Detection and Mitigation of Jamming, Meaconing, and Spoofing based on Machine Learning and Multi-Sensor Data

Philipp Bohlig, Jorge Morán García, Ramadevi Lalgudi, Jan Fischer, *ANavS GmbH*

BIOGRAPHY

Philipp Bohlig is the Head of the GNSS and Sensor Fusion group within ANavS. He has received his B.Sc. and M.Sc. degrees in Electrical Engineering and Information Technology of the Technical University of Munich, Germany. His fields of expertise and research cover tightly coupled sensor fusion, Precise Point Positioning (PPP), Integer Ambiguity Resolution (IAR) and GNSS integrity. As Head of the GNSS and Sensor Fusion group within ANavS, he coordinates and supervises technical activities in the development of multi-sensor positioning algorithms.

Ramadevi Lalgudi is a Development Engineer at ANavS GmbH, working on classification of signal anomalies using deep learning techniques to detect interference in GNSS signals. She received her master's degree in Information and Communication Engineering from TU Darmstadt and has prior experience in statistical signal processing.

Jan Fischer completed his master's degree in Mechatronics and Information Technology at the Karlsruhe Institute of Technology. Since 2022 he has been working as a data fusion and computer vision engineer at ANavS GmbH. His main focus at ANavS is the Multi-Sensor RTK and visual SLAM system.

Jorge Morán García is a Project Manager at ANavS, leading multiple European initiatives on high-accuracy and integrity localization solutions that combine GNSS (RTK and PPP-AR) with sensor fusion approaches using cameras, LiDARs, and other sensing technologies. He is also coordinating the DREAM project (DRiving aids by E-GNSS AI and Machine learning), which forms the basis for the research and results presented in this paper.

ABSTRACT

Global navigation satellite systems (GNSS) are an integral part to the functioning of various global applications including autonomous driving, aviation, and more. However, GNSS signals are often non-authenticated and weak, making them easy to override with stronger, false signals. While Galileo Open Service Navigation Message Authentication (OSNMA) tackles one element of these weaknesses, several GNSS interference types remain a challenge to be solved. Any localization application and especially if safety-critical highly depends on GNSS and thus, needs to incorporate concepts of detecting and mitigating GNSS interference, i.e., Jamming, Meaconing, and Spoofing, to provide safe and robust position, navigation, and timing (PNT) solutions.

To provide robustness against GNSS interference while maintaining a high accuracy, we propose to include a machine learning based approach which utilizes not only GNSS measurements but also additional features from unaffected sensors like an IMU. Converse to various approaches which consider only one or two features, the common processing of the features listed in the following paragraph in a recurrent neural network (RNN) reduces the risk of false alarms and missed detection, enhancing the system robustness against GNSS interference and maintaining high accuracy and availability.

The recurrent approach is used to include features of multiple consecutive epochs for the detection of an interference state of a specific epoch. Including the average Carrier-to-noise ratio (C/N_0) of a frequency and the corresponding Gain amplifier together as reported by the GNSS receiver, which implements Automatic Gain Control (AGC), abrupt changes due to interference can be detected and distinguished from a normal C/N_0 drop based on signal occlusion. Cross-band carrier-phase divergence is another phenomenon observed during e.g., spoofing. The ratio between carrier phases from signals transmitted by the same satellite in different bands changes substantially more than under nominal conditions. Finally, the feature set is complemented by raw IMU measurements, which serve as a robust cross-checking source given their immunity to GNSS interference.

The features are combined in a normalized label vector, where each sample is labeled by the affected carrier (e.g., GPS_L1 or GAL_E6), resulting in a multi-label classification problem. In this work, we compare the results of an individual labeling based on information of the type of interference (Jamming, Meaconing, Spoofing) and a common labeling as interference.

The proposed deep learning framework for multi-label GNSS interference detection consists of a convolutional neural network (CNN) encoder combined with a long short-term memory (LSTM) network. It is shown that the AI-based detector effectively classifies the interference per GNSS constellation and signal, allowing to selectively remove only the compromised signals rather than discarding the entire solution. This targeted approach enables smarter decision-making at the GNSS Kalman filter

level.
One key aspect of our work is that the network will be trained and tested with real jammed and spoofed signals, leveraging on the multi-sensor data recorded at Jammertest 2024. The well labeled collection of datasets is made publicly available together with the network itself.

I. INTRODUCTION

Global Navigation Satellite Systems (GNSS) have become indispensable for modern society, underpinning positioning, navigation, and timing (PNT) services across transportation, telecommunications, finance, and critical infrastructure. With constellations such as GPS, GLONASS, Galileo, and BeiDou providing worldwide coverage, the reliability of GNSS enables applications ranging from aviation and maritime safety to autonomous vehicles and location-based services. However, the increasing reliance on GNSS also exposes significant vulnerabilities. Due to the weak power of GNSS signals at the Earth's surface and the absence of inherent authentication mechanisms, receivers are highly susceptible to deliberate interference. The occurrence of GNSS interference is reported increasingly nowadays, not only in the military sector but also in e.g., aviation and critical infrastructure (Felux et al., 2024; OPSGROUP GPS Spoofing Workgroup, 2024).

Three primary forms of signal-in-space attacks exist: jamming, spoofing, and meaconing. Jamming is the most direct method, where an adversary transmits noise at GNSS frequencies, overwhelming the receiver and denying service. Spoofing is a more sophisticated approach in which counterfeit signals are crafted to mislead a receiver into computing false positions or timing. Meaconing differs in that authentic signals are captured and re-broadcast with a delay, tricking receivers without generating new signals. These threats can lead to severe operational consequences, from navigation errors and loss of synchronization to safety-critical failures in aviation, autonomous driving, and emergency services.

Over the past decade, numerous detection and mitigation strategies have been proposed. Traditional methods include monitoring signal-to-noise ratio (SNR), analyzing correlation peaks, and integrating data from multiple sensors or constellations. Hardware-based countermeasures such as directional antennas and array processing can also be effective but come at the cost of complexity and scalability. More recently, cryptographic solutions like Galileo's OSNMA provide authentication at the message level. While these techniques contribute to resilience, they often struggle against sophisticated spoofing and meaconing attacks or introduce trade-offs in cost, infrastructure, and deployment feasibility. Additionally, there are multiple other approaches to tackle interference detection at different stages in the processing chain or with different setups as e.g., with antenna array processing, spectral analyses, or the estimation of the direction of arrival [see e.g., (Montgomery et al., 2009; Zhang et al., 2019; Zhao et al., 2022)].

In this context and when tackling the stage of receiver measurements, machine learning and deep learning have emerged as promising approaches for GNSS interference detection. Unlike threshold-based techniques, deep models can capture subtle, high-dimensional patterns in the data, adapt to new attack strategies, and generalize across scenarios. Many researchers lack comprehensive methods capable of simultaneously detecting spoofing, jamming, and meaconing across multiple constellations in realistic conditions. In this work, we contribute at the post-correlator stage, leveraging GNSS measurements, signal characteristics, and sensor fusion data.

This paper addresses these challenges by presenting a deep learning framework for multi-label GNSS interference detection. Our model employs a convolutional neural network (CNN) encoder combined with a long short-term memory (LSTM) network. The CNN extracts relevant features of data from GNSS measurement and signal characteristics as well as from IMU measurements, and the LSTM network models the output of the CNN as a sequence. Unlike prior work that treats detection as a binary classification problem, our approach performs multi-label classification, identifying which of the GNSS signals across multiple constellations are affected. Consequently, the subsequent positioning algorithm can remove signals with interference but keep using unaffected signals to remain with high accuracy and availability even in GNSS interference scenarios. Moreover, we compare two different approaches of the multi-label classification, i.e., to treat all types of interference as one label and to differentiate between Jamming, Meaconing, and Spoofing.

The proposed method is validated using real-world datasets collected under control but realistic spoofing and jamming scenarios, at the Jammertest campaign 2024 in Andøya, Norway. In the experimental environment, we verified that the CNN-LSTM model obtained values above 0.9 for precision, recall, and F1 score respectively, demonstrating the effective classification. We also outline the challenges for future improvements.

This paper is structured as follows: Section II reviews related work on GNSS interference detection and mitigation concentrating on machine learning approaches. Section III introduces the proposed CNN-LSTM detection framework, utilized input features and information about the labeling. Section IV describes the various datasets from Jammertest 2024 and the experimental setup. Section V presents the results and analysis, followed by conclusions and an overview of the next steps in Section VI.

II. RELATED WORK

Although traditional signal processing and statistical techniques have formed the basis of spoofing and jamming detection, machine learning methods are gaining widespread popularity due to their ability to model complex patterns, adapt to unseen attack scenarios, and improve detection accuracy (Mohanty and Gao, 2024).

Conventional machine learning methods such as Support Vector Machines (SVM), Gaussian Mixture Models (GMM), and linear regression have been applied to spoofing and jamming detection with notable success. For example, (Meng et al., 2021) demonstrated that linear regression can improve spoofing detection for unmanned aerial vehicles (UAV) despite limitations in handling nonlinear relationships. (Shafique et al., 2021) employed multiple machine learning models with K-fold validation. Similarly, SVMs have been applied for both spoofing and jamming detection, with studies showing an effective classification of interference signals using various entropy features (Xu et al., 2020). GMM-based clustering methods have further been used for unsupervised spoofing detection, isolating spoofed pseudoranges from legitimate GNSS data (Feng et al., 2022).

In contrast to traditional ML approaches, deep learning techniques leverage neural architectures to automatically extract relevant features from GNSS signals. For spoofing detection, multilayer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) have been widely explored. (Shafiee et al., 2017) applied an MLP-based detector using early-late phase and signal-level features, while (Manesh et al., 2019) detected GPS spoofing with inputs consisting of pseudorange, Doppler shift, and signal-to-noise ratio. Leveraging the Cross Ambiguity Function (CAF), (Borhani Darian et al., 2020) proposes an MLP and two classes of CNNs capable of identifying multiple peaks in the CAF, which serve as indicators of GNSS spoofing. The combination of CNN and LSTM networks has been applied to the classification of transient radio frequency interference, demonstrating strong capability to identify interference sources from data in the time domain (Czech et al., 2018). Evaluated in a simulation environment with real-time UAV sensor data, the CNN-LSTM-Attention model developed by (Wu et al., 2023) proved effective in detecting both denial-of-service and GPS spoofing attacks. For jamming detection, (Janiar and Wang, 2024) proposed a deep reinforcement learning–based model capable of adapting to both static and dynamic scenarios.

Important contributions are also made in the field of labeled data which is essential for the development of any type of detector for GNSS interference. Simulation datasets as the above-mentioned work of (Wu et al., 2023) can reduce the cost of data acquisition and processing. However, they often fail to capture the full complexity and variability of real-world GNSS interference scenarios, which can lead to degraded model performance. To address this, several public datasets are available for research, including TEXBAT (Humphreys et al., 2012), OAKBAT (Albright et al., 2020), and the FGI GNSS Jamming and Spoofing Dataset Repository (FGI-JSDR) (Islam et al., 2024). However, none of them provides a set comprehensive enough for the approach presented in this paper which requires to include multi-sensor, multi-constellation data complemented with signal characteristics for effective classification of GNSS interference.

III. METHODOLOGY

Even if multiple researches on neural networks for GNSS interference detection are present including deep NN for automatic feature extractions, a comprehensive approach covering different interference types as well as multi-constellation and multi-sensor data for robustness is missing, but presented in this paper. The approach utilizes the features as outline in Sec. III.1 and is described in detail in Sec. III.2. The proposed architecture of the NN combining a CNN with an LSTM is depicted in Sec.III.3

1. Utilized Features

As mentioned in the introduction, traditional methods rely on single or dual characteristics. One common example is the joint analysis of the C/N_0 and the receiver Gain. However, based on thresholds both neural networks and heuristic approaches suffer from either lots of false alarms or missed detections if the methodology is restricted to very few types of input features. Consequently, we propose to combine multiple input features from the signal characteristics, GNSS measurements, and other sensors as outlined in Fig. 1. In addition to the already mentioned C/N_0 and Gain, we use the ratio between phase measurements of two different frequencies and double differenced measurements from multiple receivers for each tracked satellite as well as measurements from the IMU. This subsection gives an overview to the presented input features and why they are useful to learn patterns to detect GNSS interference.

Note that double differenced measurements are not covered in the data used in the experimental setup described in Sec. IV because the evaluations in this paper are restricted to static data. Dynamic data is foreseen by the concept and covered in the recorded data but will be part of future work.

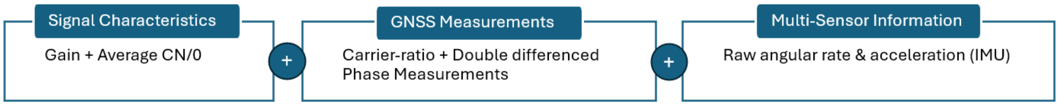

Signal Characteristics
Gain + Average CN/0
+
GNSS Measurements
Carrier-ratio + Double differenced Phase Measurements
+
Multi-Sensor Information
Raw angular rate & acceleration (IMU)

Figure 1: Summary of the proposed input features for the deep neural network.

Gain amplifier: GNSS receivers, including the selected COTS, implement Automatic Gain Control (AGC), where an adaptive gain amplifier is specifically designed to reduce quantization error. It is primarily driven by the ambient noise environment, especially when the signal power is below the thermal noise floor. When unwanted interference occurs, AGC quickly reduces gain in response to increased power within the GNSS band. Thus, the AGC gain value provided by the COTS receiver can be used for interference detection.

Carrier-to-Noise Ratio (C/N_0): We include the C/N_0 on a per-satellite basis as it is an input feature that reacts during transitions from or to an interfered state, but is also higher than usual once the interfered signal took over the receiver. Regarding the transition phase, the C/N_0 experience abrupt changes due to interference, which might exceed normal conditions. Detectors that monitor these sudden shifts can help to assess interference. However, sudden changes can also happen when the environment changes quickly under dynamic conditions. Consequently, the C/N_0 can only be used in combination with other features.

Cross-band ratio of phase measurements: The ratio between carrier phase measurements from signals transmitted by the same satellite in different bands should be relatively constant. Ionospheric delays are one source that lead to changes of the carrier ratio (CR) over time but the dynamics of these changes are very moderate compared to the reaction of the GNSS receiver under interference. Hence, clear carrier-phase divergence is an indicator of GNSS interference.

For an easier inspection, we use a modified version of the CR as input features. This modified value for satellite k and the signal combination of the m-th and the first signal is given as:

$$\text{mCR}_m^k = 10^6 \left(\frac{\lambda_1 \varphi_1^k}{\lambda_m \varphi_m^k} - 1 \right) \tag{1}$$

This modified CR is shown in Fig 2 exemplarily. The dashed red lines indicate changes of the label from nominal to interference or the other way around. For a given sequence denoted with the time of week (ToW), the reactions to interference is clearly visible, while during nominal conditions the CR is continuous and has only a slight drift induced by the change of the ionospheric slant delay.

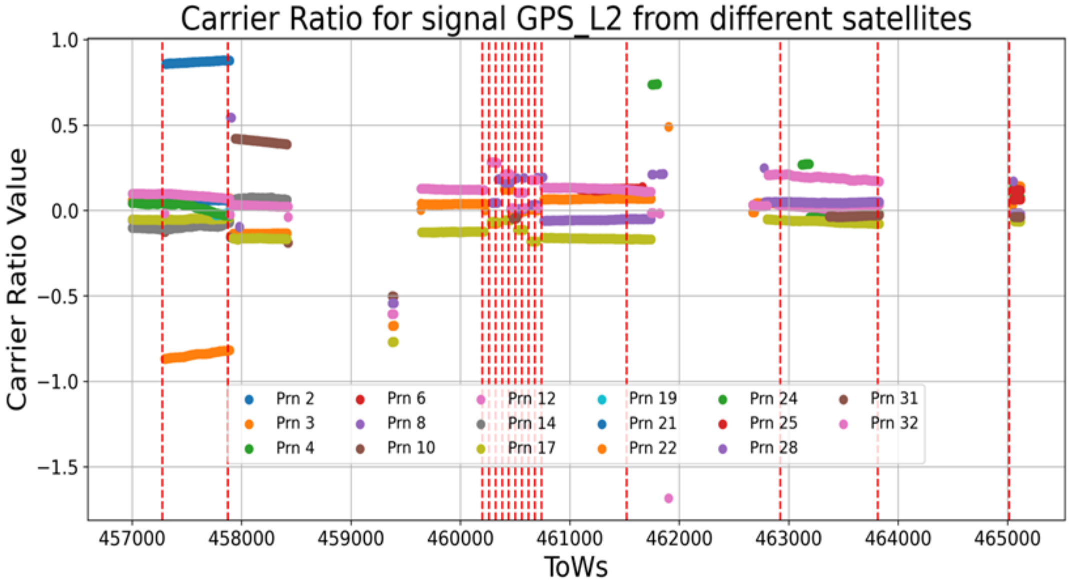


Figure 2: Behavior of the modified carrier ratio for GPS_L2 and a given sequence. Transitions from and to an interference state are indicated with vertical red dashed lines.

Dual-antenna double differences of phase measurements: A time series of double-differenced (DD) phase measurements are useful for interference detection in GNSS because it can highlight inconsistencies that arise from external disturbances. Double differences measurements inherently remove common errors between satellites and receivers due to differentiating measurements between the receivers in a multi-receiver setup and between satellites (with a reference satellite per constellation). By doing so, many effects are not inducing changes of the DD measurements, isolating other effects such as interference. This feature enhances detection accuracy by focusing on the time series of residuals, where unexpected variations are an indication for interference. To separate between unexpected variations and variations due to changes in the dynamics, i.e., the relative position between the two antennas in the navigation frame, the combination with IMU measurements listed below is essential.

IMU measurements: Informations from other non-GNSS sensors such as IMU measurements serve as a strong source for cross-checking. The features are not affected by GNSS interference and provide information about changes in the dynamics. Thus, the measurements are complementing the information about DD measurements and the CR when reacting to dynamics.

2. Two Concepts for a Multi-Label Classification for GNSS Interference

In our work, we compare two different approaches regarding the labeling and evaluate the capabilities to classify GNSS interference properly in Sec. V. These are an individual labeling based on information of the type of interference (Jamming, Meaconing, and Spoofing) and a common labeling as interference.

In the first approach, each recorded sample is annotated with both the type of interference (jamming (J), meaconing (M), or spoofing (S)) and the affected carrier (e.g., GPS_L1, GPS_L2, GPS_L5, Galileo_E1, Galileo_E5a, Galileo_E5b). This setup defines a multi-label classification problem, since a single sample may be associated with multiple active interference types across different carriers.

The label vector has a length of $N_{\text{labels}} = N_{\text{carriers}} \times N_{\text{types}}$. For instance, with 16 carriers and 3 interference types, the total length is 48. The vector is ordered as follows:

$$[\ldots,\ J/E1,\ S/E1,\ M/E1,\ J/E5a,\ S/E5a,\ M/E5a,\ \ldots,\ J/L5,\ S/L5,\ M/L5\,],$$

where the constellations are sorted alphabetically as Beidou, Galileo, Glonass, and GPS signals. Each element is a binary indicator that specifies whether a given carrier is affected by a particular interference type. For example, if spoofing occurs on Galileo E1, the element $S/E1$ is set to 1, while all other entries remain 0. Formally, the j-th element of the label vector for sample i is defined as

$$y_{i,j}^{(1)} = \begin{cases} 1 & \text{if } \{\text{type}(j), \text{carrier}(j)\} \in \text{interference}(i), \\ 0 & \text{otherwise.} \end{cases}$$

In the second approach, the interference types are not distinguished and the multi-label classification problem is reduced to labels per GNSS signal. The label vector length is now $N_{\text{labels}} = N_{\text{carriers}}$ and the j-th element of the label vector for sample i is defined as

$$y_{i,j}^{(2)} = \begin{cases} 1 & \text{if carrier}(j) \in \text{interference}(i), \\ 0 & \text{otherwise.} \end{cases}$$

3. Network Architecture

We propose a CNN-LSTM framework shown in Figure 3 to detect GNSS interference, including spoofing, jamming, and meaconing attacks. The input to the network is a matrix of size $N_s \times N_m \times 2$, where N_s is the number of GNSS signals and N_m is the number of measurements per signal. Each row corresponds to a single GNSS signal and contains the following features: C/N_0, CR, PRN, constellation identifier, frequency, AGC, sample variance, and IMU data. Note that IMU samples are available at a higher rate than GNSS and all measurements since the last GNSS epoch are filled into the rows of the signals even if unrelated to the signals themselves. Further note that the PRN, the constellation identifier, and the frequency are provided such that the network can link the input to the related measurements within the sequence.

The input matrix is constructed with a fixed size chosen to accommodate the maximum number of signals observed in the dataset. For signals or measurements that are missing, such as when a satellite is not observed or features like AGC or C/N_0 are unavailable, the corresponding entries are set to zero. An additional binary mask channel is added to indicate which values correspond to zero padding, allowing the network to distinguish valid measurements from missing data. The input is first processed by a three-layer CNN encoder. Each convolutional layer extracts feature representations from the input, capturing inter-signal correlations and local measurement patterns. The output of the final CNN layer is aggregated using average pooling to produce an embedding vector summarizing the signal-level information. For temporal modeling, embeddings corresponding to multiple consecutive timestamps are sequentially input into a LSTM network. The LSTM captures the temporal dependencies between successive embeddings, enabling the framework to exploit dynamic patterns in the signals that indicate interference. Finally, the LSTM output is passed through a fully connected layer to produce a multi-label classification output of size N_s times N_c, where N_c is the number of interference classes (spoofing, jamming, and meaconing).

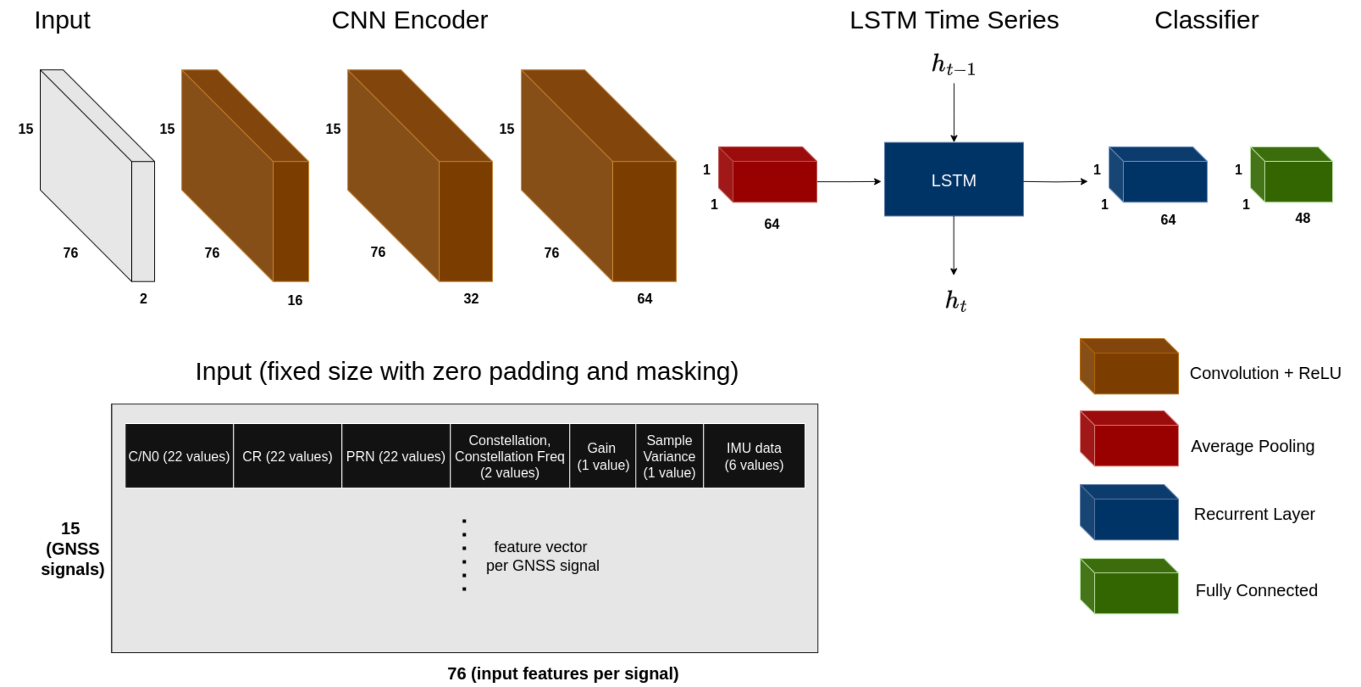


Figure 3: Architecture of the CNN-LSTM network

4. Loss Function

The Binary Cross-Entropy (BCE) loss function (2) combined with a Sigmoid activation function is used in this work. Unlike categorical cross-entropy loss, the loss computed for probability of each output label is not affected by probability values for other labels. Hence it is widely used for multi-label classification, where the probability and decision making of an input sample belonging to a certain label should not influence the decision for another label.

$$\mathcal{L} = -\sum_{i=1}^{C} w_i \left[y_i \log\left(f(s_i)\right) + (1 - y_i) \log\left(1 - f(s_i)\right) \right] \tag{2}$$

where

$\mathcal{L}$**:** The total BCE loss.

C**:** Number of labels (for multi-label classification).

w_i**:** Weight for label i (e.g., for positive label balancing).

$y_i \in \{0, 1\}$**:** Ground truth for label i.

s_i**:** Logit for label i (i.e., the output of the fully connected layer before applying the sigmoid).

$f(s_i) = \sigma(s_i) = \dfrac{1}{1 + e^{-s_i}}$**:** Sigmoid activation function.

Each label is treated as a separate binary classification (Interference or No interference detection) task so that the binary cross entropy loss is computed for each label and then averaged over all labels C. For datasets with imbalanced positive (interfered) and negative (non-interfered) samples along with some labels appearing more frequent than the others, label weights (pos_weight) are introduced in the loss function.

$$\text{pos\_weight}_c = \frac{N - P_c}{P_c} \tag{3}$$

N**:** Total number of samples in the dataset.

P_c**:** Number of samples where label c is positive (i.e., $y_{ic} = 1$).

pos_weight$_c$**:** Weight applied to the positive term for label c in the loss function.

5. Threshold selection

Dataset imbalance with some labels appearing more frequently than others can affect the prediction metrics. So, the decision-making thresholds for each sigmoid output to classify a sample as belonging to one or more labels are selected based on maximizing the F1 score on validation data, thus ensuring a balance between precision and recall for the multi-label classification problem. This design allows the model to simultaneously detect multiple types of interference across all GNSS signals, leveraging both the inter-signal features and their temporal evolution.

For each label $c \in \{1, 2, \dots, C\}$, the optimal threshold t_c^*

$$t_c^* = \arg\max_{t \in [0,1]} \mathrm{F1}_c(t) \tag{4}$$

is defined as the threshold that maximizes the F1 score

$$\mathrm{F1}_c(t) = \frac{2 \cdot \mathrm{Precision}_c(t) \cdot \mathrm{Recall}_c(t)}{\mathrm{Precision}_c(t) + \mathrm{Recall}_c(t)} \tag{5}$$

for label c at threshold t. To have an estimate of the optimal threshold for the test data, we maximize the F1 score for the validation dataset.

IV. DATASETS AND EXPERIMENTAL SETUP

1. Data from Jammertest 2024

We use recorded data from Jammertest 2024, a controlled large-scale campaign in Andøya, Norway. The event provides a predefined transmission plan that specifies the interference type, time, affected carriers, and coverage area. This enables precise ground truth labels to identify both the interference type and the affected signals. The dataset includes GNSS recordings of jamming, spoofing, meaconing, and combined attacks across different power levels and frequency bands. The recorded inputs include gain amplifier, carrier-to-noise ratio (C/N_0), cross-band carrier-phase divergence, sample variance, and IMU data, offering richer observables than existing datasets. A static antenna setup ensures consistent signal collection, and the dataset is split into training and validation subsets. The Jammertest 2024 dataset provides realistic, multi-constellation, and carrier-specific data uniquely suited for developing and evaluating our CNN-LSTM multi-label detection framework.

Ground truth labels are derived from the transmission plan and interference logs of Jammertest 2024, which specify the time, location, interference type, and affected carriers during each experiment. By aligning this information with the measurement timeline, each recorded sample is assigned the correct multi-label vector. This procedure ensures consistent and unambiguous supervision for the deep learning model.

The datasets comprise measurements collected over five days. Table 1 summarizes the total recording duration per day, as well as the duration of interference observed during the recordings.

Table 1: Recording duration and interference time during JammerTest 2024 sessions (MeasEpoch2 data).

Date	Recording Duration (hours)	Interference Duration (hours)
2024-09-09	5.3	2.2
2024-09-10	12.9	5.2
2024-09-11*	6.7*	0.5*
2024-09-12	11.8	5.9
2024-09-13	4.7	2.0

*Some of the spoofing type interference data was lost on 2024-09-11 due to issues with the data recorder based on the performed time spoofing.

2. Data Splitting

The dataset used in this work contains approximately 688,000 samples divided into training, validation, and test sets. The training set contains around 75%, the validation set 15% and the remaining 10% of the total sample size are set as the test set. The validation data samples are evaluated to select per-label thresholds and also to tune the hyper-parameters of our selected model. We use two different ways to split the datasets and evaluate the respective model performance. In the first setup, the samples are drawn at random and the input data representing different interference types is well distributed in training, validation and test datasets. The model trained on such data is expected to achieve over-optimistic performance metrics because the model has seen samples that are close in time to the test data and potentially correlated with it.

In the second setup, the split of samples used for training (75%), validation (15%), and testing (10%) is preserved. After partitioning the dataset, the recorded measurements from 2024-09-09 to 2024-09-11 are used for training, from 2024-09-12 for validation and 2024-09-13 for the test. The model trained on this setup is not expected to perform as good as the above since the distribution of test data might be different from that of training data. The completely unseen day also covers test cases that are different compared to the ones used for training and validation. This method is suited to evaluate how well the network generalizes in its detection capabilities.

3. Published data

To promote open research and facilitate further advancements in the field, all datasets and the corresponding labels supporting the results presented in this paper will be made publicly available. This open-access initiative aims to foster knowledge sharing, enable reproducible benchmarking, and accelerates the development of robust navigation solutions. The network and labeled data will be published at our GitHub repository.

V. RESULTS

When splitting the dataset randomly into training and test sequences, the model is able to reliably discriminate between the different interference types. To illustrate the model behavior in more detail, Figure 4 presents the prediction output for the random split over an exemplary two-hour test interval for the GPS_L2 band. The period covers all three different interference types. In addition to the true and predicted labels, features used as network inputs are also shown, including the average C/N_0, the AGC values, and the average modified carrier ratio as described in Sec. III.1. Note that the C/N_0 and modified CR are visualized as the signal-average but the network uses the values of each satellite individually.
The predicted labels closely follow the ground truth, showing high accuracy in distinguishing spoofing, meaconing, and jamming. These signals illustrate how interference manifests in the physical measurements and offer valuable context for understanding the model's predictions. Moreover, it shall be noted that some misclassifications are also not critical for the application as in the case where jamming is wrongly classified as meaconing. Both classifications would lead into an exclusion of the signal.

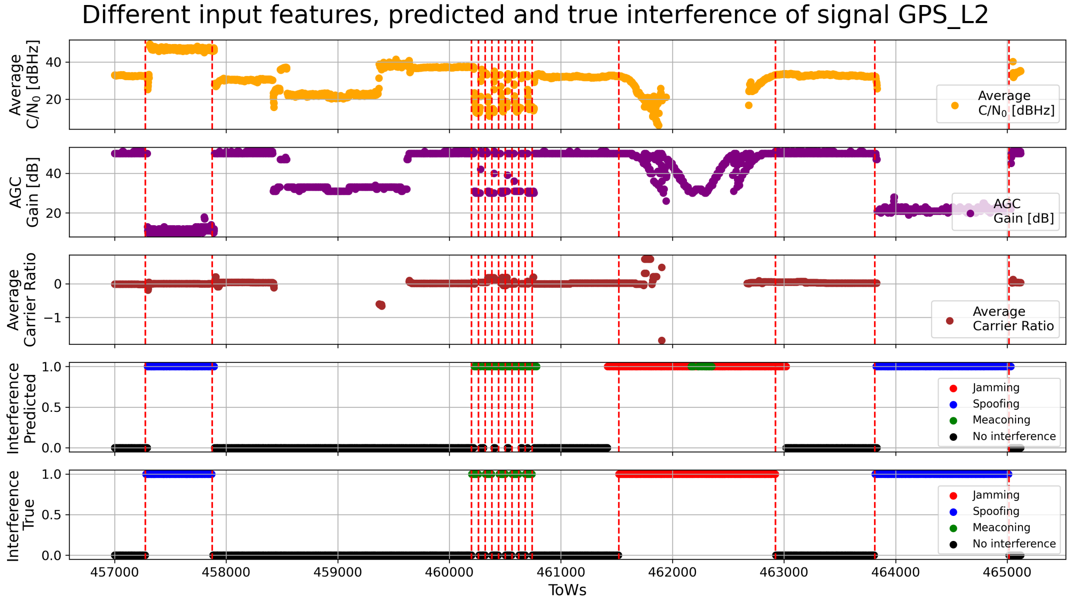


Figure 4: Model predictions for GPS_L2 test data with corresponding C/N_0, AGC, and carrier ratio signals using random split over a two-hour interval, showing close agreement with ground truth for spoofing, meaconing, and jamming

To further examine the network behavior, we present detailed time series examples of individual spoofing and jamming events on the GPS_L2 signal, highlighting how the model responds to evolving interference patterns. Figure 5 shows a spoofing example for random sampled test data. During the spoofing event, the C/N_0 increases. This behavior reflects the physical effect of spoofing, where the receiver locks onto the higher-power counterfeit signal, which is intended by the spoofer to take over GNSS receivers, leading to an apparent signal strengthening. The model prediction closely follows the ground truth but exhibits a latency. This latency arises because the ground truth is defined by the interference transmission schedule during the Jammertest, while the receiver is only affected once the spoofed signal effectively captures its tracking loops.

A similar latency can also be observed during the transition back from spoofed to non-spoofed conditions. Interestingly, the network partially learns to compensate for this effect, enabling it to predict transitions partially. This latency highlights the

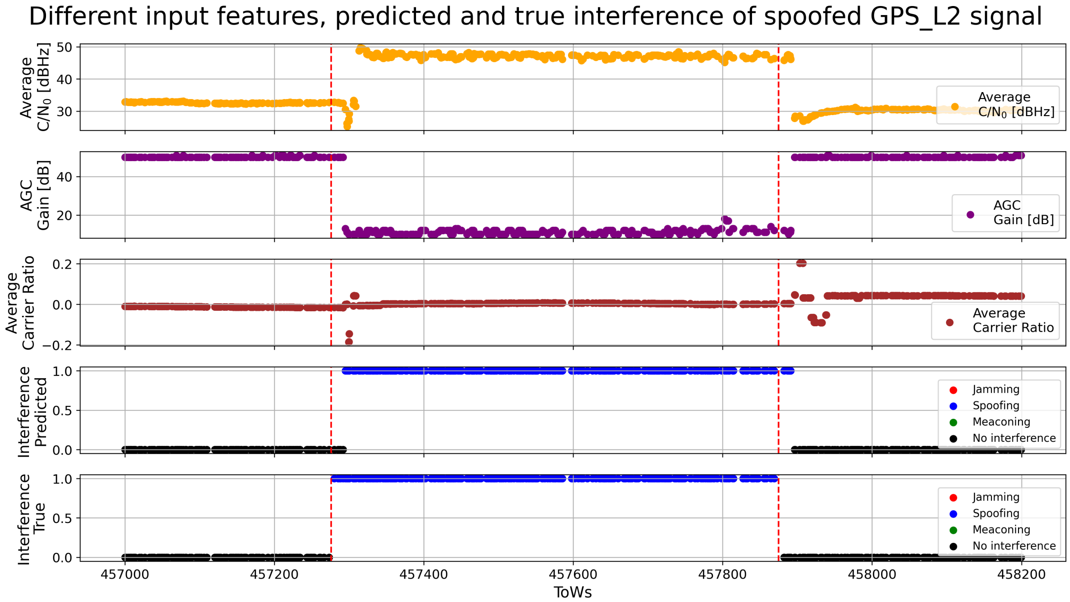


Figure 5: Model predictions for a GPS_L2 spoofing example on random split test data with corresponding C/N_0, AGC, and carrier ratio signals.

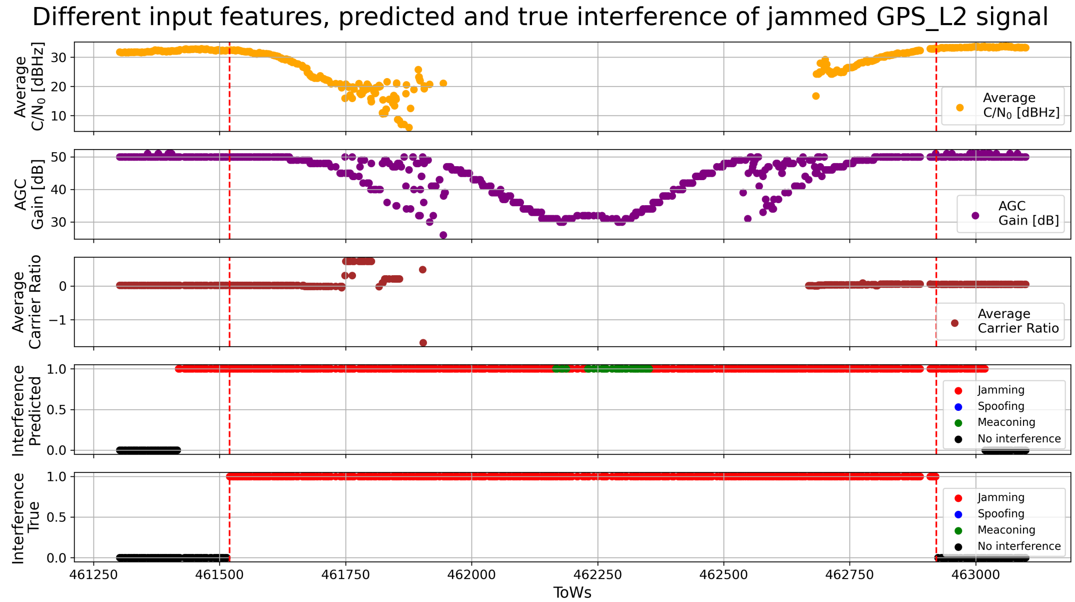
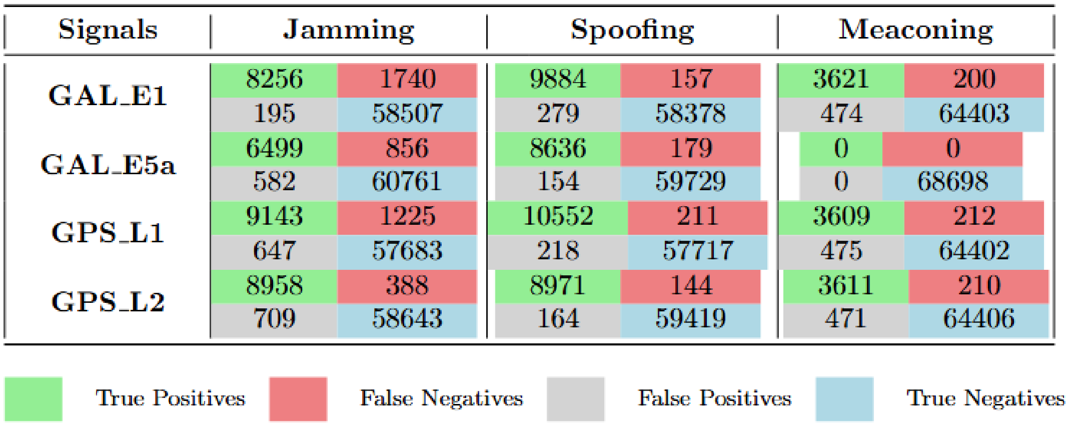


Figure 6: Model predictions for a GPS_L2 jamming example on random split test data with corresponding C/N_0, AGC, and carrier ratio signals.

importance of carefully labeled transitions in spoofing, jamming, and meaconing datasets to ensure that the network can rapidly detect interference events, supporting timely downstream processing.

Figure 6 shows a corresponding jamming example. Here, the C/N_0 values drop significantly, as expected for jamming scenarios. This reflects the physical mechanism of jamming, where wide-band interference reduces the effective signal-to-noise ratio, impairing lock maintenance in the receiver. The model predictions align closely with the ground truth, demonstrating its ability to recognize such characteristic patterns. Several delays in the reaction of the receiver are also due to slightly increasing power ramps or due to sweeps, both resulting in different times where the receiver is affected. This demonstrates the need for further research to increase the classification performance by updating the labeling based on the specifics of interference test cases.



Signals	Jamming		Spoofing		Meaconing	
GAL_E1	8256	1740	9884	157	3621	200
	195	58507	279	58378	474	64403
GAL_E5a	6499	856	8636	179	0	0
	582	60761	154	59729	0	68698
GPS_L1	9143	1225	10552	211	3609	212
	647	57683	218	57717	475	64402
GPS_L2	8958	388	8971	144	3611	210
	709	58643	164	59419	471	64406

🟩 True Positives 🟥 False Negatives ⬜ False Positives 🟦 True Negatives

Figure 7: Confusion Matrices for 4 different interfered signals when using randomly split dataset.

Nonetheless, the presented deep learning approach already classifies interference efficiently not only in specific scenarios but also in general, which is depicted in Fig. 8 where the micro metrics are shown, which combine all labels. The area under curve (AUC) for the receiver operating curve (ROC) of 0.99 and the high precision and recall values (Micro-AP = 0.952) show the effective identification of interfered signals.

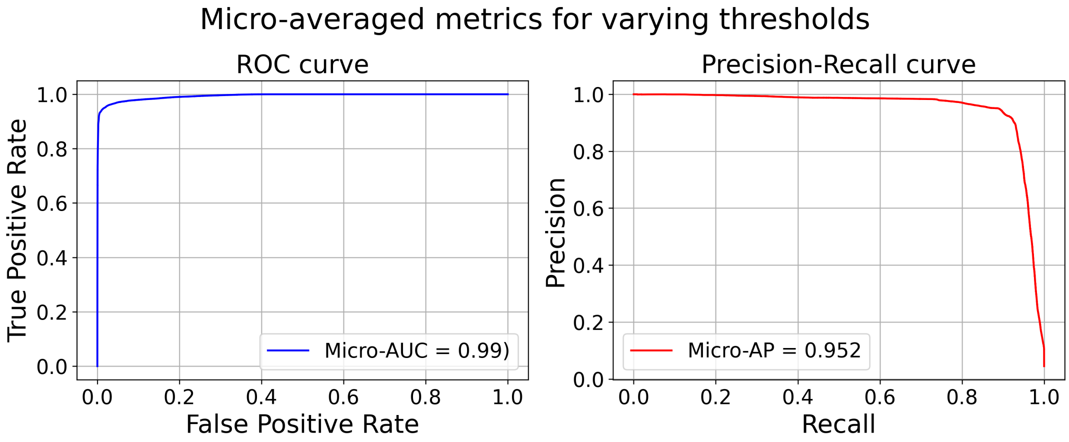


Figure 8: Micro metric combining all labels: receiver operating curve (ROC) and precision-recall (PR) curve for the random split and the labeling approach that distinguishes between interference types.

To see the classification performance for some representative signals, the confusion matrices with true and false positives along with true and false negatives as in Figure 7 show the prediction of the model for the individual interference types. The remaining 12 signal types interfered with jamming, spoofing and meaconing had similar values. It can be seen that the model trained and evaluated on randomly divided train/test dataset performs better with less number of false negatives when detecting spoofing and meaconing compared to jamming.

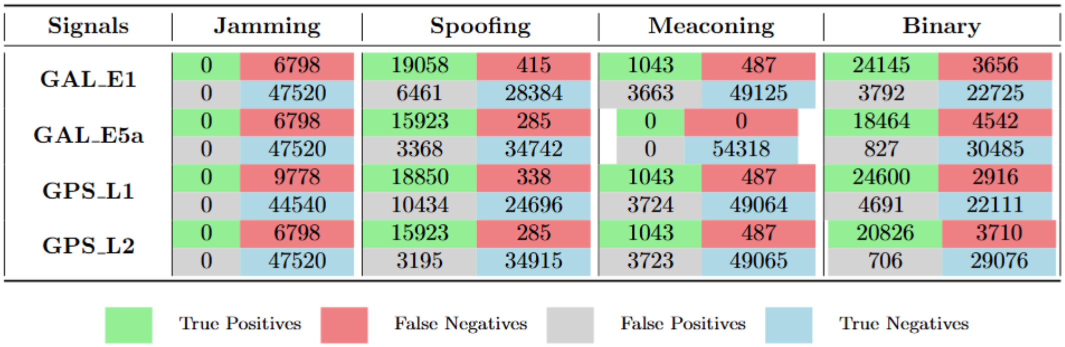
Using the random train and test data split, the model achieves precision and recall values exceeding 0.9 across the various GNSS signals and interference types, as summarized in Table 2 for representative signals including GAL_E1, GAL_E5a, GPS_L1, and GPS_L2. These results demonstrate that the network can reliably distinguish between no interference, spoofing, jamming, and meaconing for different GNSS signals when the training and test distributions are highly similar.

Figure 9 and Table 3 with confusion matrices and performance metrics respectively correspond to the model trained and evaluated on datasets with measurements from different days.
Converse to when the evaluation was performed with a temporally independent split, using data from the last day as the test set, the network's ability to distinguish between specific interference types decreases noticeably. This can be mainly seen in the bad performance of Jamming detection. However, the model remained robust in detecting whether a GNSS signal was affected by any interference, i.e., in the binary mode. When the output categories were grouped into a binary classification of interference versus no interference, the network achieved precision and recall values close to and exceeding 0.9, demonstrating that it retains

Table 2: Per-signal precision, recall, and F1-score for spoofed, jammed and meaconed signal detection when using randomly split dataset.

Signal	Spoofed			Jammed			Meaconed		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
GAL_E1	0.97	0.98	0.98	0.98	0.83	0.89	0.88	0.95	0.93
GAL_E5a	0.98	0.98	0.98	0.92	0.88	0.90	–	–	–
GPS_L1	0.99	0.99	0.99	0.93	0.88	0.91	0.88	0.94	0.91
GPS_L2	0.98	0.98	0.98	0.93	0.96	0.94	0.89	0.95	0.91

strong discriminative power for interference detection at a coarser level. It shall be emphasized that the test cases in the test datasets where different in their nature. For example, the test data had test cases with combinations of initial jamming with spoofing which were not learned by the network. Nevertheless, based on the good metrics in the binary case, the network can adapt to attacks that are new from its perspective.



Signals	Jamming		Spoofing		Meaconing		Binary	
GAL_E1	0	6798	19058	415	1043	487	24145	3656
	0	47520	6461	28384	3663	49125	3792	22725
GAL_E5a	0	6798	15923	285	0	0	18464	4542
	0	47520	3368	34742	0	54318	827	30485
GPS_L1	0	9778	18850	338	1043	487	24600	2916
	0	44540	10434	24696	3724	49064	4691	22111
GPS_L2	0	6798	15923	285	1043	487	20826	3710
	0	47520	3195	34915	3723	49065	706	29076

🟩 True Positives 🟥 False Negatives ⬜ False Positives 🟦 True Negatives

Figure 9: Confusion Matrices for 4 different interfered signals when using an independent test dataset.

Table 3: Per-signal precision, recall, and F1-score for spoofed, jammed, meaconed and binary interfered signal detection when using an independent test dataset.

Signal	Spoofed			Jammed			Meaconed			Binary		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
GAL_E1	0.75	0.98	0.85	0.0	0.0	0.0	0.22	0.68	0.33	0.86	0.87	0.87
GAL_E5a	0.83	0.98	0.93	0.0	0.0	0.0	–	–	–	0.96	0.8	0.87
GPS_L1	0.64	0.99	0.78	0.0	0.0	0.0	0.22	0.68	0.33	0.84	0.89	0.87
GPS_L2	0.83	0.98	0.9	0.0	0.0	0.0	0.22	0.68	0.33	0.97	0.85	0.9

These results have two main implications. First, for downstream GNSS data processing, a binary classification into interference versus no interference is sufficient, since the key requirement is to detect whether a signal is disturbed rather than to identify the exact interference type. Second, the choice of data splitting strategy is critical, as random splits can lead to overly optimistic performance estimates, while independent test sets provide a more realistic assessment of generalization.

VI. CONCLUSION AND FUTURE WORK

This work presented a deep learning framework for multi-label GNSS interference detection based on features including GNSS measurements, signal characteristics, and IMU measurements. The presented deep learning framework consisting of a CNN encoder combined with an LSTM network classifies interference per GNSS signal.

The evaluation is performed with well-labeled physical data recorded at the Jammertest 2024. Using these datasets, we could demonstrate that the proposed scheme could effectively detect GNSS interference on a per-signal basis, allowing to remove only the affected signals selectively. Consequently, the positioning system can leverage from detailed information, maintaining a high accuracy and availability in the presence of GNSS interference.

Comparing two presented multi-label problems, with and without distinguishing between the interference types jamming, spoofing, and meaconing, the binary classification per signal performed superior when evaluating the performance on a testing day and testing scenarios that were not covered by the test data. These findings highlight the challenge for the network if the training data samples do not represent the full variability of spoofing, jamming, and meaconing attacks, which is difficult to achieve given the wide range of possible scenarios and the challenges of reproducing and labeling them. We could demonstrate that with the binary classification this generalization to unseen conditions is feasible. Thus, new or more complex attack strategies, particularly in spoofing, might still be well detectable by the presented neural network.

Moreover, we could highlight challenges that arise due to the nature of different interference types such as power ramps and sweeps in jamming scenarios. The latency of the receiver when reacting to the interference affects the labeling and thus, the training process. This will be investigated in future research. Additional future work includes the analysis focusing more on dynamic data and the associated challenges to know when a label should be changed because of occlusion from the jammer or spoofer due to buildings or other parts of the environment. Ultimately, comparisons on accuracy, availability and more metrics of the positioning algorithm when making use of the detection scheme are also planned.

ACKNOWLEDGEMENTS

The authors express their sincere gratitude to the European Union and the European Union Agency for the Space Programme (EUSPA) for funding the DREAM project under the Fundamental Elements mechanism and for their valuable guidance and support throughout the project.

Disclaimer: However, the views and opinions expressed are those of the authors alone and do not necessarily reflect those of the European Union or the EUSPA. Neither the European Union nor the EUSPA can be held responsible for them.

REFERENCES

Albright, A., Powers, S., Bonior, J., and Combs, F. (2020). Oak ridge spoofing and interference test battery (oakbat) - gps.

Borhani Darian, P., Li, H., Wu, P., and Closas, P. (2020). Deep neural network approach to detect gnss spoofing attacks.

Czech, D., Mishra, A., and Inggs, M. (2018). A cnn and lstm-based approach to classifying transient radio frequency interference. *Astronomy and Computing*, 25:52–57.

Felux, M., Fol, P., Figuet, B., Waltert, M., and Olive, X. (2024). Impacts of global navigation satellite system jamming on aviation. *NAVIGATION: Journal of the Institute of Navigation*, 71(3).

Feng, Z., Seow, C. K., and Cao, Q. (2022). Gnss anti-spoofing detection based on gaussian mixture model machine learning. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3334–3339.

Humphreys, T., Bhatti, J., Shepard, D., and Wesson, K. (2012). The texas spoofing test battery: Toward a standard for evaluating gps signal authentication techniques. 5:3569–3583.

Islam, S., Bhuiyan, M. Z. H., Liaquat, M., Pääkkönen, I., and Kaasalainen, S. (2024). Fgi's gnss spoofing dataset repository (fgi-spoofrepo). `https://doi.org/10.23729/7a648509-2ca8-4a7d-8223-0b429182f857`. National Land Survey of Finland, FGI Dept. of Navigation and positioning.

Janiar, S. B. and Wang, P. (2024). Intelligent anti-jamming based on deep reinforcement learning and transfer learning. *IEEE Transactions on Vehicular Technology*, 73(6):8825–8834.

Manesh, M. R., Kenney, J., Hu, W. C., Devabhaktuni, V. K., and Kaabouch, N. (2019). Detection of gps spoofing attacks on unmanned aerial systems. In *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–6.

Meng, L., Yang, L., Ren, S., Tang, G., Zhang, L., Yang, F., and Yang, W. (2021). An approach of linear regression-based uav gps spoofing detection. *Wirel. Commun. Mob. Comput.*, 2021:5517500:1–5517500:16.

Mohanty, A. and Gao, G. (2024). A survey of machine learning techniques for improving global navigation satellite systems. *EURASIP Journal on Advances in Signal Processing*, 2024.

Montgomery, P. Y., Humphreys, T. E., and Ledvina, B. M. (2009). Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil gps spoofer. In *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, pages 124–130, Anaheim, CA. The Institute of Navigation.

OPSGROUP GPS Spoofing Workgroup (2024). Gps spoofing - final report of the 2024 gps spoofing workgroup. Technical report, OPSGROUP. Accessed 2025-08-28.

Shafiee, E., Mosavi, M., and Moazedi, M. (2017). Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency gps receivers. *Journal of Navigation*, 71:1–20.

Shafique, A., Mehmood, A., and Elhadef, M. (2021). Detecting signal spoofing attack in uavs using machine learning models. *IEEE Access*, 9:93803–93815.

Wu, S., Li, Y., Wang, Z., Tan, Z., and Pan, Q. (2023). A highly interpretable framework for generic low-cost uav attack detection. *IEEE Sensors Journal*, 23(7):7288–7300.

Xu, J., Ying, S., and Li, H. (2020). Gps interference signal recognition based on machine learning. *Mobile Networks and Applications*, 25.

Zhang, J., Cui, X., Xu, H., and Lu, M. (2019). A two-stage interference suppression scheme based on antenna array for gnss jamming and spoofing. *Sensors*, 19(18).

Zhao, Y., Shen, F., Qi, B., and Meng, Z. (2022). Doa estimation under gnss spoofing attacks using a coprime array: From a sparse reconstruction viewpoint. *Remote Sensing*, 14(23).