formula reduces to

$$D_{KL}(P||Q) = \frac{1}{2} \left[ \sum_{i=x,y,z} \frac{\sigma_{P,i}^2}{\sigma_{Q,i}^2} + \sum_{i=x,y,z} \frac{(\mu_{P,i} - \mu_{Q,i})^2}{\sigma_{Q,i}^2} + \ln \left( \prod_{i=x,y,z} \frac{\sigma_{P,i}^2}{\sigma_{Q,i}^2} \right) - 3 \right],$$

where  $\sigma_{K,i}^2$  refers to the covariance in the  $i^{th}$  dimension of the probability distribution  $K$  and  $\mu_{K,i}$  refers to the mean in the  $i^{th}$  dimension of the probability distribution  $K$ .

The second metric we propose for this purpose is the Wasserstein distance,<sup>60</sup> denoted as  $W$ . This metric is colloquially known as *Earth mover's distance* because of its interpretation as the cost of transforming one probability distribution into a second reference probability distribution, as one might consider when attempting to move a mound of soil to match the exact characteristics of a

<sup>59</sup> Joyce, "Kullback-Leibler Divergence."

<sup>60</sup> Panaretos and Zemel, "Statistical Aspects of Wasserstein Distances."

second, separate mound of soil nearby. Again, in the special case in which both of the considered distributions are 3D Gaussians with diagonal covariance matrices, the Wasserstein metric reduces to

$$W(P, Q) = \sum_{i=x,y,z} \left[ (\mu_{P,i} - \mu_{Q,i})^2 + \sigma_{P,i}^2 + \sigma_{Q,i}^2 - 2\sigma_{P,i} \cdot \sigma_{Q,i} \right].$$

There are key differences between  $W$  and  $D_{KL}$  that make each better suited to particular components of our methodology. One key factor we will consider in our model evaluation is whether the uncertainties produced by our model are well calibrated with respect to the model distribution's mean offset from the target distribution. Unlike  $D_{KL}$ , there are no terms within  $W$  that couple the means and standard deviations of the two respective distributions; thus,  $W$  is not well suited to evaluate the uncertainty calibration of our model.

On the other hand, for neural network model training, we also need to select a similarity metric to use as an objective function. Although  $D_{KL}$  has been used in many applications as an objective function, partly because of the natural logarithm term, it can produce more-volatile gradients that lead to unstable training. Although such methods as gradient clipping can be employed to protect against failures, they add complexity.<sup>61</sup> Within our case studies, we choose to use  $D_{KL}$  to evaluate our model performance and  $W$  as an objective function for model training. Over the experiments we conducted, we found that selecting  $W$  as an objective function for model training generally mapped to high performance on  $D_{KL}$  while also enabling fewer trials to be run in each experiment because of more stability in training loss.

## Summary

In this chapter, we highlighted existing efforts to develop AI/ML for SDA mission processes. We described our own two AI/ML case studies, which use neural networks to perform prediction and classification tasks relevant to SDA operations. Finally, we provided an overview of neural network architectures and performance metrics, both of which will be referenced in the following chapters, which describe our case studies in greater depth.

---

<sup>61</sup> Pascanu, Mikolov, and Bengio, "On the Difficulty of Training Recurrent Neural Networks."

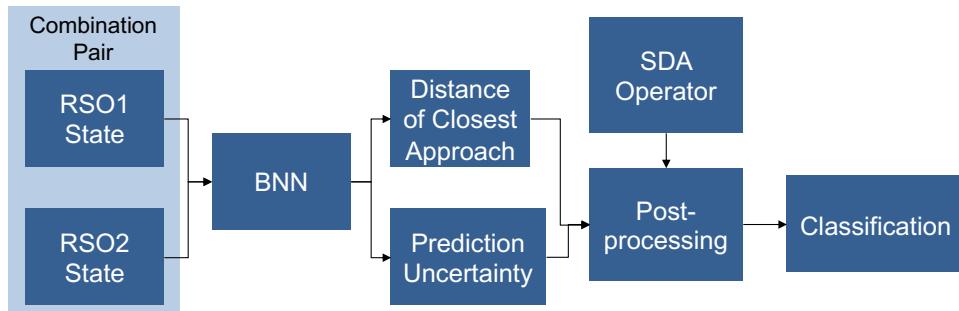
# Elliptical Screening Process Tool

We first discuss the ESP tool, a BNN designed to replicate the conjunction screening process. In Chapter 1, we described the conjunction screening process as screening through every combination pair of RSOs to identify and flag combination pairs whose future trajectories violate either an elliptical screening volume constraint or a  $P_c$  threshold. The tool presented here provides the operator with estimates of the distance of closest approach and an uncertainty. This tool may serve as a prescreen to reduce the number of RSO combination pairs fed into the computationally limited CAVENet for subsequent high-fidelity conjunction screening.

## Elliptical Screening Workflow Description

In the case study presented in this section, a BNN is provided two RSO states at a given epoch. The two RSO states may be the observation state provided directly from the sensor on the SPADOC system or the product of the DC process on the CAVENet system. The only requirement is that the two RSOs have an identical initial epoch. From these two initial states, the BNN performs a regression analysis and produces two outputs: (1) a prediction of the distance of closest approach over the next five days and (2) an estimate of uncertainty in the prediction. We then conduct postprocessing classification analysis of the BNN output to determine whether the RSO pairing should be flagged for further review. When a possible conjunction event is flagged, additional resources and more-accurate algorithms can be dedicated to better characterize the combination pair. The workflow for this use case is described in Figure 3.1.

**Figure 3.1. Case Study 1: Elliptical Screening Tool Workflow**



## Data Generation

Data generation for AI/ML involves creating a dataset that can be used to train, validate, or test a neural network model. For SDA applications, a dataset can be real-world data collected by SDA operators stored on the CAVENet system or synthetic data produced using modeling and simulation. To a certain extent, the analysis tools used by SDA operators to produce real-world data also leverage modeling and simulation, but the exact algorithm, calibration parameters, and assumptions would be considered classified information. The decision to present our analysis to a general audience is the primary motivating factor to create an entirely synthetic dataset representative of data, modeling, and simulation methods used at the 18th and 19th SDSs.

When generating a synthetic dataset for the elliptical screening tool case study, we first reference general perturbation orbit data provided by the U.S. government via CelesTrak in an Orbit Mean-Elements Message (OMM) Extensible Markup Language (XML) data format.<sup>62</sup> To scope the development effort, we constrain the RSO considered to Starlink satellites, which operate approximately between 540 and 570 km altitude—in inclinations that include 53.0, 70.0, and 97.6 degrees. Researchers have recently stated that the number of Starlink conjunction data message events with a computed  $P_c < 1e - 5$  accounted for 43,042 events; 35,047 of those were versus Starlink.<sup>63</sup> We claim that a Starlink-focused dataset would provide enough diversity in orbit to remain nontrivial and prevent overfitting while providing enough conjunction warnings to adequately train the neural network.

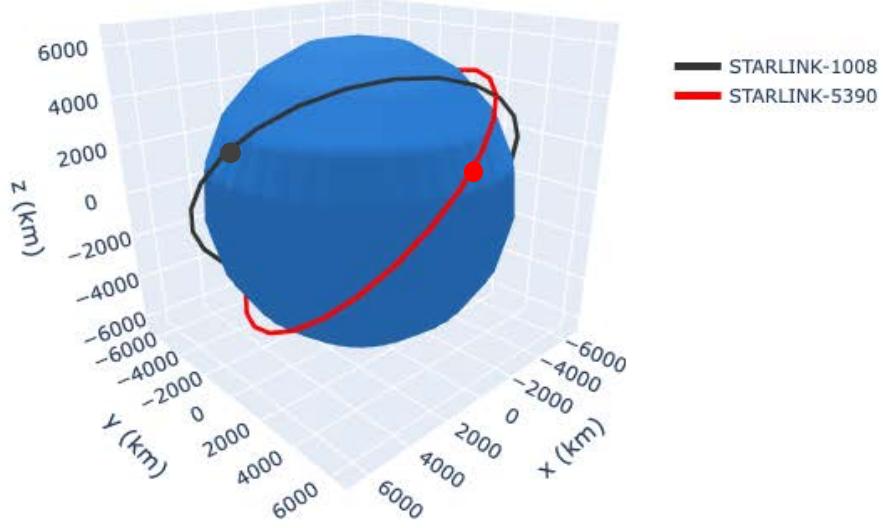
Using the general perturbation data from CelesTrak, we first sample a random combination pair of satellites. For each satellite, we then randomly assign an ephemeris contained in a single orbital period to act as the starting condition, as shown in Figure 3.2. These initial conditions provide an infinite number of permutations to construct our training dataset.

---

<sup>62</sup> Kelso, “A New Way to Obtain GP Data (aka TLEs).”

<sup>63</sup> Moomey, Falcon, and Khan, “Trending and Analysis of Payload vs. All Low Earth Conjunction Data Messages Below 1,000 km, from 2016 Through 2021.”

**Figure 3.2. Notional Combination Pair Selection**



Once the combination pairs are sampled and initial conditions are selected, we need to evaluate the distance of closest approach over a time horizon of five days. We leverage the Cowell’s method implementation in an open-source Python library for astrodynamics and orbital mechanics called Poliastro.<sup>64</sup> The algorithm accounts for the Keplerian forces in addition to higher-order dynamical forces.<sup>65</sup> The initial-state conditions are then propagated to the time horizon, and the distance between the two objects is evaluated at each step. The distance and epoch of closest approach are then stored as an output variable. This process is repeated to generate a 2 million data point training dataset.

## Model Architecture

After a synthetic dataset has been produced, the next step is to use the data for designing and optimizing a BNN to predict the distance of closest approach between the combination pair of Starlink satellites. The model uses a Gaussian prior distribution in which the Gaussian mean prediction is interpreted as the predicted distance of closest approach and the Gaussian standard deviation provides a quantification of the prediction uncertainty. Using the open-source Python package Hyperopt,<sup>66</sup> we experimented with a variety of negative log likelihood (NLL) loss functions, loss function weights, and neural network architectures. We evaluated the different NLL loss functions and neural network architectures using common metrics, including MAPE and MSE, ultimately resulting in an architecture with a batch size of 256; three hidden layers with 243, 104, and

<sup>64</sup> Poliastro—Astrodynamics in Python, homepage.

<sup>65</sup> These higher-order dynamical forces include additional celestial bodies (the sun and moon), nonsphericity of the central body terms ( $J_{2,0}$  and  $J_{3,0}$ ), atmospheric drag and lift, and solar radiation pressure, as described in Chapter 1.

<sup>66</sup> Hyperopt: Distributed Asynchronous Hyper-parameter Optimization, homepage; Bergstra et al., “Hyperopt”; Bergstra, Yamins, and Cox, “Making a Science of Model Search.”

113 neurons for each, respectively; the Adam (adaptive moment estimation) optimization algorithm,<sup>67</sup> a learning rate of approximately  $1.608e-3$ , and a modified NLL loss function shown below, where  $\mu$  is the mean prediction,  $\sigma$  is the standard deviation, and  $y$  is the true distance of closest approach.  $\Lambda$  represents a weight for how important the primary accuracy metric is, while still incorporating the standard deviation. The  $\alpha$  weight represents the importance associated directly with the MAPE metric. Both  $\lambda$  and  $\alpha$  were introduced to modify the loss function to get higher prediction accuracy while attempting to maintain uncertainty reliability. Because MAPE is skewed toward smaller distances, this model maintained its accuracy for longer distances than previous models while increasing accuracy for shorter distances. The model is expressed as

$$NLL = \sum \left( \log(\sigma) \times 0.5 \times \lambda \left( \frac{y-\mu}{\sigma} \right)^2 + \alpha \text{MAPE}(y, \mu) \right).$$

The training data were scaled using minimum-maximum scaling, and dropout regularization was set at 20 percent to prevent overfitting and increase accuracy on the validation set. The algorithm underwent 100 training epochs. Of the nearly 2 million data points we had, we allocated 30 percent as a hold-out test dataset to evaluate the final performance of our algorithm and allocated 10 percent as a validation dataset. The results below are how the model performed on the hold-out set. The algorithm predicted on the hold-out set 1,000 times, and the results are the means of both those predictions and standard deviations. The standard-deviation means accounted for the aggregate and individual model means with the following equation:

$$\Sigma = \bar{m} + \sqrt{(m - \mu)^2},$$

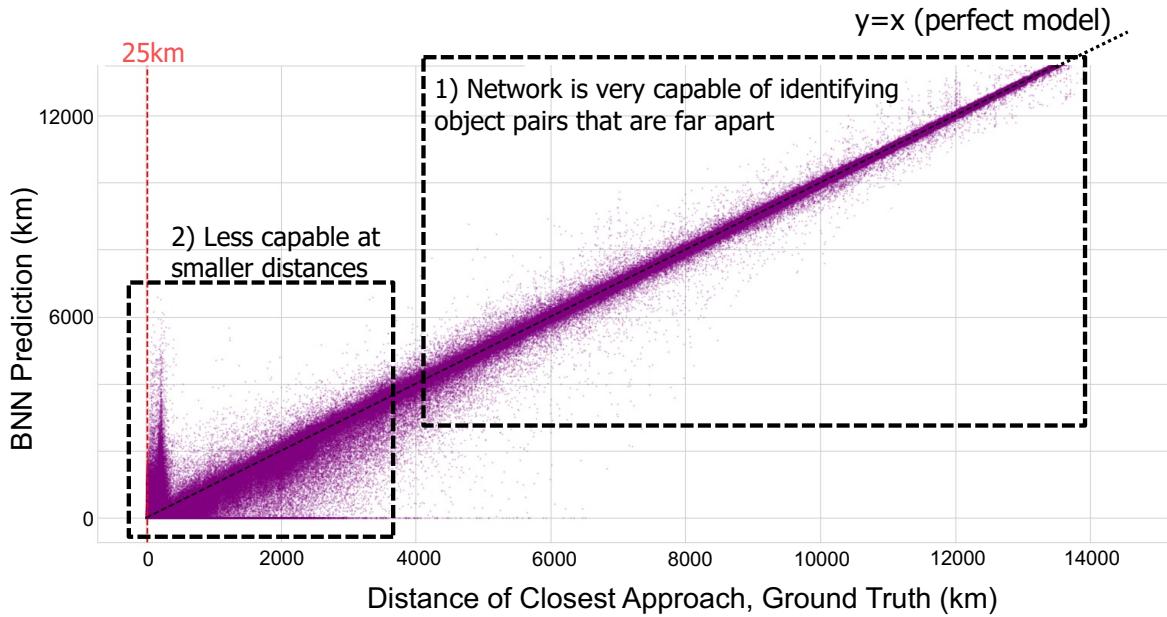
where  $\sigma$  represents the standard deviation,  $\bar{m}$  represents the mean of all the standard deviations,  $m$  represents the prediction, and  $\mu$  represents the mean prediction.

Figure 3.3 presents the accuracy of the model plotting the true values versus the predicted mean values. This figure shows how the predictions align with a perfect model significantly more when the true closest approach distance is large.

---

<sup>67</sup> Kingma and Ba, “ADAM.”

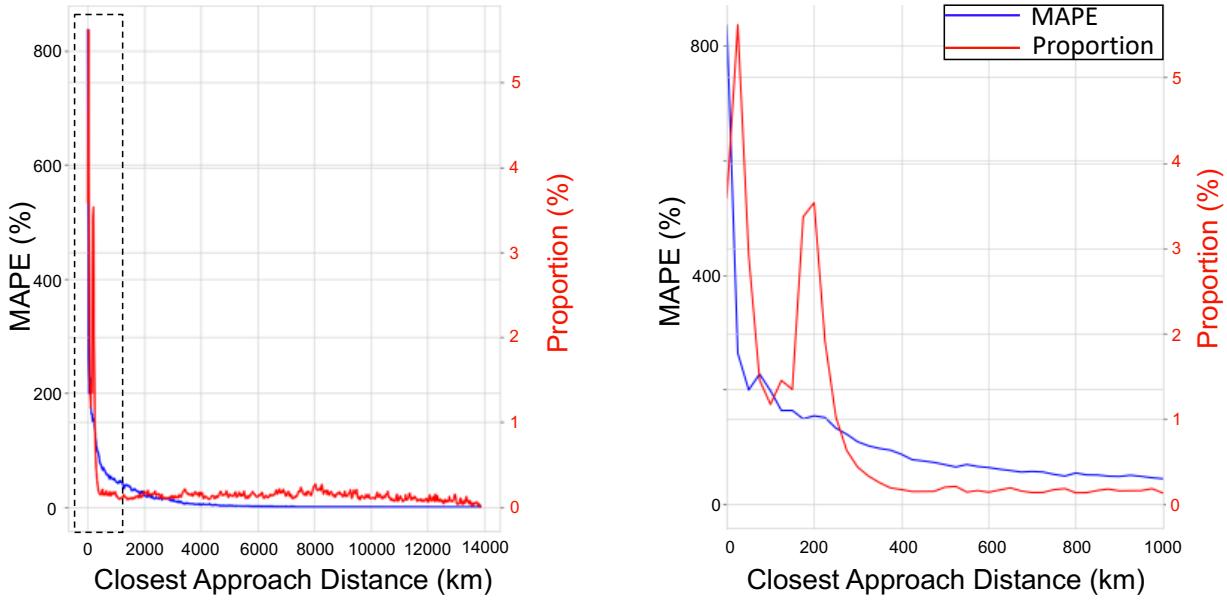
**Figure 3.3. BNN Distance of Closest-Approach Optimization**



We observe that, in general, the algorithm is capable of predicting the distance of closest approach between a combination pair of Starlink satellites. The algorithm appears to be more accurate for large distances of closest approach and less accurate for small distances of closest approach. We suspect this is due to at least two reasons: (1) There is significantly more data representation for the larger distances of closest approach, and (2) the number of features provided is limited. In Figure 3.4, we bin the data according to the true distance of closest approach and present the average MAPE in blue and the proportion of the training data represented by the bin in red. Each bin represents a 25 km distance of closest approach. We can see that the average MAPE is poor for the small distance of closest approach and approaches zero after 1,000 km. The first bin (0 to 25 km) then represents about 3.4 percent of all the training data, large relative to the proportion of other bins but small in comparison with the aggregate training dataset.

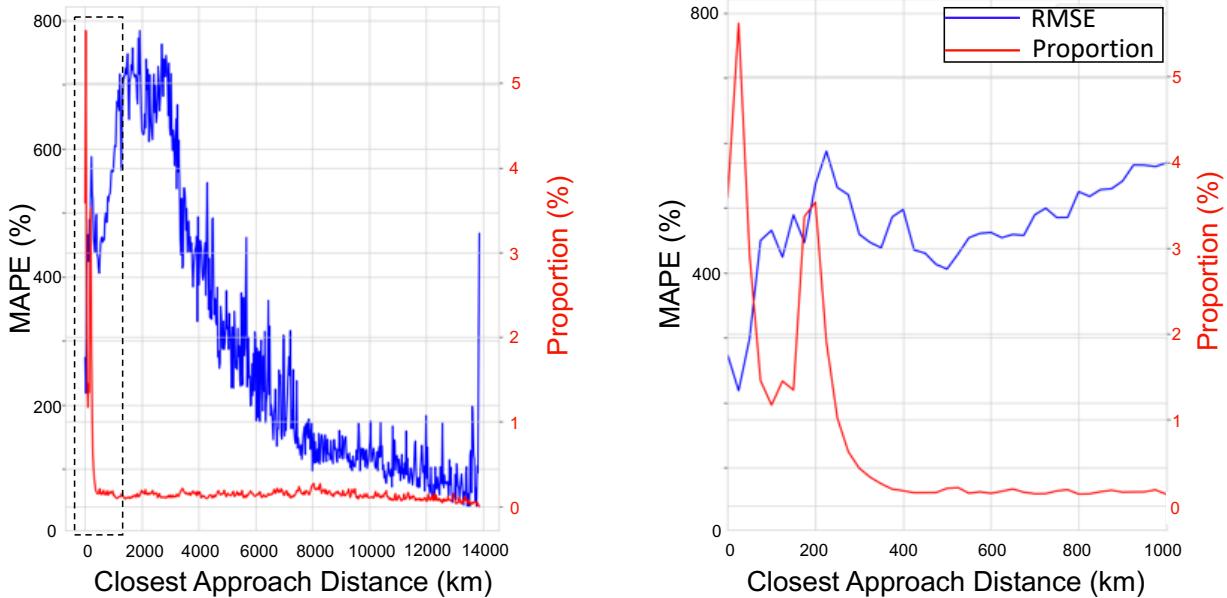
Figures 3.4 and 3.5 show two different accuracy metrics, once again portraying how the accuracy of the model significantly increases for larger distances of closest approach. For both figures, the lower error on the y-axis corresponds to higher model accuracy; as shown, this error significantly decreases as the distance of closest approach increases.

**Figure 3.4. MAPE Versus Distance of Closest Approach**



NOTE: The right-hand panel provides a close-up view of the region in the box in the left-hand panel. The plot demonstrates that the BNN MAPE performance increases for larger true distances of closest approach despite the increased proportion of data points in this region.

**Figure 3.5. Root MSE Versus Distance of Closest Approach**



NOTE: The right-hand panel provides a close-up view of the region in the box in the left-hand panel. RMSE = root mean squared error.

The BNN presented here can predict the distance of closest approach between a combination pair of Starlink satellites; however, the performance is inconsistent depending on the true distance of closest approach and in general performs worse when that distance is small. Nevertheless, the BNN mean prediction and uncertainty can then be leveraged by the postprocessing classifier to screen through combination pairs for potential conjunction events, as we will describe in the subsequent section.

## Model Output and Postprocessing

We now focus this case study discussion on the postprocessing of the BNN output and how an operator may interpret the measure of uncertainty. In Chapter 1, we described the process of screening through every combination pair of RSOs to identify and flag combination pairings that violate a 25 km elliptical volume for near-earth objects based on Table 1.2. It is important that we acknowledge that, although the BNN is attempting to make accurate predictions of the distance of closest approach, what is more operationally relevant is the correct classification of combination pairs (whether the combination pair is flagged for further analysis by an SDA operator). This is the final step of the ESP workflow described in Figure 3.1.

To prepare the BNN output for postprocessing, we first split the test data used to validate our BNN into two groups: Group A includes data points with a true closest approach less than or equal to 25 km, and group B includes data points with a true closest approach greater than 25 km. We also formulated a simple confidence metric to represent the level of uncertainty the algorithm has in a given prediction; the confidence metric is defined as the prediction standard deviation divided by the mean prediction.

The criteria for flagging a combination pair for further analysis by an SDA operator are as follows: (1) The mean prediction violates the 25 km distance of closest approach, or (2) the confidence metric violates the confidence threshold assigned by an SDA operator. If either criterion is violated, the combination pair is flagged. The ESP performance for group A is then measured by the percentage of combination pairs flagged for further review. The performance of group B is measured by the percentage of combination pairs not flagged for further review. Incorrect classification of the combination pairs is not of equal consequence, as an incorrect classification of group A results in a missed conjunction event. The incorrect classification of group B results in additional manpower required to review nonthreats. This dynamic is important for the SDA operator to consider when assigning a confidence threshold.

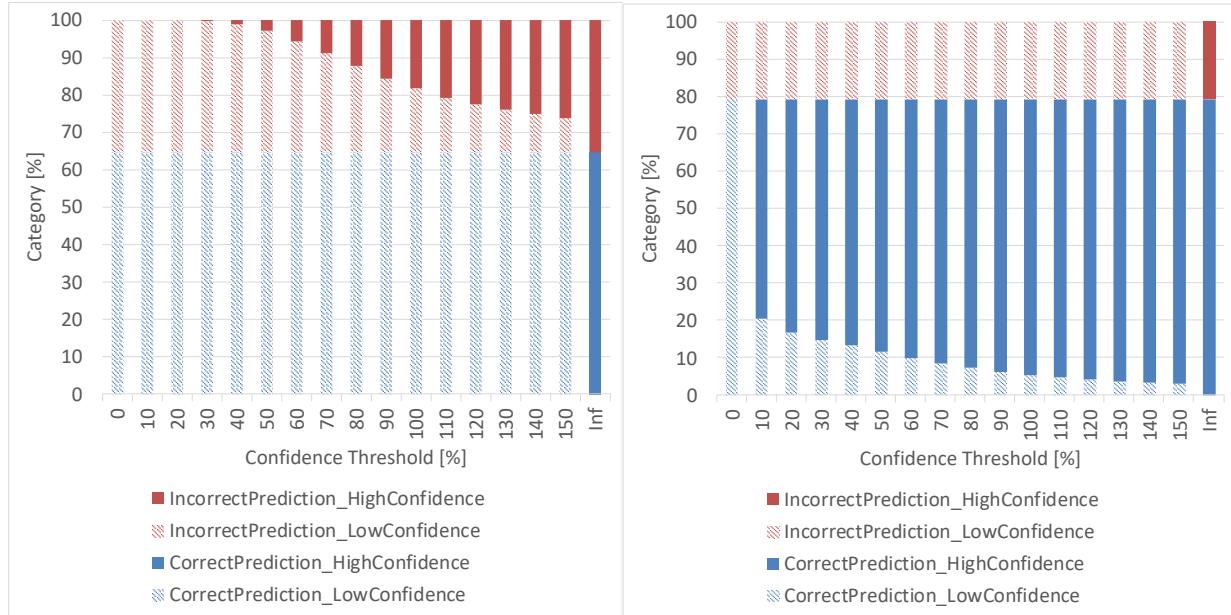
In Table 3.1, we present the classification matrix for predictions in group A and B when the confidence threshold is set to 10 percent. A correct prediction in group A is a prediction less than or equal to 25 km, and a correct prediction in group B is greater than 25 km. We first notice that all of group A is correctly categorized because of low prediction confidence despite 27.52 percent of the mean prediction being larger than 25 km. This result may be surprising considering the mean prediction performance presented in the earlier section, but, with the inclusion of a confidence metric, these combination pairs are flagged for further analysis. For group B, we note that 44.95 percent of combination pairs are incorrectly flagged for further analysis. This equates to a 55.05 percent reduction in combination pairs that require further analysis by an SDA operator.

**Table 3.1. Classification Matrix**

Group	Correct Prediction?	High Confidence?	Classification	Percentage
A	Yes	No	Flagged	72.48
	Yes	Yes	Flagged	0.00
	No	No	Flagged	27.52
	No	Yes	Not flagged	0.00
B	Yes	No	Flagged	21.74
	Yes	Yes	Not flagged	55.05
	No	No	Flagged	23.24
	No	Yes	Flagged	0.00

The parameterization of the confidence metric variable and the resulting classification matrix are shown in Figure 3.6 for group A and group B. A confidence metric of zero indicates that the SDA operator is skeptical of all BNN predictions, and thus every combination pair is flagged for further review, resulting in no added value from the BNN. On the opposite end of the spectrum, an infinite confidence metric is equivalent to an SDA operator having absolute trust in the BNN prediction, resulting in a significant number of misclassifications for group A. Overall, Figure 3.6 shows that, by introducing confidence via a BNN, we are able to effectively filter out predictions, thus reducing an analyst's workload. Using the right-hand panel, for example, by applying a 30 percent confidence metric, we have about 65 percent of the predictions in group B considered correct and confident, which can be filtered out. Figure 3.7 demonstrates this using true and false positives instead.

**Figure 3.6. Categorization of BNN Output for Group A (Left) and Group B (Right)**



NOTE: Inf = infinite.

The left panel of Figure 3.7 illustrates model performance as a function of uncertainty. In collaboration with an SDA operator, the inclusion of uncertainty quantification (x-axis) may improve the performance of prediction and classification methods. For example, if an SDA operator is willing to accept a 30 percent confidence metric, the blue curve indicates that the algorithm will correctly flag 99.7 percent (true positive rate) of combination pairings in group A, near-perfect performance. This comes at the cost of misidentifying and flagging 35.5 percent (false positive rate) of group B. As a result, this confidence threshold filters out 55.6 percent of all possible RSO pairs, reducing the workload in subsequent steps.<sup>68</sup> Contrast this with an infinite confidence metric meant to represent an SNN that performs significantly worse. Performance is depicted as the true-positive rate for the blue line ( $\leq 25$  km group) and the false-positive rate for the red line ( $> 25$  km group). The dashed lines are the asymptotic limits for a perfectly trusted model with an infinite confidence threshold. The right panel depicts this information as a receiver operating curve (ROC). Confidence thresholds are hidden in this style of plot. The area under ROC (AUROC)—the closer to 1.0, the better—indicates the overall strength of this classifier.

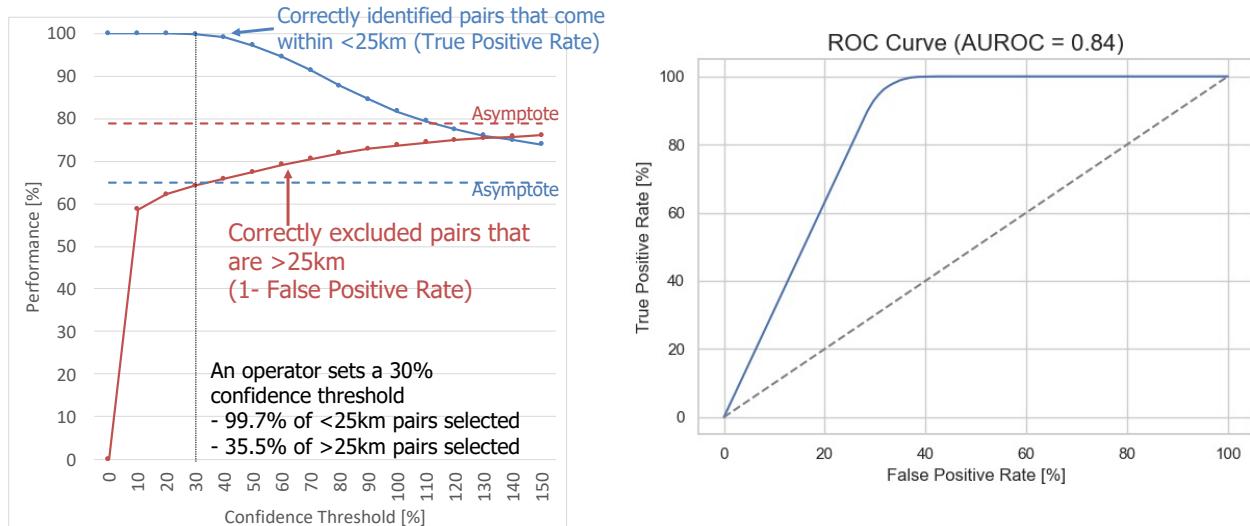
It is noteworthy to mention the computational efficiency inherent to this model because the BNN does not require an evaluation of the orbit propagation trajectory. After predicting on 599,985 unique combination pairs, 1,000 times, the average time the model took to generate predictions on the hold-out set was only 0.03418 seconds.<sup>69</sup> After linearly extrapolating the computational time to reflect a

<sup>68</sup> The reduction of the total workload metric is relative to the ratio of RSO pairs in group A to group B for our synthetic dataset. The number of RSO pairs flagged in an operational setting will vary and affect the metric performance. For reference, 3.58 percent of the synthetic dataset generated belongs to group A.

<sup>69</sup> The algorithm performance is evaluated on a NVIDIA T4 Tensor Core graphics processing unit (GPU).

more expansive dataset representative of what may be currently required by SDA operators,<sup>70</sup> we found that the estimated time for predictive analysis is approximately 56.2 seconds. When compared with the process time frames presented in Chapter 2, the BNN presents an opportunity to reduce the computational load, increase flexibility, and increase responsiveness by identifying and flagging likely conjunction events without requiring reference to the HAC trajectories.

**Figure 3.7. Performance Optimization**



NOTE: ROC = receiver operating curve; AUROC = area under ROC.

## Implementation Challenges and Future Work

We first would like to acknowledge the limitations in available training data and resources to optimize the BNN performance. These limitations are clear on inspection of Figure 3.7. Despite minimal hyperparameter tuning and a modest modification and optimization of the loss function, the workflow we present provides surprisingly robust categorizations. This may be a testament to the added value provided by the BNN uncertainty quantification in contrast to an SNN. A clear opportunity for improvement is continued refinement of the BNN algorithm and, subsequently, characterization of the performance optimization curve for the improved algorithm.

In the analysis presented here, we constrain the initial orbital state parameter space to the Starlink constellation satellites for either RSO in the combination pair. We expect the BNN to perform worse for both predictions of closest approach and prediction uncertainty in regions of phase space where it is untrained. Although a large prediction uncertainty may signal to an SDA operator that the BNN is outside the region of expertise, the degree of correlation between prediction precision and accuracy would still require exploration. Expanding the initial orbital state parameter space would require more

---

<sup>70</sup> Recent estimates state the 18th SDS maintains a catalog tracking more than 44,000 RSO (Space-Track.org, homepage). The number of unique combination pairs is then 967,978,000.

data than we were able to synthetically generate on the project timeline. We expect organizations such as the 18th SDS and the 19th SDS to have historical data that could be leveraged for this purpose.

Another opportunity for increasing the capability of the BNN is to include a prediction of the TCA for flagged combination pairs. The value of this capability will become more relevant when used in combination with the second case study, on the orbital state propagator (OSP) tool, described in the next chapter.

## Summary

In this chapter, we demonstrated feasibility of an AI/ML case study that flags RSO pairs for possible conjunctions. The ESP tool provides the operator with estimates of the distance of closest approach and an uncertainty, enabling decisionmaking based on the operator's risk acceptance level. The model is fast, and we estimate that the time to analyze all current RSO combination pairs in the unclassified space catalog is approximately 56.2 seconds. The tool is also accurate in classifying combination pairs of concern when accounting for both the distance of closest approach prediction and uncertainty. A tool similar to what we have demonstrated here trained on data in an operational environment may serve as a prescreen to reduce the number of RSO combination pairs fed into the computationally limited CAVENet for subsequent high-fidelity conjunction screening. It may also serve as a screening process for high-priority combination pairs identified outside the typical eight-hour conjunction process timelines.

# Orbital State Propagator Tool

We now discuss the OSP tool, the second case study designed by the RAND team to predict the future state and covariance from an initial RSO state and covariance. There are many instances within SDA processes in which the future state and covariance of a space object is required for analyses or task execution. Examples include (1) knowing where to point ground-station sensors for observations and (2) the calculation of  $P_c$ . Covariance calculations are currently predominantly performed through SP calculations that offer high-fidelity performance but are also time and resource intensive. The ability to quickly evaluate future states with minimal computational load without requiring access to the central repository of trajectories stored on CAVENet has implications for the ability to perform analysis at the tactical edge and for creating flexible timelines for covariance-related analysis not amenable to current processes.<sup>71</sup>

This task aims to assess whether AI/ML can accurately replicate low-tolerance SP calculations. The opportunities for the orbital state propagator to provide operational value increase with better performance. For this reason, our attention here is focused on producing a well performing BNN as opposed to designing a workflow that is integrated with SDA operators. Although it is challenging to establish minimum operational requirements for such a tool because of individual-level comfort with AI, close-enough rapid covariance propagation can accelerate many SDA tasks.

In the following sections, we describe our general methodology for propagating covariance forward in time, present our proof-of-concept results, and present two frameworks by which an OSP model could be operationalized.

## Covariance Propagation Workflow Description

As shown in Figure 4.1, the workflow process begins with an initial state and covariance of a space object. The initial state and covariance are a probability distribution; the parameters associated with this distribution are input to the neural network, which in turn produces the output state and covariance of the object at a specified time in the future. In the following sections, we will discuss two different neural network methods explored by the RAND team for state and covariance propagation and discuss the metrics used to evaluate the performance of each neural network algorithm.

For this prototype, we train the BNN to output an RSO's covariance at a fixed future time (time of flight = 1,000 seconds). We use various statistical measures to gauge performance and assess how

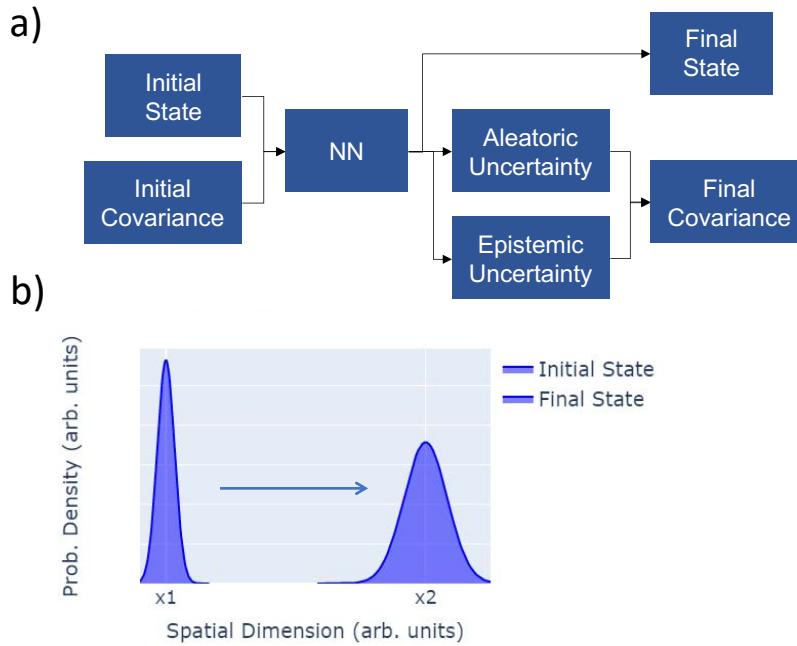
---

<sup>71</sup> We also considered leveraging AI for propagation of trajectory positions, without any covariance predictions. However, high-fidelity, yet computationally expensive, numerical techniques already exist for trajectory propagation, and lower-fidelity, yet lower-resource, techniques (such as general perturbation methods) are also well established. Unlike with covariance propagation, these numerical alternatives make it challenging to find a “Goldilocks” zone in which an AI surrogate model could offer an operational advantage in computational time and fidelity space.

closely the BNN uncertainty matches the true covariance (as informed by Monte Carlo simulations of orbit propagation).

However, operators often require knowledge of a space object's future covariances across a wide variety of time horizons. In a later section, we also discuss (1) how a fixed time-of-flight model may be operationalized to provide covariance estimates across a range of flight and (2) the feasibility of variable time-of-flight models.

**Figure 4.1. OSP Tool Workflow**



NOTE: NN = neural network; arb. = arbitrary.

## Data Generation

Like the ESP case study, we lacked access to operationally relevant data and chose to generate a synthetic dataset for the OSP tool. The OSP tool consists of a neural network that ingests an initial RSO state and covariance and predicts the RSO future state and covariance. Generating a training dataset for this case study required an algorithm to propagate orbit trajectories and covariances in a way that is representative of how the process is performed by SDA operators in a system such as CAVENet. Because we did not have access to the specific algorithms implemented on CAVENet, it was challenging to compare and benchmark our case study performance against operational performance.

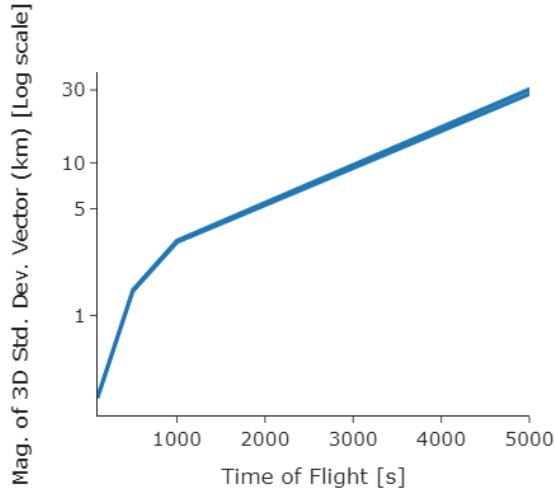
When generating a synthetic dataset for the OSP, we first referenced general perturbation orbit data provided by the U.S. government via CelesTrak in an OMM XML data format. To scope the development effort, we constrained the RSOs considered to Starlink satellites, which operate approximately between 540 and 570 km altitude, in inclinations that include 53.0, 70.0, and 97.6

degrees. We maintained the same assumptions from the previous chapter—that a Starlink-focused dataset would provide enough diversity in orbit to remain nontrivial and prevent overfitting. These initial conditions also provided an infinite number of permutations from which to construct our training dataset.

Using the general perturbation data from CelesTrak, we sampled a random Starlink satellite and assigned an ephemeris arbitrarily selected in a single orbital period of the satellite to act as the mean starting condition. We then conducted a Monte Carlo sample of a distribution of initial conditions around the mean to represent the covariance. The initial-state (position and velocity) distribution's standard deviation is spherical. The standard deviation for the positional measurement is 50 m and is 5 m per second for the velocity, which we think is representative of U.S. government ground sensor capabilities. The initial state could represent either a sensor observation or the product of the DC process, and the source of this initial state has implications for the accuracy and precision of the measurement, as described in Chapter 1.

The initial-state distribution was then propagated for a fixed time of flight to an end-state distribution. Consistent with the SP implementation at the 18th and 19th SDSs, we observed an expansion of the state covariance as the propagation time increases. Statistics such as mean and standard deviation of the end-state distribution were determined and provided as output of our numerical orbit propagation algorithm. As a simplification, we chose to extract only the diagonal elements of the output covariance matrix within our simulations. However, in future iterations, our process can be modified to extract all nine elements of the output covariance matrix, so these quantities may be forecasted by our OSP tool as well.

**Figure 4.2. Increase in Covariance Versus Time of Flight**



Similar to the synthetic data generated for the ESP tool, we leveraged the Cowell's method implementation in an open-source Python library for astrodynamics and orbital mechanics called

Poliastro to propagate the orbit state.<sup>72</sup> The algorithm accounts for the Keplerian forces in addition to higher-order dynamical forces.<sup>73</sup> The numerical algorithm used is the Dormand-Prince integration method of order 8(5,3),<sup>74</sup> with a relative tolerance of  $10^{-9}$ . The aleatoric uncertainties and the rate at which the covariance expands are specific to the numerical algorithm used; therefore, our methodology is reflective of current SDA processes, but there will be discrepancies between our training dataset and an operational dataset.

For the majority of experiments, we randomly sampled across our input parameter space to generate a training set of  $\sim 140,000$  events and subsequently trained a variety of models on this set. However, because the training data for our models were generated via simulation, dynamic active-learning training protocols can also be leveraged, a possibility we explore in further detail in a later section.

An important parameter to consider was how many Monte Carlo samples are required to accurately represent the uncertainty for each set of input parameters. Enough Monte Carlo trials need to be run to fully explore the range of possible output states of the model and thereby extract accurate final position and covariance states. On the other hand, computational time increases linearly with trial number. To identify an acceptable trade-off point between these two factors, we monitored how the variance in the X, Y, and Z output positions changed as a function of Monte Carlo trial number. We conducted this analysis for a set of 1,000 randomly chosen input states. Following standard practice,<sup>75</sup> for each of our input states, we extracted the trial number at which the variances of the output positions in the X, Y, and Z dimensions appeared to converge. The convergence point was extracted by assuming the exponential convergence profile and fitting such functions to the data to extract  $1/e^3$  decay constants. The results of this convergence analysis for a time of flight of 1,000 s are presented in Figure 4.3. Ultimately, we chose to conduct 1,500 Monte Carlo trials per input to construct our training set, a trial number that was associated with convergence in  $\sim 80$  percent of inputs in our convergence analysis. When building the training set, we parallelized simulation runs across eight computing cores and were able to produce event data at a rate of  $\sim 1,500$  events per hour.

---

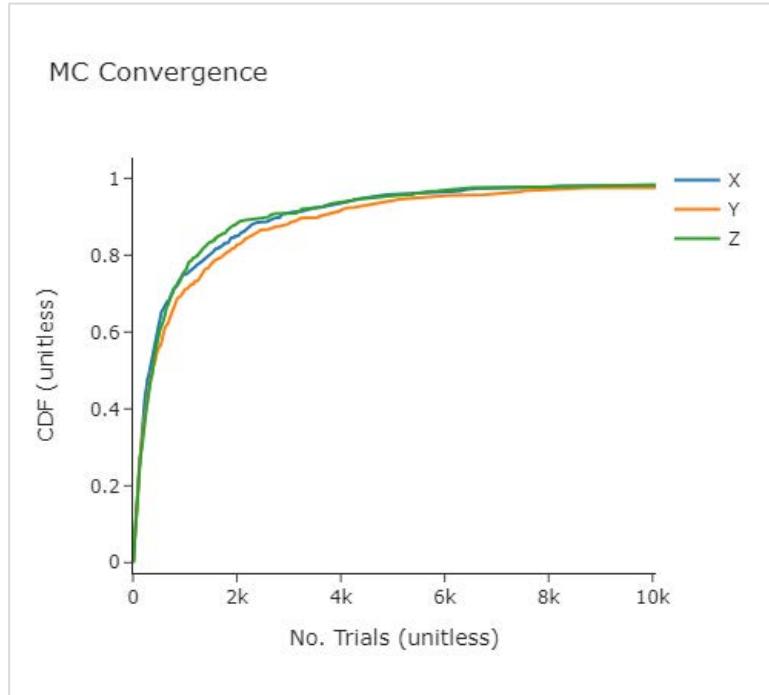
<sup>72</sup> Poliastro—Astrodynamics in Python, homepage.

<sup>73</sup> These higher-order dynamical forces include the nonsphericity of the central body terms ( $J_{2,0}$  and  $J_{3,0}$ ), atmospheric drag and lift, and solar radiation pressure, as described in Chapter 1.

<sup>74</sup> SciPy, “scipy.integrate.DOP853.”

<sup>75</sup> Myong, Karchani, and Ejtehadi, “A Review and Perspective on a Convergence Analysis of the Direct Simulation Monte Carlo and Solution Verification.”

**Figure 4.3. Monte Carlo Trial Convergence Versus Trial Number**



NOTE: CDF = cumulative distribution function; MC = Monte Carlo.

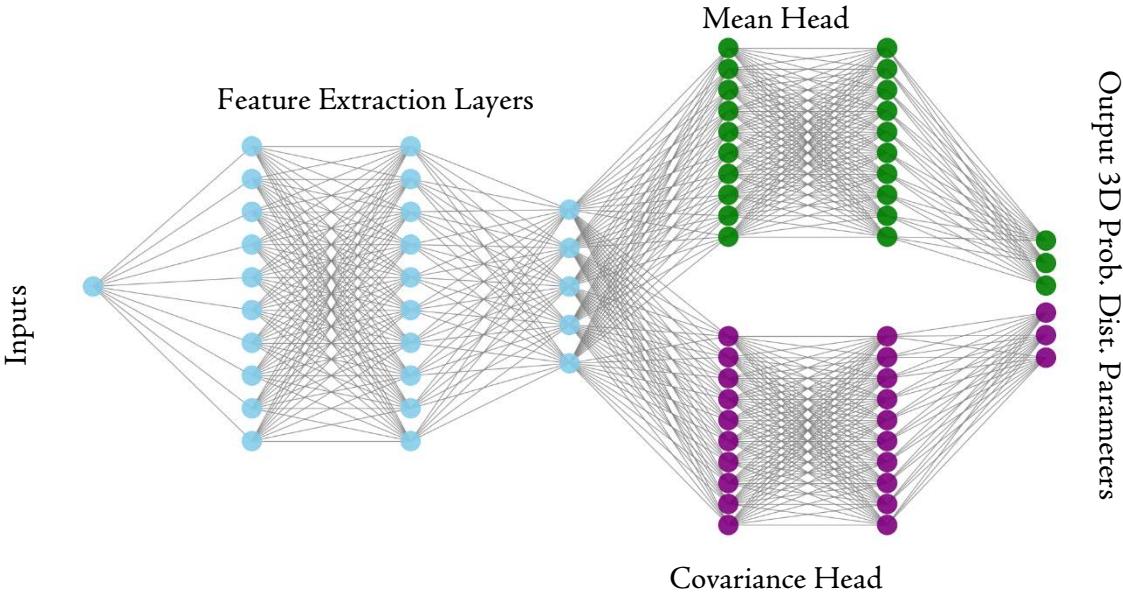
## Model Architecture

The model inputs consist of the following sets of data arranged in a tabular format: an input 3D position vector, an input 3D position covariance matrix, an input 3D velocity vector, an input 3D velocity covariance matrix, a space object's cross-sectional area, a space object's drag coefficient, and, lastly, a time of flight (for fixed time-of-flight models, this input is excluded). The BNN is a five-layer feedforward network, constructed via the BLiTZ Python package,<sup>76</sup> that ingests these inputs and produces six outputs, representing an output 3D-position vector and an output 3D-position covariance. As shown in Figure 4.4, the model possesses a twin architecture consisting of three fully connected layers, followed by two separate two-layer heads, one for covariance calculations and one for mean position calculations. The output of each head consists of three parameters that specify the 3D mean position and positional covariance of the final-state multivariate Gaussian distribution. Note that, because our training dataset was built under the assumption that all input-output covariance matrices were diagonal, we made the same assumption with our BNN inputs and outputs.

---

<sup>76</sup> Esposito, “BLiTZ—Bayesian Layers in Torch Zoo.”

**Figure 4.4. Notional Neural Network Model Architecture**



NOTE: Prob. Dist. = probability distribution.

The number of nodes within each layer was determined by leveraging the Hyperopt package to perform a hyperparameter search over the node number in each layer.<sup>77</sup> Here, 300 nodes were chosen as an upper bound for our search space to keep the total model parameter number less than the total number of training data rows available (~140,000). The learning rate of the Adam optimizer used for model training and the batch size were also optimized in this manner. After the above parameters had been optimized, training was carried out over 1,000 epochs, with a validation set included to monitor overfitting.

Once the model was trained, the BNN was evaluated on a hold-out test set by running  $N = 100$  forward passes through the model for each input state. This resulted in 100 generally distinct output 3D probability distributions that were aggregated together to form a final output state.

The aggregate positional mean  $\hat{x}$  from these samples was extracted as

$$\hat{x} = \frac{1}{N} \sum x_i,$$

where  $x_i$  is the 3D position vector associated with BNN sample  $i$ . The covariances were calculated as

$$\hat{\sigma} = \frac{1}{N} \sum \sigma_i + \sqrt{(x_i - \hat{x})^2}.$$

The variable  $\hat{\sigma}$  is the aggregate BNN standard-deviation output vector, and  $\sigma_i$  is the 3D standard-deviation vector associated with BNN sample  $i$ . The second term is included to consider each BNN

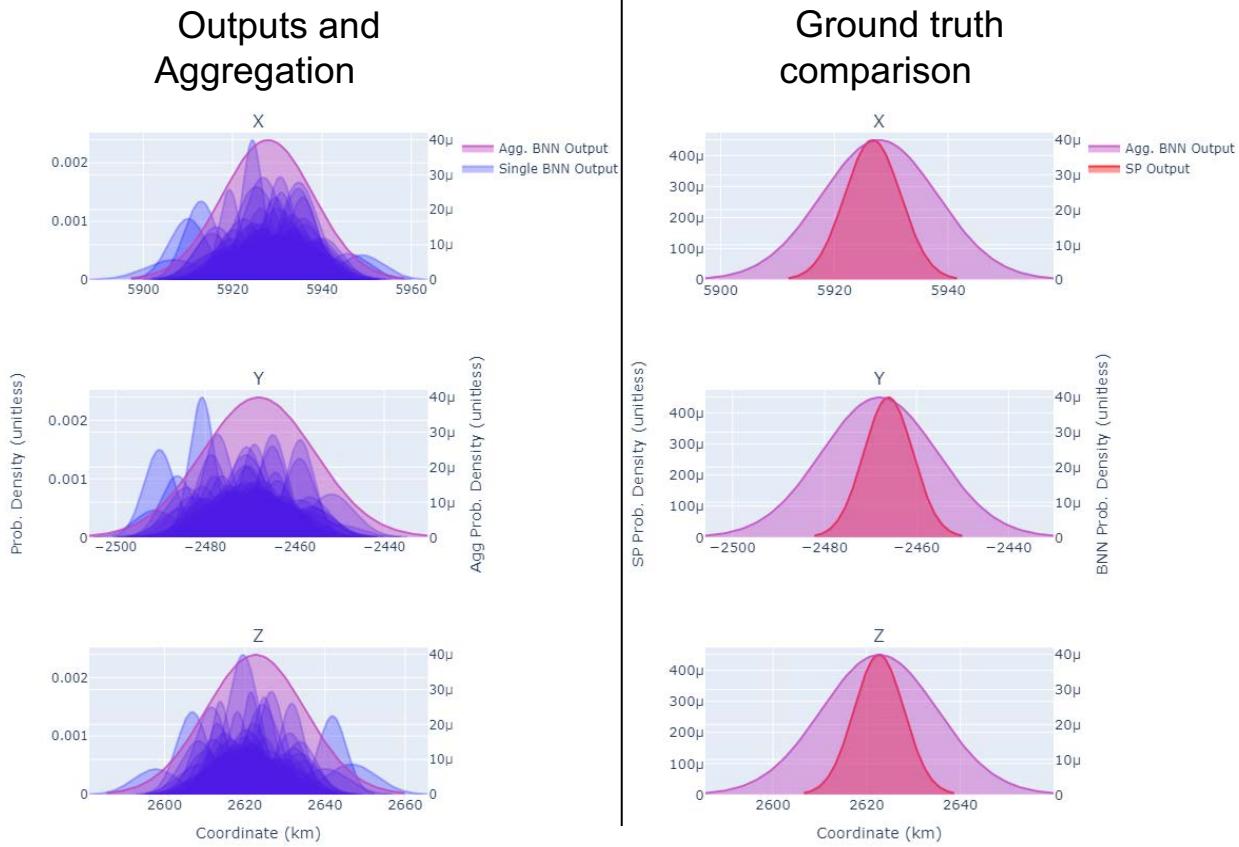
---

<sup>77</sup> Bergstra, Yamins, and Cox, "Making a Science of Model Search."

sample's positional offset from the  $\hat{x}$  in the standard-deviation calculation, thus providing more-informative uncertainty estimates. Figure 4.5 provides a visual example of this process for a single prediction in each of the three dimensions. The three panels on the left show the various single BNN output samples being aggregated into a mean prediction and standard deviation; the three panels on the right compare the aggregate prediction against the target data.

We followed the same general training procedure as above for a non-Bayesian SNN for comparison purposes, with the modification that no output aggregation was necessary because of the deterministic nature of inference within these models. The results are discussed in the next section.

**Figure 4.5. BNN Sample Output**



NOTE: Agg. = aggregate; Prob. = probability. For a given RSO input state, we generate 100 output state predictions using the BNN and then aggregate the results into a single distribution (left panel). This aggregated 3D distribution is compared against the numerical model output, labeled "SP Output," in the right panel.

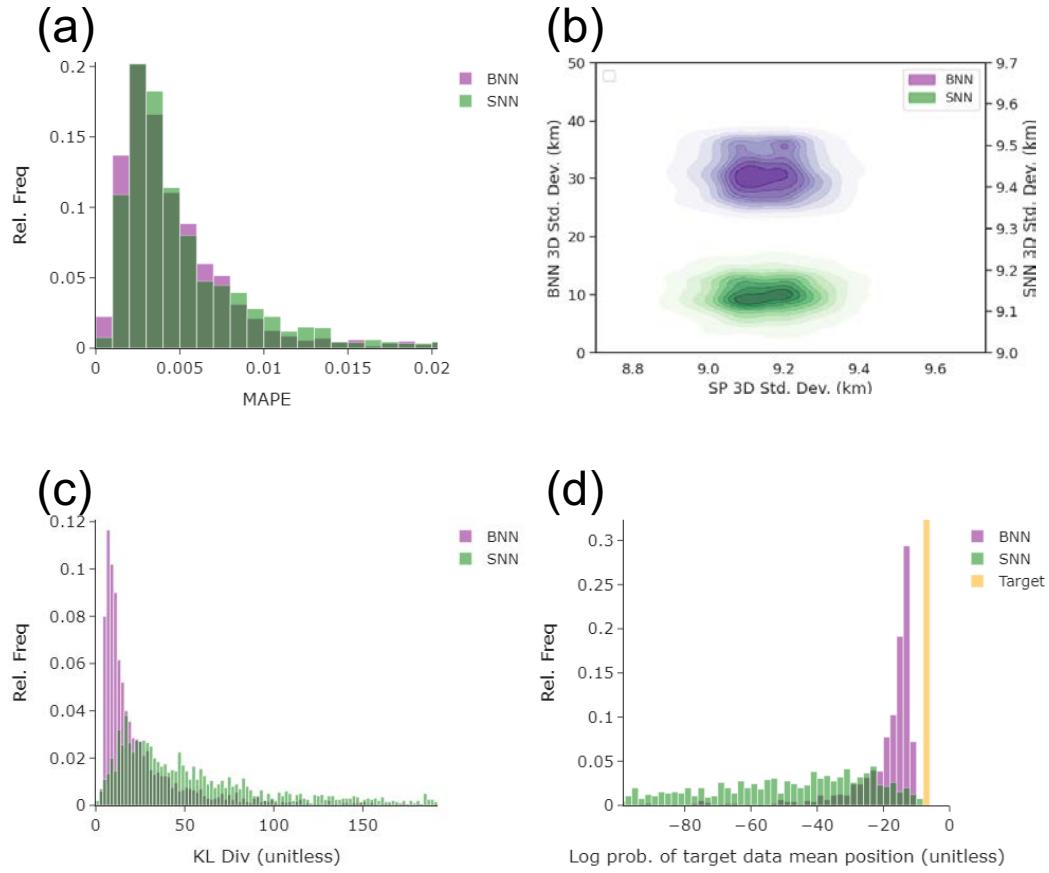
## Model Performance

By repeating the data-generation process, we assembled a hold-out test of  $\sim 100,000$  events on which to evaluate both our SNN and BNN models. The strengths and weaknesses of the two models are described in the text below and in Table 4.1. The following discussion references results presented in Figure 4.6.

**Table 4.1. BNN and SNN Comparative Performance**

Metrics	BNN	SNN	Optimal Model
MAPE	Low	Low	Low
Covariance root MSE	High	Low	Low
KL divergence	Low	High	Low
Training time	High	Low	Low
Probability of target position	High	Low	High

**Figure 4.6. BNN and SNN Proof-of-Concept Performance**



NOTE: The panels depict histograms for four metrics on 100,000 RSOs in the validation set. The BNN and SNN architectures are compared. Panel A shows MAPE. Panel B shows normalized standard-deviation distributions with different y-axes presented for BNN and SNN traces to allow for distributional resolution of both despite the large difference in their produced standard-deviation magnitudes. Panel C shows KL divergence. Panel D shows the probability (log scale) of the true mean in each of the predicted distributions. Div = divergence; Prob. = probability; Rel Freq = relative frequency.

## SNNs

SNNs produce MAPEs on the same order as BNNs (median value =  $\sim 0.3$  percent) and produce relatively lower covariance root MSEs; however, SNNs also produce high KL divergences. The produced covariances are generally too small relative to the positional error of the model. If the output probability distribution of the SNN is evaluated at the actual position of the final-state vector target data, it typically yields a probability lower than the probability from a corresponding BNN by  $\gg 10$  orders of magnitude (bottom-right panel of Figure 4.6). If operators were using such probabilities to guide risk calculations associated with conjunction events or other position-dependent scenarios, they would have a skewed understanding of the actual state of the probed space object. This poses challenges to implementing such models in scenarios in which position-based risk calculations are critical.

## BNNs

As mentioned above, the BNN-produced MAPE is on the same order as the SNN. However, in contrast to the SNN, the BNN produces a higher covariance root MSE but lower KL divergences. Rather than underapproximating the covariances, the BNN approach produces covariance values that are more consistent with their positional errors with respect to SP calculations. In most cases, if the output probability distribution of a BNN is evaluated at the actual position of the final-state vector's target data, the produced probabilities would be much higher than if calculated by the SNN—but still lower than if this probability were calculated using the target probability distributions themselves.<sup>78</sup>

However, BNNs do generally produce positional standard deviations a factor of approximately two to three times higher than the target data, a feature that would need to be mitigated for to achieve optimal operational performance. Besides the performance criteria listed above, BNNs possess additional features, such as out-of-distribution data detection, that are difficult to extract with an SNN and may offer additional operational advantages (further discussed below). On the other hand, a BNN contains roughly twice as many parameters as an SNN and is more computationally expensive to train and deploy. Over the data regime we studied, the BNN required more than approximately ten times the training periods—meaning that additional engineering would likely be needed to scale the model to larger data regimes because of the increase in model complexity.

## Performance Summary

Our proof-of-concept model demonstrates that covariances can be propagated forward using a neural network; however, there is still much room for improvement. An idealized model would jointly satisfy all evaluation metrics mentioned above. This would result in a model whose final-state covariances and mean predictions both closely matched those of the underlying training data (see the bottom-right panel of Figure 4.6 for a notional example), which in turn would translate to near-zero KL divergences.

---

<sup>78</sup> Evaluating the target probability distributions at the position of the target mean yields probabilities  $\sim 10^{-8}$ .

Over the data regime we studied, the BNN generally offered a higher performance across the discussed performance criteria compared with SNN models. However, further research must be conducted to understand the upper performance bounds of both these methods in operational data regimes, as well as any trade-offs that may be encountered because of the BNN’s comparatively longer training times. In addition, both of these performance bounds must be considered with respect to minimum operational requirements, which we could not establish during this study. Once both the performance bounds of the considered AI models and minimum operational performance requirements have been characterized, an assessment can be made on whether AI models offer a comparative advantage for rapid covariance propagation over alternative techniques, such as low-tolerance SP calculations or SGP models.

## Additional BNN Features

Apart from the ability to produce more-realistic uncertainties, BNNs also offer other advantages that may make them operationally advantageous.

### Out-of-Distribution Detection

Because of their inclusion of epistemic uncertainty, BNNs can express when they are ingesting data that are out of distribution from their training set,<sup>79</sup> producing high-uncertainty estimates in such scenarios that could serve as an alert flag.

Operationally, this situation may arise if the BNN was fed inputs from a space object orbiting in an orbital regime outside the BNN training set. Since the model was not trained on orbits of this nature, it is unlikely to produce accurate results. Ideally, no model would be deployed outside its operating requirements; however, because of process complexities, it is possible that an operator would not always be aware of the input limitations of a black-box model. In such scenarios, operators risk receiving unknowingly inaccurate information. The high output errors that can be produced by a BNN in such scenarios can act as a flag for operators to reevaluate model results.

To demonstrate this feature, we evaluated our optimized BNN model both on one set of inputs from the Starlink constellation, which the model was trained on, and on a second set of inputs from the OneWeb constellation, which the model was not exposed to during training (Figure 4.7). The MAPE performance metric on the OneWeb constellation (~20 percent) was much higher than the corresponding MAPE for the Starlink constellation (~0.05 percent), as expected. However, the produced uncertainties were also generally higher for the OneWeb constellation, suggesting that such inputs might not be processed by the model accurately. Ideally, the error distributions between these two sets would be even more distinct, and such metrics can be considered in future optimizations.

Even within the constellation that a BNN is trained on, there may be certain regions of phase space in which the model has poor training-set coverage, especially considering the near-infinite possibilities of input states. Inputs evaluated from these regions are likely to lead to poorer-quality

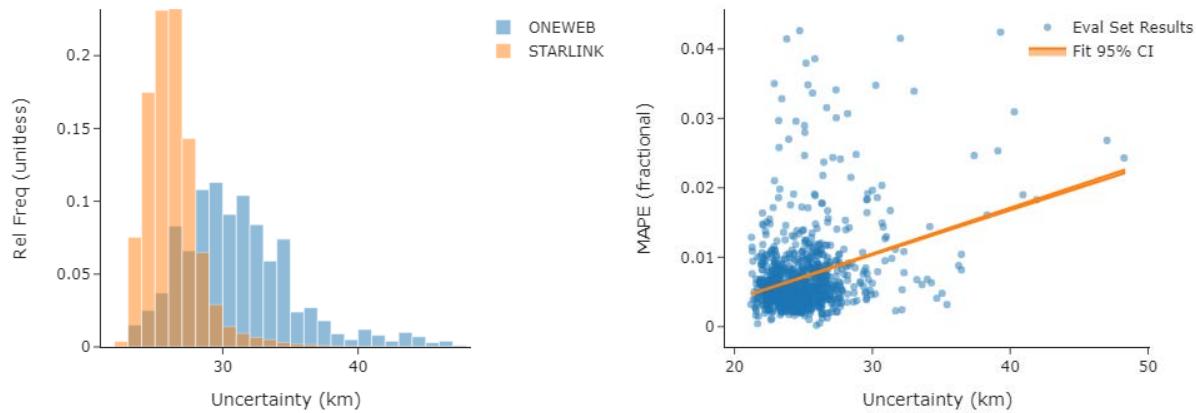
---

<sup>79</sup> Jospin et al., “Hands-On Bayesian Neural Networks—a Tutorial for Deep Learning Users.”

output states. Once again, a BNN may be able to indicate that it is operating in an out-of-distribution zone by producing large epistemic errors in these scenarios.

To evaluate this property, over our Starlink constellation evaluation set, we plotted the overall 3D uncertainty for each prediction (the magnitude of the covariance vector) on the x-axis and the positional MAPE on the y-axis. We see that there is positive, yet weak, correlation between uncertainty and MAPE, indicating that the BNN generally produces higher uncertainties for the output states that it predicts less accurately. Over our experiments, the strength of this correlation did vary, and, in certain instances, no correlation could be observed for specific training epochs. Further optimization could be conducted if this property is advantageous or necessary for an operational use case.

**Figure 4.7. BNN Error Flagging, Relative Frequency (Left) and MAPE (Right)**



NOTE: We demonstrated out-of-distribution performance for the BNN by evaluating predictions on data outside the scope of the BNN training set. The BNN was primarily trained on Starlink satellites, and we evaluated BNN predictions of OneWeb satellites. We see that out-of-distribution predictions have higher uncertainties (left). Additionally, within the Starlink dataset, there is a correlation between high uncertainties and higher percentage error (right). Put together, these two plots provide evidence that the BNN is able to (1) detect out-of-distribution scenarios and (2) flag in-distribution scenarios in which model performance is poor, both serving as potential alerts to operators. CI = confidence interval; Eval = evaluation; Rel Freq = relative frequency.

## Active Learning

Given that the training data for the OSP are generated synthetically using a Monte Carlo simulation, one could theoretically continuously generate additional training data to augment the training process. However, generating data randomly across the input parameter space might not be as effective as focusing on regions of the parameter space in which the model is likely to benefit most.

This type of dynamic adjustment of training sets is known as *active learning*. BNNs have been suggested as more-effective active learners than SNNs in certain scenarios.<sup>80</sup> If integrated into the OSP tool training process, active learning may prevent *model holes*—regions of input parameter space in which model performance is subpar—from forming. Further, this process could be applied successively to millions of data points to continuously increase model quality.

As a proof of concept, we trained an active-learning BNN by executing the following active-learning pseudocode:

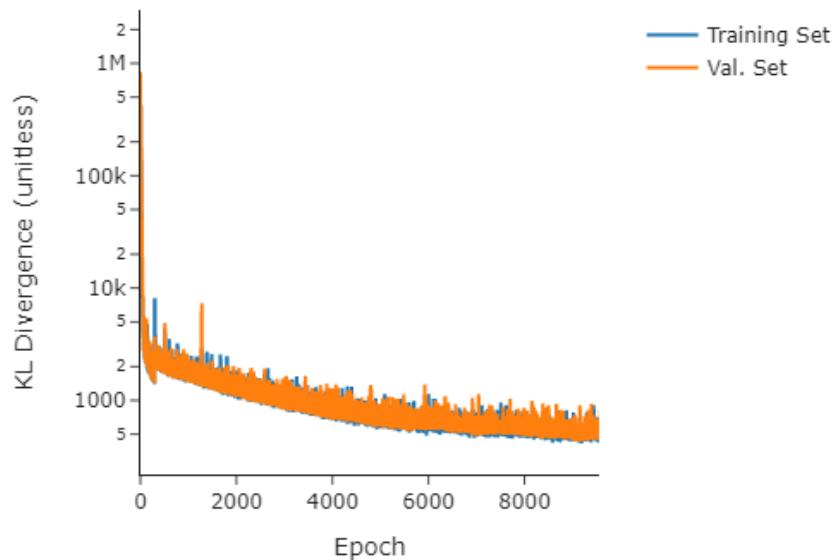
- For  $k$  in  $N$ :
  - Process  $k$  input states, generated randomly across parameter space, through SP simulations to generate  $k$  output states.
  - Split these input-output pairs into a training set,  $T$ , and validation set,  $V$ .
- If  $k! = 1$ ,
  - Concatenate  $T$  with  $F$  from the  $k-1$ th step.
  - Train the BNN for  $E$  epochs on  $T$ .
  - Sort  $V$  by the KL divergence:  $D_{KL}$ .
  - Select the top  $H$  elements from  $V$  and generate  $F$  additional input states randomly perturbed from  $H$  by  $\Delta$  percent.

We present the training curve for a model built using the above procedure in Figure 4.8, with parameters  $K = 5,000$ ,  $E = 300$ ,  $H = 1,250$ ,  $F = 1,250$ , and  $\Delta = 10$  percent. As can be seen from the plot, the training and validation loss continue to decrease as a function of epoch number, indicating successful training. We also present a grid plot of the KL divergence of the trained model on an evaluation set, binning across the XY, XZ, and YZ spatial dimensions of parameter space. See Figure 4.9. The plots are produced for training epoch 1 (top row) and training epoch 1,000 (bottom row). As can be seen from the plot, there are several areas of position space in which the model performs poorly, producing large KL divergences, at epoch 1. At epoch 1,000, these regions of parameter space have lower KL divergences, indicating that the training process was successful at addressing these poor-performance zones.

---

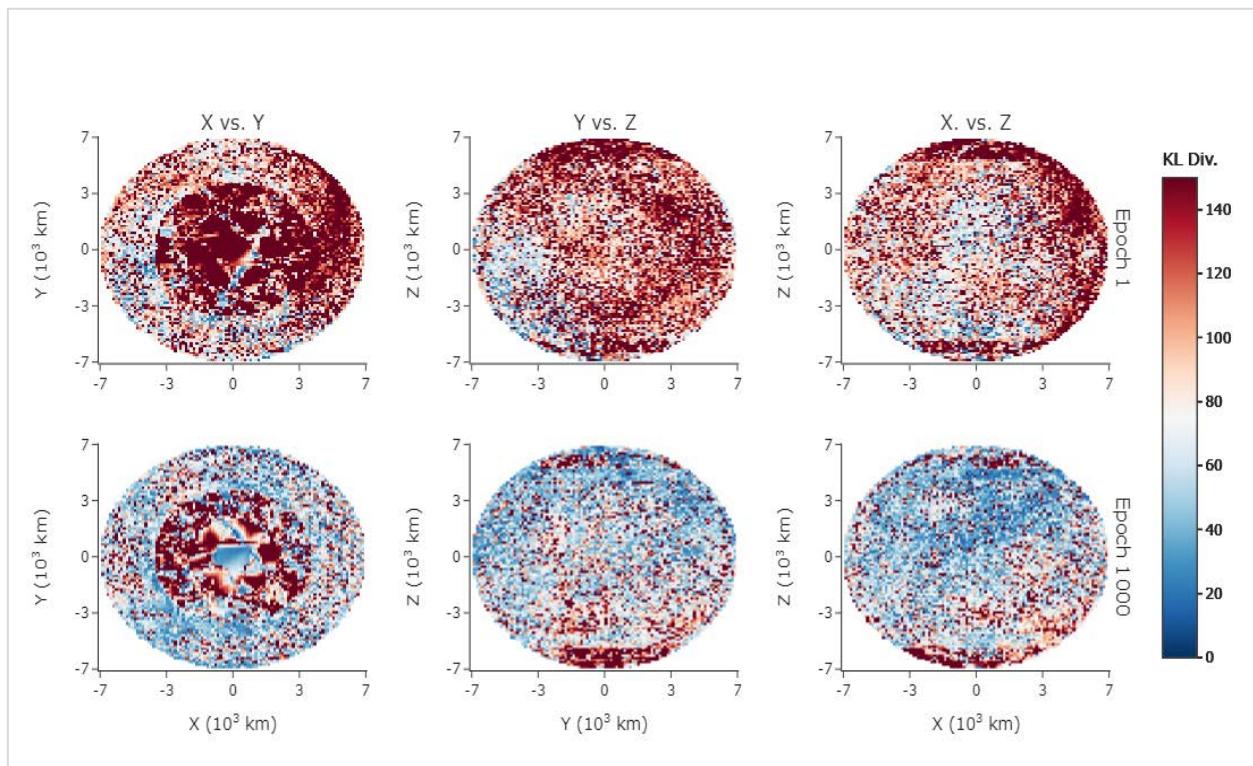
<sup>80</sup> Rakesh and Jain, “Efficacy of Bayesian Neural Networks in Active Learning.”

**Figure 4.8. Active-Learning Training Curve**



NOTE: Val. = value.

**Figure 4.9. Active-Learning Spatial Distribution of KL Divergences Versus Training Epoch**



NOTE: Div = divergence.

Active learning is a well-studied field,<sup>81</sup> and there are many ways in which this simple proof-of-concept procedure could be optimized, depending on operator needs. Because the SDA data infrastructure provides regular updates to state and covariance for all RSOs with consistent data formatting and labeling, active learning may be an attractive AI/ML feature when paired with SDA mission processes. In addition to having access to the abundance of RSO data, active learning has the advantage of reducing maintenance and resource cost, being more adaptable to the rapidly changing space environment, and being more scalable than the alternative.

## Operational Frameworks

Our proof-of-concept model would need to be modified in several ways to fit operational use cases. To offer operational value, the OSP tool would likely need to provide covariance propagation across a wide range of time horizons (calculated in approximate days). Currently, our model propagates only at fixed time intervals of 1,000 seconds. We propose two frameworks to modify this functionality.

### Recursive Propagation

The outputs of our model could be fed into the inputs of our model to extend our predictions for longer time horizons. For example, ten successive 1,000-second propagation model calls could be daisy-chained together in this way to propagate a covariance forward 10,000 seconds.

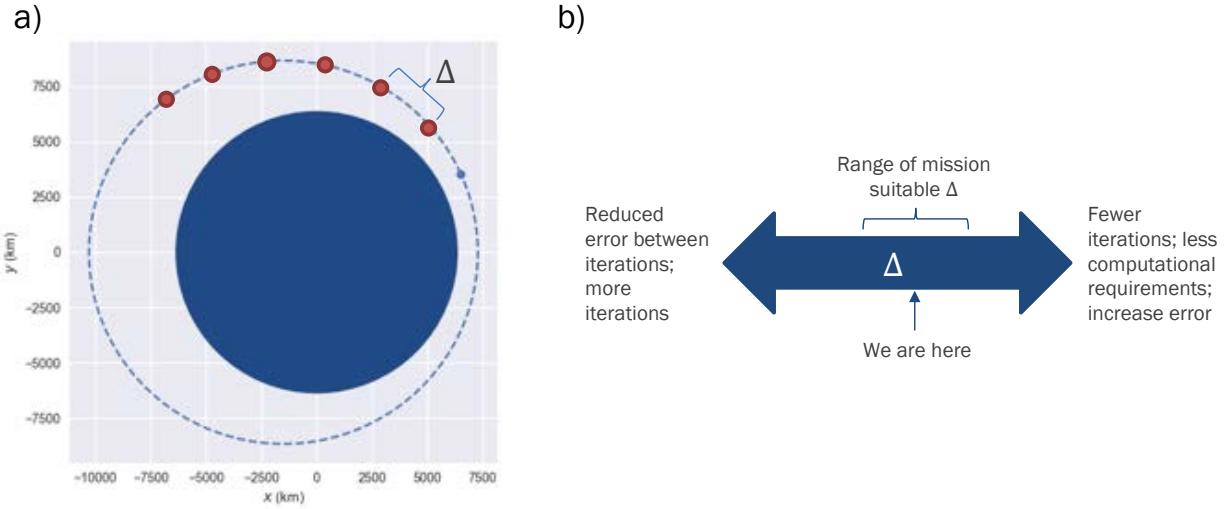
There are a few challenges and drawbacks to this method. First, further research would need to be conducted on how model error propagates across recursive calls similar to numerical propagation techniques. Our current model was trained on a limited set of input states. A daisy-chained model would need the ability to ingest a wide range of input states with differing covariances; adding such flexibility to a model may introduce error that would cause recursive propagation to produce covariances too large for operational use.

Second, recursive propagation offers only time resolutions that are a fixed multiple of the model propagation time step (in this case, 1,000 seconds). Model time steps could be scaled up or down to offer coarser or finer resolutions, respectively. If finer time steps are chosen, timing resolution will be increased, but more recursive calls would be needed to propagate over a given time horizon, increasing computational time and also potentially increasing model error. On the other hand, coarse time steps reduce computational time but offer reduced timing resolution. This trade-off is illustrated in Figure 4.10.

---

<sup>81</sup> Ren et al., “A Survey of Deep Active Learning.”

**Figure 4.10. Notional Recursive Propagation**



NOTE: A recursive propagation of orbit trajectory may use a much coarser time step when compared with numerical propagation techniques, as shown in the left panel. The trade-off for using larger or smaller time steps is presented in the right panel.

Lastly, although our proof-of-concept model produced only 3D positions and 3D position covariances as outputs, a recursive model would also need to output all of its inputs so that these quantities can be forecasted for future model calls. In our case, this would require also producing 3D velocity values and covariances.

With these challenges in mind, BNNs do offer advantages for recursive iteration compared with SNNs. In recursive propagation, a single severely miscalculated time step could cause massive deviations to the final produced output state. BNNs, with the ability to flag out-of-distribution inputs and to produce epistemic uncertainties, could produce warnings for operators in such situations. Hybrid BNN-SP systems could be developed that switch BNN propagation over to SP methods whenever epistemic uncertainty is raised above a certain threshold, allowing the speed advantages of AI/ML methods to be partially realized while also ensuring some degree of error tolerance.

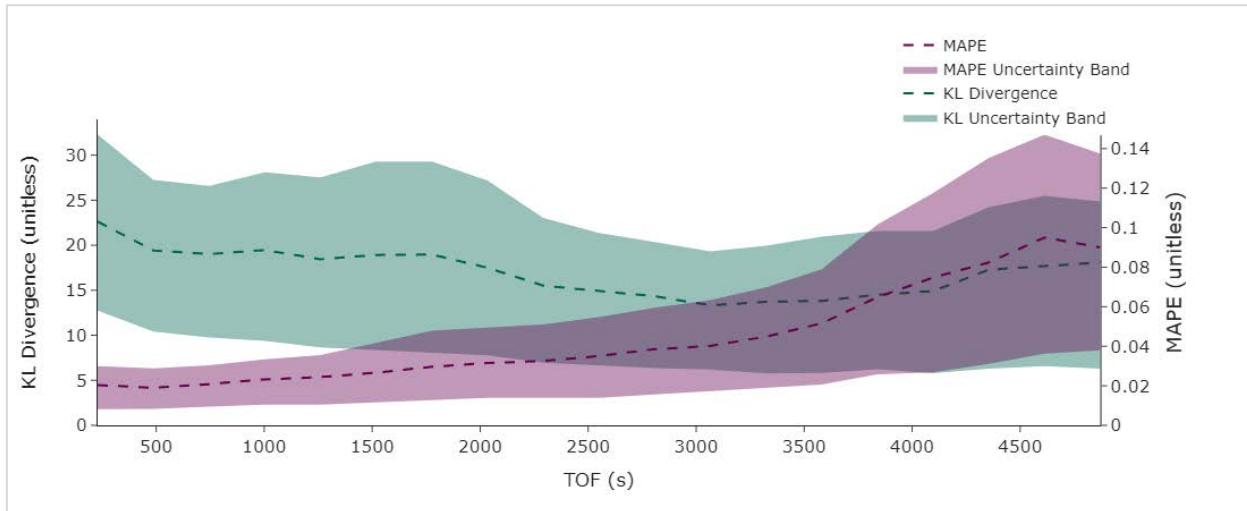
## Flexible Time-Step Models

Another option is to produce a model that can predict covariances over flexible time horizons, increasing the timing resolution of covariance estimation and potentially reducing the need for recursive calls. To provide a proof of concept, we constructed a BNN that predicted future covariance states across a range of times from 100 to 5,000 seconds by ingesting time of flight as an additional model input. The MAPE and KL divergences of our prototype model are shown in Figure 4.11.

As can be seen in Figure 4.11, MAPE grows with increasing time of flight, but the model does offer relatively consistent performance across certain time-horizon zones. The model was trained on 90,000 examples, as compared with the ~140,000 for our fixed time-step model. The average MAPE for this model is ~4 percent, higher than the corresponding value of ~0.5 percent for our fixed time-

step model, because of both the smaller training set size and the increased difficulty of the prediction task.

**Figure 4.11. Flexible Time-Step BNN Performance**



NOTE: TOF = time of flight.

Research must be done on how the following concepts are related: (1) the time horizon over which a flexible time-step model can make predictions, (2) the upper performance bounds of such a model, and (3) the amount of training data needed to achieve stated performance bounds. Flexible time-step models with wide prediction time horizons offer opportunities for the faster covariance propagation and reduce the need for error-inducing recursive calls; however, the extent to which such models can be trained effectively is unknown.

Ultimately, discrete time-step models, flexible time-step models, and recursive model calls may be used in conjunction with one another rather than being considered separately. For example, one could deploy a coarse discrete time-step model to recursively propagate through the bulk of a desired time horizon. Next, an operator could deploy a fine-grain flexible time-step model to propagate the final state to a more specific time of interest within the time resolution of the coarse model.

## Implementation Challenges and Future Work

Although we are able to demonstrate that an OSP is feasible using a BNN algorithm, it is challenging to compare our case study performance (speed or accuracy) against operational performance. We are unaware of existing research that has characterized SP model performance. SP methods are currently the only algorithm available to SDA operators for propagating states and covariances in the HAC. The performance of SP methods is interdependent on three key components: (1) the relative or absolute error tolerances used, (2) the physical assumptions made by the algorithm, and (3) the computing hardware capabilities at SDA facilities, including those operated by the 18th and 19th SDSs. Despite the lack of a comparative study, we have demonstrated that

BNNs are generally faster than the numerical methods they approximate and may exhibit “good enough” estimates to be useful for SDA. Further analysis should be done to benchmark the BNN prediction with real, operationally relevant data.

## Summary

In our second case study, we attempted to predict the future states and covariance of an RSO over various time horizons. We explored the use of two algorithms, BNNs and SNNs, and evaluated the performance of these algorithms with different metrics. We found that both algorithms have relative strengths and weaknesses, but, over the explored data regime, BNNs offer higher performance over metrics most likely to be relevant to risk-based SDA decisionmaking. Ultimately, the upper performance bounds of both methods, as well as minimum operational requirements, must be better established before an assessment can be made of operational feasibility.

In addition, we explored how other features of BNNs, such as out-of-distribution detection and active learning, may be leveraged to further optimize our prototype model. Lastly, we investigated how variable time-step models, recursive propagation, and discrete time-step models may each be leveraged to unlock a wider variety of operational use cases for the OSP tool.

## Chapter 5

# Conclusion

This report provided a detailed look at the technical approaches we took to demonstrate the feasibility of AI/ML within SDA. Managing uncertainty and predicting the future are the core mathematical principles underlying SDA, and we tested AI/ML algorithms, particularly BNNs, to determine their suitability for accomplishing these tasks.

Through the case studies presented in this report, we demonstrated the feasibility of using AI/ML applications to improve the CA process and predict the future states and covariance of RSOs. Our first case study showed that an AI/ML application can flag RSO pairs for possible conjunctions, providing operators with estimates of the distance of closest approach and an uncertainty. In our second case study, we explored the use of BNN and SNN to predict the future states and covariance of RSOs over various time horizons. We found that BNNs offer higher performance on metrics most likely to be relevant to risk-based SDA decisionmaking.

## Key Findings

**Neural networks are capable approximators for complex nonlinear functions. We demonstrated that prediction and classification capabilities can be applied to SDA mission processes with SNNs and BNNs.** Our findings suggest that AI/ML applications can significantly improve the CA process and enhance SDA capabilities. We demonstrated feasibility and pathways to incorporate these tools into existing SDA processes. We also explored how our BNN models can be applied in a variety of ways, such as variable time-step models, recurse propagation, and discrete time step models. However, there is significant work to be done before operationalizing these applications.

**AI/ML tool developers should focus on tools that are compatible with the operational SDA architecture, data infrastructure, and processes.** Given our limited access to operational datasets, we designed a methodology that is representative of the SDA mission processes and tools used by such organizations as the 18th and 19th SDSs to produce our model training sets. Although we cannot directly compare our training sets with those that would be produced by current operational systems, in general more-realistic training data can reduce model uncertainty and result in better predictions and classifications. Any operational AI/ML tool would also benefit from access to the operational SDA tools for clearer performance benchmarking.

**Quantification of risk and uncertainty tolerance provides an operator or analyst with the capability to provide feedback to an AI/ML model by setting thresholds based on acceptable risk.** In our first case study, developing an elliptical screening tool, we explored how an SDA operator or analyst might leverage the uncertainty quantification produced by a BNN. We characterized the performance of our postprocessing classifier based on the threshold of acceptable risk set by an operator or analyst. In doing so, we found significant variance for the classification of both true

positives and false negatives. This performance optimization curve provides an operator with insight into the model discrimination performance versus optimal models, threshold selection, and trade-offs between operating conditions.

**Quantification of risk and uncertainty tolerance can support improved performance of AI/ML tools focused on prediction and classification when compared with SNN approaches.** In both case studies we observed better prediction and classification performance when using methods that account for uncertainty quantification. This is not a general statement, as there may be instances where the reverse is possible. We do note that BNNs are preferable to SNNs when uncertainty is a necessary output, when there are limited available training data, and when the detection of outlier predictions is important to flag for an operator.

**Active learning may be an attractive AI/ML feature when paired with SDA mission processes.** The SDA mission processes and supporting data infrastructure provide regular updates to state and covariance for all RSOs, with consistent data formatting and labeling. Provided with this abundance of real-time RSO data, active-learning protocols may offer such advantages as reducing maintenance and resource costs, providing adaptability to rapidly changing space environments, and offering more scalability than alternative model training techniques.

## Recommendations

**The development of AI/ML tools requires significant investment in ensuring high-quality training data. The Office of the Chief Scientist of the U.S. Air Force, with input from SDA operators, should consider the availability of such data for AI/ML investment.** The availability of high-quality training data is another enabler for AI/ML impact. AI/ML can be powerful at learning complex relationships between inputs and outputs of a task, but AI/ML performance is often limited by the quality of model training sets. To develop effective AI/ML applications for SDA tasks, it is essential to start collecting high-quality input-output data from analysis tasks to curate a database that can train future systems. This enables AI/ML to learn from best analyst practices and slowly automate more-complex tasks as the database grows. Given the evolving nature of the space environment, this effort will need to be continuous.

**AI/ML tool developers should consider SDA processes and limitations that have implications for tool design, including the selection of algorithms, metrics of performance, and benchmark requirements.** Since AI/ML tools are only as good as the rules they are given, more clarification is necessary about what these rules should be based on in terms of mission effectiveness. This approach requires a detailed look at what these metrics should be moving forward. Continued support for these efforts would enable higher impact from AI/ML tools designed to support sensor tasking and prioritization and would avoid developing AI/ML solutions that operators cannot implement and use.

**DoD should continue to consider the value of uncertainty quantification methodologies when developing and deploying operational AI/ML tools.** Uncertainty quantification has the potential to improve decisionmaking, risk management, model selection and optimization, and robustness. However, using these algorithms may be challenging because of the complexities of these models and their computational overhead requirements. AI/ML practitioners within DoD at large should be knowledgeable of these trade-offs when considering the development of AI/ML tools.

# Abbreviations

3D	three-dimensional
ADC	Automatic Differential Corrections
AFSPC	Air Force Space Command
AI	artificial intelligence
ASP	Automatic Special Perturbations Ephemeris Generation
ASW	Astrodynamics Support Workstation
AutoCA	Automatic Conjunction Assessment
BNN	Bayesian neural network
C2	command and control
CA	conjunction assessment
CAVENet	Command, Analysis, Verification, and Ephemeris Network
COM	composite
DC	differential corrections
DoD	U.S. Department of Defense
EGP	extended general perturbations
ESP	elliptical screening process
HAC	High Accuracy Catalog
KL	Kullback-Leibler
LEO	low earth orbit
MAPE	mean absolute percentage error
ML	machine learning
MSE	mean squared error
NASA	National Aeronautics and Space Administration
O/O	owner-operator
OA	orbital analyst
OMM	Orbit Mean-Elements Message
OSA	orbital safety analyst
OSP	orbital state propagator
P <sub>c</sub>	probability of collision
RSO	resident space object
SDA	space domain awareness
SDS	space defense squadron
SGP	simplified general perturbation
SNN	standard neural network

SP	special perturbation
SPADOC	Space Defense Operations Center
SSA	space situational awareness
SSN	Space Surveillance Network
SuperCOMBO	Super Computation of Miss Between OrbitS
TCA	time of closest approach
TLE	two-line element
USSF	U.S. Space Force
USSPACECOM	U.S. Space Command
XML	Extensible Markup Language

# References

- 18th and 19th SDSs—See 18th and 19th Space Defense Squadrons.
- 18th and 19th Space Defense Squadrons, Combined Force Space Component Command, *Spaceflight Safety Handbook for Satellite Operators*, version 1.7, April 2023.
- Albon, Courtney, “US Space Force Preparing to Decommission Legacy Command and Control System,” C4ISRNET, March 16, 2022.
- Alfano, Salvatore, “Relating Position Uncertainty to Maximum Conjunction Probability,” *Journal of the Astronautical Sciences*, Vol. 53, No. 2, 2005.
- Alfano, Salvatore, and Daniel Oltrogge, “Probability of Collision: Valuation, Variability, Visualization, and Validity,” *Acta Astronautica*, Vol. 148, July 2018.
- Alfriend, Kyle T., Maruthi R. Akella, Joseph Frisbee, James L. Foster, Deok-Jin Lee, and Matthew Wilkins, “Probability of Collision Error Analysis,” *Space Debris*, Vol. 1, March 1999.
- Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaria, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan, “Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions,” *Journal of Big Data*, Vol. 8, No. 53, 2021.
- Aptima, Inc., “Space Domain Awareness,” webpage, undated. As of September 20, 2023:  
<https://www.apptima.com/solutions/capabilities/space-domain-awareness/>
- Astromaterials Research and Exploration Science, National Aeronautics and Space Administration, “Legend: 3D/OD Evolutionary Model,” webpage, undated. As of April 17, 2024:  
<https://orbitaldebris.jsc.nasa.gov/modeling/legend.html>
- Bergstra, James, Brent Komer, Chris Eliasmith, Dan Yamins, and David D. Cox, “Hyperopt: A Python Library for Model Selection and Hyperparameter Optimization,” *Computational Science and Discovery*, Vol. 8, No. 1, 2015.
- Bergstra, James, Daniel Yamins, and David Cox, “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures,” *Proceedings of the 30th International Conference on Machine Learning*, Vol. 8, 2013.
- Booz Allen Hamilton, “Artificial Intelligence for Space Missions,” webpage, undated. As of April 1, 2024:  
<https://www.boozallen.com/markets/space/artificial-intelligence-for-space-missions.html>
- Chandra, Rohitash, and Yixuan He, “Bayesian Neural Networks for Stock Price Forecasting Before and During COVID-19 Pandemic,” *PLOS One*, Vol. 16, No. 7, July 2021.
- Clark, Colin, “What About JMS? Air Force Reanimates ‘Old Clunker’ Space Tracking System,” *Breaking Defense*, April 8, 2019. As of August 16, 2023:  
<https://breakingdefense.com/2019/04/what-about-jms-air-force-reanimates-old-clunker-space-tracking-system/>

- Conkey, Dave, and Mitchell Zielinski, *Assessing Performance Characteristics of the SGP4-XP Propagation Algorithm*, a.i. solutions, September 2022.
- Der, Gim J., and Roy Danchick, "Conversion of Osculating Orbital Elements to Mean Orbital Elements," paper presented at the Flight Mechanics/Estimation Theory Symposium 1996, May 1, 1996.
- Erwin, Sandra, "Space Force to Increase Spending on Technologies to Turn Data into Knowledge," *SpaceNews*, May 17, 2021. As of September 20, 2023:  
<https://spacenews.com/space-force-to-spend-years-and-billions-of-dollars-to-convert-data-into-knowledge/>
- Erwin, Sandra, "Scout Wins Defense Contract to Demonstrate Utility of Commercial Data from Sensors in Space," *SpaceNews*, July 22, 2022. As of September 20, 2023:  
<https://spacenews.com/scout-wins-defense-contract-to-demonstrate-utility-of-commercial-data-from-sensors-in-space/>
- Esposito, Piero, "BLiTZ—Bayesian Layers in Torch Zoo," GitHub, released May 20, 2020. As of August 21, 2023:  
<https://github.com/piEsposito/blitz-bayesian-deep-learning/>
- Fletcher, Justin, "AI and Autonomy for SDA: Progress and Prospects," Space Systems Command, Space Domain Awareness Division, August 12, 2022, Not available to the general public.
- Fortuin, Vincent, "Priors in Bayesian Deep Learning: A Review," *International Statistical Review*, Vol. 90, No. 3, 2022.
- Fortuin, Vincent, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Rätsch, Richard E. Turner, Mark van der Wilk, and Laurence Aitchison, "Bayesian Neural Network Priors Revisited," arXiv, arXiv:2102.06571, February 2021.
- Hilton, C. G., and J. R. Kuhlman, "Mathematical Models for the Space Defense Center," Philco-Ford, November 1966.
- Hoots, Felix R., and Ronald L. Roehrich, *Spacetrack Report No. 3: Models for Propagation of the NORAD Element Sets*, U.S. Air Force Aerospace Defense Command, 1980.
- Hüllermeier, Eyke, and Willem Waegeman, "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods," *Machine Learning*, Vol. 110, 2021.
- Hyperopt: Distributed Asynchronous Hyper-parameter Optimization, homepage, undated. As of August 1, 2023:  
<http://hyperopt.github.io/hyperopt/>
- Ingram, Christopher, Jaycie Bishop, Phillip Cunio, Douglas Hendrix, and Mark W. Jeffries, Jr., "ExoALERT: 1 Year of AI-Enabled Space Traffic Management Services at GEO," Advanced Maui Optical and Space Surveillance Technologies Conference, September 2022.
- Joint Publication 3-14, *Space Operations*, U.S. Joint Chiefs of Staff, April 10, 2018, change 1, October 26, 2020.
- Jospin, Laurent Valentin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun, "Hands-On Bayesian Neural Networks—a Tutorial for Deep Learning Users," *IEEE Computational Intelligence Magazine*, Vol. 17, No. 2, April 2022.
- Joyce, James M., "Kullback-Leibler Divergence," in Miodrag Lovric, ed., *International Encyclopedia of Statistical Science*, Springer, 2011.

- Kelso, T. S., "A New Way to Obtain GP Data (aka TLEs)," CelesTrak, last updated September 9, 2023. As of September 1, 2023:  
<https://celestak.org/NORAD/documentation/gp-data-formats.php>
- Kingma, Diederik P., and Jimmy Lei Ba, "ADAM: A Method for Stochastic Optimization," paper presented at the 3rd International Conference on Learning Representations, 2015.
- Kozai, Y., "The Motion of a Close Earth Satellite," *Astronomical Journal*, Vol. 64, 1959.
- Lane, Ben, Mark Poole, Matt Camp, and Jeremy Murray-Krezan, "Using Machine Learning for Advanced Anomaly Detection and Classification," paper presented at the Advanced Maui Optical and Space Surveillance Technologies Conference, 2016.
- McLachlan, Scott, Kudakwashe Dube, Graham A. Hitman, Norman E. Fenton, and Evangelia Kyrimi, "Bayesian Networks in Healthcare: Distribution by Medical Condition," *Artificial Intelligence in Medicine*, Vol. 107, July 2020.
- Michelmore, Rhiannon, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska, "Uncertainty Quantification with Statistical Guarantees in End-to-End Autonomous Driving Control," paper presented at the 2020 IEEE International Conference on Robotics and Automation, May 31, 2020.
- Montenbruck, Oliver, and Eberhard Gill, *Satellite Orbits: Models, Methods and Applications*, Springer, 2000.
- Moomey, Daniel, Rachael Falcon, and Arbab Khan, "Trending and Analysis of Payload vs. All Low Earth Conjunction Data Messages Below 1,000 km, from 2016 Through 2021," paper presented at the Advanced Maui Optical and Space Surveillance Technologies Conference, September 2022.
- Moulton, Forest Ray, *Introduction to Celestial Mechanics*, 2nd rev. ed., Dover Publications, 1984.
- Myong, R. S., A. Karchani, and O. Ejtehadi, "A Review and Perspective on a Convergence Analysis of the Direct Simulation Monte Carlo and Solution Verification," *Physics of Fluids*, Vol. 31, No. 6, June 2019.
- NASA—See National Aeronautics and Space Administration.
- National Aeronautics and Space Administration, *Application and Analysis of Satellite Orbit Prediction Techniques*, JSC-12571 Internal Note No. 77-FM-19, April 1977. As of March 21, 2024:  
<https://ntrs.nasa.gov/api/citations/19790020037/downloads/19790020037.pdf>
- National Aeronautics and Space Administration, *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*, NASA/SP-20205011318, 2020.
- National Coordination Office for Space-Based Positioning, Navigation, and Timing, "Control Segment," webpage, GPS.gov, last updated August 9, 2021. As of July 5, 2024:  
<https://www.gps.gov/systems/gps/control/>
- Omitron, "Astrodynamics Support Workstation (ASW)," webpage, undated. As of July 5, 2024:  
<https://www.omitron.com/astrodynamics-support-workstation/>
- Panaretos, Victor M., and Yoav Zemel, "Statistical Aspects of Wasserstein Distances," *Annual Review of Statistics and Its Application*, Vol. 6, 2019.
- Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio, "On the Difficulty of Training Recurrent Neural Networks," in *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28, 2013.

Perovich, Nicholas, Zachary Folcik, and Rafael Jaimes, "Satellite Maneuver Detection Using Machine Learning and Neural Network Methods Behaviors," paper presented at the IEEE Conference on Aerospace, March 5–12, 2022.

Peterson Schriever Space Force Base, U.S. Space Force, "19th Space Defense Squadron," fact sheet, December 2023. As of September 18, 2024:

<https://www.petersonschriever.spaceforce.mil/About-Us/Fact-Sheets/Display/article/1060278/>

Poliastro—Astrodynamics in Python, homepage, undated. As of September 20, 2023:

<https://docs.poliastro.space/en/stable/>

Poore, Aubrey B., Jeffrey M. Aristoff, and Joshua T. Horwood, eds., *Covariance and Uncertainty Realism in Space Surveillance and Tracking*, Working Group on Covariance Realism, Air Force Space Command Astrodynamics Innovation Committee, June 2016.

Rakesh, Vineeth, and Swayambhoo Jain, "Efficacy of Bayesian Neural Networks in Active Learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang, "A Survey of Deep Active Learning," *ACM Computing Surveys*, Vol. 54, No. 9, 2021.

Russell, Stuart, and Peter Norvig, eds., *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice Hall, 2010.

SciPy, "scipy.integrate.DOP853," webpage, undated. As of September 21, 2023:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.DOP853.html>

SES Space and Defense, "U.S. Space Command Deputy Commander: There Have Been Three Distinct Space Ages," August 23, 2023. As of April 17, 2024:

<https://sessd.com/gsr/u-s-space-command-deputy-commander-there-have-been-three-distinct-space-ages>

Shaw, John E., "Space Domain Awareness (SDA), Memorandum," Air Force Space Command, October 4, 2019.

Space Systems Command, "Space Systems Command's New TAP Lab Kicks Off with Project Apollo to Accelerate Innovation in Space Domain Awareness," media release, September 25, 2023.

Space-Track.org, homepage, undated. As of September 21, 2023:

<https://www.space-track.org/auth/login>

Temple, Dwight, "Synthetic Heterogeneous Anomaly and Maneuver—Neural Network Event Winnowing System (SHAM NNEWS)," paper presented at the Advanced Maui Optical and Space Surveillance Technologies Conference, 2018.

U.S. Code, Title 15; Chapter 119, National Artificial Intelligence Initiative; Section 9401, Definitions.

U.S. Department of Defense, *Defense Space Strategy Summary*, June 2020.

U.S. Government Accountability Office, *Space Command and Control: Comprehensive Planning and Oversight Could Help DOD Acquire Critical Capabilities and Address Challenges*, GAO-20-146, October 2019.

U.S. Government Accountability Office, *Space Situational Awareness: DOD Should Evaluate How It Can Use Commercial Data*, GAO-23-105565, April 2023.

U.S. Space Force, *Spacepower: Doctrine for Space Forces*, June 2020.

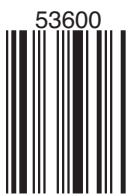
U.S. Space Force, "18th Space Defense Squadron," fact sheet, May 2, 2022.

- Vallado, David A., *Fundamentals of Astrodynamics and Applications*, 4th ed., Microcosm Press, 2013.
- Vallado, David A., and Paul Crawford, "SGP4 Orbit Determination," American Institute of Aeronautics and Astronautics, 2008.
- Weeden, Brian, "2007 Chinese Anti-Satellite Test Fact Sheet," Secure World Foundation, last updated November 23, 2010. As of July 5, 2024:  
[https://swfound.org/media/9550/chinese\\_asat\\_fact\\_sheet\\_updated\\_2012.pdf](https://swfound.org/media/9550/chinese_asat_fact_sheet_updated_2012.pdf)
- Weeden, Brian, "Space Situational Awareness Fact Sheet," Secure World Foundation, last updated May 2017. As of July 5, 2024:  
[https://swfound.org/media/205874/swf\\_ssa\\_fact\\_sheet.pdf](https://swfound.org/media/205874/swf_ssa_fact_sheet.pdf)
- Wehbe, L., and A. Ramdas, "Introduction to Machine Learning," lecture, Carnegie Mellon University, 2019.
- Wilson, Cynthia, and Diana McKissock, "18 SDS Small Satellite Support," paper presented at the 36th Annual Small Satellite Conference, August 2022.
- Woodie, Alex, "Why AI Is a Critical Capability for the US Space Force," *Datanami*, March 2, 2020. As of August 21, 2023:  
<https://www.datanami.com/2020/03/02/why-ai-is-a-critical-capability-for-the-us-space-force/>
- Woodward, Douglas, Madison Hobbs, James Andrew Gilbertson, and Nicholas Cohen, "Uncertainty Quantification for Trusted Machine Learning in Space System Cyber Security," paper presented at the IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT), July 2021.
- Yu, Haofan, Aldyandra Hami Seno, Zahra Sharif Khodaei, and M. H. Ferri Aliabadi, "Structural Health Monitoring Impact Classification Method Based on Bayesian Neural Network," *Polymers*, Vol. 14, No. 19, 2022.
- Zehnder, Jeff, "CU Boulder Lands \$5.5 Million Air Force Project to Advance Orbital and AI Research," University of Colorado, Boulder, August 23, 2022. As of August 21, 2023:  
<https://www.colorado.edu/aerospace/2022/08/23/cu-boulder-lands-55-million-air-force-project-advance-orbital-and-ai-research>
- Zhang, Li Ang, Krista Langeland, Jonathan Tran, Jordan Logue, Prateek Puri, George Nacouzi, Anthony Jacques, and Gary J. Briggs, *Artificial Intelligence and Machine Learning for Space Domain Awareness: Characterizing the Impact on Mission Effectiveness*, RAND Corporation, RR-A2318-1, forthcoming.

To address the growing demands of operating in the space domain, the U.S. Space Force and space domain awareness (SDA) operators must determine how to prioritize sensor observations more effectively, scale up to meet the sheer volume of resident space objects, and develop analytic capabilities that reflect the complexity of orbital mechanics and space operations, all while maintaining the responsiveness necessitated by operations in a warfighting domain. Although artificial intelligence and machine learning (AI/ML) tools have the potential to help meet these SDA challenges, the impact of these tools on the overall success of the SDA mission is not well understood, and this lack of understanding is a barrier to plan for and optimize the integration of these tools..

This report documents technical approaches to demonstrating the feasibility of AI/ML when applied to the U.S. Space Force and the space domain awareness mission. The ability to estimate the current and future states and uncertainties of a resident space object using mathematical and numerical techniques is fundamental for current SDA processes. The authors tested AI/ML algorithms, particularly Bayesian neural networks, to determine their suitability for these tasks. The research presented in this report focuses on the highly resource-intensive conjunction assessment mission under the broader SDA mission set. The two case studies focus on prediction and classification capabilities of neural networks and the use of these capabilities to improve the conjunction assessment. The authors found that Bayesian neural networks offer suitable performance trade-offs on metrics most likely to be relevant to risk-based SDA decisionmaking when compared with traditional processes and greater performance on metrics when compared with standard neural networks.

\$36.00

ISBN-10 1-9774-1414-1  
ISBN-13 978-1-9774-1414-4

9 781977 414144

[www.rand.org](http://www.rand.org)