POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Deep Neural Network Approach for Classification of GNSS Interference and Jammer

Original
A Deep Neural Network Approach for Classification of GNSS Interference and Jammer / Mehr, I.E., Dovis, F.. - In: IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS. - ISSN 0018-9251. - STAMPA. - (2025), pp. 1-18. [10.1109/taes.2024.3462662]

Availability:
This version is available at: 11583/2992616 since: 2024-09-19T10:01:34Z

Publisher:
IEEE

Published
DOI:10.1109/taes.2024.3462662

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

A Deep Neural Network Approach for Classification of GNSS Interference and Jamming

IMAN EBRAHIMI MEHR, Member, IEEE
FABIO DOVIS, Member, IEEE
Politecnico di Torino, Torino, Italy

Global navigation satellite systems (GNSSs) are one of the most important infrastructures in the modern world for positioning and timing, also enabling many critical applications that require the reliability of the received signals. However, it is well known that the power of the GNSS signals at the receiver's antenna is extremely weak, and radio-frequency interference affecting the GNSS bandwidths might lead to reduced positioning and timing accuracy or even a complete lack of the navigation solution. Therefore, in order to mitigate interference in the GNSS receivers and guarantee reliable solutions, interference classification becomes of paramount importance. This article proposes an approach for the automatic and accurate classification of the most common interference and jammers based on the use of convolutional neural networks (CNN). The input for the network is the time-frequency representation of the received signal, together with features in the time and frequency domains. The time-frequency representation is obtained using both the Wigner-Ville and the short-time Fourier transforms. Moreover, the performance of the proposed method is compared using two different CNN architectures, AlexNet and ResNet. The effectiveness of the method is shown in two case studies: monitoring and classification by a terrestrial station and from a low Earth orbit (LEO) satellite. The results reveal that the proposed method achieves a high accuracy of 99.69% in classifying interference, even with low interference power, and can be implemented as a real-time tool for monitoring jammers.

Received 3 February 2024; revised 19 May 2024 and 27 July 2024; accepted 7 September 2024. Date of publication 17 September 2024; date of current version 14 April 2025.

DOI. No. 10.1109/TAES.2024.3462662

Refereeing of this contribution was handled by J. Dunik.

This work is part of the project NODES which was supported by the MUR M4C2 1.5 of PNRR, funded by the European Union - NextGenerationEU under Grant Agreement ECS00000036. The Ph.D. work of Iman Ebrahimi Mehr was supported by the grant DOT1332092 CUP E11B21006430005 funded within the Italian Programma Operativo Nazionale (PON) Ricerca e Innovazione 2014-2020, Asse IV "Istruzione e ricerca per il recupero" con riferimento all' Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell' innovazione" e all' Azione IV.5 "Dottorati su tematiche green" DM 1061/2021.

Authors' address: Iman Ebrahimi Mehr and Fabio Dovis are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy, E-mail: (iman.ebrahimi@polito.it; fabio.dovis@polito.it). *(Corresponding author: Iman Ebrahimi Mehr.)*

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

I. INTRODUCTION

The number of global navigation satellite systems (GNSSs) users is increasing enormously due to the high accuracy and precision of position determination and timing. Additionally, the widespread availability and free access to most GNSS signals have further fueled this surge in users. The widespread use of GNSS-based positioning and timing services includes several applications for which the availability and continuity of such services are of paramount importance. However, it is well known that the received power of GNSS signals at the receiver antenna is very weak, thus they are vulnerable to radio frequency interference (RFI) [1]. RFI can be unintentionally caused by signals broadcasted from other telecommunications infrastructures with carrier frequencies within the GNSS band, like distance measuring equipment (DME) and tactical air navigation (TACAN) or by higher order harmonics and intermodulation outside the GNSS band, such as digital TV channels [2]. Intentional RFI is instead based on signals transmitted by personal privacy devices (PPD) to disrupt GNSS receivers by purpose. This type of interference can take various forms, including spoofing, meaconing, or jamming, with spoofing aiming at transmitting counterfeit yet plausible GNSS signals to induce false position, velocity, and time (PVT) estimations at the targeted victim receiver [3]. Meaconing typically involves acquiring genuine navigation signals by capturing and retransmitting them, leading receivers to inaccurately determine PVT [4], [5]. On the other hand, jamming refers to transmitting radio frequency (RF) signals to disturb the receiver's RF chain. GNSS jammers can be implemented by transmitting frequency-modulated tones of a proper power level, spanning the entire GNSS band in a specific area [6].

The intense power of interference or jamming signals might completely saturate the initial stages of a GNSS receiver front-end, in which the receiver can be blinded and stop functioning, often due to intentional jammers. While in many cases, intermediate or low power interferences degrade the PVT estimation performed by a GNSS receiver, thus threatening the applications relying on it. These power levels are particularly dangerous, as they sometimes go undetected by the receiver [2]. The potential impact of RFI includes being a source of error for the signal processing stages of receivers and increasing the error of code and phase measurements. Consequently, this can lead to biases and additional noise for the constructed pseudoranges [7]. Therefore, in order to achieve high reliability and prevent a complete lack of positioning, GNSS applications need to detect interferences and further classify the types of interference for mitigation. Early detection and detailed

RFI characteristic analysis are crucial to implementing effective mitigation strategies. For instance, techniques such as pulse blanking for narrow sweep chirps [8] and pulsed interference [9], adaptive notch filtering (ANF) for continuous wave and sawtooth chirps [10], [11], and robust filtering for tick chirps have proven effective [12]. The ANF has demonstrated effective performance against continuous wave and chirp signals. However, its effectiveness heavily relies on its configuration, which needs adjustment based on the type of interference encountered, highlighting the importance of accurate interference classification [6].

In recent years, there has been considerable interest in the use of machine learning (ML) methods in the GNSS context. The authors in [13] and [14] presented some applications in GNSS that used ML to provide a new solution or service. Generally speaking, any ML method builds a model based on an appropriate set of features that can be used to achieve the desired classification goal [15]. Limiting our interest to RFI detection and classification, ML introduces a new paradigm for detecting and classifying GNSS threats, such as the presence of interference, spoofing, scintillation events, and multipath. This study aims to introduce an effective method that utilizes a convolutional neural network (CNN) to classify RFI signals in an automatic way, improving accuracy with respect to other similar methods presented in the literature. The proposed method works at the precorrelation stage, and the input is the stream of raw samples from a radio front-end. This pipeline enables real-time interference monitoring and classification for the GNSS monitoring station [16], which provides intermediate-frequency (IF) and in-phase/quadrature (IQ) raw sample. The inputs to the CNN are the visual representation of the raw sample in the time-frequency domain, together with the statistical parameters and metrics derived from the raw samples in the time and frequency domain. The preliminary work [17] focused on the time-frequency representation of raw samples only. While in this article, the CNN model is aided by the extracted features to improve the performance, particularly in classifying low-power interference. This article investigates the classification and characterization of 16 different types of RFI, namely, the 11 most common chirp signals found in [18] and [19], frequency hopping (FH) jammers, continuous wave interference (CWI) [or amplitude modulation (AM) jammer], narrowband jammers, frequency modulation (FM) jammers, and pulsed interference (DME-like). Moreover, the short-time Fourier transform (STFT) and the Wigner-Ville transform (WVT) are the two studied transformations to analyze the signal in the time-frequency domain. In addition, the performance of two different CNN architectures, AlexNet and ResNet, is evaluated and discussed in terms of accuracy, F1-score, inference time, and the size of each model on disk. The effectiveness of the proposed method is demonstrated in two case-study scenarios: the monitoring and classification by a terrestrial station and from a low Earth orbit (LEO) satellite. To summarize, the key contributions of this article are as follows.

1) The classification of a wider variety of interference signals, specifically covering 16 different types, by means of an efficient CNN in terms of performance.
2) Demonstrating the effectiveness at identifying low-power RFI signals that might still affect positioning performance.

The rest of this article is organized as follows. Section II reviews the related works. Section III describes the proposed methodology. Section IV introduces the two case-study scenarios, training phase, and evaluation metrics. Section V reports the simulation result and Section VI concludes this article. Moreover, Appendix A provides detailed mathematical models and descriptions of the GNSS interferences studied in this article. Appendix B offers a summary and background on how CNNs operate on input images.

II. RELATED WORKS

Given that the presence of interferences disrupts GNSS receiver operation, successful threat detection is relatively easy to achieve. However, if an effective countermeasure has to be applied for interference mitigation, their characteristics must be assessed in terms of time and frequency behavior, not just as an ON–OFF presence. This is a typical *classification* task that can be performed using traditional techniques or ML-based methods thanks to the increased computational capabilities of GNSS receiver processors.

A. Traditional Techniques

In general, traditional interference classification requires multiple techniques and an "interference alert" mechanism to integrate and unify the input from different methods for a confirmed decision on the presence and harmfulness of detected interference. This is because traditional detection techniques are tailored to specific interference characteristics, balancing sensitivity to a particular type against generality for all types. Typically, the RFI detection techniques are based on recognizing the deviations from normal conditions (distortions) caused by interference affecting the GNSS antenna [2]. These techniques can be performed using measurements from different stages of receivers, which can generally be categorized into two groups: precorrelation and postcorrelation techniques. Several techniques have been proposed in the literature, such as automatic gain control (AGC) monitoring [20], time-domain statistical analysis [21], as well as signal monitoring using transformed domain techniques (spectral monitoring [22] or time-frequency analysis [23], [24]) that work in the front-end stage of the GNSS receiver (precorrelation techniques). Monitoring the power density ratio between carrier and noise (C/N_0) [25], statistical analysis of the correlation result [26], pseudorange and PVT accuracy monitoring [2] are other approaches to detect RFI in the postcorrelation stage. The postcorrelation techniques may be less effective in situations with significant interference power because they depend on the precondition of typical signal acquisition and tracking [2]. Furthermore, some of the mentioned

techniques, such as AGC or C/N_0 monitoring, raise an alarm for detected power in every case of interference. For example, with no differentiation between a chirp signal or a continuous wave as interference; therefore, they may not be useful for classification tasks.

The traditional techniques utilize maximum likelihood for classification [27], [28] and demand significant human involvement in design, requiring the development and fine-tuning of multiple classification stages for each type of interference [29]. However, traditional techniques may be limited when it comes to automatically classifying interference types, especially when dealing with chirp signals due to their diversity. For example, as noted in [18], characterizing chirp signals may show similarities in the time or frequency domains, but the distinctive swept-frequency nature of chirp signals can be identified only through time-frequency analysis. Classifying different types of chirps typically necessitates postprocessing and human-driven analysis, integrating multiple facts from various domains [19]. In addition, as software-defined radio techniques advance, there is a concern about more complex jamming attacks emerging, such as smart jamming, which can dynamically change jammer type during attacks [30]. Hence, automated classification of different kinds of interferences, especially chirp signals, using traditional techniques is challenging due to their diversity.

B. Machine Learning Techniques

Leveraging ML algorithms can facilitate the development of a unified model capable of learning intricate patterns of interference signals and adapting to dynamic interference scenarios. ML-based methods have been recently proposed for the classification of different kinds of threats to GNSS signals, such as jamming, spoofing, and multipath, with a unified model [31], [32], [33], [34], [35], [36]. Since our proposed methodology focuses solely on jamming and unintentional interference, excluding spoofing and multipath, we do not delve into the details of these studies. Several studies have addressed the use of ML for RFI detection and classification, processing the time series of digital samples at the output of the receiver front-end. Qin and Dovis [19] investigated K-nearest neighbor algorithms for classifying chirp jamming signals as reported in [18] using the features extracted from the received signal at the precorrelation stage as input of the model, with an overall accuracy of 92.6%. Xu et al. [37] extracted the combined entropy feature of three types of interference, namely FM, AM, and phase modulation (PM). They then implemented support vector machine (SVM) and random forest (RF) for the classification task and achieved an average accuracy of 90%. There have been similar studies that focus on extracting features from the raw samples of received signals or raw observation (feature-based approaches) and utilizing ML models for classification tasks [38], [39]. Different from previous mentioned studies, Ferre et al. [40] proposed a method for classification by analyzing spectrogram images from five different types of jammers (AM jammers, FM jammers, linear chirp jammers, pulse jammers, and narrowband jammers). They performed SVM and CNN to do the classification task with an accuracy of 94.90% and 91.36%, respectively. Similarly, Swinney and Woods [41] performed a CNN for feature extraction from a concatenation image of the spectrogram, power spectral density (PSD), histogram, and raw IQ constellation of jamming signals (the same types of jammers as [40]). Next, SVM, RF, and logistic regression are evaluated for classification, with SVM having the best accuracy of 98%. Ferre et al.[40] investigated the use of federated learning for training the CNN model proposed by, achieving an accuracy of 93%. Recently, the authors in [42] and [43] proposed a multistage framework processing time series (C/N_0, number of satellites and, etc.) and spectrogram data separately. They performed a late and intermediate fusion of the features extracted from the spectrogram by a CNN architecture (ResNet18) and the time series by the TS-Transformer. With an accuracy rate of 95.32%, they classified six different types of interferences.

The higher power interfering signals are relatively easy to detect by the methods of ML, but the performance of the GNSS receiver might be degraded even if the interfering signal has low power and is hidden under the noise. However, in such a case, it is not easy to detect and classify it. The mentioned studies do not specifically address low power of interference, being of course more concerned with more disruptive signals. Nevertheless, robust algorithms with higher sensitivity are essential in identifying low power RFI for critical and high-end GNSS applications. Furthermore, the other key question is the computational complexity of such algorithms in terms of resources (CPU and RAM) and the time-to-detect for identifying the threat, which depends on the preprocessing of raw samples and the time required by the model to make a classification decision. Moreover, the authors in [40], [41], and [42] used two different ML models, one for feature extraction and the other for classification, which means higher complexity of the algorithm since each model requires separate tuning for optimal performance. From the mentioned research, it can be seen that the algorithms of ML, including the traditional ML and deep neural networks or their concatenation, have introduced novel approaches for detecting and classifying interference. It is also shown that neural network methods provide the best results among all ML methods, thus justifying their investigation in the framework of this work.

III. METHODOLOGY

Intentional interference by jammers or unintentional interference can adversely affect GNSS receivers in a given operational area, and may make them inoperable. Chirp signals commonly used by jammers behave differently in terms of time and frequency characteristics, and proper time-frequency analysis of the signals allows the detection and classification of different kinds of jamming signals. An appropriate time-frequency representation (TFR) can provide temporal and spectral information simultaneously [44]. Since the TFR of signals can be stored as images, the

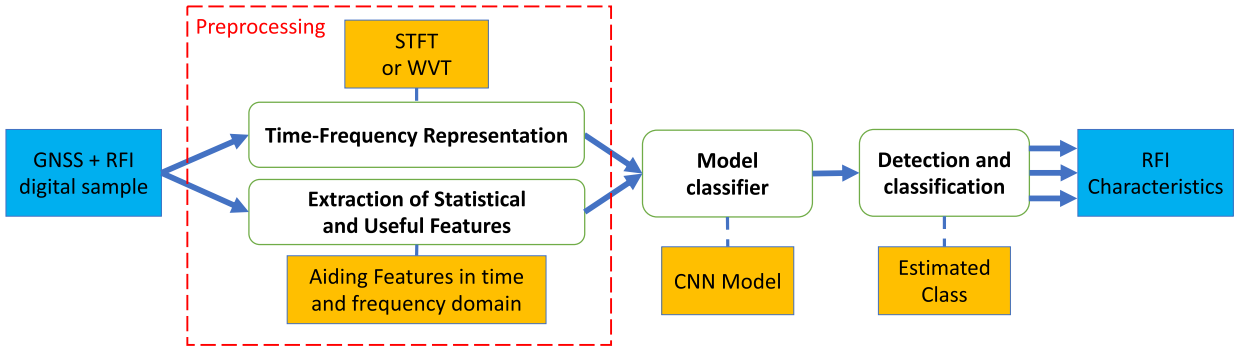


Fig. 1. Proposed methodology—feature-aided CNN classifier.

proposed method for categorizing different types of interference relies on techniques used in the field of ML-based image classification. Accordingly, the interference classification task is transformed into an image classification, where a model is trained to recognize the target classes of pixels or vectors within an image. The image classification task can be addressed using ML models such as decision trees, RF, and SVM, where hand-crafted features are manually derived from the images and subsequently utilized as input for the classification model. The challenge with this type of model lies in selecting the most important features for each specific image, and as the number of classes to classify grows, feature extraction becomes increasingly cumbersome [45]. In response to these challenges, deep learning algorithms, especially CNN, automatically extract the most salient and informative features for each specific class [46]. Therefore, in this work, a CNN-based method is chosen as the model classifier, which is the most popular deep learning method. The CNN models are widely used in the domain of image classification and have achieved state-of-the-art results in computer vision tasks [47], [48], [49]. Nevertheless, when dealing with low-power interference, the signal might be masked by the noise, and it is not easy to find a pattern in TFR as an image by ML methods. To overcome this limitation, other relevant signal features in the time and frequency domains are derived to aid classification. This article discusses the effectiveness of the proposed method named as *feature-aided CNN classifier*.

A. Feature-Aided CNN Classifier

As explained earlier, the method is performed at the precorrelation stage, and its concept block diagram depicted in Fig. 1. The input data comprise a digitized version of the received GNSS signal with possible interference that the explanation of its simulation will be provided in Section IV. The following step involves preprocessing and consists of two distinct phases, aimed at extracting pertinent information for the model classifier through the analysis of the input signal across time, frequency, and time-frequency domains. First, the input signal is examined within the time-frequency domain, resulting in the creation of a TFR that is preserved as an image. This transformation helps capture the dynamic spectral content of the signal over time, providing a detailed view of both the GNSS signal and potential interference. Within this framework, two distinct transformations are employed and investigated, and their specifics will be explained further in Section III-B. Second, acting with the identical time series of input signals, some statistics and metrics, called aiding features, are derived from both the time and frequency domains. In the time domain, common statistical values such as the mean, median, standard deviation, mean absolute deviation, root mean square, 25th percentile, 75th percentile, and interpercentile range are used to measure the dispersion and central tendency of the raw samples. These features offer insights into the overall behavior and variability of the received signal. Additionally, higher order statistics like skewness and kurtosis [50] are employed to measure the asymmetry and outlier-prone nature of the probability distribution, respectively. Entropy is included to quantify the uncertainty and randomness of the samples, providing information about the signal's complexity. In the frequency domain, the features focus on the power characteristics of the signal. Since the received power of GNSS signals is typically below the noise floor, it is not expected to have high power spectral components in the frequency domain in the presence of GNSS signals only. Therefore, any spectral component with significant power levels in the frequency domain is likely due to interference signals. The PSD then provides information about the location and bandwidth in the frequency of the RFI and its relative power level with respect to the GNSS signals. Mean power provides an average measure of the signal's power across all frequencies. These frequency domain features help differentiate different kinds of interference. To capture these frequency domain features, the PSD is computed using Welch's method [51] with the following parameters: the segment length (nfft) is set to 512, a boxcar window with the same length of nfft, zero overlap between segments, and the PSD is computed for both positive and negative frequencies. Additionally, the raw signal is mean-centered before the calculation to remove any dc offset. Table I presents selected features in the time and frequency domains that are employed to describe the characteristics of interferences. These features are widely and commonly used as, for example, in [19], [43], and [52]. The subsequent step after preprocessing involves feeding the TFR image to the CNN

TABLE I
List of Features in Time and Frequency Domains [19]

Domain	Features	Description
Time domain	Mean value	Common statistical values of raw sample to measure the dispersion of the samples
	Median value	
	Standard deviation	
	Mean absolute deviation	
	Root Mean Square	
	25th percentile	
	75th percentile	
	Inter percentile range	
	Skewness	Measure the asymmetry of the probability distribution.
	Kurtosis	Measure the outlier-prone of the probability distribution.
	Entropy	Measure the uncertainty and randomness of the samples.
Frequency domain	Frequency of max power	Frequency of the maximum power located.
	Maximum power	Maximum power obtained in frequency domain.
	Mean power	Mean power obtained in frequency domain.

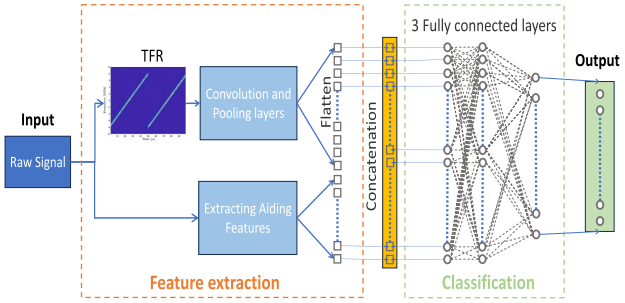


Fig. 2. Features fusion of proposed methodology.

for feature extraction. The CNN processes the TFR image to learn and extract high-level features that represent the signal characteristics. Following this, the extracted features from the CNN model are first flattened, which converts the 2-D feature maps into a 1-D feature vector. These flattened CNN features are then concatenated with aiding features derived from the time and frequency domains as illustrated in Fig. 2. This fusion of heterogeneous features generates a combined feature vector, which captures the temporal, spectral, and learned representations in time-frequency domain of the signal. The fully connected (FC) layers perform the final classification by processing the combined features and making predictions about the interference types. Finally, the raw output scores (logits) from the last FC layer go to a softmax layer [53], which provides the probabilities associated with each interference type as the output of methodology. Further details regarding CNN models and their architectures explained in Section III-C of this chapter.

B. Time-Frequency Representation as Image

Most of the effective jammers exhibit dynamic frequency behavior, where their instantaneous frequency changes over time such as chirp signals. Despite chirp signals possessing a periodic pattern, their frequency content varies, requiring a time-frequency representation to capture their characteristics. For the time-frequency analysis, STFT and WVT are investigated. Both of these transforms allow for finding a pattern of interference. For example, the linear chirp jammer, widely used in PPD jamming, can get various forms characterized by frequency sweeps across different bandwidths. These linear chirp signals can have either wide or narrow bandwidths and differ in sweep rate, categorized into four groups: slow, medium, fast, and rapid. Each type exhibits progressively faster and more pronounced frequency changes, with steeper slopes reflecting the increased transition rate that can be visualized in TFR.

1) *STFT:* It considers a window function of fixed length for segmenting the nonstationary signal into shorter segments (frames), and the Fourier transform of each frame is separately assessed [54]. The formula for STFT can be written as [55]

$$\mathrm{STFT}_x[n,k] = \sum_{m=0}^{M-1} x[n+m]w[m]e^{-j2\pi mk/N} \tag{1}$$

where n and k, respectively, denote time frame and the frequency bin indices, M is the length of each window segment, and $w[m]$ represents the window function. The spectrogram is a representation of the signal's power spectral density, which is derived from the squared magnitude of the STFT [56]. The STFT employs 128 FFT points in each analysis window, providing detailed frequency resolution. To ensure smooth transitions and continuity between consecutive analysis windows, the windows overlap by 99% of the FFT points, resulting in an overlap of approximately 127 samples. This high degree of overlap enhances the time resolution of the spectrogram by providing more frequent updates

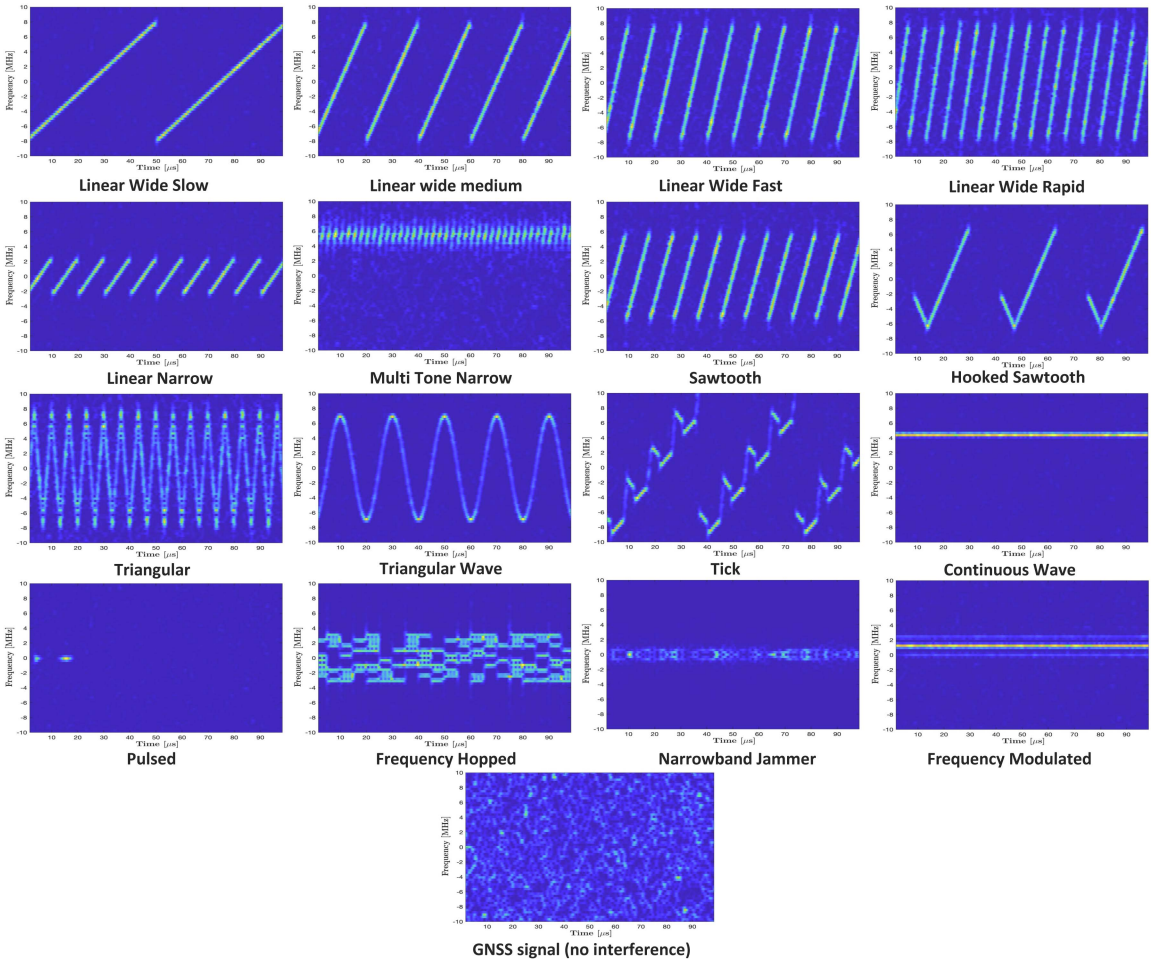


Fig. 3. Spectrogram of common interferences used in this context.

to the time-frequency representation, reducing artifacts, and ensuring a smoother and more continuous spectrogram. Fig. 3 provides an example of a spectrogram from each interference, where C/N_0 of the GNSS signal is 40 dBHz and the interference's power is -120 dBW.

2) *WVT:* There are several presentations in literature for the discrete WVT, the general form [57] for a signal $x[t]$ considering a window of M samples, can be written as

$$WV_x[n,k] = \sum_{m=-M}^{M} x\left[n+\frac{m}{2}\right] x^*\left[n-\frac{m}{2}\right] e^{-j2\pi mk/N} \quad (2)$$

where x^* denotes the conjugate of x, n, and k denote the index numbers for the time and frequency vectors, respectively. In order to avoid the cross-term effect in quadratic form, smoothed pseudo WVT (SPWVT) [58], [59] is used to analyze the signal, which uses independent window functions to smooth in time and frequency as

$$\mathrm{SPWV}_x[n,k] = \sum_{m=-M}^{M} h[m] \sum_{p=-P}^{P} g[p-n] x\left[p+\frac{m}{2}\right] x^*\left[p-\frac{m}{2}\right] e^{-j2\pi mk/N} \quad (3)$$

where $h[.]$ and $g[.]$ are arbitrary windows, and their shape and width will affect the smoothing performed in both frequency and time on WVT. The WVT provides a high-resolution representation in both time and frequency for nonstationary signals. Furthermore, it has the special properties of satisfying the time and frequency marginal in terms of the instantaneous power in time, energy spectrum in frequency, and total energy of the signal in the time and frequency plane [60].

C. Convolutional Neural Networks

This article evaluates the performance of two CNN architectures for classifying GNSS interferences: AlexNet (a classical architecture) and ResNet (a modern architecture) [61], [62]. The selection of these models is informed by their performance on the ImageNet dataset [63]. AlexNet is chosen for its superior inference time, while ResNet is preferred for its higher accuracy in classification tasks. The general architecture of both models remains largely unchanged, with a minor modification that adds aiding features to the architecture and replaces the FC layers with newly designed ones. Specifically, this modification involves combining the aiding features with the output of the final convolutional

layer. These concatenated features are then fed into three FC layers for classification. Each of these layers maintains the same input and output dimensions, except for the final layer, which matches the number of interference classes. The parameters of each model, such as weights and biases, undergo random initialization. The details regarding how CNN operates are provided in Appendix B and the main characteristics of both chosen networks are briefly summarized in the following.

1) *AlexNet:* AlexNet is one of the most studied and famous architectures designed by Alex Krizhevsky and achieved high accuracies in image classification. The input to AlexNet can be an image of size $227 \times 227 \times 3$, and the last dimension refers to the depth or channel of input [61]. AlexNet has eight layers of depth and 60 million parameters. It comprises five convolutional layers in which the activation function is ReLU, and max-pooling layers follow the first, second, and last one. AlexNet also utilizes dropout regularization in the FC layers to reduce overfitting. Dropout is a technique that randomly drops a fraction of neurons in a layer from the neural network during training [64].

2) *ResNet:* Recently, CNN achieved state-of-the-art results in solving complex tasks in computer vision by increasing the network's depth (layers' number) or the size and number of the kernel. Nevertheless, it has been observed that as the network goes deeper, accuracy gets saturated, the training is more complicated, and then the network tends to degrade [65]. ResNet, short for residual network, was presented by He et al. [62] to overcome these problems. The idea is to directly connect the input of a layer to further layers by skipping some layers of models, known as skip connection, shown in [62, Fig. 2]. There are variant versions of ResNet that follow the same concept but with different numbers of layers. Among them, ResNet-50 reported being suitable in terms of depth, training time, and accuracy. The ResNet-50 has 23 million parameters and 49 convolution layers, where the last one connects to average pooling layers.

IV. SCENARIOS, TRAINING PHASE, AND PERFORMANCE METRICS

A. Scenarios and Datasets

As previously introduced, the proposed technique is intended for the digitized version of received signals, at the output of a GNSS receiver's front-end. The front-end typically amplifies and filters the GNSS analog signal coming from the antenna to minimize out-of-band contributions. Then GNSS signal is downconverted to IF and passed to the analog to digital (A/D) conversion. Assuming a situation where the interference signal exists, the discrete-time signal at the IF after the front-end stage can be written as [2]

$$y_{\text{IF}}[n] = y_{\text{IF}}(nT_s) = \sum_{i=0}^{L-1} s_{\text{IF},i}(nT_s) + i_{\text{IF}}(nT_s) + w(nT_s) \tag{4}$$

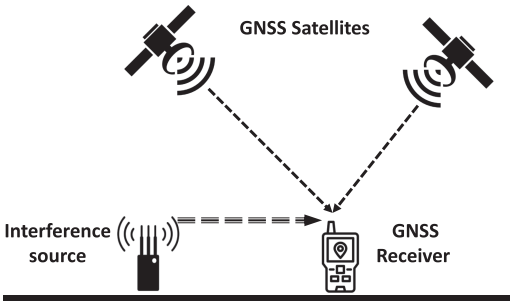


Fig. 4. Terrestrial interference monitoring scenario.

where $s_{\text{IF},i}(nT_s)$ is the received GNSS signal of interest from the lth satellite, $i_{\text{IF}}(nT_s)$ is the additive interference, $w(nT_s)$ are the samples of the complex additive white Gaussian noise (AWGN) (thermal noise), T_s is the sampling interval, and n is the discrete time index. The suggested methodology is being evaluated in two different scenarios: detecting and classifying interference on the ground and from space. These two scenarios are simulated in realistic conditions of signal-to-noise ratio for a set of possible interference ($i_{\text{IF}}[n]$).

1) *Interference Classification on the Ground:* Fig. 4 depicts the scenario of terrestrial interference monitoring, where the receiver remains static and receives both GNSS and interference signals at its antenna. The process of simulating GNSS and interference signals and the generation of datasets are described in the following.

GNSS signals: The GNSS signals in this scenario are from the GPS constellation and belong to the L1 band, with a central frequency of 1575.42 MHz. The C/N_0 for these signals is between 25 and 50 dBHz at the step of 1 dB where $N_0 = -202$ dBW/Hz (The signal power ranges from -177 to -152 dBW). The 25 to 50 dBHz range indicates that the quality of the signals ranges from bad reception to good one and may be susceptible to interference in different ways. The GPS signals are simulated utilizing FULL Educational Library of Signals for Navigation (N-FUELS) [66], a signal/disturbances generator. N-FUELS is implemented as a series of nonreal-time MATLAB scripts that can simulate samples of a GNSS signal as seen by the receiver after the ADC (front-end output). N-FUELS indeed take into account the effect of the front-end filter. However, possible in-band harmonics generated by the saturation effects that might be induced by the jammers' strong received power, are not modeled. The GPS signal is processed through an 8th-order Butterworth filter with a bandwidth of 20.46 MHz. These signals are subsequently sampled at a rate of $f_s = 40$ MHz in the I/Q components, with a 64-b floating point precision (double type: simulated floating-point signal with no ADC limitations).

Interference signals: GNSS signals are combined with interference signals to simulate a realistic scenario where disruption is present. This is achieved mathematically by adding the GNSS and interference signals together, as stated

in (4). The models and characteristic of the simulated interference are reported in Appendix A of this article. In this scenario, the interference power ranges from −142 dBW to −107 dBW at the step of 1 dB, resulting in an interference to noise ratio (INR) [67] of −13–22 dB. In this work, the term low power interference refers to the negative values of the INR. Considering the combination of C/N_0 for the GNSS signals between 25 and 50 dBHz (26 cases), and interference signal power varying between −142 and −107 dBW (36 cases), different conditions of interference-to-signal ratios (also known as jamming-to-signal ratio J/S) ranging from 10 to 70 dB are considered in this scenario. Furthermore, in the Monte Carlo analysis, at each step of the interference-to-signal ratio, two other factors are considered for interferences as follows.

1) Considering real-world scenarios where the center frequency of interference may vary with respect to the center frequency of GNSS bands, we have analyzed different interference center frequencies, randomly chosen from −9 to 9 MHz relative to the center frequency of GPS L1 (5 cases).
2) Different cases (7 cases) of the initial relative phase between the interference and GNSS signal are simulated.

Datasets: Two datasets are generated for this scenario, each representing a different time-frequency analysis (STFT and WVT). These datasets are generated using a Monte Carlo analysis, combining various GNSS signals in terms of C/N_0 with interference signals at various power levels, center frequencies, and initial relative phase. This provides a realistic classification scenario and ensures that all possible cases where the GNSS signal is at high power and interference power is low and vice versa are included. Each dataset consists of 17 classes, of which one refers to the GNSS signal with no interference, while the other 16 classes represent GNSS signals in the presence of various types of interference. Each class of dataset comprises $N_{\mathrm{MC}} = 32\ 760$ elements that were determined by combining different parameter values as

$$N_{\mathrm{MC}} = \alpha \times \beta \times \gamma \times \delta \tag{5}$$

where α represents the number of GNSS signal power cases, with 26 different power levels considered. β is the number of interference power cases, which is 36. γ is the number of randomly chosen center frequencies, with five different values used. Lastly δ is the number of cases to have different initial relative phase, set to 7. Each element in the dataset corresponds to a unique combination of these four parameters, which comprises a TFR calculated over a 100 μs snapshot of raw signal samples (IQ) and statistical features extracted over a window of 200 μs.

Additionally, within the scenario of terrestrial interference monitoring, a lower number of bits quantization has been investigated. These scenarios are simulated under identical parameters explained so far using STFT as the TFR technique and taking into account the impact of ADC quantization employing 8 and 4-b for both the in-phase and quadrature components of the signal.

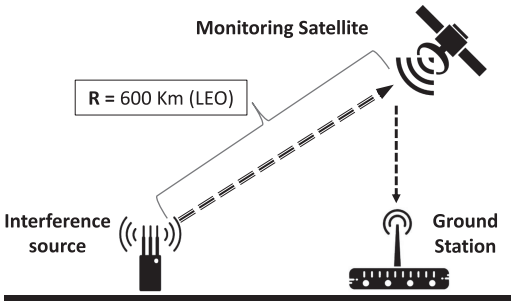


Fig. 5. Space-based interference monitoring scenario.

2) *Interference Classification From the Space:* Recent works address the idea of monitoring the presence of GNSS interference from space [68], [69], [70], as a promising way to cover large areas. It refers to the process of detecting and monitoring disruptive signals that can impact the performance of GNSS from a space-based perspective. Moreover, the classification of detected interference is needed for further investigation and provides more details related to their frequency and time domain.

Fig. 5 illustrates the simulation of this scenario, and the main distinction of this scenario from the prior scenario is that the antenna on-board of the satellite is solely receiving the interference signal and background noise. The architecture of this monitoring system consists essentially of a NADIR antenna, a radio front-end (RFE), and a software processing unit. The NADIR antenna is pointed toward the Earth's surface to grab potential interference generated on the ground. The RFE is responsible for converting the signal from analog to digital. It first amplifies and filters the analog signal and then downconverts to an intermediate frequency to allow digital conversion. Further details regarding the architecture can be found in [70].

Relying on the limitations and a scarcity of resources on-board of satellite, the raw samples of the received interfering signals may be either processed on-board or transmitted to a ground station for implementation of the classification algorithms. Depending also on the storage resources and the downlink channel capacity in case of remote processing, the RFE bandwidths (B_{IF}) and the number of bits of the ADC might be constrained. Therefore, three possible B_{IF} of 5, 10, and 20 MHz are evaluated in this scenario. To simulate the received signal on-board of the satellite, the interference signals are generated with a sampling frequency of $f_s = 2.2 \times B_{\mathrm{IF}}$ to have some margin against the Nyquist frequency. In addition, the interference signal is embedded in complex AWGN. The variance of the thermal noise over the bandwidth B_{IF} has been accounted for in the margin as well: $\sigma^2_{\mathrm{IF}} = N_0 \times B_{\mathrm{IF}}$ where N_0 is the power spectral density of the noise, estimated to be $N_0 = -205$ dBW/Hz. In this specific scenario, the minimum target power that can be detected for an interfering signal received by the LEO satellite has been set to −142 dBW. Within this context, three distinct datasets are generated by varying the chosen RFE bandwidth (5, 10,

TABLE II
Classification Result of Interference on the Ground

Dataset	Methodology	CNN Architecture	Accuracy [%] (Test dataset)	F1-score [%] (Test dataset)	Accuracy [%] (challenging dataset)	Inference time [ms]	Task Completion time [ms]
STFT	CNN	AlexNet	91.69	91.35	77.67	11.85	98.13
		ResNet	96.85	96.30	91.02	69.71	155.99
	Feature-aided CNN	AlexNet	96.41	96.12	90.19	13.11	99.39
		ResNet	99.69	99.57	99.46	71.39	157.67
WVT	CNN	AlexNet	95.20	94.80	88.15	12.47	2088.55
		ResNet	98.84	98.43	96.74	73.18	2149.26
	Feature-aided CNN	AlexNet	95.82	95.67	89.71	13.43	2089.51
		ResNet	98.87	98.49	96.85	73.39	2149.47

and 20 MHz). The interference power levels within each dataset range from -142 to -125 dBW, incrementing by one dB (18 different cases). Similar to the previous scenario, we have also considered the variation of interference center frequency in this scenario. Specifically, the interference center frequency has been randomly chosen (5 cases) between $-B_{\text{IF}}/2$ and $B_{\text{IF}}/2$, ensuring a comprehensive exploration of different interference conditions. For the Monte Carlo analysis, various combinations of initial relative phase (7 cases) and additional random AWGN (15 cases) with the same noise floor are simulated. Consequently, each dataset comprises 151 200 elements, evenly distributed among the different interference types ($18 \times 7 \times 15 \times 5 = 9,450$ elements for each class).

B. Training Phase

Since the training process of ML methods requires a lot of computational power, memory, and data storage, the training process is performed in the Google Colaboratory (Colab) environment [71]. The available resources for the training process include a single GPU core of Tesla T4 (Intel(R) Xeon(R) @ 2.00 GHz) and 13 GB of RAM memory. Further details of the training are as follows.

1) *Framework:* CNN models are developed using the TensorFlow library in the Python language. TensorFlow was released by Google in 2015 and is an open-source framework for building and deploying various types of ML and artificial intelligence.
2) *Splitting dataset:* The image datasets are divided into three subsets, and the training process uses 50% of the dataset. 15% of the dataset is utilized for tuning the parameters and finding the best model (validation dataset), while 35% is used for unbiased evaluation of the best model. For partitioning the dataset, the random state function controls the shuffle process, ensuring that all samples in the training, validation, and test datasets are the same for both CNN models.
3) *Optimization:* The training process uses the Adam optimization algorithm [72] to update the neural network parameters with a batch size of 64 examples, and the initial learning rate is 0.001. The adaptive learning rate is even considered, and if the performance does not improve with the current learning rate after two epochs, the learning rate is divided into two. The used loss function to quantify the dissimilarity or error between the predicted output and the actual ground truth is the categorical cross entropy (softmax loss) [73].
4) *Training cycles:* The epoch number for training is 25, and all samples in the training dataset pass through the model 25 times. In addition, the early stopping technique [74] is utilized (set at 3 epochs), which involves terminating the training process when there is no further improvement in performance on the validation dataset.

TABLE III
Confusion Matrix for Binary Classification

True label \ Predicted label		
	True positive (TP)	False negative (FN)
	False positive (FP)	True negative (TN)

C. Evaluation Metrics

The following evaluation metrics are used to measure the performance of the proposed methodology with different architecture of CNN.

1) *Confusion matrix:* It shows the number of correct and incorrect predictions for each class, which is counted to provide a summary of the results of a classification task. The confusion matrix demonstrates how the classification model makes predictions while it is confused. Table III shows a confusion matrix for binary classification.
   True positive (TP) is a correct positive prediction where the model correctly identifies a positive class. False negative (FN) is an incorrect negative prediction where the model fails to identify a positive class. False positive (FP) is a wrong positive prediction, where the model identifies a positive class, but it's actually negative. True negative (TN) is a correct negative prediction where the model correctly identifies a negative class.

TABLE IV
Performance of Feature-Aided CNN Classifier for the Space-Based Classification

RFE Bandwidth [MHz]	CNN Architecture	Accuracy [%] (Test dataset)	F1-score [%] (Test dataset)	Accuracy [%] (challenging dataset)	Preprocessing time [ms]	Task Completion time [ms]
5	AlexNet	91.33	90.65	80.25	65.11	78.22
	ResNet	94.70	94.43	89.32	65.11	136.5
10	AlexNet	93.07	92.82	83.55	77.6	90.71
	ResNet	97.76	97.43	93.07	77.6	148.99
20	AlexNet	98.47	97.99	96.32	89.05	102.16
	ResNet	99.21	98.89	98.44	89.05	160.44

2) *Overall accuracy:* Overall accuracy is determined by counting the total number of correctly classified samples and dividing by the total number of samples. It can be expressed as follows:

$$\text{Acc} = \sum_{i=1}^{n_c} \frac{\text{TP}_i + \text{TN}_i}{n_s} \tag{6}$$

where n_c is the number of classes, n_s is the total number of samples, and TP_i and TN_i denote true positive and true negative for class i, respectively.

3) *F1-score:* The F1-score's objective is to create a single metric by combining the precision (p) and recall (r) criteria and is defined as the harmonic mean of precision and recall

$$F1 = \frac{2 \times p \times r}{p + r} \tag{7}$$

where precision measures the proportion of correctly predicted samples out of the total number of predicted samples in a positive class ($p = \text{TP}/(\text{TP} + \text{FP})$) and recall measures the fraction of positive samples that are correctly classified ($r = \text{TP}/(\text{TP} + \text{TN})$) [75].

4) *Inference time:* The time that a trained model takes to make a decision based on input data is called the inference time or network latency. It is a critical metric for deploying a deep network in a real-world application.

5) *Task completion time:* Here, task completion time (also known in the literature as job completion time) refers to the time required by the entire methodology (shown in Fig. 1) to complete the task (from the beginning to the end). It is basically the sum of preprocessing and inference time.

V. SIMULATION RESULTS

The findings from the experiments, conducted following the methodology detailed in Section III and using the datasets described in Section IV, are presented in this section. To enable comparability between the various methodologies and the CNN architectures fairly and consistently, the same training dataset and test samples are used for all experiments. As remarked in Section III, a snapshot of 200 μs is used to get the inputs for the model (spectrograms and features), and inference and task completion time reported in Tables II and IV are based on one single input. The evaluation of these time metrics occurs on a local system equipped with an Intel Core i5 CPU running at 3 GHz and 4 GB of memory.

1) *Classification Result on the Ground:* Table II summarizes the results of the feature-aided CNN classifier and the CNN classifier presented in [17] for the two produced image datasets using STFT and WVT analyses. The accuracy results state that the use of statistical features of raw samples improves classification performance. For instance, within the STFT dataset, the accuracy of feature-aided CNN classifiers in both the AlexNet and ResNet models has seen improvements of 5% and 3%, respectively. In fact, the interference signals with low power are buried under the noise floor, and it is difficult for the model to detect them using only the time-frequency representation. However, the extracted features aid the model in predicting the correct class.

The accuracy of the challenging dataset in Table II refers to the methodology's performance for a dataset including only the low-power of interferences from −142 to −130 dBHz (negative values of INR) which were picked up from the test and validation dataset. These datasets are fed to the algorithm to better evaluate the models at lower power of interferences. From the results, the ResNet architecture utilizing the extracted features and STFT achieves the best performance with 99.46%. As an example of a challenging dataset, Fig. 6 illustrates a tick interference with the power of −142 dBHz where the ResNet model predicts the suitable class.

From the results in Table II, the feature-aided CNN classifier with ResNet architecture and STFT gains the best result in terms of accuracy for the test and challenging set. Fig. 7 shows the confusion matrix of this model for the test dataset. The confusion matrix shows that the model is able to detect the presence of interference signals with 99.99%, although there are some cases where the type of interference is misclassified. For instance, the model's accuracy for the triangular chirp signal is 98.4%, but in 183 cases (1.5%), the model misclassified with linear wide rapid chirp signal. Furthermore, the confusion matrix in Fig. 7 shows consistently high accuracy across all classes, with values exceeding 99%.

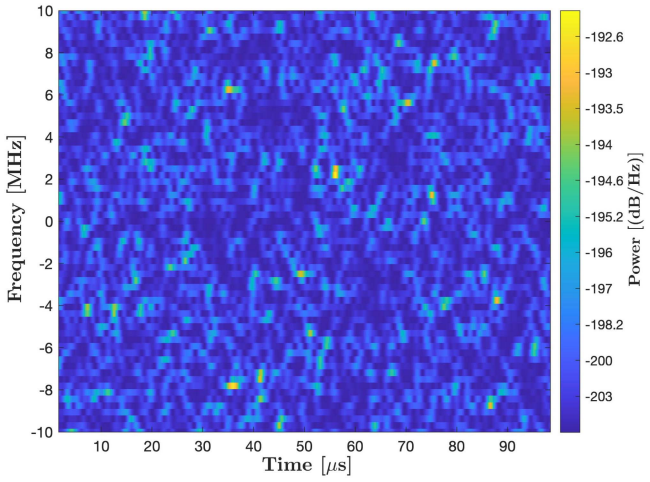


Fig. 6. Sample of tick interference where the model predicts an appropriate class.

This demonstrates the model's effectiveness in accurately classifying each type of interference.

Regarding the inference time of each CNN model, it is clear that more convolutional layers in the ResNet architecture lead to a longer inference time (71.39 ms) compared to the AlexNet architecture (11.85 ms). Nevertheless, on the other hand, ResNet has better performance, and finding the best model is a tradeoff between the inference time and accuracy. A comparative assessment of inference time is vital for understanding the computational complexity and resource utilization of our proposed method in various environments, particularly in resource-constrained scenarios such as satellite vehicles. We conducted measurements across various hardware architectures, on both CPU and GPU platforms. Specifically, the inference time for the ResNet model was recorded as 3.99 ms on the GPU and 71.39 ms on the CPU. What's more, the model of each architecture with the trained weights, parameters, and model configuration is stored in a singular file format (.h5) designed for structured data storage. The size on disk of the AlexNet model is 227 MB, while the ResNet model occupies 127 MB.

Moreover, the current study confirms the findings of our previous work [17] that the two transformations STFT and WVT have almost the same accuracy in classification. However, the task completion time shows that the computation of WVT takes more time compared to STFT. The preprocessing time with WVT and STFT is 2076.08 and 86.28 ms, respectively, in which feature extraction takes 0.08 ms. In addition, the findings pertaining to task completion time revealed that the proposed methodology using STFT could be suitable for real-time classification.

Table V provides a comparison of how the suggested approach based on STFT and ResNet architecture performs with different numbers of bits for ADC quantization. The results indicate that the overall accuracy in the case of 8-b still performs well, with only a modest reduction of 1%, while in 4-b quantization the performance is decreased by 5%. Furthermore, when employing 4-b quantization, we observed an 8% decrease in the model's accuracy for challenging dataset, highlighting the influence of reduced bit quantization on the classification of low-power interference signals.

TABLE V
Performance Evaluation With Bit-Quantization in Interference Classification

ADC quantization	Accuracy [%] (Test dataset)	F1-score [%] (Test dataset)	Accuracy [%] (Challenging dataset)
64-bit	99.69	99.57	99.46
8-bit	98.88	98.57	97.29
4-bit	94.47	94.20	90.92

In addition, as mentioned in Section IV-B, we used various techniques, such as early stopping and adaptive learning rates, which helped decrease the number of epochs needed for training, resulting in a much shorter training time. Additionally, performing the training process on the cloud using a GPU significantly decreased the training time by 17 times compared to using a CPU.

2) *Classification Result From Space:* Due to the superior performance observed in the previous scenario when employing the proposed methodology with STFT, our focus in this scenario is solely on evaluating the performance of the feature-aided CNN classifier using STFT. Consequently, we train and validate a total of six distinct ML models, with three datasets assigned to each bandwidth: 5, 10, and 20 MHz.

Considering that the inference time of the AlexNet (13.11 ms) and ResNet (71.39 ms) models remains nearly unchanged from the previous scenario, Table IV shows just preprocessing and task completion time. The results reveal that as the bandwidth increases, the sampling frequency also increases, leading to a rise in the preprocessing time. On the other hand, high accuracy is obtained when the selected RFE bandwidth is higher since the entire bandwidth of the interfering signal enters into the RFE.

Interestingly, the feature-aided CNN classifier still has good accuracy in the case of lower bandwidth. For instance, AlexNet architecture using the statistical features has an accuracy of 91.33% in the 5 MHz case, which is a significant improvement over the previous study, which was 80.18% [17].

VI. CONCLUSION

In this study, a comprehensive analysis method was employed to assess the time-frequency representation and the temporal and spectral characteristics of the GNSS signal. This analysis was utilized to identify and classify potential interference using a CNN model. ResNet as a CNN model has shown promising results with accuracy of 99.69% and thus could be a powerful tool in the field of interference classification. The main achievement of this study may be the effectiveness of the feature-aided CNN classifier methodology for classifying lower power of interference.

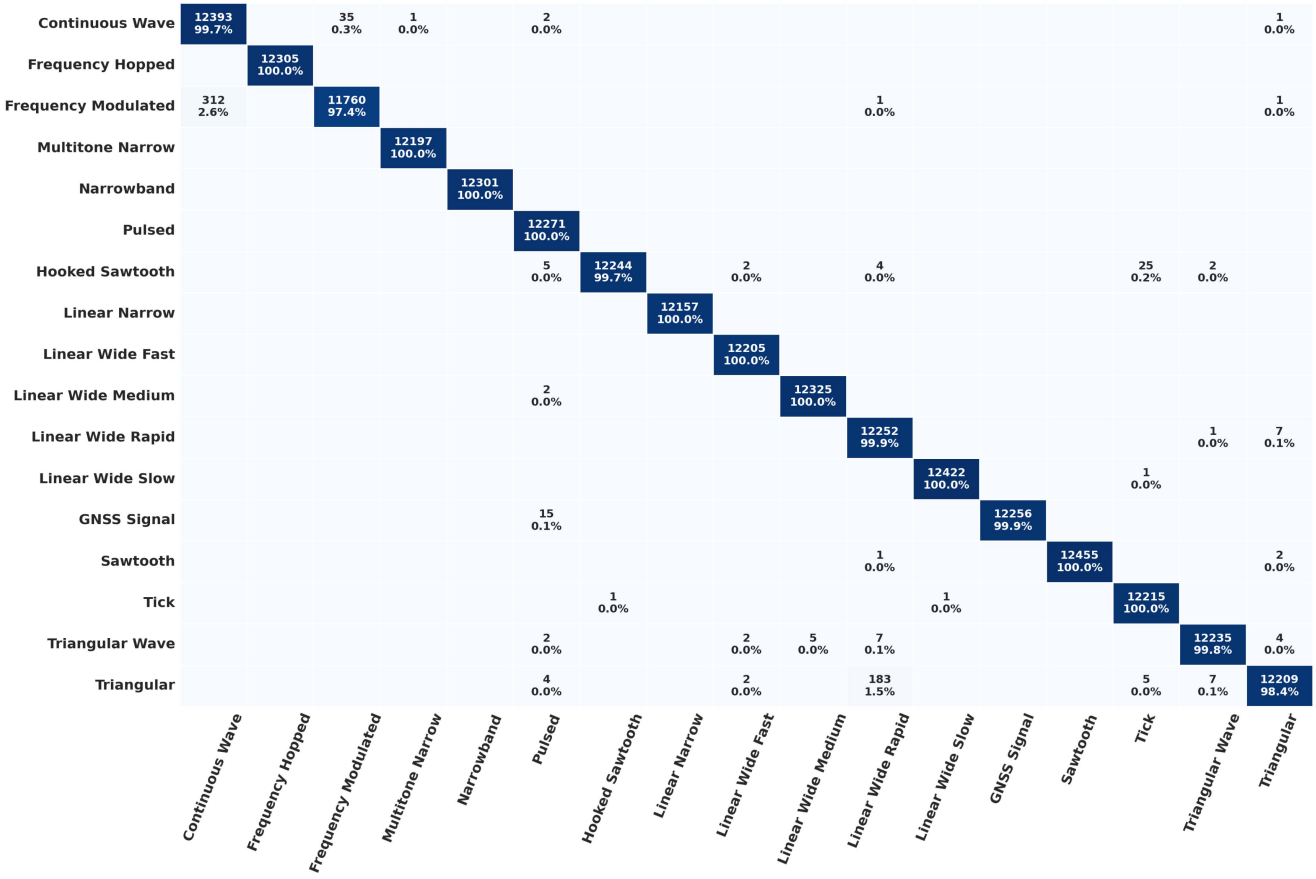


Fig. 7. Confusion matrix of feature-aided CNN classifier using Resnet architecture.

These findings could provide new insights into real-time monitoring interference. As mentioned in the related work section, traditional interference classification methods rely on predefined rules and thresholds, and mostly require ad hoc algorithms for each type of RFI signal structure. On the other hand, ML-based approaches offer a model to be trained for the accurate and automatic classification of interferences. A further advantage is that in the case of new scenarios or types of interference, the same model by retraining, can be adapted to these changes. One drawback of ML is that implementing it in mass-market devices can be challenging, often requiring more computational power or even changes to hardware design. It is worth noting that in GNSS interference detection and classification, where the output of the front-end can be used as the input data for the ML approach, variations in the output of each front-end architecture can occur despite consistency in factors such as bandwidth, sampling frequency, and bit quantization. Consequently, it can be inferred that for real-world implementation and deployment, ML algorithms should be trained with consideration for the specific front-end hardware, utilizing signal instances originating from that particular front-end. Additionally, the proposed methodology involves extensive preprocessing, particularly the calculation of STFT and conversion to images, which accounts for approximately 98% of the preprocessing time. This duration is significantly longer than that of models using only statistical features and metrics from a raw sample of signals. Adding to the earlier discussion of drawbacks, the training process of CNN architecture requires substantial computational resources and can be cumbersome, as is typical with deep learning models. Nevertheless, there are aspects to be addressed when implementing the algorithm in a system, such as a GNSS monitoring station, in order to tune the algorithm to cope with the constraints of the specific hardware (e.g., number of quantization bits, filtering effects of the front-end, etc.). The model itself can be evolved and extended to more interference classes as well as to deal with multiple interferences.

APPENDIX

A. GNSS Interferences

Various types of interference can arise from different sources, in this study, a comprehensive examination is conducted to determine their presence within the GNSS frequency band and classify them. The types of interference targeted by our method have been extensively researched and selected based on their prevalence and characteristics. Specifically, as documented in the STRIKE3 project [18], the commonly detected chirps can be classified into 11 distinct types, each characterized by unique features in

TABLE A1
Characteristics of Chirp Signals

Chirp Name	Sweep Rate (per 100 μs)	Sweep Range (Bandwidth MHz)
Wide sweep/slow	2	16
Wide sweep/medium	5	16
Wide sweep/fast	10	16
Wide sweep/rapid	15	16
Narrow sweep	10	5
Triangular	15	14
Triangular wave	5	16
Sawtooth	11	12
Hooked sawtooth	3	14
Tick	3	16
Multi tone	40	3

the spectrum and spectrogram. These types of signals are widely used as jammer devices in the market, making them a primary focus for interference classification. In addition to these common chirps, we have identified and included other types of interference based on a comprehensive literature review of previous work in the field of interference classification using machine learning. These additional interference types have been chosen to ensure that our model can detect a diverse range of interference scenarios encountered in real-world GNSS applications. The following sections describe each interference and jamming signal. Note that the amplitude remains constant for all simulated interferences in this study.

1) *Chirp Signals:* Chirp signals are among the most common disruptive or interfering signals, capable of disassembling or blocking a specific portion of the GNSS signal band and disrupting receiver operation. Chirp, also known as a swept-frequency signal, is a signal in which the frequency changes as a function of time; therefore, the frequency increase or decrease is referred to as up-chirp or low-chirp, respectively. In general, a chirp signal can be expressed as [76]

$$i(t) = A \cdot \exp\left\{ j\left(2\pi \int_0^t f(t)\,dt + \phi\right)\right\} \tag{8}$$

where A is the chirp amplitude and ϕ is the initial phase of the signal. $f(t)$ is the instantaneous frequency, and in the case of the linear chirp model, $f(t) = f_0 + kt$ where the frequency variation is linear over time. Swept-frequency jammers are characterized by their ability to generate overwhelming signals with carrier frequencies that vary over GNSS signal bands [77]. According to the survey in [18], the characteristics of chirp signals can be classified by shape, sweep range, sweep rate, and power. Table A1 shows the characteristics of each chirp signal analyzed in this study. Fig. 3 includes the spectrogram of 11 common chirp types, and most likely, the receiver will encounter them.

2) *Continuous Wave Interference:* The term continuous wave interference (CWI) refers to one of the main forms of interference and is made up of pure sine waves of constant amplitude and frequency and appears as a single tone in the frequency domain. Jamming devices or wireless telecommunications infrastructure such as VHF and UHF TV, instrument landing systems (ILS), and very high-frequency omnidirectional range (VOR) systems can be the source of common CWI [78], [79]. The components of complex CWI can be expressed as

$$i(t) = A \cdot \exp\{j(2\pi f_i t + \phi_i)\} \tag{9}$$

where A is the amplitude, f_i and ϕ_i are the interference frequency and phase.

3) *DME (Pulsed) Interference:* Pulsed interference refers to an interference signal consisting of multiple short pulses repeated over a period of time. This type of interference can occur in the L5/E5 bands, which are also occupied by aeronautical radio navigation services (ARNS) such as DME and TACAN [80]. DME interference is composed of a pair of Gaussian pulses modulated by a cosine, which can be modeled as [81]

$$i(t) = A \cdot \sum_{n=1}^{k} \left(e^{-\frac{\alpha(t-t_k)^2}{2}} + e^{-\frac{\alpha(t-t_k-\Delta t)^2}{2}} \right) \exp\{j(2\pi f_i t + \theta_i)\} \tag{10}$$

where A is the interference power, α determines the pulse width, Δt is the interpulse interval, t_k is the emission time of the kth pulse pair [81]. For example, the radiated signal of an individual DME station has $\alpha = 4.5 \times 10^{-11} s^{-2}$, $\Delta t = 12\ \mu s$ and a maximum repetition of 3600 pulses per second [2]. Simplified DME signals assumed for this study consist of a single pulse pair occurring inside the observation window (image) without collisions. This assumption is made to showcase the baseline performance of the ML approach. Advanced DME studies, such as signal collisions and detections, are suggested for future research.

4) *Frequency Hopped (FH) Jamming Signal:* Frequency hopping jamming introduces discontinuities in instantaneous frequencies by repeatedly switching the carrier frequency. According to [82], FH jamming has been observed on the island of Lampedusa (Italy). Pica et al.[82] provided a signal model for this interference as

$$i(t) = A \cdot \sum_{m=1}^{W} exp\{j2\pi f_m(t)t\} \tag{11}$$

where $f_m(t)$ is a function to create a set of m subtones randomly that occupy a specific frequency range, A is the amplitude, and W is the number of random generation trials. Table A2 shows the parameters used to simulate FH jamming in this context.

TABLE A2
Simulation Parameters of FH Jamming

Parameters	Value	Remark
T_i	100 μs	Signal duration
m	10	number of subcarriers
B_i	± 3 MHz	Subcarriers range
T_s	5 μs	duration of hopping
W	20	Random generation trials (T_i/T_s)

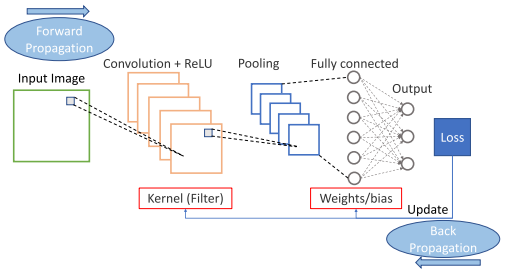


Fig. 8. Simple architecture of CNN.

5) *Frequency Modulated Interference:* Frequency modulation is a form of angular modulation in which the instantaneous frequency of a carrier wave is varied linearly, and it can be used by jammers. FM interference with a sinusoidal modulation signal can be represented as

$$i(t) = A \cdot \exp\{j(2\pi f_i t + \beta \sin(2\pi f_m t))\} \tag{12}$$

where A is the amplitude of the carrier, f_i is the interference frequency, f_m is the frequency of the modulated signal, and β denotes the modulation index [83], [37]. In this article, the frequency of the modulated signal and β are assumed to be in the range of [1:5] MHz and [0.3:0.9], respectively.

6) *Narrowband Interference:* Narrowband interference (NBI) or jammers are interferences whose spectral occupation is smaller than the GNSS signal bandwidth. NBI can be referred to narrowband Gaussian, narrowband phase and frequency modulation, and narrowband swept-frequency interference [84]. In this context, a narrowband Gaussian jammer is utilized, which essentially comprises a white Gaussian noise (WGN) filtered through a rectangular pulse shape with a bandwidth of 2 MHz. The formulation for this jammer is as

$$i(t) = A \cdot (g(t) * h(t))\exp\{j(2\pi f_i t + \theta_i)\} \tag{13}$$

where $g(t)$ is a complex Gaussian process and $h(t)$ is a narrow band filter function.

B. Convolutional Neural Networks

Deep learning constitutes a subset of machine learning techniques that leverage artificial neural networks to emulate the neural network of the human brain, enabling analysis and learning processes [47]. The artificial neural network consists of multiple nodes distributed in various layers, and each node implements an instruction called algorithms that guides the machine in recognizing patterns in the dataset and solving common problems [85]. CNNs have significantly impacted the field of image classification, and the name comes from the convolutional operator used in the filtering domain. A CNN is a multilayer neural network proposed to identify patterns from images represented by pixels with the least preprocessing. It comprises several layers: convolution, pooling, and fully connected (FC), briefly described in the following together with the role of the activation function. Fig. 8 demonstrates a simple CNN architecture.

1) *Convolutional layer:* The convolution layer is the fundamental component of CNN and takes most of the computations. The convolution operation is used for feature extraction by performing an elementwise product between a small matrix called kernel (filter) and a subset of input, then summing up the result. This operation slides across the whole input (known as a sliding window) to find a specific type of feature in the input, and the outputs produce a matrix of new features called a feature map [46]. Two parameters can define convolutions, the size, and the number of the kernel that apply to the input.
2) *Activation function:* The activation function is the core of each neural network, deciding which neuron of the layer should fire to the next layer. It increases the expression's ability of the network to learn the complex pattern of data by adding nonlinearity into the network. Generally, nonlinear activation layers come after each learnable layer, such as the convolution layer [86]. The rectified linear unit (ReLU), softmax, hyperbolic tangent, and sigmoid functions are commonly used that can be chosen with respect to classification task. ReLU, as the most common activation functions applies to the feature map and replaces all the negative values with zero, avoiding the values from summing up to zero.
3) *Pooling layer:* A pooling layer is essentially a downsampling layer and reduces the dimensions of created feature map from the previous layer into a new one. The pooling operation aggregates values of a given matrix. The most typical one is max pooling, where it selects the maximum value from each of the windows of the feature map. Therefore, the output values of the pooling layer are the most critical features of the previous feature map.
4) *FC layers:* Commonly, the last layers of CNN architecture are FC layers (dense layers) that perform the classification task. Each neuron of this layer is connected to the output of previous layers. The feature map created by the last convolution or pooling layer is flattened into a vector and will feed to the FC layer [87]. Typically, the last FC layer has the same number of nodes as the number of classes in the task. Furthermore, it incorporates a softmax activation function which applies to the multilabel classification task, where provides a probability distribution over the different classes valued from 0 to 1, indicating the likelihood of the input belonging to each class (all values sum to 1) [53].

Fig. 9 illustrates a one-layer CNN that consists of a convolution layer with a filter of size 2×2, ReLU activation function, and max-pooling layer.

CNN becomes a valuable solution for different problems by performing local receptive fields (the way the convolutional layer work), pooling layer, and shared weights/biases, which implies that all weights and biases in

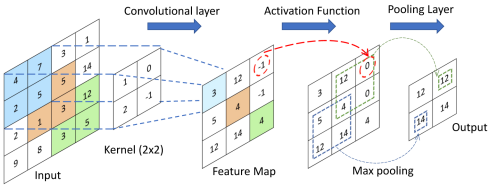


Fig. 9. Numerical example of one-layer CNN.

each window of neurons are the same. It is evident that using these concepts in CNN causes the number of parameters to be extremely decreased [87].

ACKNOWLEDGMENT

The authors would like to thank A. Minetto and M. T. Gamba for their invaluable contributions and assistance.

REFERENCES

[1] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Waltham, MA, USA: Ganga-Jamuna Press, 2011.

[2] F. Dovis, *GNSS Interference Threats and Countermeasures, Ser. The GNSS Technology and Application Series*. Norwood, MA, USA: Artech House, 2015.

[3] A. Rustamov, A. Minetto, and F. Dovis, "Improving GNSS spoofing awareness in smartphones via statistical processing of raw measurements," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 873–891, 2023.

[4] A. Minetto et al., "Nanosecond-level resilient GNSS-based time synchronization in telecommunication networks through WR-PTP HA," *IEEE Syst. J.*, vol. 18, no. 1, pp. 327–338, Mar. 2024.

[5] A. Minetto, B. D. Polidori, M. Pini, and F. Dovis, "Investigation on the actual robustness of GNSS-based timing distribution under meaconing and spoofing interferences," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 3848–3862.

[6] W. Qin, M. T. Gamba, E. Falletti, and F. Dovis, "An assessment of impact of adaptive notch filters for interference removal on signal processing stages of a GNSS receiver," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 4067–82, Oct. 2020.

[7] D. Borio, F. Dovis, H. Kuusniemi, and L. Lo, "Presti Impact and detection of GNSS jammers on consumer grade satellite navigation receivers," *Proc. IEEE*, vol. 104, no. 6, pp. 1233–1245, Jun. 2016.

[8] D. Borio, "Swept GNSS jamming mitigation through pulse blanking," in *Proc. Eur. Navigation Conf.*, 2016, pp. 1–8.

[9] C. Hegarty, D. Bobyn, M. Tran, T. Kim, and J. Grabowski, "Suppression of pulsed interference through blanking," in *Proc. IAIN World Congr. 56th Annu. Meeting Inst. Navigation*, 2000, pp. 399–408.

[10] D. Borio, L. Camoriano, and L. Lo Presti, "Two-pole and multi-pole notch filters: A computationally effective solution for GNSS interference detection and mitigation," *IEEE Syst. J.*, vol. 2, no. 1, pp. 38–47, Mar. 2008.

[11] E. Falletti, M. T. Gamba, and M. Pini, "Design and analysis of activation strategies for adaptive notch filters to suppress GNSS jamming," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3718–3734, Oct. 2020.

[12] D. Borio and C. Gioia, "Mitigation of frequency-hopped tick jamming signals," in *Proc. IEEE/ION Position Location Navigation Symp.*, 2020, pp. 624–630.

[13] A. Siemuri, H. Kuusniemi, M. S. Elmusrati, P. Välisuo, and A. Shamsuzzoha, "Machine learning utilization in GNSS-use cases, challenges and future applications," in *Proc. Int. Conf. Localization GNSS*, 2021, pp. 1–6.

[14] A. Nardin et al., "On the use of machine learning algorithms to improve GNSS products," in *Proc. IEEE/ION Position Location Navigation Symp.*, 2023, pp. 216–227.

[15] P. Flach, *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge, U.K.: Cambridge Univ. Press, 01 2012.

[16] I. Ebrahimi Mehr, A. Minetto, F. Dovis, E. Pica, C. Cesaroni, and V. Romano, "An open architecture for signal monitoring and recording based on SDR and docker containers: A GNSS use case," in *Proc. 20th Int. Conf. Smart Technol.*, 2023, pp. 66–71, doi: 10.1109/EUROCON56442.2023.10199078.

[17] I. Ebrahimi Mehr and F. Dovis, "Detection and classification of GNSS jammers using convolutional neural networks," in *Proc. Int. Conf. Localization*, 2022, pp. 1–6, doi: 10.1109/ICL-GNSS54081.2022.9797030.

[18] M. Pattinson et al., "D4.2: Draft standards for receiver testing against threats STRIKE3 public deliverable," Nov. 2017.

[19] W. Qin and F. Dovis, "Situational awareness of chirp jamming threats to GNSS based on supervised machine learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 3, pp. 1707–1720, Jun. 2022.

[20] H. Borowski, O. Isoz, F. Eklof, S. Lo, and D. Akos, "Detecting false signals: With automatic gain control," *GPS World*, vol. 23, pp. 38–43, 2012.

[21] B. Motella and L. L. Presti, "Methods of goodness of fit for GNSS interference detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 1690–1700, Jul. 2014.

[22] A. Tani and R. Fantacci, "Performance evaluation of a precorrelation interference detection algorithm for the GNSS based on nonparametrical spectral estimation," *IEEE Syst. J.*, vol. 2, no. 1, pp. 20–26, Mar. 2008.

[23] X. Ouyang and M. Amin, "Short-time fourier transform receiver for nonstationary interference excision in direct sequence spread spectrum communications," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 851–863, Apr. 2001.

[24] D. Borio, L. Camoriano, S. Savasta, and L. L. Presti, "Time-frequency excision for GNSS applications," *IEEE Syst. J.*, vol. 2, no. 1, pp. 27–37, Mar. 2008.

[25] E. Falletti, M. Pini, and L. L. Presti, "Low complexity carrier-to-noise ratio estimators for GNSS digital receivers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 1, pp. 420–437, Jan. 2011.

[26] F. Bastide, E. Chatre, and C. Macabiau, "GPS interference detection and identification using multicorrelator receivers," in *Proc. 14th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2001, pp. 872–881.

[27] J. Gross and T. Humphreys, "GNSS spoofing, jamming, and multipath interference classification using a maximum-likelihood multitap multipath estimator," in *Proc. Int. Tech. Meeting Inst. Navigation*, 2017, pp. 662–670.

[28] J. R. van der Merwe et al., "GNSS interference monitoring and characterisation station," in *Proc. Eur. Navigation Conf.*, 2017, pp. 170–178.

[29] J. R. van der Merwe, D. C. Franco, T. Feigl, and A. Rügamer, "Optimal machine learning and signal processing synergies for low-resource GNSS interference classification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 3, pp. 2705–2721, Jun. 2024.

[30] J. T. Curran, M. Bavaro, P. Closas, and M. Navarro, "On the threat of systematic jamming of GNSS," in *Proc. 29th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2016, pp. 313–321.

[31] X. Chen, D. He, X. Yan, W. Yu, and T.-K Truong, "GNSS interference type recognition with fingerprint spectrum DNN method," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4745–4760, Oct. 2022.

[32] A. R. Kazemi, S. Tohidi, and M. R. Mosavi, "Enhancing classification performance between different GNSS interferences using neural networks trained by TAC-PSO algorithm," in *Proc. 10th Int. Symp. Telecommun.*, 2020, pp. 150–154.

[33] R. R. Yakkati, B. Pardhasaradhi, J. Zhou, and L. R. Cenkeramaddi, "A machine learning based GNSS signal classification," in *Proc. IEEE Int. Symp. Smart Electron. Syst.*, 2022, pp. 532–535.

[34] A. Elango, S. Ujan, and L. Ruotsalainen, "Disruptive GNSS signal detection and classification at different power levels using advanced deep-learning approach," in *Proc. Int. Conf. Localization*, 2022, pp. 1–7.

[35] N. Spens, D.-K. Lee, F. Nedelkov, and D. Akos, "Detecting GNSS jamming and spoofing on android devices," *J. Inst. Navigation*, vol. 69, no. 3, 2022, doi: 10.33012/navi.537.
[36] Y. Ding and K Pham, "1 GNSS interference identification beyond jammer classification," in *Proc. IEEE Aerosp. Conf.*, 2023, pp. 1–8.
[37] J. Xu, S. Ying, and H. Li, "GPS interference signal recognition based on machine learning," *Mobile Netw. Appl.*, vol. 25, pp. 2336–2350, 2020.
[38] J. Shu, Y. Liao, and X. Luan, "An interference recognition method based on improved genetic algorithm," in *Proc. 7th Int. Conf. Comput. Commun.*, 2021, pp. 496–500.
[39] N. Raichur et al., "Machine learning-assisted GNSS interference monitoring through crowdsourcing," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 19–23.
[40] R. Ferre, A. Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, 2019, Art. no. 4841.
[41] C. J. Swinney and J. C. Woods, "GNSS jamming classification via CNN, transfer learning & the novel concatenation of signal representations," in *Proc. Int. Conf. Cyber Situational Awareness Data Analytics Assessment*, 2021, pp. 1–9.
[42] T. Brieger et al., "Multimodal learning for reliable interference classification in GNSS signals," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 3210–3234.
[43] J. R. van der Merwe, D. C. Franco, D. Jdidi, T. Feigl, A. Rügamer, and W. Felber, "Low-cost COTS GNSS interference detection and classification platform: Initial results," in *Proc. Int. Conf. Localization*, 2022, pp. 1–8.
[44] I. Shafi, J. Ahmad, S. I. Shah, and F. M. Kashif, "Techniques to obtain good resolution and concentrated time-frequency distributions: A review," *J. Adv. Signal Process.*, vol. 1, pp. 1–43, 2009.
[45] N. O. Mahony et al., "Deep learning versus traditional computer vision," in *Proc. Comput. Vis. Conf.*, 2019, pp. 128–144.
[46] K. O'Shea and R. Nash, "*An introduction to convolutional neural networks*," 2015, *arXiv:1511.08458*.
[47] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit. Lett.*, vol. 141, pp. 61–67, 2021.
[48] D. Lu and Q. Weng, "A. survey of image classification methods and techniques for improving classification performance," *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 823–870, 2007.
[49] M. Xin and Y. Wang, "Research on image classification model based on deep convolution neural network," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, pp. 1–11, 2019.
[50] M. DeGroot and M. Schervish, *Probability and Statistics*. Reading, MA, USA: Addison-Wesley, 2012.
[51] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, Jun. 1967.
[52] M. Nicola, G. Falco, R. Morales Ferre, E.-S. Lohan, A. de la Fuente, and E. Falletti, "Collaborative solutions for interference management in GNSS-based aircraft navigation," *Sensors*, vol. 20, no. 15, 2020, Art. no. 4085.
[53] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, 2018.
[54] C. Mateo and J. Talavera, "Short-time fourier transform with the window size fixed in the frequency domain," *Digit. Signal Process.*, vol. 77, pp. 13–21, 2018.
[55] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.
[56] E. Klejmova and J. Pomenkova, "Identification of a time-varying curve in spectrogram," *Radioengineering*, vol. 26, no. 1, pp. 291–298, 2017.
[57] L. Cohen and P. Loughlin, "Leid1. time-frequency analysis: Theory and applications," *J. Acoustical Soc. America*, vol. 134, no. 5, pp. 4002–4002, 2013.
[58] L. Stankovic, "A method for time-frequency analysis," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 225–229, Jan. 1994.
[59] P. Flandrin, "Some features of time-frequency representations of multicomponent signals," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 1984, pp. 266–269.
[60] L. Debnath and F. A. Shah, "Shah the Wigner-Ville distribution and time-frequency signal analysis," in *Wavelet Transforms and Their Applications*. Berlin, Germany: Springer, 2015.
[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, May 2017.
[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
[65] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5353–5360.
[66] E. Falletti, D. Margaria, M. Nicola, G. Povero, and M. Troglia, "Gamba N-FUELS and SOPRANO: Educational tools for simulation, analysis and processing of satellite navigation signals," in *Proc. IEEE Front. Educ. Conf.*, 2013, pp. 303–308.
[67] A. Nardin, T. Imbiriba, and P. Closas, "Jamming source localization using augmented physics-based model," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[68] M. J. Murrian et al., "First results from three years of GNSS interference monitoring from low earth orbit navigation," *J. Inst. Navigation*, vol. 68, no. 4, pp. 673–685, 2021.
[69] Z. Clements, T. E. Humphreys, and P. Ellis, "Dual-satellite geolocation of terrestrial GNS jammers from low earth orbit," in *Proc. IEEE/ION Position Location Navigation Symp.*, 2023, pp. 458–469.
[70] M. Troglia Gamba, B. D. Polidori, A. Minetto, F. Dovis, E. Banfi, and F. Dominici, "GNSS radio frequency interference monitoring from LEO satellites: An in-laboratory prototype," *Sensors*, vol. 24, no. 2, 2024, Art. no. 508. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/508
[71] E. Bisong, *Google Colaboratory*. New York, NY, USA: Apress, 2019, pp. 59–64. [Online]. Available: https://research.google.com/colaboratory/faq.html
[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 7–9, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980
[73] E. Gordon-Rodriguez, G. Loaiza-Ganem, G. Pleiss, and J. P. Cunningham, "Cunningham Uses and abuses of the cross-entropy loss: Case studies in modern deep learning," in *Proc. "I Can't Believe It's Not Better!" NeurIPS Workshops*, PMLR, 2020, pp. 1–10.
[74] L. Prechelt, *Early Stopping' But When*? Berlin, Germany: Springer, 2012, pp. 53–67, doi: 10.1007/978-3-642-35289-8_5.
[75] M. Hossin and S. M.N, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, pp. 1–11, 2015.
[76] T. Kraus, R. Bauernfeind, and B. Eissfeller, "Survey of in-car jammers - analysis and modeling of the RF signals and IF samples (suitable for active signal cancelation)," in *Proc. 24th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2011, pp. 430–435.
[77] W. Qin, M. Troglia Gamba, E. Falletti, and F. Dovis, "Effects of optimized mitigation techniques for swept-frequency jammers on tracking loops," in *Proc. 32nd Int. Tech. Meeting Satell. Division Inst. Navigation*, 2019, pp. 3275–3284.

[78] D. Borio, L. Camoriano, L. Lo Presti, and P. Mulassano, "Analysis of the one-pole notch filter for interference mitigation: Wiener solution and loss estimations," in *Proc. 19th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2006, pp. 1849–1860.
[79] R. Landry Jr. and A. Renard, "Analysis of potential interference sources and assessment of present solutions for GPS/GNSS receivers," in *Proc. 4th Int. Conf. Integr. Navigation Syst.*, St. Petersburg, Russia, May 26–28, 1997.
[80] L. Musumeci, J. Samson, and F. Dovis, "Performance assessment of pulse blanking mitigation in presence of multiple distance measuring equipment/tactical air navigation interference on GNSS signals," *IET Radar, Sonar Navigation*, vol. 8, no. 6, pp. 647–657, 2014.
[81] A. Garcia-Pena, C. Macabiau, O. Julien, M. Mabilleau, and P. Durel, "impact of DME/TACAN on GNSS L5/E5a receiver," in *Proc. Int. Tech. Meeting Inst. Navigation*, 2020, pp. 207–221.
[82] E. Pica, A. Minetto, C. Cesaroni, and F. Dovis, "Analysis and characterization of an unclassified RFI affecting ionospheric amplitude scintillation index over the mediterranean area," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 8230–8248, 2023.
[83] T. S. Rappaport et al. *Wireless Communications: Principles and Practice*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[84] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: Principles and Applications*. Norwood, MA, USA: Artech House, 2017.
[85] A. Geron, *Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2019.
[86] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, 2020, Art. no. 1897.
[87] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications," *Future Directions J. Big Data*, vol. 8, no. 1, pp. 1–74, 2021.



Iman Ebrahimi Mehr (Member, IEEE) was born in Tehran, Iran, in 1989. He received the B.Sc. degree in electronic engineering from Azad University, Tehran, Iran, and the M.Sc. degree in ICT for smart society from the Politecnico di Torino, Turin, Italy, in 2012 and 2021, respectively. He is currently working toward the Ph.D. degree with the Navigation Signal Analysis and Simulation Group, Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy.

His research interests include Global Navigation Satellite Systems (GNSS) and artificial intelligence applied to positioning, navigation, and timing, as well as to interference detection and mitigation.



Fabio Dovis (Member, IEEE) was born in Bruino, Italy, in 1970. He received the M.Sc. degree in electronics engineering and the Ph.D. degree in electronics and communication engineering from the Politecnico di Torino, Turin, Italy, in 1996 and 2000, respectively.

In 2004, he joined the Department of Electronics and Telecommunications, Politecnico di Torino as an Assistant Professor, where he has been a Full Professor since 2021. He coordinates the Navigation Signal Analysis and Simulation Research Group. His research interests include design of GPS and Galileo receivers and advanced signal processing for interference and multipath detection and mitigation, and also ionospheric monitoring.

Dr. Dovis is a Member of the IEEE Aerospace and Electronics Systems Society Navigation Systems Panel.

Open Access provided by 'Politecnico di Torino' within the CRUI CARE Agreement