This is a repository copy of *Deep temporal semi-supervised one-class classification for GNSS radio frequency interference detection*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/217151/

Version: Published Version

Article:

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Deep temporal semi-supervised one-class classification for GNSS radio frequency interference detection

Viktor Ivanov,[1]* Maurizio Scaramuzza,[2] and Richard. C. Wilson[1]

[1] Department of Computer Science, University of York, York, UK

[2] Skyguide, Swiss Air Navigation Services Ltd, Zurich, Switzerland.
*Corresponding author: Viktor Ivanov; Email: vii500@york.ac.uk

Abstract
We present a deep learning approach for near real-time detection of Global Navigation Satellite System (GNSS) radio frequency interference (RFI) based on a large amount of aircraft data collected onboard from the Global Positioning System (GPS) and Attitude and Heading Reference System (AHRS). Our approach enables detection of GNSS RFI in the absence of total GPS failure, i.e. while the receiver is still able to estimate a position, which means RFI sources with low power or at larger distance can be detected. We demonstrate how deep one-class classification can be used to detect GNSS RFI. Furthermore, thanks to a unique dataset from the Swiss Air Force and Swiss Air-Rescue (Rega), preprocessed by Swiss Air Navigation Services Ltd. (Skyguide), we demonstrate application of deep learning for GNSS RFI detection on real-world large scale aircraft data containing flight recordings impacted by real jamming. The approach we present is highly general and can be used as a foundation for solving various automated decision-making problems based on different types of Communications, Navigation and Surveillance (CNS) and Air Traffic Management (ATM) streaming data. The experimental results indicate that our system successfully detects GNSS RFI with $83 \cdot 5\%$ accuracy. Extensive empirical studies demonstrate that the proposed method outperforms strong machine learning and rule-based baselines.

1. Introduction

Radio frequency interference (RFI) is a major challenge in aviation when using the Global Navigation Satellite System (GNSS) for navigation and surveillance purposes (Scaramuzza et al., 2014, 2015, 2016, 2017, 2019; Truffer et al., 2017; Ala'Darabseh and Tedongmo, 2019; Jonáš and Vitan, 2019; Morales Ferre et al., 2019; Liu et al., 2020a, 2020b, 2021, 2022; Lukeš et al., 2020; Eurocontrol, 2021; Swinney and Woods, 2021; Mehr and Dovis, 2022; Ebrahimi Mehr and Dovis, 2023). Aircraft flying under instrument flight rules (IFR) increasingly depend on GNSS as a main navigational aid. GNSS is becoming crucial for approach and departure operations as well as for the flight's en-route phase. Near the ground, aircraft rely on GNSS signals that are more likely to be jammed, whether intentionally or unintentionally. The low power of GNSS signals at the receiver antenna makes them easily degraded in the presence of even low-power or distant jammers. GNSS-based safety-critical applications exposed to RFI may suffer unacceptable performance degradation. Therefore, it is crucial to develop advanced capabilities for continuous quantitative risk assessment of GNSS RFI over large regions, even for cases without total loss of Global Positioning System (GPS) signals.

The availability of large amounts of high-quality Communications, Navigation and Surveillance (CNS) and Air Traffic Management (ATM) streaming data processed and stored by air navigation service providers (ANSPs) or airspace operators presents a unique opportunity for building innovative

deep learning systems for automated decision making. These systems may be able to autonomously extract valuable insights from current CNS, ATM and airspace operator databases, and they will make use of that insights to offer statistically likely solutions to issues that are challenging or impossible to resolve using conventional engineering techniques.

In real-world aviation operations, there are a huge number of systems producing large volumes of GNSS-related measurements continuously in time. To monitor GNSS working conditions in near real-time, it is vital to be able to efficiently detect abnormalities in time series data so that potential risks related to GNSS RFI can be mitigated on time.

A high-performing GNSS time series anomaly detection model should be able to generalise well to unknown anomalies while learning complex nonlinear temporal patterns of the aircraft measurements' expected characteristics. GNSS time series have complex nonlinear temporal dependencies on other aircraft onboard signals. Additionally, GNSS RFI anomalies are highly infrequent and manually identifying, and annotating these patterns is extremely labour-intensive. Hence, GNSS time series anomaly detection is typically phrased as an unsupervised learning task. However, in reality, we might also have a limited number of observations that have been annotated as normal or abnormal, in addition to a huge set of unlabelled examples. Conventional unsupervised anomaly detection methods are not able to take effective advantage of such labelled data, which is a main limitation for achieving high detection catch rate and low false positive rate.

In our work, we concentrate on the challenge of GNSS RFI detection. We built a deep learning method for near real-time detection of GNSS RFI from large amounts of aircraft onboard data collected from GPS and AHRS. Our approach initially learns a deep anomaly detection model from historical aircraft onboard data, learning from known anomalies, and then applies this model to detect GNSS RFI in new, unseen flight recordings. Our method is highly general and can be used as a foundation for solving different types of anomaly detection tasks based on various kinds of CNS and ATM streaming data.

The key technical insight of our work is to phrase the problem of inferring GNSS RFI as semi-supervised temporal one-class classification in deep learning. We have demonstrated how deep one-class classification can be used to detect GNSS RFI. Our formulation enables us to use powerful methods to learn from historical flight recordings data without manual feature engineering and to perform detection of GNSS RFI.

Inspired by the Temporal Hierarchical One-Class Network (Shen et al., 2020) and Deep Semi-Supervised Anomaly Detection (Ruff et al., 2019), we propose Deep Temporal Semi-Supervised One-Class Classification. In contrast to classic anomaly detection methods that often fail in high-dimensional datasets and typically require substantial amounts of manual feature engineering, our deep learning approach presents a way to learn informative temporal features automatically from raw aircraft measurements data, with outstanding success over classical machine learning and rule-based methods. Our semi-supervised one-class loss function enables the model to learn from known annotated anomalies, which is superior to classic unsupervised anomaly detection approaches that use only normal data.

A standard assumption for anomaly detection models is that clean training data are available that represent normal patterns accurately (Qiu et al., 2022). In practice, this assumption is often wrong, i.e. datasets are often contaminated, i.e. contain anomalies among the normal samples. For instance, a large aircraft onboard streaming dataset may already contain RFI anomalies. Simply training an unsupervised anomaly detection model on such contaminated data will likely lead to poor detection accuracy, i.e. low catch rate and high false positive rate. In our work, we address the data contamination problem by leveraging GNSS RFI annotations from an RFI detection method developed by Scaramuzza et al. (2014, 2015) that is based on carrier to noise (C/No) ratio and aerial aircraft attitude measurements.

To illustrate our approach, we built a scalable prediction engine for detecting GNSS RFI in Switzerland based on a unique dataset collected during a project called Helicopter Recording Random Flights (HRRF) executed by Skyguide, Swiss Air Force and Swiss Air-Rescue (Rega) (Scaramuzza et al., 2014, 2015, 2016, 2017, 2019; Truffer et al., 2017), where data from around thirty helicopters operated by the Swiss Air Force and Rega were recorded. For all flights, data from the Flight Management System

(FMS), Global Positioning System (GPS) and Attitude and Heading Reference System (AHRS) were logged over a number of years and in normal operating circumstances. Large portions of Switzerland were covered. Low flight altitudes are a recurring feature of all of these helicopter operations. As a result, it is anticipated that their likelihood of being impacted by radio frequency interference is greater compared with aircraft operating at higher altitudes. GPS carrier to noise measurements combined with other onboard aircraft signals from FMS and AHRS enable an effective statistical learning for identification of RFI exposures. While laboratory studies are essential for comprehending the effects of jamming on GPS receivers, they frequently do not reflect the real world (Truffer et al., 2017). This dataset is distinctive in that it includes a number of flight recordings from actual field trials of military and civilian aircraft engaged in live jamming exercises conducted by Skyguide and the Swiss Air Force (Truffer et al., 2017). Data from GPS, AHRS and FMS that were recorded during the flights clearly represent how the jamming signals affect various GPS receivers on different aircraft types. The flight recordings from the jamming trials are used as real anomaly examples in our machine learning approach, and thus used for testing of our model. Experimentally, our method detects GNSS RFI with $83 \cdot 5\%$ accuracy. Extensive empirical studies demonstrate that the proposed method outperforms strong machine learning and rule-based baselines.

A key limitation of most of the existing approaches for GNSS RFI detection, especially those based on ADS-B, is that, for achieving high detection accuracy, interference has to be large enough to totally disrupt the reception of GPS signals or remarkably deteriorate the position accuracy. Our approach enables detection of GNSS RFI in the absence of total GPS signal loss, i.e. while the receiver is still able to determine a position, which means RFI sources with low power or at larger distance could be detected.

By formulating the problem of detecting GNSS RFI as deep temporal semi-supervised one-class classification and demonstrating how to perform model training and anomaly detection in onboard aircraft streaming data, our work opens up new opportunities for addressing a wide range of challenges in the context of aviation including anomaly detection, anomaly source localisation and risk assessment.

The main contributions of our work are as follows.

- A deep learning approach for near real-time detection of GNSS RFI based on statistical anomaly detection with deep temporal semi-supervised one-class classification.
- A general framework for anomaly detection in CNS and ATM data.
- A system that is a concrete implementation of our method used for detecting GNSS RFI in Switzerland based on recorded helicopter flights.
- An evaluation on real-world large-scale aircraft data from the Swiss Air Force and Rega, preprocessed by Skyguide, collected onboard and containing flight recordings impacted by a real jammer. The experimental results indicate that our system successfully detects GNSS RFI with $83 \cdot 5\%$ accuracy. Extensive empirical studies demonstrate that the proposed method outperforms strong machine learning and rule-based baselines.

2. Related work

Before introducing our method, we briefly review previous approaches to GNSS RFI detection using machine learning as well as state-of-the-art one-class classification models for anomaly detection.

2.1. GNSS RFI detection

Conventional techniques for interference detection and localisation, e.g. using radio direction finding, are expensive and time-consuming. Recently, there has been an increased interest in creating statistical learning models for GNSS RFI detection. However, there exists very limited previous research on applying machine learning and especially deep learning methods in this domain. The majority of earlier developments are based on conventional statistical analysis. The approach described by Lukeš et al. (2020) analyses probability distribution changes of ADS-B Navigation Accuracy Category - Position

(NACp). A recent method outlined by Liu et al. (2021) provides a machine learning solution for detecting GNSS RFI that is also based on ADS-B data. The authors used out-of-the-box neural networks that learn from ADS-B data and generate a classification outcome indicating if the aircraft has been jammed. Navigation Integrity Category (NIC) is one of the primary raw features used by these models and RFI events from Cypriot airspace are used for positive examples. Due to the limited features in ADS-B messages, a key limitation of most of these approaches is that often interference has to be large enough to totally disrupt the reception of GPS signals or remarkably deteriorate the position accuracy which is not useful for situations where the interference impact is weak. Another line of research (Morales Ferre et al., 2019; Swinney and Woods, 2021; Mehr and Dovis, 2022; Ebrahimi Mehr and Dovis, 2023) focuses on approaches to interference detection based on convolutional neural networks that learn from visual time-frequency representation of the received GNSS signal. Many of these approaches are based on artificially generated datasets that are not well representative for real-world flight measurements' dynamics. Last but not least, recorded aircraft measurements impacted by real jamming are hardly available which makes it difficult for applying conventional supervised machine learning methods and performing robust performance evaluation. The works of Scaramuzza et al. (2014, 2015, 2016, 2017, 2019) address the GNSS RFI detection problem through a non-machine learning approach and it is based on the same dataset used in the current work. The method is based on conventional signal processing techniques and enables detection of potential radio frequency interference based on C/No and aerial vehicle attitude measurements. C/No attenuation due to the antenna pattern and antenna environment is taken into account. Interference affecting the C/No signals by only a few decibels could be detected with this model.

Our work presents a well-performing deep learning GNSS RFI detection method developed and tested based on large-scale real-world aircraft measurements data containing flight recordings impacted by real jamming which enables a rigorous assessment of the whole solution. Our proposed solution outperforms the method described by Scaramuzza et al. (2014, 2015, 2016, 2017, 2019). A key advantage of our method is that it can use a much larger input context, i.e. in addition to C/No and aerial vehicle attitude measurements, it can use other input parameters such as heading, velocity, etc. Furthermore, our method does not have to derive an antenna diagram with over 100 h of data and then go on to search for jamming events, instead, it detects a jamming event directly.

2.2. Tree-based supervised and unsupervised learning

Recently, the group of machine learning algorithms based on decision trees has gained solid traction across academia and especially industry. Light GBM (Ke et al., 2017), XGBoost (Chen and Guestrin, 2016) and Random Forest (Breiman, 2001) are proven to be some of the most effective and efficient classification methods available today when dealing with labelled tabular data. Isolation Forest (Liu et al., 2008) is usually one of the go to options when we deal with unlabelled tabular data and aim to perform anomaly detection.

The main limitation of these methods is that they all require manual feature engineering. Our solution is superior, in that sense, since it provides automatic feature learning capability from raw data and thus it is not dependent on manual feature engineering.

2.3. Kernel-based one-class classification

One-class classification (Moya et al., 1993; Schölkopf et al., 2001; Tax, 2002; Tax and Duin, 2004) is a discriminative anomaly detection approach where the key technical assumption is that most of the data are normal (genuine) and a model can be trained to learn normal behaviour. When a given sample cannot be adequately explained by the model, the observation is considered abnormal. One-class classification methods directly learn a decision boundary and avoid full density estimation as an intermediate step to anomaly detection in contrast to generative anomaly detection approaches. The most common one-class classification methods are kernel-based OC-SVM (One-Class Support Vector Machine) (Schölkopf

et al., 2001) that learns a hyperplane to discriminate genuine observations from abnormal ones, and SVDD (Support Vector Data Descriptor) (Tax, 2002; Tax and Duin, 2004) that learns a hypersphere to enclose the normal data. Both of these models are based on the so-called kernel trick (Schölkopf et al., 2002), which consists of projecting model input to a higher dimensional kernel space for better discrimination. Given a set of N observations $x_1, \ldots, x_N$, where the majority are genuine and some are abnormal, the support vector data descriptor objective is to learn an as small as possible hypersphere with radius R and centre $\mathbf{c}$ to enclose the normal samples. The following optimisation problem definition is a direct realisation of that concept:

$$\min_{\mathbf{c}, R, \xi} R^2 + \frac{1}{\nu N} \sum_{i=1}^{N} \xi_i \tag{2.1}$$
$$\text{such that } \|\phi_k(\mathbf{x}_i) - \mathbf{c}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \; \forall_i \in 1, \ldots, N,$$

where $\phi(\cdot)$ represents a kernel feature function, ξ_i are slack variables enabling a soft decision boundary, and ν is controlling the trade-off between the volume of the sphere and the slack variables.

Even though these methods have been successfully applied in many applications (Chen et al., 2001; Liu et al., 2013; Zhao et al., 2013), they are highly dependent on manual feature engineering and thus limited to data settings where normal patterns can be easily learned. Our method enables automatic learning of representations and thus eliminates the need for manual feature engineering.

2.4. Deep one-class classification

Recently, there has been an increased interest in integrating deep learning (LeCun et al., 2015) into traditional one-class classification approaches. A deep support vector descriptor (Ruff et al., 2018; Ruff, 2021) improves the classic support vector descriptor method by replacing the kernel feature function $\phi(\cdot)$ with a trainable deep neural network. Similarly to Equation (2.1), the optimisation problem is defined as

$$\min_{R, \mathcal{W}} R^2 + \frac{1}{\nu N} \sum_{i=1}^{N} \max\{0, \|NN(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|^2 - R^2\} + \lambda \Omega(\mathcal{W}). \tag{2.2}$$

Here, $NN(\cdot; \mathcal{W})$ represents a neural net with L hidden layers and its corresponding parameters $\mathcal{W} = \{\mathbf{W}^1, \ldots, \mathbf{W}^L\}$, where $\Omega(\mathcal{W})$ is a regularisation function like ℓ_2 regularisation. The entire model is trained end-to-end.

The main limitation of this method is that it is not able to learn from labelled examples and thus, in situations where such examples are available, this method will fall short against systems capable to leverage annotated data. This is one of the key advantages of our solution compared with deep one-class classification.

2.5. Deep semi-supervised one-class classification

Deep semi-supervised one-class classification (Ruff et al., 2019, 2021) represents an enhancement of the deep one-class classification method where the key technical assumption is that there are m labelled examples $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ available together with the n unlabelled examples $x_1, \ldots, x_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^D$ and $\mathcal{Y} = \{-1, +1\}$. Here, $\tilde{y} = -1$ means anomalous examples and $\tilde{y} = +1$ genuine or unlabelled examples. In Deep Semi-Supervised Anomaly Detection (Deep SAD), the objective is formulated in the following way (Ruff et al., 2019, 2021):

$$\min_{\mathcal{W}} \frac{1}{n+m} \sum_{i=1}^{n} \|\phi(x_i, \mathcal{W}) - \mathbf{c}\|^2 + \frac{\eta}{n+m} \sum_{j=1}^{m} (\|\phi(\tilde{x}_j; \mathcal{W}) - \mathbf{c}\|^2)^{\tilde{y}_j} + \frac{\lambda}{2} \sum_{\ell=1}^{L} \|\mathbf{W}^\ell\|_F^2. \tag{2.3}$$

In that particular case, we use the same objective function as in the deep support vector descriptor method (Ruff et al., 2018) for the unlabelled samples, i.e. we recover the deep support vector descriptor as a concrete case when there are no labelled examples, i.e. ($m = 0$). The parameter $\eta > 0$ controls the balance between the labelled and the unlabelled term, where $\eta > 1$ assigns more weight on the labelled data and $\eta < 1$ assigns more weight on the unlabelled data.

This approach enables learning from labelled examples; however, its limitation in the context of GNSS RFI detection is that it is designed for fixed dimensional input while GNSS time series data are dynamic. The superiority of our proposed method compared with deep semi-supervised one-class classification is exactly in our ability to operate with time series data effectively.

2.6. Deep temporal one-class classification

Overall, traditional and deep one-class classification methods are intended for fixed dimensional input. How to expand these techniques for time series anomaly detection is still an open research challenge. Running a sliding window on the time series data in a basic manner would generate a fixed-dimensional feature vector containing the required context and serving as an input to a one-class classification model. This method, however, falls short of fully capturing the underlying temporal dependencies. Numerous time series anomaly detection models based on recurrent networks have recently been proposed as solutions to this issue. An LSTM (Long Short-Term Memory) encoder-decoder (Malhotra et al., 2016) defines an anomaly score derived from the time series reconstruction error. The problem with that approach is that the model experiences error accumulation when decoding lengthy sequences. Deep generative models have also been proposed, including the recurrent variational autoencoder (Su et al., 2019) and different types of generative adversarial networks (GANs) (Li et al., 2019; Zhou et al., 2019); however, designing an effective discriminator and generator is very challenging in practice (Kodali et al., 2017). Temporal Hierarchical One-Class (THOC) (Shen et al., 2020) leverages a dilated recurrent neural network (Chang et al., 2017) to learn temporal features from the time series. THOC uses features from all internal network layers, and multiple hyperspheres at each layer represent typical data patterns. The model uses a multiscale support vector data description one-class loss function determined by the difference between hypersphere centres and the final features.

The drawback of THOC is that no label information is leveraged during training – it only takes into account the unsupervised learning setting. Our approach represents an advancement over THOC in a way that it enables both learning from time series data as well as learning from labelled examples.

2.7. Anomaly detection with contaminated data

A basic assumption in most of the conventional unsupervised anomaly detection models is that the number of anomalies in the training dataset is minimal and the model will use inlier priority (Wang et al., 2019). Quite often, no special treatment is performed to deal with data contamination which is the situation where our dataset contains hidden anomalies among the normal samples. A technique that eliminates potential anomalies from the training data is described by Yoon et al. (2021), where an ensemble of one-class classifiers is used for anomalies removal. There are variations where this method is combined with autoencoders (Xia et al., 2015; Beggel et al., 2020) or with latent SVDD (Görnitz et al., 2014). These techniques do not, however, take advantage of the outlier exposure (Hendrycks et al., 2018), insight which imposes a limitation considering that anomalies could be a highly important training signal for improving overall model accuracy. Zhou and Paffenroth (2017) employed an autoencoder to recognise anomalous data points, but their method necessitates training a fresh model every time to spot anomalies, which is impractical in the majority of near real-time situations. Hendrycks et al. (2018) suggest to synthesise artificial anomalies by getting samples from a related domain. Qiu et al. (2022) offer a general method to enhance the training mechanism of many deep learning based anomaly detection models. Their work aims to take advantage of unlabelled anomalies in the training data while outlier exposure assumes labelled anomalies.
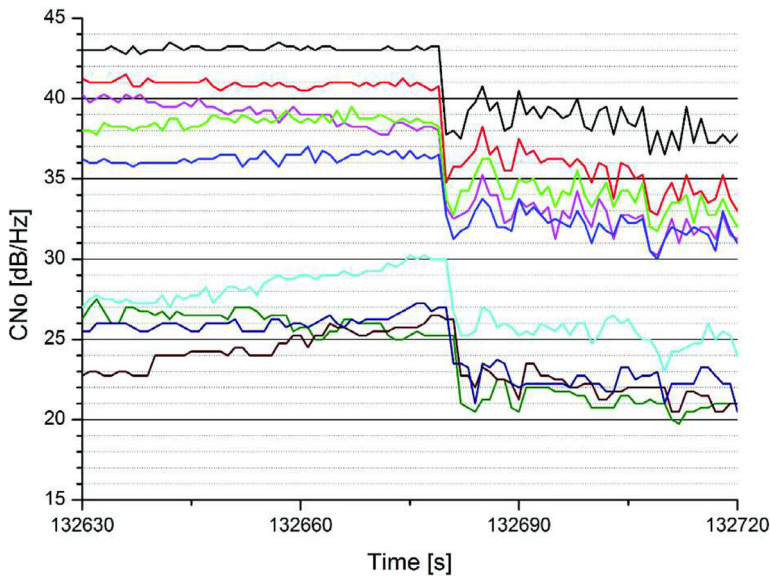
Figure 1. *RFI on a stationary GNSS receiver.*

In our work, we employ a domain-specific method for dealing with contaminated datasets that is based on the work of Scaramuzza et al. (2014, 2015).

3. Deep temporal semi-supervised one-class classification for GNSS RFI detection

Next we introduce our anomaly detection method for GNSS RFI detection. The core idea is to phrase the problem of inferring GNSS RFI as semi-supervised temporal one-class classification in deep learning. We demonstrate how the statistical learning framing is done, what deep learning objective we use and how we deal with data contamination. The anomaly detection framework presented here is highly general and can be easily instantiated to many types of challenges in CNS and ATM data.

3.1. The GNSS RFI detection problem

Carrier to noise ratios of all tracked satellites are the primary factors used for GNSS RFI detection in our work. Those are augmented with additional measurements such as heading, roll and pitch, as well as ground speed and true air speed measurement.

Interference on a stationary GPS receiver. Analysing statistical properties of C/No values might reveal a potential radio frequency interference (Scaramuzza et al., 2014). Since RFI affects the entire GPS receiving antenna equally, all C/No ratios should theoretically decrease by a near constant level when exposed to interference (Scaramuzza et al., 2014). This hypothesis is empirically validated by Scaramuzza et al. (2014) where, in a laboratory environment, a real interference is assessed on a stationary GPS receiver, which is illustrated in Figure 1. When interference started, all C/No measurements dropped by five dB.

Interference detection on a live GNSS receiver. We could address in an analogous way the detection of interference for a GNSS receiver running in a live setting; however, there are some challenges of which to be aware. In contrast to the laboratory example in Figure 1, a flying aircraft that approaches an interference source would typically be impacted gradually and the C/No ratio would smoothly decrease (Scaramuzza et al., 2014). Second, the C/No is impacted by the satellite positions relative to the antenna (Scaramuzza et al., 2014). Figure 2 illustrates a real scenario assessed by Scaramuzza et al. (2014), where changes in the aircraft's attitude are shown to have an impact on some C/No values.
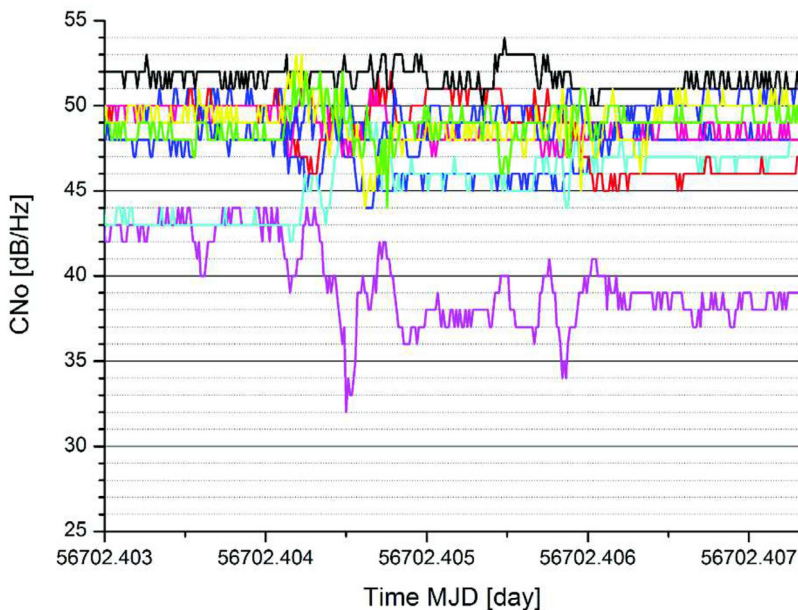
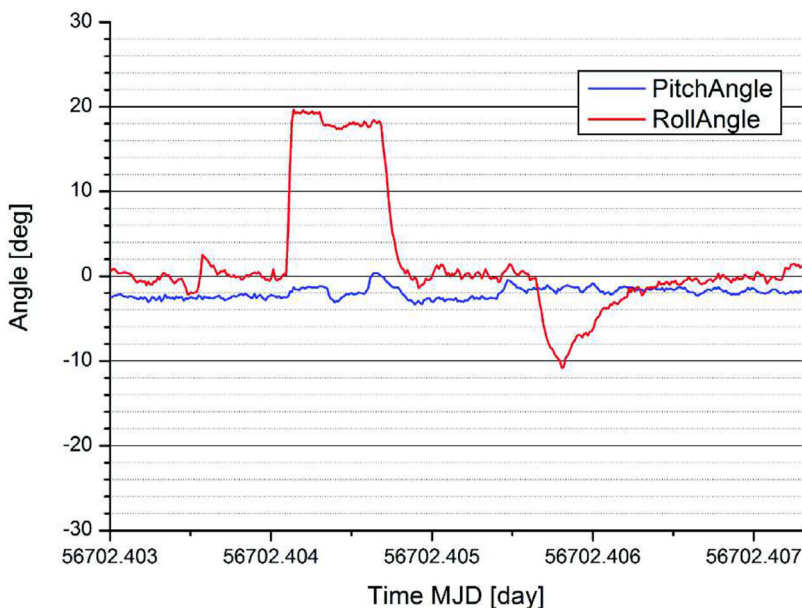**Figure 2.** *Helicopter manoeuvre affecting a GNSS satellite's C/No.*

Figure 3. *Roll and pitch angles affecting the carrier to noise ratio shown in Figure 2.*

Considering that some C/No ratios are decreasing and some are increasing due to the interference, the drop at 56702.404 time cannot be attributed to radio frequency interference (Scaramuzza et al., 2014). According to the measured roll and pitch angles, a manoeuvre has been performed (Figure 3).

Overall, the following factors, in addition to radio frequency interference, could affect the carrier to noise measurements (Scaramuzza et al., 2014):

1. declining signal due to multipath from surroundings outside the airframe;
2. declining signal due to airframe multipath;

3. signal fading due to air frame;
4. antenna gain pattern;
5. deviation in fading of cabling and amplifiers gain;
6. troposphere;
7. ionosphere.

By restricting the measurements to areas where the helicopter travels at a minimum ground speed, factor 1 could be solved (Scaramuzza et al., 2014). That way, we can prevent scenarios where the geometry between GPS antenna, reflector and satellite stays static for a prolonged time frame, and thus signal decline is minor (Scaramuzza et al., 2014).

Factors 2, 3 and 4 must always be taken into consideration. These factors have one thing in common: the antenna's local coordinate system and the satellite's position determine how much the signal is reduced (Scaramuzza et al., 2014).

Factor 5 could be ignored considering that these amplifications and losses are applicable to all tracked satellites (Scaramuzza et al., 2014).

Similar to factors 2 to 4, factors 6 and 7 are always present but are unaffected by the attitude of the aircraft (Scaramuzza et al., 2014). An estimate of the signal attenuation caused by the troposphere and ionosphere is pertinent in this situation. The low signal attenuation caused by the atmosphere is one of the primary factors in the choice of the L-band for GNSS applications (Scaramuzza et al., 2014). For signal paths entirely within the troposphere, the tropospheric attenuation is much lower than 1 dB (Essentials, 2012; Recommendation, 2013). For signal paths from space to Earth, it is even lower. According to Christie et al. (1996), the ionospheric attenuation is thought to be insignificant. Special attention should be made in case of ionospheric scintillation considering that in such situations, attenuation can reach levels above twenty dB (Series, 2015, 2016). The geomagnetic equator is where ionospheric scintillation is at its highest and mid-latitude regions (Series, 2015) are where it is at its lowest. Since our experimental work is based on datasets covering Switzerland, regardless of Switzerland's location at middle latitudes, we should not record interference detection when ionospheric scintillation is present (Scaramuzza et al., 2014).

The factors that matter most for interference detection are factors 2, 3 and 4 (Scaramuzza et al., 2014).

3.2. *Statistical learning problem definition*

Let $\mathcal{D} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be a set of time series representing recorded flights. Each $\mathbf{X}_s \in \mathcal{D}$ represents a flight of length T_s epochs, and the flight measurements vector at time t is $x_{t,s} \in X \subseteq \mathbb{R}^D$. The GNSS RFI detection problem is to estimate if the measurements $x_{t,s}$ are impacted by radio frequency interference, based on the sequence of observations $x_{1:t,s}$ that has been observed so far.

We phrase the GNSS RFI detection problem as anomaly detection in time series data, and thus we generalise the GNSS RFI detection task as determining if the measurements $x_{t,s}$ are anomalous.

Our objective is to develop an anomaly detection model that reduces false positives and missed true anomalies. Conceptually, attaining a low (or even zero) false positive rate is straightforward: given a large number of instances of normal data, we can simply define a decision boundary that encompasses all of the instances, e.g. a large hypersphere that contains all data observations. Of course, maintaining a low miss rate while not drawing this boundary too broadly is the challenge here. Therefore, the main technical problem consists in reducing miss rate for a predefined false positive rate when we have zero or just a few anomalous samples.

Let $Y \in \{\pm 1\}$ be the target variable considering that $Y = +1$ means genuine flight measurement observations and $Y = -1$ means abnormal flight measurement observations. Additionally, we define $\ell : \mathbb{R} \times \{\pm 1\} \rightarrow \mathbb{R}$ as a binary classification loss and $f : X \rightarrow \mathbb{R}$ to be a real-valued scoring function. Having an unlabelled dataset of flight measurement vectors $x_1, \ldots, x_n \in X$ and, if available, an annotated dataset of flight measurement vectors $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)$, and following the empirical risk minimisation concept, we formulate our one-class classification learning objective in Equation (3.1).

The assumption that the n unlabelled training samples are non-anomalous is incorporated into the implicit labelling $y = 1$ in the first term.

$$\min_{f} \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i), +1) + \frac{1}{m} \sum_{j=1}^{m} \ell(f(\tilde{\mathbf{x}}_j), \tilde{y}_j) + \mathcal{R}. \tag{3.1}$$

In the unsupervised setting, the second term is a zero sum. To signify and encompass regularisation, we add the term $\mathcal{R}$ as an additional component in our loss function.

3.3. Deep temporal semi-supervised one-class classification

Next, we present our approach for deep temporal semi-supervised anomaly detection. Consider that we have m labelled examples $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ together with the n unlabelled examples $x_1, \ldots, x_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^D$ and $\mathcal{Y} = \{-1, +1\}$, considering that $\tilde{y} = +1$ means genuine observations and $\tilde{y} = -1$ known abnormalities. Our model architecture is built on top of the THOC Network (Schölkopf et al., 2001). Similar to THOC, our model architecture uses a dilated recurrent neural network (Chang et al., 2017) to effectively learn temporal representations from time series data. Multiple hyperspheres at each network layer are used to learn representations of normal patterns which enables the model to learn complex relationships in GNSS time series data and is much more effective than conventional one-class classification approaches based on just a single hypersphere (Shen et al., 2020). We designed a semi-supervised one-class loss function to enable the model to learn from known annotated anomalies and deal with contaminated data which is superior to classic unsupervised anomaly detection approaches that use only normal data. We define our deep temporal semi-supervised one-class classification learning objective as follows:

$$\ell = \frac{1}{NK^L} \sum_{s=1}^{N} \frac{1}{T_s} \sum_{t=1}^{T_s} \sum_{j=1}^{K^L} R_{t,j,s}^L d(\mathbf{f}_{t,j}^L, \mathbf{c}_j^L)^{y_{t,s}} + \lambda \Omega(\mathcal{W}). \tag{3.2}$$

It is based on the difference between hyperspheres' centres $\{\mathbf{c}_1^L, \ldots, \mathbf{c}_{K^L}^L\}$ and final calculated features $\{\bar{\mathbf{f}}_{t,j}^L\}$ at the last layer of the deep network, where t runs over the time points and T_s is number of epochs for a given time series, j runs over the hyperspheres at a given layer, and L is the number of network layers (3 in our reference implementation). Given a concrete scale $l \in \{1, \ldots, L\}$, we have a specific layer of K^l hyperspheres where K^l is the number of hyperspheres at layer l. Cosine distance is used for $d(\bar{\mathbf{f}}_{t,j}^L, \mathbf{c}_k^L)$. Here, $\mathcal{W}$ are the trainable network weights and $\Omega(\mathcal{W})$ is ℓ_2 regularisation. The subscript s is added for samples and N is the number of samples. While conventional one-class classification has just a single hypersphere, our model incorporates multiple network layers each with multiple hyperspheres' centres. $R_{t,j,s}^L$ represents how much the measurement $\mathbf{x}_{t,s}$ is similar to centre $\mathbf{c}_j^L$ and, in our case, it is based on a cosine similarity function. The term is used as an importance weighting of $d(\bar{\mathbf{f}}_{t,j}^L, \mathbf{c}_k^L)$. It is computed in a recursive manner for each layer and for a given layer l, it is calculated by a softmax over all centres in the given layer. For simplicity, we remove the subscript s:

$$R_{t,j}^l = \frac{\exp(\tilde{R}_{t,j}^l)}{\sum_{i=1}^{K^l} \exp(\tilde{R}_{t,i}^l)}, \quad \text{where } \tilde{R}_{t,j}^l = \begin{cases} P_{t,i,j}^l & \text{if } l = 1, \\ \sum_{i=1}^{K^{l-1}} P_{t,i,j}^l R_{t,i}^{l-1} & \text{if } 1 < l \le L. \end{cases} \tag{3.3}$$

Here the term $P_{t,i,j}^l$ is defined as

$$P_{t,i,j}^l = \frac{\exp(\cos(\bar{\mathbf{f}}_{t,i}^{l-1}, \mathbf{c}_j^l))}{\sum_{k=1}^{K^l} \exp(\cos(\bar{\mathbf{f}}_{t,i}^{l-1}, \mathbf{c}_k^l))}. \tag{3.4}$$

To introduce semi-supervision, we penalise the inverse of the distances by $d(\bar{\mathbf{f}}_{t,j}^{L}, \mathbf{c}_{k}^{L})^{y_{t,s}}$ for the labelled anomalies ($\tilde{y} = -1$), requiring anomalies to be mapped further from the centre. This is consistent with the widely held belief that anomalies do not cluster (Schölkopf et al., 2002; Steinwart et al., 2005). This component enables the model to learn from just a few labelled anomalies.

The THOC Network is recovered as a concrete case when there are no available annotated training samples ($m = 0$) by using the same objective for the unlabelled data ($\tilde{y} = +1$) in our loss function. We also take into account the fact that the majority of the unlabelled data is normal when doing this.

With a trained deep learning model against our objective, let x_t be the measurement at time t for a new and unseen time series X. We define a quantitative anomaly score that determines how an observation at a given time epoch differs from the regular data patterns represented by the hyperspheres. The score is defined as follows: $f(x_t) = \sum_{j=1}^{K^{L}} R_{t,j,s}^{L} \cdot d(\mathbf{f}_{t,j}^{L}, \mathbf{c}_{j}^{L})$. In combination with a user-defined threshold selected empirically based on target precision and recall or simply based on the value that produces the highest F1 score on a validation set, the score is used for anomaly detection in unseen time series. Thus, considering a threshold δ, we then classify x_t as anomalous in case $f(x_t) > \delta$.

3.4. Addressing data contamination

Now, we present our domain-specific strategy for generating pseudolabels and dealing with data contamination. Considering that our datasets are likely contaminated and contain various types of RFI anomalies, we implemented a strategy to address this problem by leveraging GNSS RFI annotations from the RFI detection method developed by Scaramuzza et al. (2014, 2015) that is based on carrier to noise measurements and aerial aircraft attitude. This approach enables us to generate anomaly pseudolabels ($\tilde{y} = -1$) for particular time steps that our semi-supervised model can use during training.

The core idea behind the method developed by Scaramuzza et al. (2014, 2015) is to assess the C/No values distribution of each tracked GPS satellite (Scaramuzza et al., 2014, 2015) taking into account that the GPS receiving antenna is entirely impacted by the same level of interference, an RFI occurrence would cause the C/No to decrease by a near constant amount at each individual time step (Scaramuzza et al., 2014, 2015). To do so, the method estimates the GPS antenna pattern which affects the GPS C/No depending on the satellite position referred to the GPS antenna. This allows to normalise the C/No and minimise most signal attenuation not related to RFI. Finally, it is possible to determine whether a constant decrease of all C/No is present, indicating a potential RFI, or not. Figure 4 depicts all steps of the method detecting any potential RFI based on Scaramuzza et al. (2014, 2015).

The synopsis in Figure 4 shows the context in which the man-made RFI is detected. It is understood that RFI is not the only electromagnetic type of interference to have an adverse impact on the quality of the GNSS signals. Only non-intentional and intentional man-made RFI is subject of RFI detection, i.e. multipath and natural RFI is out of scope (Scaramuzza et al., 2014, 2015).

4. Implementation

In this section, we provide implementation details around our solution. Figure 5 summarises the high level network architecture. The system operates based on a sliding window approach. We use a long window length of 100 epochs for better performance. The first 100 epochs are used to washout since they have not enough context or, in other words, we assume that they are normal samples. The sliding step is 20 and it is less than the window size which helps training. If some points are detected multiple times, we compute their anomaly scores' mean value as the final score. We use a batch size of 32, learning rate of 0·001 and weight decay of 1×10^{-6}. Our dilated recurrent neural network has three layers and dilations [1, 12, 34]. Respectively, the number of clusters at each layer is as follows: [12, 6, 6]. Hidden layers size is 64 and input size is 37. The model uses an Adam optimiser. For clustering, we use k-means. Data are scaled with a robust scaler. Model hyperparameters are fine-tuned with a random search. The
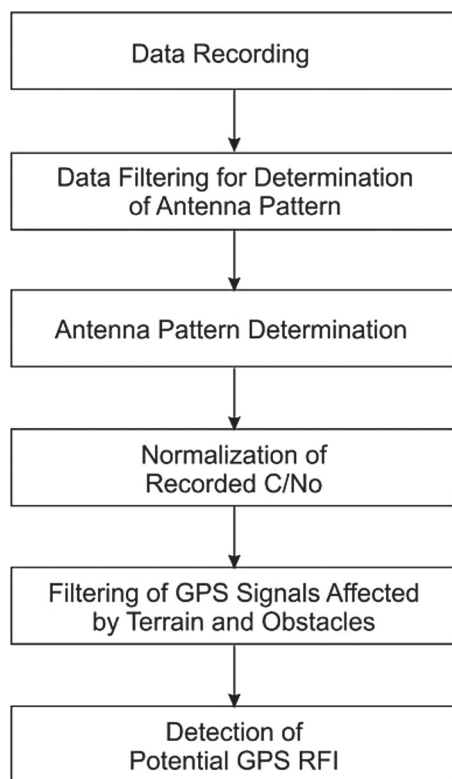
Figure 4. *Flow chart of all required steps to detect potential GPS RFI based on a method developed by Scaramuzza et al. (2014, 2015).*

implementation is done in Python 3.9.7 using PyTorch 1.11.0. The source code is available at https://github.com/vii500/dtssoccgnssrfid.

5. Experimental evaluation

In this section, we describe the evaluation of the proposed deep learning model. Our main objective is to study how effective our approach is for GNSS RFI detection based on a real-world dataset containing onboard measurements of jammed aircraft.

5.1. Dataset

Skyguide has provided the data for this research. There are 54,402 recorded flights in the dataset. The information was gathered by Skyguide as part of the Helicopter Recording Random Flights (HRRF) project, where data recorders were installed on board of three dozens of helicopters operated by Swiss Air Force and Rega (Scaramuzza et al., 2014, 2015, 2016, 2017, 2019; Truffer et al., 2017). For all flights, data from GPS, AHRS and FMS were logged over the course of six years and in normal operating circumstances. Large portions of Switzerland were thus covered. Low flight altitudes are a recurring feature of all of these helicopter operations. As a result, it is anticipated that their likelihood of being exposed to GNSS RFI is higher than that of aircraft flying at higher altitudes. Through the recorded C/No values or position losses, any exposure of this kind can be identified (Scaramuzza et al., 2014, 2015). The additional recorded data enable more effective statistical learning based on richer input features.
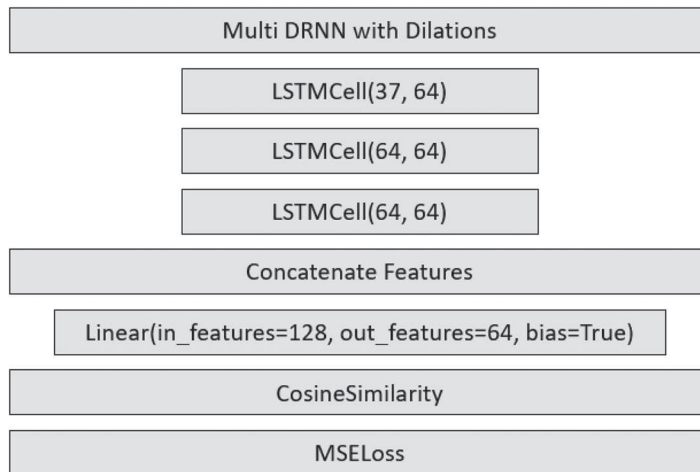
Figure 5. *Deep temporal semi-supervised one-class classification high level network architecture.*

Figure 6. *EC145 of the HEMS operator Rega. The GPS antenna is attached on the top of the fin in front of the strobe light (courtesy Rega).*

The Swiss Air Force operated 18 EC-635 helicopters (Figure 7) and Rega operated 11 AW109SP and 6 EC-145 helicopters (Figure 6), making up the entire fleet of helicopters with recording units. Fixed installations were implemented due to the many years data collection timeline.

Installation. We briefly describe the technical setup from Scaramuzza et al. (2014, 2015). A mini Quick Access Recorder (mQAR), depending on the architecture of the aircraft, is connected to its ARINC bus or RS-232 interface as the technical solution. The mQAR is a lightweight and compact unit. An installed mQAR is shown in Figure 8, as indicated by the red arrow. The mQAR begins recording data automatically as soon as the helicopter is powered on and continues doing so until the power is turned off. Therefore, there is no need for the pilot or ground crew to interact. An SD (Secure Digital) memory card serves as the storage medium and, under normal circumstances, can store several weeks' worth of

Figure 7. *EC635 of the Swiss Air Force. The GPS antenna is mounted analogous to the EC145 (courtesy VBS).*

flight data. Each helicopter base's ground staff were given instructions to download the recorded data and upload it to a shared data storage every two to four weeks.

Recorded data. Overall, we have a large volume of data collected onboard. From EC-145/635, we are leveraging data from the GPS, AHRS and FMS, and from AW109SP, we have FMS and GPS only. EC145/635 GPS data include satellite position, GPS position, vertical and horizontal integrity limits and figure of merits, pseudo range and pseudo range rate, C/No ratios and different status indicators. Only on AW109SP do we have position domain data. AHRS data include roll, heading and pitch measurements. The GPS and AHRS sampling interval is 1 Hz.

Interference detection input. The key measurement used for GNSS interference detection is the carrier to noise ratio of each tracked satellite. Radio frequency interference will impact all carrier to noise ratios. The carrier to noise input (C/No_1–C/No_{32}) is augmented with heading, roll and pitch measurements as well as ground speed, i.e. velocity over ground measurement (no wind effects taken into account) and true air speed measurement, i.e. velocity relative to air. This represents a much richer context compared with the conventional ADS-B setups.

True interference labels. Two flight recordings from actual field trials of military and civilian aircraft engaged in live jamming situations carried out by Skyguide and the Swiss Air Force (Truffer et al., 2017) are included in our dataset. The recorded data help us to understand how real-world jamming affects various GPS receivers on different types of aircraft. The jamming trials are used for model testing, i.e. they are used in our deep learning approach to extract true examples of anomalies. We specifically annotate the epochs within the jamming time range as anomalous and the epochs outside the jamming time range as normal based on the known start and stop times of the jammer. We have two fully annotated time series with actual anomalies, allowing us to run a robust test.

5.2. Evaluation setup

We now provide details on how model evaluation is performed.

Training data. For model training, we use the flight recordings of 17 of the helicopters available in the dataset provided by Skyguide with anomaly pseudolabels derived from a method developed by
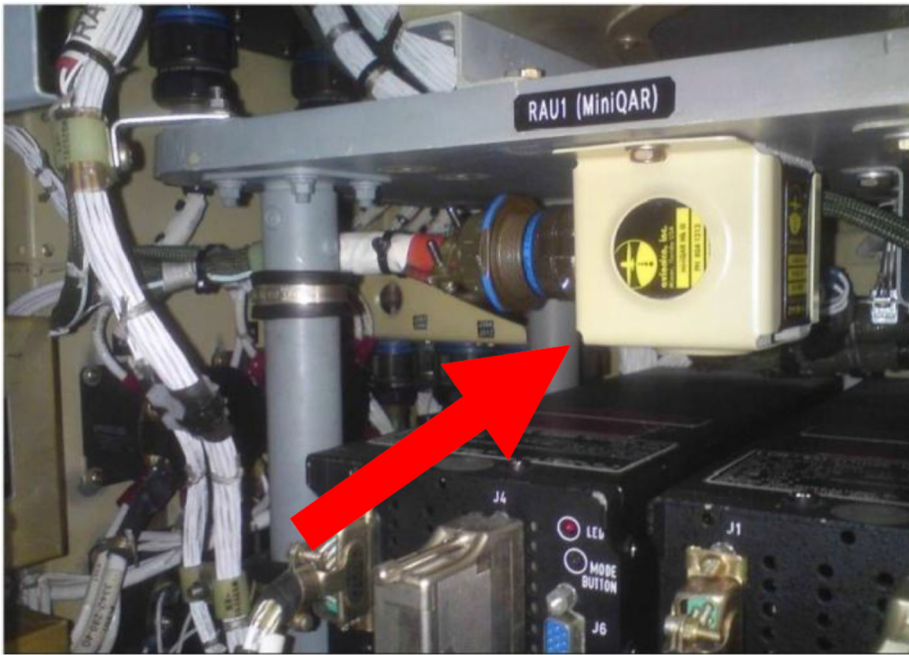
Figure 8. *Installed Avionica mQAR in Swiss Air Force and Rega helicopters (red arrow).*

Table 1. *Known start and stop jammed epochs in jammed flight 1.*

Jammer mode	Jamming start epoch	Jamming end epoch
Noise	1,939	2,355
CW	2,370	2,674
Pulse	3,091	3,526
Sweep	3,691	3,942

Table 2. *Known start and stop jammed epochs in jammed flight 2.*

Jammer mode	Jamming start epoch	Jamming end epoch
Noise	994	1,269
CW	1,294	1,651
Pulse	2,387	2,807
Sweep	2,867	2,927
Sweep	3,047	3,176

Scaramuzza et al. (2014, 2015). We explicitly exclude all flights of the two helicopters participated in the jamming trials to prevent potential overfitting on specific aircraft. We have 43,696 number of flights (time series) in total where the average number of epochs per flight (time series length) is 1,700. We have 0·05% anomalous epochs in the training data.

Test data. For model evaluation, we use the two flights from the jamming trials having anomaly labels based on the known start and stop times of the jammer. Jammed flight 1 has 4,002 epochs of which 2,592 flagged as normal and 1,410 flagged as abnormal. Jammed flight 2 has 3,782 epochs of which 2,536 flagged as normal and 1,246 flagged as anomalous. Tables 1 and 2 present the jamming annotations based on the known start and stop times of the jammer.

***Table 3.** Model evaluation on jammed flight 1.*

Method	Precision	Recall	F1 score	Accuracy
Our method (semi-supervised mode)	$0 \cdot 80$	$0 \cdot 78$	$0 \cdot 79$	$0 \cdot 85$
Our method (unsupervised mode)	$0 \cdot 37$	$1 \cdot 00$	$0 \cdot 54$	$0 \cdot 40$
THOC network	$0 \cdot 37$	$1 \cdot 00$	$0 \cdot 54$	$0 \cdot 40$
Scaramuzza et al. (2014)	$0 \cdot 96$	$0 \cdot 25$	$0 \cdot 39$	$0 \cdot 73$
Isolation forest	$0 \cdot 36$	$0 \cdot 98$	$0 \cdot 53$	$0 \cdot 38$
Light GBM	$0 \cdot 98$	$0 \cdot 18$	$0 \cdot 30$	$0 \cdot 71$
Random forest	$0 \cdot 96$	$0 \cdot 22$	$0 \cdot 36$	$0 \cdot 72$

Baselines for comparison. The following set of anomaly detection algorithms are benchmarked with the suggested model. The first group contains models based on supervised machine learning. These include random forest (Breiman, 2001) and light gradient boosting machine (LGBM) (Ke et al., 2017). The second group contains anomaly detectors for general multivariate tabular data. This group is represented by isolation forest (Liu et al., 2008). The third group contains deep learning anomaly detectors for time series data. This group is represented by THOC Network (Shen et al., 2020). Finally, we compare with the method developed by Scaramuzza et al. (2014).

Performance metrics. We use four standard metrics for evaluating our method against the baselines: precision, recall, F1 score and accuracy:

- **Precision** is calculated as TP/(TP + FP);
- **Recall** calculated as TP/(TP + FN);
- **F1 score** is calculated as $2 \times$ (precision $\times$ recall)/(precision + recall);
- **Accuracy** is calculated as (TP + TN)/(TP + TN + FP + FN),

where TP refers to true positive which is the outcome where the model correctly predicts the anomaly class, TN refers to true negative which is the outcome where the model correctly predicts the normal class, FP false positive which is the outcome where the model incorrectly predicts the anomaly class and FN refers to false negative which is the outcome where the model incorrectly predicts the normal class.

5.3. Hardware

We ran our experiments on a standard desktop workstation with a Intel Core i7-7700 CPU $3 \cdot 60$ GHz processor, 32GB RAM, a solid-state drive storage and running 64-bit Windows 10 Home.

5.4. Evaluation results

Table 3 shows the results on jammed flight 1 and Table 4 shows the results on jammed flight 2. The main comparison metric is F1 score which is commonly used as a standard benchmark metric in imbalanced binary classification since it integrates precision and recall into a single metric, and thus provides a better understanding of model discriminative power. Precision, recall and accuracy are reported for completeness of evaluation. To calculate the performance metrics, we select a predefined threshold for our model. The threshold is selected out of a large range of consecutive options based on the highest F1 score produced by the model.

Isolation Forest performs poorly on the time series, which is not surprising given that it is not well suited to capture the underlying temporal dependencies in the data. Even though it achieves high recall, its precision is very low, and thus overall F1 score and accuracy are both low. This model will produce a high number of false positives and thus is impractical for real usage.

Method	Precision	Recall	F1 score	Accuracy
Our method (semi-supervised mode)	0·72	0·72	0·72	0·82
Our method (unsupervised mode)	0·35	1·00	0·52	0·38
THOC network	0·35	1·00	0·52	0·38
Scaramuzza et al. (2014)	1·0	0·1	0·18	0·70
Isolation forest	0·32	0·55	0·40	0·46
Light GBM	1·00	0·06	0·11	0·69
Random forest	1·00	0·07	0·14	0·70

THOC Network does not perform well in general. Similarly to Isolation Forest, its precision is very low. F1 score and accuracy are both low. This model will also exhibit a high false positive rate that is unacceptable for real application.
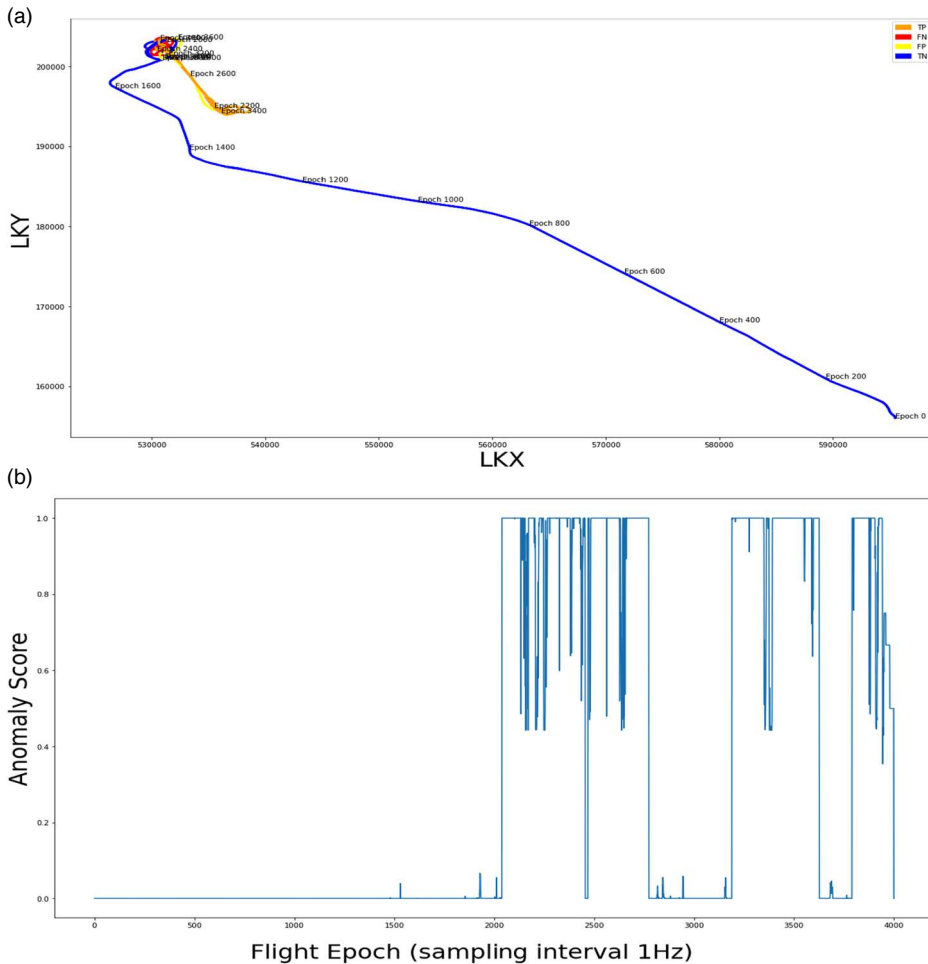

Figure 9. *Model evaluation on jammed flight 1. (a) Flight trajectory and marked true positive (TP), false positive (FP), true negative (TN) and false negative (FN) epochs. (b) Model anomaly scores of all epochs. (c) Values of all C/No. (d) Mean value of all C/No (blue line) and marked TP, FP, TN and FN epochs.*
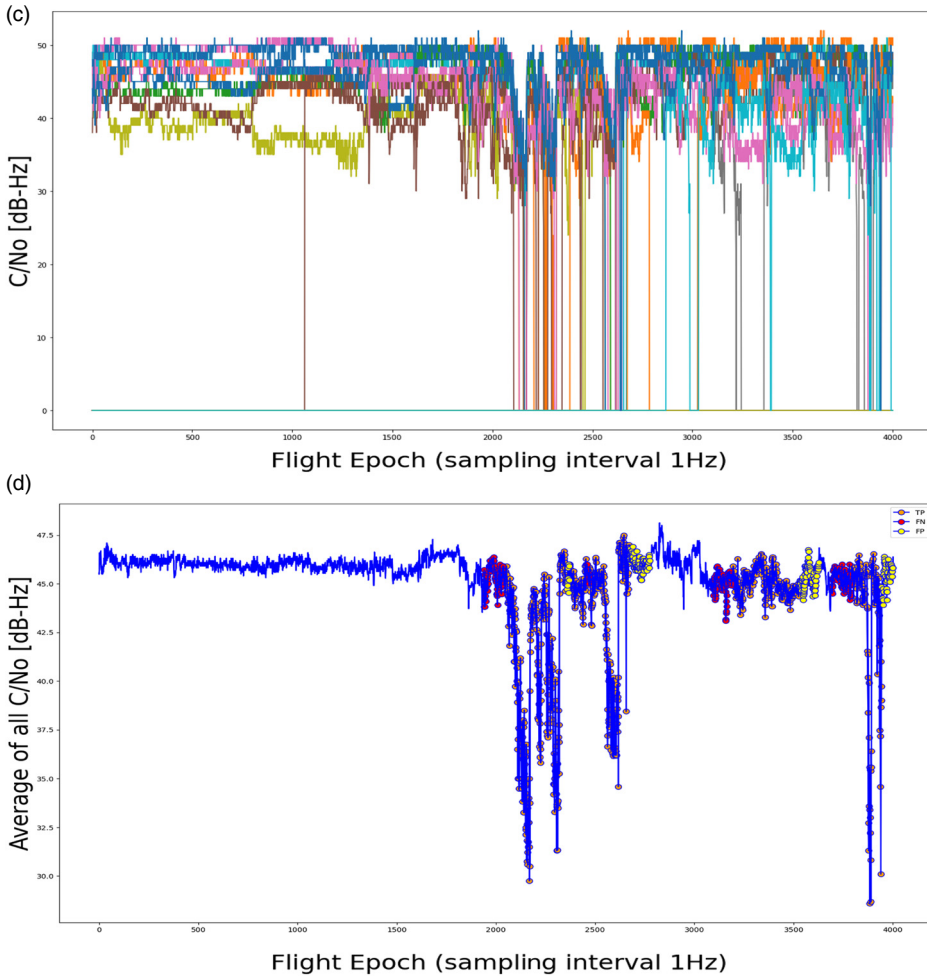
Figure 9. *Continued.*

Random Forest and *Light GBM* perform better considering that they leverage information from annotated examples in the training data. They achieve very high precision, but their recall is very low, and the F1 score and accuracy are both low. These models will miss most of the potential RFIs in real-world application.

The method of Scaramuzza et al. (2014) performs best out of all baseline methods. Similarly to Random Forest and Light GBM, it achieves very high precision but the recall is low. F1 score and accuracy are both low. This model is missing the majority of potential RFIs.

Our proposed model operating in unsupervised learning mode, i.e. when data contamination is not addressed and labelled anomalies are not used, recovers the performance of the THOC network and performs poorly.

Our proposed model operating in semi-supervised learning mode outperforms all the baselines on both flights in F1 score and accuracy. The model is able to learn complex temporal dynamics in time series data and at the same time to use known anomalies and extract relevant information from them. It achieves both high precision and high recall resulting in highest F1 score and accuracy among all models. It is detecting the majority of the RFIs with high precision.

Based on our test set, the estimated average inference time of our method is $0 \cdot 125$ s per flight where the average number of epochs in a flight is 1698 considering that the epoch sampling interval is 1 Hz.

Figures 9 and 10 illustrate the performance of our model on jammed flight 1 and flight 2. We can see jammed flight 1 and 2 trajectories with marked true positives, false positives, true negatives and false negatives in Figures 9(a) and 10(a). Correlation between anomaly scores and C/No drops can be observed in Figures 9(b) and 9(c) for jammed flight 1, and Figures 10(b) and 10(c) for jammed flight 2. Finally, we present marked true positive, false negative and false positive epochs on mean C/No plot in Figure 9(d) for jammed flight 1, and Figure 10(d) for jammed flight 2.

5.5. Discussion

Next, we discuss two systematic errors made by our model that are clearly observable in the empirical evaluation.

Systematic false negatives. We observe a systematic error mode of our model in both flights leading to false negatives or missed anomalies. Concretely, in jammed flight 1, we see a concentration of false negatives around flight epochs 2000 and 3200. Similarly, we observe the same type of concentration of
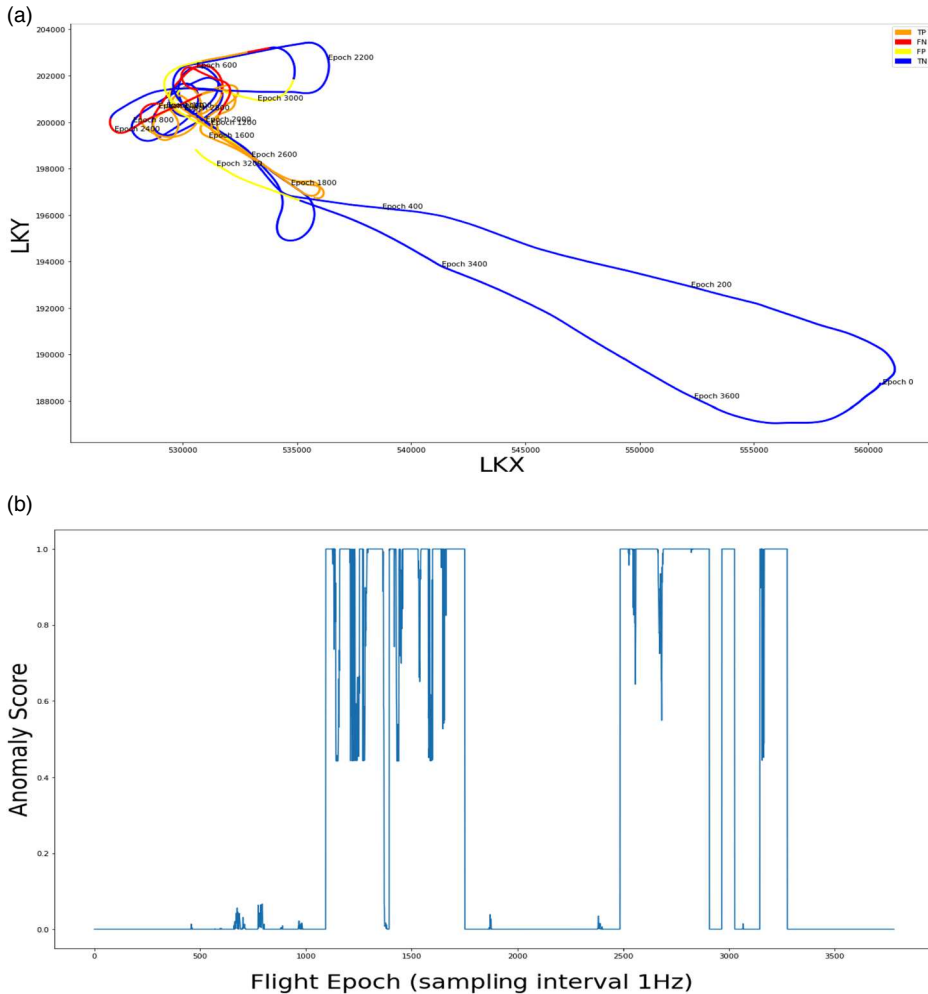

Figure 10. *Model evaluation on jammed flight 2. (a) Flight trajectory and marked true positive (TP), false positive (FP), true negative (TN) and false negative (FN) epochs. (b) Model anomaly scores of all epochs. (c) Values of all C/No. (d) Mean value of all C/No (blue line) and marked TP, FP, TN and FN epochs.*
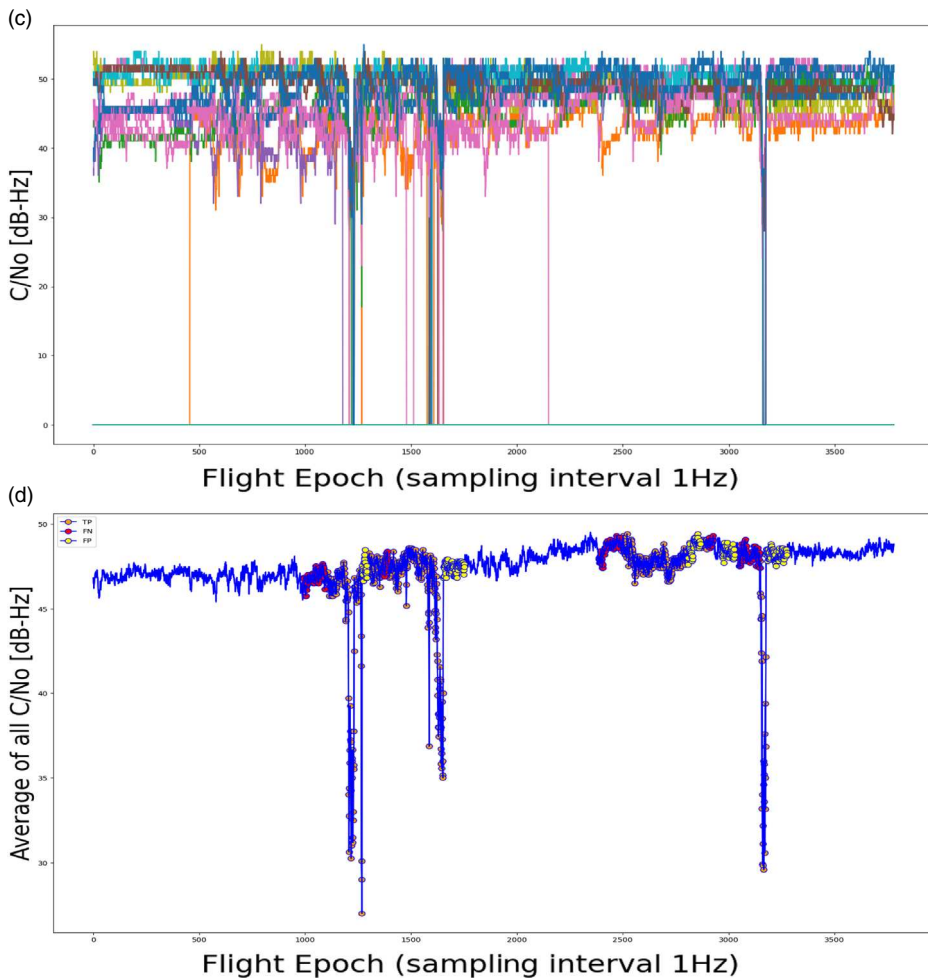
(c)

(d)

Figure 10. *Continued.*

false negatives in jammed flight 2 around flight epochs 1000 and 2500. In both cases, these are epochs impacted by RFI following a long period of normal recordings. The intuition behind these errors is that the model is estimating potential RFI for a given epoch based on the past recordings and so it is unable to accumulate sufficient anomalous context to estimate potential RFI quickly enough. Therefore, our model might be ineffective in situations where the jamming interval is really short. Furthermore, we should be aware that given a detected RFI, we should always expect that flight recordings prior the first successful detection might also be impacted by RFI but not detected by the model.

Systematic false positives. We observe another systematic error mode of our model in both flights resulting in false positives. These are the situations straight after the jammer is turned off and before it is turned on again. Overall, these are questionable situations that also depend on the GPS recovery time but, in the general case, we might expect that after a period of RFI impacting the flight recordings, the model might produce false positives for some period shortly after the RFI is over considering that the epochs assessed by the model straight after the RFI will still have a historical context of recordings impacted by the RFI.

Application to vehicles on the ground. In principle, our approach could also be used for vehicles on the ground, not only for helicopters, but a poorer performance is to be expected. The multipath effects will be stronger from surrounding environments and are likely to be problematic. In the case of

helicopters, the multipath effect is largely reduced, as data were only used if the velocity was larger than 10 m/s. It can therefore be expected that multipath effects only occur for a short time period and therefore hardly influence the result (exceptions can be when the helicopter flies over a smooth water surface, but this was rarely the case). Furthermore, the helicopters usually fly higher above the ground if they are moving at more than 10 m/s and therefore potential reflectors are further away from the receiving antenna. Depending on the correlator spacing of the receiver used, the multipath error collapses from a certain distance and therefore also contributes to a better result. In the conservative case of BPSK1 and 1 chip correlator spacing, the error is practically zero with a multipath delay of just over 400 m. For a vehicle on the ground, it can be assumed that these two influences are stronger than for a helicopter. Especially when, for example, a car is moving slowly in city traffic and is surrounded by reflectors such as building walls and the ground, this method is likely to lead to degraded results.

Training with synthetic jamming signals. Considering that real-world jamming data are hard to acquire, users of our model could, in principle, use a software-based approach to add synthetically generated jamming signals to real measurements of GNSS, and use the modified GNSS measurements to train the model. In such a case, the jammer has to be modelled and this highly depends on the environment (terrain, obstacles, etc.), the jammer antenna characteristics, etc. and depending on the applied model, the result might be better or not.

6. Conclusion and future work

We presented a deep learning approach for detection of GNSS RFI from large amounts of aircraft data. Our solution is easily transferable and can be used as a foundation for different anomaly detection tasks in CNS and ATM data. Extensive empirical studies demonstrate that the proposed method outperforms strong baselines. As a next step, we plan to evolve our solution towards localisation of GNSS interference sources.

References

Ala'Darabseh, E. B. and Tedongmo, B. (2019). Detecting GPS Jamming Incidents in Opensky Data. In: *Proceedings of the 7th OpenSky Workshop*, 97–108.

Beggel, L., Pfeiffer, M. and Bischl, B. (2020). Robust Anomaly Detection in Images using Adversarial Autoencoders. In: *Machine Learning and Knowledge Discovery in Databases European Conference*, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I. Springer, 206–222. Available at: https://doi.org/10.1007/978-3-030-46150-8_13.

Breiman, L. (2001). Random forests. *Machine Learning*, **45**, 5–32.

Chang, S., et al. (2017). Dilated Recurrent Neural Networks. *Advances in Neural Information Processing Systems 30*.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. Available at: https://doi.org/10.1145/2939672.2939785.

Chen, Y., Zhou, X. S. and Huang, T. S. (2001). One-Class SVM for Learning in Image Retrieval. In: *Proceedings 2001 International Conference on Image Processing* (Cat. No. 01CH37205). IEEE, 34–37.

Christie, J. R. I., Parkinson, B. W. and Enge, P. K. (1996). The effects of the ionosphere and C/A frequency on GPS signal shape: considerations for GNSS-2. Institute of Navigation, Alexandria, VA, USA. Citeseer, 1, 647–653.

Ebrahimi Mehr, I. A. and Dovis, F. (2023). A Deep Neural Network Approach for Detection and Classification of GNSS Interference and Jammer. TechRxiv.

Essentials, R. (2012). *A Concise Handbook for Radar Design and Performance G*. Richard Curry Institution of Engineering and Technology.

Eurocontrol, A. I. U. (2021). *Does radio frequency interference to satellite navigation pose an increasing threat to network efficiency, cost-effectiveness and ultimately safety?* Tech. Rep. 2021-2003, March, Brussels, Belgium.

Görnitz, N., et al. (2014). Learning and Evaluation in Presence of Non-i.i.d. Label Noise. In: *Artificial Intelligence and Statistics*. PMLR, 293–302.

Hendrycks, D., Mazeika, M. and Dietterich, T. (2018). Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606.

Jonáš, P. and Vitan, V. (2019). Detection and Localization of GNSS Radio Interference using ADS-B Data. In *2019 International Conference on Military Technologies (ICMT)*. IEEE, 1–5. Available at: https://doi.org/10.1109/MILTECHS.2019.8870034.

Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems 30*.

Kodali, N., et al. (2017). On convergence and stability of GANs. arXiv preprint arXiv:1705.07215.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436–444. doi:10.1038/nature14539

Li, D., et al. (2019). MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In: *Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks*, Munich, Germany, September 17–19, 2019, Proceedings, Part IV. Springer, 703–716. Available at: https://doi.org/10.1007/978-3-030-30490-4_56.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2008). Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422. Available at: https://doi.org/10.1109/ICDM.2008.17.

Liu, B., et al. (2013). SVDD-based outlier detection on uncertain data. *Knowledge and Information Systems*, **34**, 597–618. doi:10.1007/s10115-012-0484-y

Liu, Z., Lo, S. and Walter, T. (2020a). Characterization of ADS-B Performance under GNSS Interference. In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 3581–3591. Available at: https://doi.org/10.33012/2020.17675.

Liu, Z., Lo, S. and Walter, T. (2020b). GNSS interference characterization and localization using opensky ADS-B data. *Proceedings*, **59**, 10. doi:10.3390/proceedings2020059010

Liu, Z., Lo, S. and Walter, T. (2021). GNSS Interference Detection using Machine Learning Algorithms on ADS-B Data. In: *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 4305–4315. Available at: https://doi.org/10.33012/2021.18111.

Liu, Z., Lo, S. and Walter, T. (2022). GNSS Interference Source Localization using ADS-B Data. In: *Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*, 158–167. Available at: https://doi.org/10.33012/2022.18241.

Lukeš, P., et al. (2020). Recognition of GNSS Jamming Patterns in ADS-B Data. In: *2020 New Trends in Civil Aviation (NTCA)*. IEEE, 9–15. Available at: https://doi.org/10.23919/NTCA50409.2020.9291039.

Malhotra, P., et al. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148.

Mehr, I. E. and Dovis, F. (2022). Detection and Classification of GNSS Jammers Using Convolutional Neural Networks. In: *2022 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 1–6. Available at: https://doi.org/10.1109/ICL-GNSS54081.2022.9797030.

Morales Ferre, R., de la Fuente, A. and Lohan, E. S. (2019). Jammer classification in GNSS bands via machine learning algorithms. *Sensors*, **19**(22), 4841. doi:10.3390/s19224841

Moya, M. M., Koch, M. W. and Hostetler, L. D. (1993). *One-class classifier networks for target recognition applications*. NASA STI/Recon Technical Report N, 93, 24043.

Qiu, C., et al. (2022). Latent Outlier Exposure for Anomaly Detection with Contaminated Data. In: *International Conference on Machine Learning*. PMLR, 18153–18167.

Recommendation, I. (2013). Attenuation by atmospheric gases P Series Radiowave propagation. Geneva.

Ruff, L. (2021). Deep one-class learning: a deep learning approach to anomaly detection. Technische Universitaet Berlin, Germany.

Ruff, L., et al. (2018). Deep one-Class Classification. In: *International Conference on Machine Learning*. PMLR, 4393–4402.

Ruff, L., et al. (2019). Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906. 02694.

Ruff, L., et al. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, **109**(5), 756–795. doi:10.1109/JPROC.2021.3052449

Scaramuzza, M., et al. (2014). RFI detection in Switzerland based on helicopter recording random flights, Oklahoma IFIS2014.

Scaramuzza, M., et al. (2015). GNSS RFI Detection: Finding the Needle in the Haystack. In: *Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, 1617–1624.

Scaramuzza, M., et al. (2016). Empirical Assessment and Modelling of RFI Impact on Aviation GPS/SBAS Receiver Performance. In: *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, 3063–3069. Available at: https://doi.org/10.33012/2016.14586. .

Scaramuzza, M., et al. (2017). Quality Assessment of GNSS Simulations for Flight Procedures based on Onboard Recorded Flight Data. In: *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 1633–1643. Available at: https://doi.org/10.33012/2017.15129.

Scaramuzza, M., et al. (2019). Investigation of Spatial and Temporal RFI Events Distribution within the Swiss Airspace. In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 1392–1400. Available at: https://doi.org/10.33012/2019.16905.

Schölkopf, B., et al. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, **13**(7), 1443–1471. doi:10.1162/089976601750264965

Schölkopf, B., et al. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press. Available at: https://doi.org/10.7551/mitpress/4175.001.0001SEP.

Series, P. (2015). Propagation data and prediction methods required for the design of Earth-space telecommunication systems. Recommendation ITU-R, 618–612.

Series, P. (2016). Ionospheric propagation data and prediction methods required for the design of satellite services and systems. Recommendation ITU-R, 531–513.

Shen, L., Li, Z. and Kwok, J. (2020). Timeseries Anomaly Detection using Temporal Hierarchical One-class Network. *Advances in Neural Information Processing Systems 33*, 13016–13026.

Steinwart, I., Hush, D. and Scovel, C. (2005). A classification framework for anomaly detection. *Journal of Machine Learning Research*, **6**(2), 211–232.

Su, Y., et al. (2019). Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2828–2837. Available at: https://doi.org/10.1145/3292500.3330672.

Swinney, C. J. and Woods, J. C. (2021). GNSS Jamming Classification via CNN, Transfer Learning & the Novel Concatenation of Signal Representations. In: *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. IEEE, 1–9. Available at: https://doi.org/10.1109/CyberSA52016.2021.9478250.

Tax, D. M. J. (2002). One-class classification: concept learning in the absence of counter-examples.

Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Machine Learning*, **54**, 45–66. doi:10.1023/B:MACH.0000008084.60811.49

Truffer, P., et al. (2017). Jamming of Aviation GPS Receivers: Investigation of Field Trials Performed with Civil and Military Aircraft. In *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 1258–1266. Available at: https://doi.org/10.33012/2017.15303.

Wang, S., et al. (2019). Effective End-to-End Unsupervised Outlier Detection via Inlier Priority of Discriminative Network. *Advances in Neural Information Processing Systems 32*.

Xia, Y., et al. (2015). Learning Discriminative Reconstructions for Unsupervised Outlier Removal. In: *Proceedings of the IEEE International Conference on Computer Vision*, 1511–1519. Available at: https://doi.org/10.1109/ICCV.2015.177.

Yoon, J., et al. (2021). Self-supervise, refine, repeat: improving unsupervised anomaly detection. arXiv preprint arXiv:2106.06115.

Zhao, Y., Wang, S. and Xiao, F. (2013). Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD). *Applied Energy*, **112**, 1041–1048. doi:10.1016/j.apenergy.2012.12.043

Zhou, C. and Paffenroth, R. C. (2017). Anomaly Detection with Robust Deep Autoencoders. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 665–674. Available at: https://doi.org/10.1145/3097983.3098052.

Zhou, B., et al. (2019). BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series. In: *IJCAI*, 4433–4439. Available at: https://doi.org/10.24963/ijcai.2019/616.