Adversarial Machine Learning for Privacy-Preserving Space Situational Awareness

Conference Paper · October 2024

DOI: 10.1109/MobiSecServ63327.2024.10760053

Neha Chauhan
Wichita State University

Adversarial Machine Learning for Privacy-preserving Space Situational Awareness

Neha Chauhan
School of Computing
Wichita State University
Wichita, KS, USA
nxchauhan@shockers.wichita.edu

Sergio A. Salinas
School of Computing
Wichita State University
Wichita, KS, USA
sergio.salinasmonroy@wichita.edu

***Abstract*—Satellite operators rely on space situational awareness (SSA) data to avoid collisions with other space objects. However, SSA data can also be used by adversaries to launch both kinetic and non-kinetic attacks against satellites. An existing approach to prevent these attacks consists in labeling a sensitive satellite as a debris in the SSA data. Its objective is to hide the satellite's true label from potential adversaries. Unfortunately, adversaries can easily determine the true label of a satellite based on the orbit information provided by the SSA system. In this paper, we propose a novel adversarial machine learning approach to conceal sensitive satellites in SSA data from potential adversaries. Our method conceals the satellite by changing both its label and the orbit information. The resulting SSA data entry cannot be used to determine that the space object has been mislabeled. Our experimental results show that our proposed method reduces the ability of an adversary to determine the true label a of a concealed satellite to less than 50%.**

***Index Terms*—SSA, Adversarial Machine Learning, Satellite Information Security, Data Concealment, Space Debris.**

I. Introduction

The space surrounding Earth is becoming increasingly crowded with both satellites and debris, and thus more challenging to navigate. There are over 2,000 in-orbit satellites and more than 1 million debris orbiting Earth. These numbers are expected to grow in the next decade due to an increase in commercial missions [1].

Satellite operators rely on space situational awareness (SSA) data to plan their missions in a way that avoids collisions with other space objects. SSA systems detect, track, and identify both satellites and debris using radar and telescope data from both ground-based and in-orbit equipment [2]. Space operators can access SSA data by either operating their own SSA systems (i.e., the United States Space Force operates its own SSA system), or by purchasing it from a commercial third-party provider.

However, SSA data can also be used by adversaries to launch jamming attacks using ground-based or orbit-based transmitters, eavesdropping attacks using other satellites, and kinetic attacks using energy weapons or other satellites [3], [4]. Besides protecting the satellites themselves, commercial operators are interested in protecting proprietary information such as radio coverage and other details of their operations that can be leaked through SSA data. We show an example architecture of an SSA system in Fig. 1.

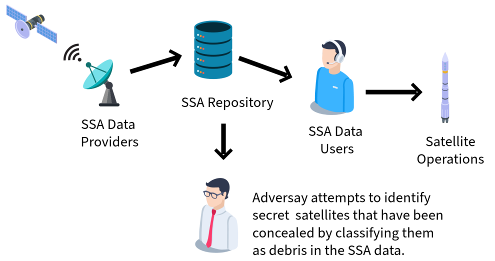

SSA Data Providers
SSA Repository
SSA Data Users
Satellite Operations
Adversay attempts to identify secret satellites that have been concealed by classifying them as debris in the SSA data.

Fig. 1: A system architecture and threat model for SSA.

There are some existing techniques that aim to prevent such attacks against satellites. These techniques can be broadly classified into orbit data obfuscation [5] and classification label flipping [6]. Orbit data obfuscation techniques add random noise to the orbit determination data provided by the SSA, while classification label flipping simply labels a satellite as a debris.

Unfortunately, these existing approaches are insufficient to conceal the true nature of a satellite from adversaries. Orbit data obfuscation techniques only aim at protecting the orbit without providing a way to conceal the true label of a satellite. Moreover, adversaries can determine the true label of a space object with high accuracy by using a machine learning model trained with orbit determination data from an SSA repository [6].

To address this challenge, we propose an adversarial machine learning approach that effectively conceals the true nature of a space object. Specifically, our proposed approach not only modifies the label associated with the space object, but also modifies the orbit determination data. We observe that under our proposed attacks the accuracy of the best performing machine learning model used to determine the type of space object can be reduced to below 50%, that is, worse than a random guess.

II. Related works

1) Privacy-preserving SSA Systems: Although there is a rich literature on SSA systems [7], [8], there are only a few works on approaches that can be used to conceal the true classification of a satellite in an SSA repository.

Das and Bhattacharya [5] propose to add artificial noise to the measurements used by a Kalman filter to estimate the position of a satellite. However, this approach only aims to conceal the orbit of the object. It does not consider the problem of concealing the true label of a sensitive satellite.

More recently, Pavur et al. [6] propose a label-flipping approach that can be used to conceal sensitive satellites in the SSA repository. Specifically, the authors propose to change the object type label of a sensitive satellite from satellite to debris. This method aims to deceive potential adversaries while allowing for safe navigation as the orbit determination data is left unchanged. However, as the authors show in the same work [6], it is possible for an adversary to easily spot the label-flipping attacks using a random forest classifier. In their experiments, the authors show that an adversary using the classifier can detect the attack with 98% accuracy.

Consequently, we observe that there is a need for a mechanism that allows operators to conceal the true nature of their sensitive satellites. In the rest of this paper, we propose an adversarial machine learning approach that can be used by operators to conceal the true nature of their satellite from adversaries.

2) Adversarial Machine Learning: In this subsection, we provide a brief overview of adversarial machine learning (AML). AML refers to attacks against machine learning models. These attacks can be broadly classified into inference attacks and training data poisoning attacks. In this work, we focus on showing that it is possible for the SSA repository to design inference attacks that can allow it to maintain its secret satellites concealed from machine learning models that can reveal their true nature. We leave poisoning attacks against the training phase of machine learning models for future work.

There are several works that design inference attacks against machine learning models [9]. Some of them focus on inference attacks against random forests [10], [11]. Specifically, Aiken et al. [11] first propose a random forest to detect network intrusion attacks. They then craft malicious samples that cause the random forest classifier to make incorrect predictions. Their main approach is to find malicious samples is to observe the random forest's prediction confidence under varying values of certain input features. Owezarski [10] also designs adversarial samples against a random forest designed to detect network attacks with similar techniques.

Although previous works address similar problems, there are no works that consider inference attacks against SSA data.

III. Background on Orbit Determination

Before describing our proposed adversarial machine learning methods to conceal the true label of a sensitive satellite, we provide a brief overview about SSA data and how it is used to determine the orbit of a space object.

A. Orbit determination with TLE data

The current and future position of a space object can be determined through their two-line element (TLE) set. The TLE is a vector provided by the SSA repository that identifies an object's orbit around the Earth. TLEs are calculated based on multiple observations of the space object's position through a process called orbit determination [12].

The random forest model that we consider in Section V takes the TLE vector of an object as input and outputs a label prediction: satellite or debris. In the rest of this section, we describe each of the orbital elements that form a TLE.

1) Mean Anomaly (M): The mean anomaly defines the angular position of the object at a particular point in time. This is the only value in the TLE that is constantly changing. The rest of the values are assumed to be constant.

2) Mean Motion (n): The mean motion indicates the average angular velocity.

3) Eccentricity (e): The eccentricity indicates how elliptical or circular is the object's orbit.

4) Inclination (i): Inclination refers to the tilt of the orbit with respect to the Earth's equator.

5) Right Ascension of the Ascending Nodes (RAAN) (Ω): The RAAN indicates the angle between the reference direction (usually the vernal equinox) and the point where the satellite or debris orbit intersects the Earth's equatorial plane during its ascending node.

6) Argument of periapsis (AoP) (ω).: The AoP denotes the angular position of the periapsis with respect to the ascending node. The AoP is also referred to as Argument of pericenter.

Besides the above six orbital elements that determine an object's current orbit, TLEs are often reported with two additional elements that are used to predict how an object's orbit will change due to atmospheric drag [13]. Namely,

7) Epoch (t_0): The epoch denotes time at which the TLE is valid.

8) Revolutions at epoch (R): This value is the total number of revolutions completed by the object since the epoch.

9) Ballistic Coefficient or Drag term (B^*): The ballistic coefficient is a measure of the atmospheric drag experienced by the space object.

Finally, the TLE data includes a label for each object [13]. The label is used to indicate if the object is debris, a satellite, or a rocket used to launch satellites. In this paper, we only focus on TLEs with debris and satellite labels. The reason is that satellite operators focus on concealing their satellites, not their rockets.

B. SSA Systems

There are only a few SSA systems in existence today due to the extremely large cost of operating an SSA infrastructure. For example, the United States Space Surveillance Network (SSN) consists of more than 20 dedicated location plus additional military installations [14]. The US government further enhances its SSA database by purchasing data from commercial providers [15].

Although modern satellites can report their own positions, this information is usually only available to the satellite operator. Most position observations are obtained through radar sensors and electro-optical telescopes [16].

IV. System Model

In this section, we describe in details our considered scenario, including the SSA system architecture, the adversary aims and capabilities, and the random forest classifier used to detect label-flipping in SSA data.

A. System Architecture

We consider a system architecture formed by an SSA data repository, SSA data providers, and SSA data users as shown in Fig. 1. The SSA data repository maintains a database of labeled TLE data as described in Section III. The SSA data providers sell their data to the SSA data repository to augment its dataset. The SSA data users employ the data in the SSA repository to safely navigate their satellites in space.

We assume that the SSA data repository is also a satellite operator aiming to conceal the true label of its sensitive satellites. To achieve this, it flips the labels of its sensitive satellites from satellite to debris. We also assume that SSA data providers could be satellite operators with sensitive satellites, and thus also flip the labels of their satellites.

B. Threat Model

The adversary's objective is to find the sensitive satellites in the data published by the SSA repository that have been labeled as debris. To find the sensitive satellites, the adversary trains a machine learning model that takes TLEs as input and predicts whether the TLE belongs to a satellite or a debris. The adversary uses the trained model to predict the labels of the TLEs published by the SSA data repository. If the predicted label differs from the label reported by the SSA data repository, the adversary determines the TLE label has been flipped. Otherwise, it assumes the label is true.

We assume the adversary uses a random forest classifier since it is the machine learning model that offers the best performance for label-flipping detection in SSA data [6]. We assume the adversary uses a trusted SSA data set to train the random forest classifier. For example, the adversary may have access to historical SSA data or operates its own telemetry network. Furthermore, the adversary assumes that the SSA data repository and the SSA data providers only flip the labels and leave the rest of the sensitive satellite's TLE entries unchanged.

1) The Adversary's Random Forest: Formally, the adversary' random forest classifier can be defined as follows. Let the TLEs in the adversary's dataset be $X \in \mathbb{R}^{n \times m}$, where n is the number of TLEs and m is the number of orbital elements in a TLE. Let the TLE labels in the adversary dataset be $Y \in \mathbb{S}, \mathbb{D}^n$, where S denotes a satellite and D denotes a debris.

Then, the random forest classifier problem is given by $f : \mathbb{R}^{n \times m} \rightarrow S, D$. To find random forest f, the adversary uses the standard random forest training algorithms such as the ones provided by the Scikit-learn library [17]. Following standard practice, the adversary trains the random forest using a majority of X and the corresponding labels in Y as the training dataset, e.g., 80%. The remaining samples are used for testing.

V. Privacy-preserving Adversarial TLEs to Evade Label-flipping Detection Models

In this section, we describe an adversarial machine learning technique that allows satellite operators to craft TLEs for their sensitive satellites to evade detection by the adversary's random forest classifier described in IV-B. These TLEs can then be published by the SSA data repository.

Our proposed approach involves two steps. In the first step, we identify the orbital elements that have the largest influence in the decisions of the random forest model. We use the Gini index of the random forest model to rank the features by importance. Although it is possible to change the value of all orbital elements of a sensitive satellite's TLE, we are interested in finding the smallest changes that will evade the label-flipping detection model. The reason is that the TLEs published by the SSA data repository should still be useful for safe navigation.

Once the most influential orbital elements have been identified, the second step involves crafting the sensitive satellite's TLEs. Our approach replaces the values of the most influential orbital elements in a TLE with debris-like values. We explore both constant and probabilistic approaches to replace the original TLE values with debris-like values.

A. Dataset

To better explain our adversarial sample crafting approach, we first describe a real-world SSA dataset. Specifically, we utilize the public SSA dataset from the United States Space Surveillance Network (SSN) catalog [13]. Although this repository includes TLEs since the launch of Sputnik I in 1957, we only download data between March 1, 2019, and March 1, 2020. The reason is that this will allow us to replicate the random forest described in [6] which achieves a 98% accuracy (see Section VI). The dataset includes approximately 2 million TLEs describing around 19 thousand unique objects. The dataset includes both satellites and orbital debris, with approximately 40% representing satellites and 60% representing debris.

We conduct an exploratory data analysis to illustrate the main differences between the orbital element values of the satellites and those belonging to debris. We show histograms of all the orbital elements for both satellite and debris TLEs in Fig. 2. The columns of the histogram have been normalized so that they add to one.

We see that several of the TLE value frequencies show ranges where debris objects are significantly more common than for satellite objects, and vice versa. For example, we see that for angle of periapsis ω there is a peak in the frequency of debris values between 80 and 101 degrees while value frequency for satellites peaks between 260 and 273. These differences in frequency values are used by the random forest to make predictions. We exploit these differences to craft adversarial samples.

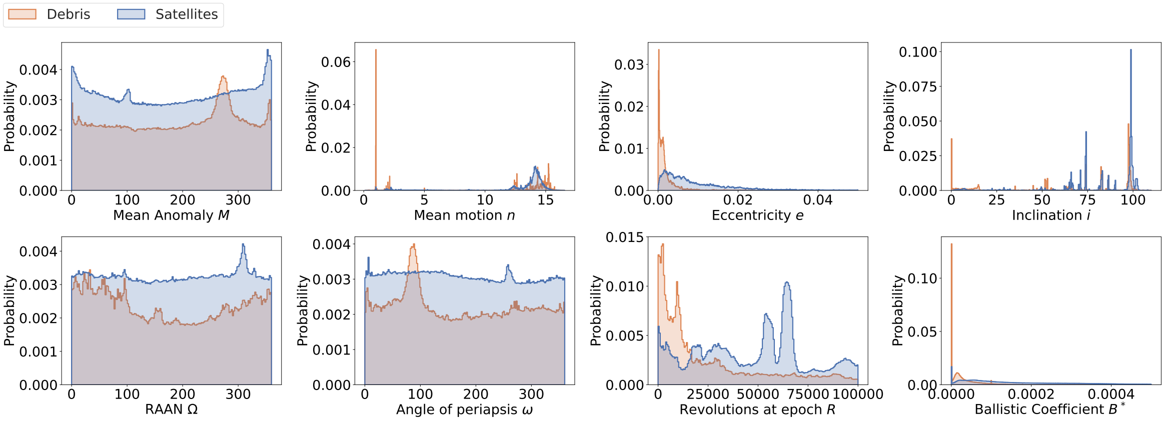


Fig. 2: Histograms of orbital elements.

B. Malicious Feature Design using a Constant Number

Now that we have seen the histogram of TLE values, we are ready to craft the adversarial TLEs. Our first technique is to replace one of the original values of the sensitive satellite's TLE by the value that occurs most often in the debris samples. Specifically, let $\mathbf{x}_r$ be a vector formed with the values of the sensitive satellite's TLE. Let x_r^j be the jth orbital element in $\mathbf{x}_r$. Let $\mathbf{h}_D^j = [h_1^j, h_2^j, \ldots, h_K^j]$ be the histogram for the j^{th} orbital element for TLEs with label D. The histogram $\mathbf{h}_D^j$ contains K equally sized bins. Then, the jth orbital element of the sensitive satellite's adversarial TLE is given by

$$\hat{x}_r^j = v_{max}^j \tag{1}$$

where v_{max}^j is the value that corresponds to the bin with most elements in $\mathbf{h}_D^j$.

As we show in Section VI, this is an effective attack that significantly reduces the ability of the random forest model to find satellites whose label has been modified.

C. Malicious Feature Design using a Uniform Random Distribution

In the previous technique, we only consider the most common value that occurs in debris TLEs without considering if it is also a common value for the satellite TLEs. However, there are some orbital elements with ranges that are more common only for debris. Based on these observations, we design adversarial samples by replacing the original orbital element value of a target satellite TLE with a value chosen uniformly at random within the range of values that is more common for debris TLEs.

Formally, let $[a, b]_j$ be the interval of common values for the jth orbital element for TLEs with label debris D. Then, the attacker maliciously replaces the jth orbital feature x_r^j as follows:

$$\hat{x}_r^j = u \tag{2}$$

where $u \sim \mathcal{U}(a, b)$, and $\mathcal{U}$ denotes the uniform distribution with a and b as the upper and lower limits, respectively.

To find the interval $[a, b]_j$, we use the histogram of the jth orbital element for both the satellite and debris samples. Let $\mathbf{h}_i^j = [h_1^j, h_2^j, \ldots, h_K^j]$ be the histogram for the j^{th} orbital element for label $i \in \{S, D\}$ with K equally sized bins. We first calculate the element-wise difference between the debris and satellite histograms for the jth orbital element:

$$\Delta\mathbf{h}^j = \mathbf{h}_D^j - \mathbf{h}_S^j \tag{3}$$

We then find the bin that corresponds to the value where the difference between debris and satellite TLEs is largest in (3):

$$k_{max}^j = \arg\max_{k^j} \{\Delta h_k^j \in \Delta\mathbf{h}^j : \Delta h_k^j > 0\} \tag{4}$$

We can now set the lower limit a and upper limit b to the values that corresponds to α bins to the left and to the right of the k_{max}^j, respectively. The parameter α can be set by the adversary to control the number of values available to craft the malicious samples.

D. Malicious Feature Design with Gaussian Noise

Replacing the values of the target satellite's TLEs as described in our two previous approaches, risks making large changes to the original value. In this subsection, we describe an attack that adds a small random number drawn from a Gaussian distribution to the target TLEs orbital element. In particular, the jth orbital element of the target TLE can be replaced by a normally-distributed random number as follows:

$$\hat{x}_j^r = r, \tag{5}$$

where $r \sim \mathcal{N}(\mu_D^j, \sigma_D^j)$. The mean μ_D^j and variance σ_D^j denote the mean and variance of the jth orbital element of the debris TLEs.

VI. Experimental Results

In this section, we investigate the performance degradation suffered by the adversary's label-flipping detection random forest under our proposed adversarial TLE crafting approaches in Section V. We implement both the detection random forest model and our proposed attacks using Python. We use

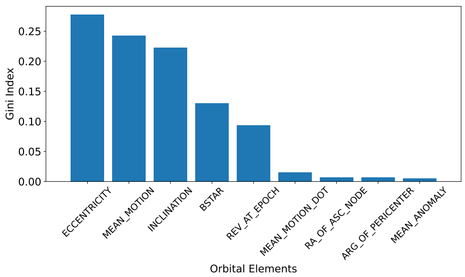


Fig. 3: Importance of orbital elements for RF classifier.

the `RandomForestClassification` class in Scikit-learn [18] to implement the label-flipping attack detection random forest. We ran all of the experiments on a high-performance computing cluster. Our experiments had access to 48GB RAM, and 6 CPUs at 2.60 Ghz.

A. *Label-flipping Detection Training*

We divide the SSA dataset described in Section V-A into a training dataset X_{train} and dataset X_{test} following the steps described in V. Our X_{train} dataset contains close to $700,000$ TLEs representing about $17,000$ unique space objects. The testing dataset X_{test} contains around $140,000$ TLEs from around $4,000$ unique space objects. We use this dataset to train a random forest as described by Pavur et al. in [6]. The random forest is the target machine learning model of our adversarial samples.

B. *Feature Importance Measurements*

To choose the orbital elements from the sensitive satellite TLEs that will be changed, we first find the orbital elements that have the largest influence on the random forest model's decisions. We measure the orbital element influence in terms of the Gini index provided by the Scikit-learn's `RandomForestClassification` class. We show the orbital element importance in Fig. 3. We see that the most influential orbital elements are eccentricity e, mean motion n, and i inclination. We thus choose to attack all combinations of the three orbital elements.

C. *Performance Degradation of the Label-flipping Detection Model*

Next, we measure the performance degradation of the label-flipping detection random forest model under our proposed privacy-preserving TLE adversarial samples. The main idea is to observe the ability of potential adversaries to identify sensitive satellite's TLEs in the SSA data repository.

Specifically, we measure the random forest performance using the receiver operating curve (ROC) when using a testing data set called X_{sim}. The dataset X_{sim} is formed by $N = 2$ sensitive satellite's TLEs that have been modified according to our proposed method, and $M = 398$ randomly chosen debris TLEs that have not been modified. By using a dataset with two concealed sensitive satellites, we can closely replicate a dataset that will be observed by an adversary.

We implement our proposed approach by modifying the top-three most influential orbital elements according to the Gini index under seven different attack combinations: one orbital element at a time, pairs of orbital elements, and all three orbital elements. We calculate the mean ROC curve for $10,000$ different X_{sim}s datasets for each combination of attacks.

Fig. 4 shows the ROC curve of the label-flipping detection model when only one orbital element is modified at a time. Based on the area under the curve (AUC), we observe that in the three cases the most effective approach to conceal the sensitive satellite is the constant attack, then the uniform random attack, and finally the Gaussian random attack. We also see that the constant attack causes the largest drop in performance when modifying the inclination i orbital element. The uniform random attack causes the most performance drop with the eccentricity e orbital element. The Gaussian random attack causes about the same performance drop when modifying the three orbital elements.

We show the ROC curve of the label-flipping detection model when pairs of orbital elements are attacked in Fig. 5. We observe that the performance degradation when compromising pairs of orbital elements is more severe compared to the case where only one element is attacked at a time. The attack with that causes the most performance degradation is when the inclination i and mean motion m orbital elements are modified using the constant attack. We also see that attacking the eccentricity e and inclination and mean motion m allow the Random and Gaussian attacks to cause higher performance degradation compared to the other two pairs of attacks.

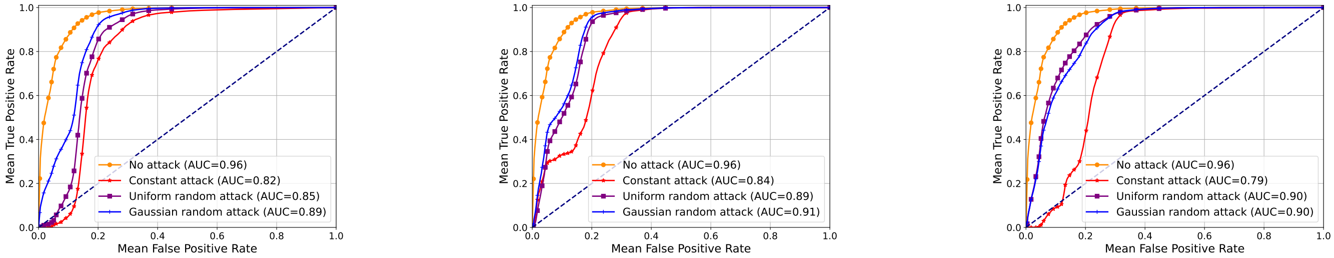
Fig. 6 shows the results when all three of the orbital elements are compromised. We see that our proposed attacks achieves maximum damage when the three orbital elements are compromised. The label-flipping detection model becomes worse than random guessing under the constant attack. The uniform random attack reduces the AUC to 0.72 and the Gaussian random attack reduces the AUC to 0.83. This is expected as it is harder for the label-flipping detection model to correctly identify the compromised TLEs as more orbital elements are compromised.

VII. Conclusions

The growing congestion in Earth's orbit necessitates robust space situational awareness (SSA) systems for safe navigation. However, SSA data can be used by adversaries to launch both kinetic and non-kinetic attacks against sensitive satellites. Existing techniques such as labeling satellites as debris can be easily defeated by an adversary using machine learning. In this paper, we have introduced an adversarial machine learning technique that effectively conceals the true nature of a sensitive satellite in SSA data. By modifying both labels and orbit determination data, our approach prevents adversaries from finding satellites that has been labeled as a debris.

References

[1] U. of Concerned Scientists, "Ucs satellite database," 2005, accessed on June 5th, 2023. [Online]. Available: https://www.ucsusa.org/resources/satellite-database

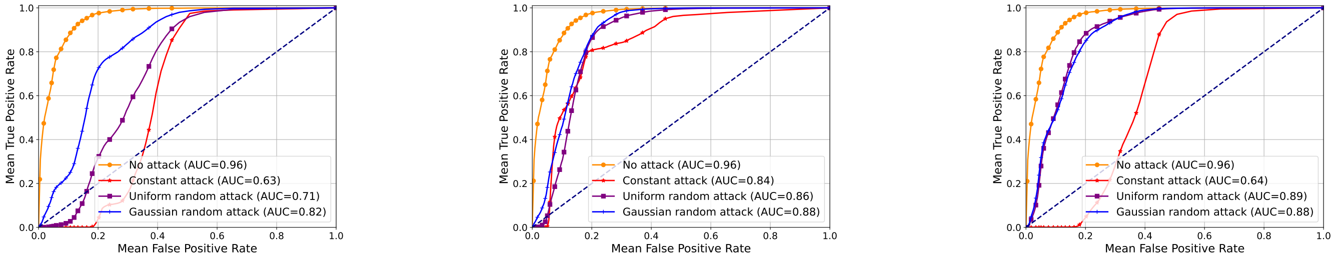


(a) Eccentricity e is attacked.

(b) Mean motion m is attacked.

(c) Inclination i is attacked.

Fig. 4: ROC curve for the label-flipping detection model when only one orbital element is attacked.



(a) Eccentricity e and mean motion m are attacked.

(b) Eccentricity e and inclination i are attacked.

(c) Inclination i and mean motion m are attacked.

Fig. 5: ROC curve for the label-flipping detection model when pairs of orbital elements are attacked.

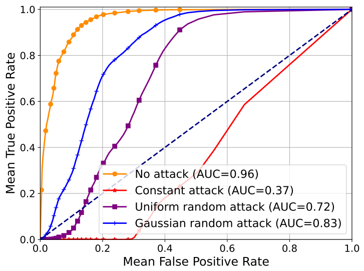


Fig. 6: ROC curve for the label-flipping detection model when all three orbital elements are attacked.

[2] u. National Aeronautics and Space Agency , title = NASA Space Situational Awareness (SSA) Overview .

[3] J. N. Pelton and J. N. Pelton, “Space weapons, the threat of war in space and planetary defense,” *Space 2.0: Revolutionary Advances in the Space Industry*, pp. 115–128, 2019.

[4] W. J. Hennigan. (2020) Russian craft shadowing u.s. spy satellite, space force commander says. [Online]. Available: https://time.com/5779315/russian-spacecraft-spy-satellite-space-force/

[5] N. Das and R. Bhattacharya, “Privacy and utility aware data sharing for space situational awareness from ensemble and unscented kalman filtering perspective,” *arXiv preprint arXiv:1912.03775*, 2019.

[6] J. Pavur and I. Martinovic, “On detecting deception in space situational awareness,” in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 280–291.

[7] B. Wang, S. Li, J. Mu, X. Hao, W. Zhu, and J. Hu, “Research advancements in key technologies for space-based situational awareness,” *Space: Science & Technology*, 2022.

[8] H. Yunpeng, L. Kebo, L. Yan’gang, and C. Lei, “Review on strategies of space-based optical space situational awareness,” *Journal of Systems Engineering and Electronics*, vol. 32, no. 5, pp. 1152–1166, 2021.

[9] P. Bountakas, A. Zarras, A. Lekidis, and C. Xenakis, “Defense strategies for adversarial machine learning: A survey,” *Comput. Sci. Rev.*, vol. 49, p. 100573, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260846357

[10] P. Owezarski, “Investigating adversarial attacks against random forest-based network attack detection systems,” *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259217269

[11] J. Aiken and S. Scott-Hayward, “Investigating adversarial attacks against network intrusion detection systems in sdns,” *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–7, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:209096174

[12] P. Escobal, “Methods of orbit determination.” *Methods of orbit determination*, 1970.

[13] Space Track, “Ssa sharing and orbital data requests,” 2020, accessed on June 5th, 2023. [Online]. Available: https://www.space-track.org/documentation

[14] B. Lal, A. Balakrishnan, B. M. Caldwell, R. S. Buenconsejo, and S. A. Carioscia, “Global trends in space situational awareness (ssa) and space traffic management (stm),” *Science and Technology Policy Institute*, vol. 10, 2018.

[15] S. Erwin, “Space force to lean on private sector for space tracking data,” 2024. [Online]. Available: https://spacenews.com/space-force-to-lean-on-private-sector-for-space-tracking-data/

[16] T. S. Kelso, “Orbit determination,” 1995, accessed on June 8th, 2023. [Online]. Available: https://celestrak.org/columns/v01n06/

[17] S. learn developers, “Random forest,” 2019, accessed on Feb 16th, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForest Classifier.html

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.