Optimal Machine Learning and Signal Processing Synergies for Low-Resource GNSS Interference Classification

JOHANNES ROSSOUW VAN DER MERWE, Senior Member, IEEE
DAVID CONTRERAS FRANCO, Student Member, IEEE
TOBIAS FEIGL, Member, IEEE
ALEXANDER RÜGAMER
Fraunhofer IIS, Nuremberg, Germany

Interference signals degrade the performance of a global navigation satellite system receiver. Classification of these interference signals allows better situational awareness and facilitates appropriate countermeasures. However, classification is challenging and processing-intensive, especially in severe multipath environments. This article proposes a low-resource interference classification approach that combines conventional statistical signal processing approaches with machine learning (ML). It leverages the processing efficiency of conventional statistical signal processing by summarizing, e.g., a short-time Fourier transform, with statistical measures. Furthermore, the ML design space is bounded as the signal is preprocessed. It results in fewer opportunities for ML but facilitates faster convergence and the use of simpler architectures. Therefore, this approach has lower ML training complexity and lower processing and memory requirements. Results show competitive classification capabilities to more complex approaches. It demonstrates that more efficient architectures can be developed using existing signal-processing approaches.

Manuscript received 1 March 2023; revised 15 September 2023 and 18 December 2023; accepted 30 December 2023. Date of publication 2 January 2024; date of current version 11 June 2024.

DOI. No. 10.1109/TAES.2023.3349360

Refereeing of this contribution was handled by M. Braasch.

This work was supported in part by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the DARCY project under Grant 50NA2017 and in part by German Aerospace Center (DLR).

Authors' addresses: Johannes Rossouw van der Merwe, David Contreras Franco, and Alexander Rügamer are with Satellite Based Positioning Systems Department, Fraunhofer IIS, 90411 Nuremberg, Germany, E-mail: (j.r.vandermerwe@ieee.org, david.contrerasfranco@ieee.org, alexander.ruegamer@iis.fraunhofer.de); Tobias Feigl is with Precise Positioning and Analytics Department, Fraunhofer IIS, 90411 Nuremberg, Germany, E-mail: (tobias.feigl@iis.fraunhofer.de). *(Corresponding author: Johannes Rossouw van der Merwe.)*

I. INTRODUCTION

Interference signals degrade global navigation satellite system (GNSS) services [1]. Ideally, the interference sources should be found and removed. If the interference signal is additionally classified, the purpose and likely origin of the interference may be uncovered [2]. For example, a chirp signal may indicate a privacy protection device (PPD) that is an intentional disruptor [3]; a single tone may indicate an unintentional harmonic from other electronic equipment [4]; a Gaussian pulsed signal in the *L*5-band may be an aeronautical distance measurement equipment signal [5]; or a modulated signal in the *E*6-band may indicate an amateur radio broadcast [6]. Therefore, understanding the signal waveform helps to decide what to do with or where to look for it. Furthermore, the current interference situation is revealed if the classification is combined with interference monitoring [7], [8], [9], [10]. The capabilities are further amplified if it is networked [11]. It facilitates appropriate development in interference mitigation techniques for resilient GNSS receivers [1], [12], [13]. Finally, signal classification is necessary for general spectrum monitoring and policing [14].

The classic approaches use maximum likelihood for classification [15], [16]. However, these approaches require high human design involvement, with several classification stages to be developed and tuned for each interference class. It results in a significant development overhead. Furthermore, these methods are less reliable in strong multipath conditions where frequency selective fading shapes the interference signals. Several simultaneous interference signals are also challenging to deal with [17].

Machine learning (ML) is a popular modern choice for signal classification and shows good performance in various applications [2], [18], [19]. It especially shows improved resilience to classification in scenarios where the interference signals are affected by multipath [20], [21]. However, it has several limitations. First, training requires a significant effort. Complex models are often challenging to develop as an appropriate model must be identified, the models need to be optimized, and they could take significant processing time before they converge. Second, generalization versus overtraining of a model could limit reusability when applied to new datasets and environments. A common approach is to (partially or fully) retrain the model for new data (it is often not possible or practical) or to use reinforcement learning or unsupervised approaches (more complex algorithms to develop) [20]. Third, ML could be excessive to solve a problem, as many well-known signal-processing methods are available, and there is no need to reinvent them. It motivates the use of ML in conjunction with known classical approaches that have already been shown optimal in the signal processing community. Fourth, ML models may become big, limiting their low size, weight, power, and cost (SWAP-C) use for edge devices. For example,

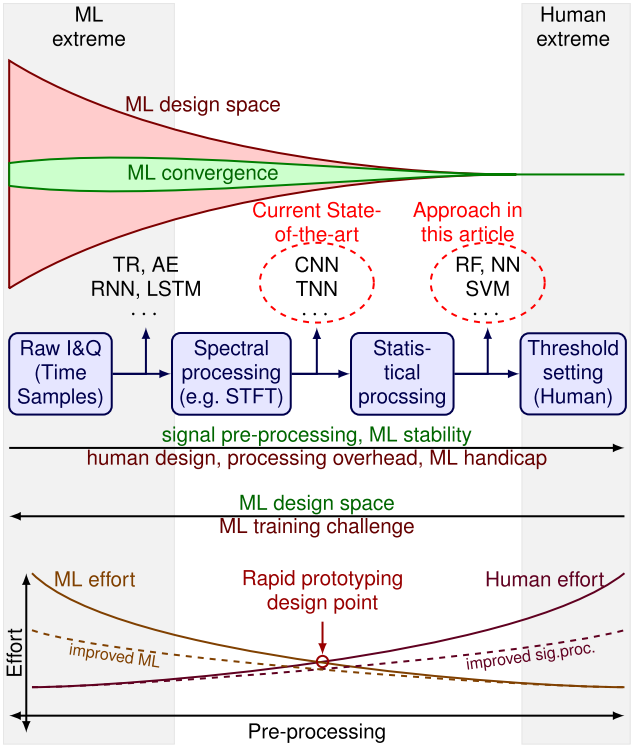


Fig. 1. Pipeline comparison of human intervention to machine learning in the design process. This graph does not comment on the potential performance of the different methods.

convolutional neural networks (CNNs) are a popular choice for signal classification [19], [20], but they require multiple multidimensional convolutions to process the data, which is not particularly useful when optimizing for low-power systems like a single-board computer (SBC). A counter around it sends the data to computer clusters for processing, but it introduces communication overhead. Fifth, ML often results in a "black box" that is not transparent nor gives context of the underlying algorithm. It makes it difficult for certification [22] and hinders the adaptation of new techniques. In conclusion, ML provides many performance improvements but requires appropriate training data and care to develop practical models for low SWAP-C targeted devices.

Fig. 1 highlights the core philosophy this article presents and how it applies to low SWAP-C architectures. How much of a pipeline relies on ML and how much on conventional methods when designing architecture? An ideal pipeline leverages the strengths of both in this tradeoff.

The left extreme of Fig. 1 represents the case where everything is done with ML. The raw in-phase and quadrature-phase (IQ) samples are passed directly to the ML pipeline (leftmost blue block, in the middle). The benefit of this approach is that the human effort is potentially low (assuming that no ML pipeline redesign is required), and all the design effort is placed in ML (graph at the bottom). Therefore, with sufficient computing power, this approach could result in low human effort for development. Applicable architectures in this regard include transformer (TR) [20], autoencoder (AE) [23], recurrent neural network (RNN) [24], [25], and long short-term memory (LSTM) [24], [25]. It results in the largest ML design space and opportunities (red area at the top) but has a significant risk that the algorithms may not converge sensibly (green area at the top). This extreme suffers from current ML training approaches, which may result in no sensible convergence and often require large ML models. Furthermore, data handling with the raw IQ becomes challenging.

The right extreme of Fig. 1 is the classical approach, where everything is hand-designed and tuned. It has no ML design space and no convergence issues. The benefit of this approach is that there is no "black box" design. However, the human effort is exceptionally high as every threshold is hand-tuned (right most blue box in the middle). This extreme suffers from human involvement and has high design costs with low generalization (i.e., a new GNSS receiver may require some thresholds to be optimized again).

In the center of Fig. 1, the current ML approaches are shown. Some preprocessing, e.g., calculating the short-time Fourier transform (STFT) of the data, is applied. It allows the interference signal to be transformed into a domain with more meaningful information. Because the data are processed, the ML design space reduces (red area at the top), but the convergence likelihood (green area at the top) improves. The ML training is more stable and has less training overhead compared to the left extreme, but it also has more opportunities than the right extreme. The current state-of-the-art design uses approaches such as CNNs [18], [19], [20] and twin neural networks (TNNs) [21]. However, the preprocessing still results in large dimensions, i.e., the STFT is still significant. The approach in this article is still further to the right, where the results are further statistically processed. It collapses the dimensions of the STFT to be much smaller. As a result, training is even faster and ensures improved convergence, but ML opportunities are severely restricted. However, it is still significantly less effort for a human, as less threshold tuning or algorithm development is required if the appropriate ML architecture is already known.[1] Some ML approaches that can be used include random forest (RF) [2], support vector machine (SVM) [26], or neural network (NN) [27].

The first contribution of this article is to provide a framework for a human-to-machine tradeoff, as demonstrated in Fig. 1. The core principle in the design is to balance the human and ML effort (bottom graph in Fig. 1). It allows for efficient system design and is ideal for rapid prototyping applications. Furthermore, the design becomes more efficient as improved ML and signal-processing methods become available.

The second contribution of this article is an analysis of signal processing approaches that facilitate improved

[1]Practical ML pipeline development may require significant development effort if an appropriate architecture is not yet identified or hyperparameter tuning is still required. Therefore, it is assumed in this discussion that appropriate models for the problem have already been identified, and only training on a new dataset is required.

interference classification for ML-based architectures. This article demonstrates the benefits of signal preprocessing to enhance ML capabilities and limit overall processing. Using the appropriate preprocessing from known classical best practices in conjunction with ML provides superior performance, as the benefits of both are leveraged. The ML simplifies threshold settings and classification logic, and the conventional methods provide optimal signal processing foundations.

The third contribution is to show that exploiting applicable signal processing methods facilitate low size, weight, and power (SWAP) and low-training effort ML pipelines. The presented architecture in this article requires significantly fewer processing resources for inference (three orders of magnitude less than an equivalent CNN, see Appendix B) and fewer features for rapid ML training (two orders of magnitude less) making it especially applicable for low SWAP platforms. It, in turn, allows the architecture to run in real time on an SBC platform [2].

In the first half of the article (see Sections IV and V), basic signal detection and classification are demonstrated by using the statistical metrics from Gaussian and sinusoid signals in continuous and pulsed operations. Monte Carlo runs vary the interference-to-noise ratio (INR), and the pulsed duty cycle evaluates the metrics. It provides a foundation for signal classification. Building on it, the second half of the article extends the classification using multispectral approaches (see Sections VI–VIII). It creates an analytical framework to demonstrate signal classification with a reduced feature space, facilitating low SWAP classification. Results indicate that complex signal waveforms are detectable and classifiable, even with rudimentary ML approaches.

The rest of this article is organized as follows. Section II gives an overview of detection and classification approaches, and Section III describes popular detection metrics and domain transformations. Section IV presents signal models and the expected statistics. Monte Carlo results demonstrate the potential for a single channel in Section V. Section VI describes the proposed processing model, and the entire system results are shown in Section VII, and discussed in Section VIII. Finally, Section IX concludes this article.

II. DETECTION AND CLASSIFICATION APPROACHES

This section provides a basis for the detection[2] (i.e., binary classification) and classification problems. The subsequent section extends by providing algorithm design and a review of the current state-of-the-art.

In classical approaches, the signals are preprocessed to obtain a suitable metric [15], [16]

$$s = f(\mathbf{x}) \tag{1}$$

where s is the metric, $\mathbf{x}$ is a complex vector of the received signal samples, and $f(\cdot)$ is a function that maps the samples to the metric. The metric is compared to a threshold to detect interference (detection) or do a binary classification

$$s \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda \tag{2}$$

where $\mathcal{H}_0$ is the null hypothesis, $\mathcal{H}_1$ is the first alternative hypothesis, and λ is the decision metric. Multiple hypotheses and metrics extend this for multiple classes. For example, such a classification algorithm with three hypotheses and two metrics may be created

$$\mathcal{H}_0 \ : \ s_1 < \lambda_1 \tag{3}$$

$$\mathcal{H}_1 \ : \ s_1 \geq \lambda_1 \ \wedge \ s_2 < \lambda_2 \tag{4}$$

$$\mathcal{H}_2 \ : \ s_1 \geq \lambda_1 \ \wedge \ s_2 \geq \lambda_2 \tag{5}$$

where $\wedge$ is the logical union.

The probability of detection P_{D} and the probability of false alarm P_{FA} as a function of the threshold λ for the simple case is defined as [28]

$$P_{\mathrm{D}}(\lambda) = P(s \geq \lambda \mid \mathcal{H}_1) \qquad P_{\mathrm{FA}}(\lambda) = P(s \geq \lambda \mid \mathcal{H}_0) \tag{6}$$

where $P(\cdot \mid \cdot)$ is the conditional probability.

Developers find and optimize these metrics (s) and tune the thresholds (λ), requiring significant effort to derive, optimize, and tune, making them costly and often impractical. Furthermore, most of these metrics are not publicly revealed due to security trade-secret reasons, limiting the development and advancement of the field. However, such metrics often build on sound signal processing principles using well-documented mathematics [28], such as maximum likelihood approaches. Finally, the complexity of the preprocessing $f(\cdot)$ impacts the computing and memory resources required for the design and often needs to be optimized to meet the specifications of the target platform. Some examples of metrics used include using the automatic gain control gain of a receiver and other GNSS receiver outputs [29], using the Fourier analysis of the signal samples [16], applying power or energy detections [30], or the power distortion of the GNSS correlators [31].

In deep learning approaches, the metrics s and obtaining them $f(\cdot)$ are left for the ML algorithm to determine, and it sets the appropriate thresholds λ. It outsources the work from a developer and streamlines development, but some ML architecture design is still required. However, the problem is commonly defined openly (i.e., a limited context of the problem is provided) to give maximum optimization room for the ML algorithm. However, in doing so, it often discards the well-known signal processing principles for simplicity. The challenge is optimally integrating known signal processing approaches with ML for superior solutions [32].

A common approach for ML-based GNSS interference classification is first to transform the data into a meaningful and representative dimension, for example, to calculate an STFT to generate a spectrogram. Next, a CNN is applied to the spectrogram data for classification [18], [19], [20]. As

[2]Note that detection in this context refers to binary classification, according to the classical signal detection literature [28]. Detection within the ML context has a significantly different meaning.

the spectrogram is processed as a 2-D image, this approach benefits from the large body of research developed for image classification [33]. Other approaches use RNN, typically LSTM-based NN architectures, to embed the temporal signal information [24], [25]. Other approaches, including TNN, Siamese neural network, and multimodal learning (MTL), have also been considered [20].

III. CLASSIFICATION METRICS

In this section, several metrics are considered for interference classification. Most of these are built on statistics and data science approaches. In the following, $x[k]$ is the received complex digital signal.

A. Energy

Energy detection is the classical interference detection (i.e., binary classification) and is shown to be optimal for an interference signal in additive white Gaussian noise (AWGN) [28], [34]. It calculates the energy in the received signal

$$s_{\mathrm{e}} = T_{\mathrm{s}} \sum_{k=0}^{K-1} p[k] = T_{\mathrm{s}} \sum_{k=0}^{K-1} x[k]x^*[k] \tag{7}$$

where s_{e} is the energy metric, T_{s} is the sample period, and $p[k]$ is the instantaneous power. If it is normalized to the integration time, it is a (mean) power detector

$$s_{\mathrm{p}} = \frac{s_{\mathrm{e}}}{KT_{\mathrm{s}}} = \frac{1}{K} \sum_{k=0}^{K-1} p[k] = \frac{1}{K} \sum_{k=0}^{K-1} x[k]x^*[k]. \tag{8}$$

As the energy and power detectors are linearly related through a scale function, they are often used interchangeably in the literature. The energy detector is popular for interference detection in GNSS [30], [35]. It can also be modified to function after correlation in the tracking stage of a GNSS receiver [31]. Energy detection is a popular, simple, optimal, and classic approach. However, it has limited classification capabilities.

B. Kurtosis

The kurtosis is the fourth statistical moment and measures the tailedness of a distribution [36]. The kurtosis increases the more values are away from the mean, i.e., outliers, which makes it great to detect mixed distributions. Therefore, it is often used to detect sudden changes in a signal, such as pulsed signals or discontinuities [35], [37], [38].

The natural biased kurtosis is defined as [36], [39]

$$\kappa_{\mathrm{n}} = \frac{\mathrm{E}\left[(X-\overline{X})^4\right]}{\mathrm{E}\left[(X-\overline{X})^2\right]^2} \approx \frac{\frac{1}{K}\sum_{k=0}^{K-1} |x[k]-\overline{x}|^4}{\left(\frac{1}{K}\sum_{k=0}^{K-1} |x[k]-\overline{x}|^2\right)^2} \tag{9}$$

where κ_n is the natural biased kurtosis, $\mathrm{E}[\cdot]$ is the statistical expectation, X is the random variable (RV) for the data samples $x[n]$, $\overline{X}$ and $\overline{x}$ are the respective means of the RV and signal, and $|\cdot|$ is the absolute value. The absolute value is required here, as it facilitates using complex signals.

A real-valued Gaussian distribution has a kurtosis of $x \in \mathbb{R} : \kappa_{\mathrm{n}} = 3$ [36]. Therefore, the excess kurtosis subtracts this value. However, for a complex signal, the kurtosis is $x \in \mathbb{C} : \kappa_{\mathrm{n}} = 2$ [38]. The excess kurtosis is defined as

$$x \in \mathbb{R} : \ \kappa_{\mathrm{r}} = \kappa_{\mathrm{n}} - 3 \, ; \qquad x \in \mathbb{C} : \ \kappa_{\mathrm{c}} = \kappa_{\mathrm{n}} - 2 \tag{10}$$

where κ_{r} is the real excess kurtosis, and κ_{c} is the complex excess kurtosis. The fact that a complex signal has a different excess kurtosis is often incorrectly disregarded [35].

The spectral kurtosis is a popular extension [40]. In this architecture, a Fourier transform of the data is first calculated, typically with an STFT, and then, the kurtosis of the spectrum is determined [41]

$$y[m,l] = \mathcal{F}\{x[n]\}[l] \tag{11}$$

$$= \sum_{k=0}^{K-1} x[(m-1)K+k] \cdot w[k] \cdot \mathrm{e}^{-2\pi \, \mathrm{j} \, lk/K} \tag{12}$$

$$\kappa_{\mathrm{s}}[l] = \frac{\frac{1}{M}\sum_{m=0}^{M-1} |y[m,l]-\overline{y}[l]|^4}{\left(\frac{1}{M}\sum_{m=0}^{M-1} |y[m,l]-\overline{y}[l]|^2\right)^2} - 2 \tag{13}$$

where $\mathcal{F}\{\cdot\}$ is the Fourier transform, $w[k]$ is a window function, $y[m,l]$ is the complex spectral values for the mth time bin and lth frequency bin, $\overline{y}$ is the mean spectral value, and κ_{s} is the spectral entropy. Note that the spectral kurtosis is correctly normalized as a complex excess kurtosis. The spectral kurtosis uses the Fourier transform, but it can be adapted to any domain transform, like those highlighted in Section III-D.

The spectral kurtosis helps detect band-limited interferences and is popular for detection and classification [42]. It isolates the interference signal to specific frequency bands for analysis due to its capability to reveal nonstationary behavior. The spectral kurtosis is a practical example of how a simple statistical measure is significantly more useful when combined with appropriate signal preprocessing.

C. Entropy

Entropy is widely used for detection and classification problems [43], [44], including interference detection in GNSS [35] and spectrum management [45]. There are multiple families of entropy metrics [44]. For example, Rényi-entropy-based is used in spectrum sensing for cognitive radio [46]. However, presenting all entropy metrics is well outside the scope of this article. The focus is primarily on the Shannon entropy.

Shannon entropy measures the information (often referred to as the surprise or uncertainly) in an RV [43]. It requires the probability density function (pdf) of the signal. Therefore, it is often approximated with a histogram. The Shannon entropy is defined as

$$s_{\mathrm{h}} = \mathrm{E}\left[-\log_b P(X)\right] \approx -\sum_{k=0}^{K-1} p[k] \log_b p[k] \tag{14}$$

where s_{h} is the entropy, b is the base of the signal and is often selected as the natural base (i.e., $b = e$) or binary ($b =$

2), $P(X)$ is the probability of X, and $p[k]$ is the discrete approximated likelihood.

A Fourier transform can improve the entropy similar to the spectral kurtosis. The spectral entropy considers the Welch power spectrum as the pdf of the signal [47]. It first calculates the power spectral density (PSD) of the spectrum

$$p_y[l] = \frac{1}{k}\sum_{k=0}^{K-1} |y[k,l]|^2 = \frac{1}{k}\sum_{k=0}^{K-1} |\mathcal{F}\{x[n]\}[l]|^2 \quad (15)$$

$$s_y[m] = \frac{p_y[m]}{\sum_{k=0}^{K-1} p_y[k]} \quad (16)$$

where $p_y[m]$ is the power spectrum, and $s_y[m]$ is the PSD. Next, it uses the power spectrum to calculate the spectral entropy

$$s_{\text{sh}} = -\sum_{k=0}^{K-1} s_y[k] \log_b s_y[k]. \quad (17)$$

Finally, it is normalized to the maximal spectral entropy of white noise

$$s_{\text{shn}} = \frac{s_{\text{sent}}}{\log_b K}. \quad (18)$$

Like the spectral kurtosis, it also measures how uniform (flat) or sparse (spikey) the spectrum is. Additionally, it can be adapted to any domain transform, as showcased in Section III-D. Some examples include singular spectral entropy, which uses eigenvalue decomposition [48], and wavelet energy spectral entropy, which uses wavelet decomposition [49].

D. Spectral Methods

The spectral kurtosis and spectral entropy demonstrated that domain transformation provides new perspectives to the data that enhance detection and classification. In modern signal processing, many transformations exist, and the most applicable ones to interference signal detection are summarized in this section.

The most popular approach is discrete Fourier transform (DFT)-based transforms [7], [16], [19], [20], [50]. The popularity is attributed chiefly to the processing efficient fast Fourier transform (FFT) implementation of the DFT. It makes these transformations practical and fast. The DFT presents the frequency information of the signal and is especially useful for signals not spectrally flat that distinguish them from spectrally flat AWGN. The STFT calculates several smaller time-delayed FFTs to have a time-to-frequency tradeoff. It has a coarser frequency resolution but retains some time resolution, and it is popular for classification tasks [16], [51]. A limitation of the DFT is the frequency isolation capability and sidelobes. An approach to improve these is to use appropriate windowing functions on the data before the transform [50]. These could suppress the sidelobe level but broaden the main lobe, which reduces frequency accuracy. More advanced approaches, such as using a polyphase filter bank [52], suppress the sidelobes but result in significantly more processing as additional filter infrastructure is required.

Wavelet-based methods decompose the signals with wavelet functions that have both time and frequency information [1], [53]. The wavelet packet decomposition decomposes the signal in a full binary tree using high-pass and low-pass wavelet filters and is suitable for linearly-spaced frequency data. Under the appropriate selection of wavelet functions, this implementation approaches Fourier methods. In contrast, the discrete wavelet transform only iteratively decomposes the low-pass filter branches to form an asymmetric tree. It is popular for logarithmic-spaced frequency data such as audio-visual compression [54] but is less useful for radio-frequency data. As the wavelet function has more flexibility and presents a time and frequency tradeoff, it is popular for modern spectral detection and classification approaches [55]. However, it is not as processing efficient as the FFT, even though it is parallelizable.

Autocorrelation approaches have enhanced detection capabilities for periodic signals [56]. The Wigner–Ville distribution (WVD) autocorrelates the data before using a DFT to translate to the frequency domain [52]. It has improved frequency resolution to the DFT, which is ideal for accurate tone detection. However, it requires more processing and has a significant problem with cross-correlation products that limit the dynamic range. Nevertheless, it performs well for signal classification tasks [56], [57]. The Choi–Williams distribution (CWD) improves the WVD by using appropriate weighting (i.e., an exponential kernel function) to reduce the cross-correlation components [56]. However, it reduces the frequency precision. The CWD facilitates a modern classification approach for autocorrelation transformations [58].

The Teager–Kaiser (TK) estimates the energy of an oscillating system but has been adapted as a tone estimator for digital systems [59]. It is useful for some detection and classification tasks [35], [60]. The TK operator must be calculated for every frequency that needs to be analyzed, which makes it processing intensive. However, it has a superior frequency resolution than the DFT.

The Karhunen–Loève transform (KLT) decomposes a signal to the maximally sparse domain using eigenvalue decomposition [1]. It is processing intensive (i.e., eigenvalue decomposition) and often requires prior information to make it practical. However, it transforms the data into an ideal representation that is great for classification [61] but unpractical.

Gabor-based methods introduce Gaussian time-weighting to existing transforms. The Gabor transform alters the STFT and has improved tonal classification [62], [63]. Similarly, the Gabor–Wigner transform weights the WVD transform and has an interesting performance [64].

Many modern GNSS receivers use complex baseband samples as the efficiency of complex-valued signal processing improves later GNSS processing. Several real-valued transformations exist, like the discrete cosine transform.

TABLE I
Summary of Statistical Properties of the Five Signal Types

Metric	No int	GC	CC	GP	CP
s_e	$T_s K \sigma_n^2$	$T_s K B^2$	$T_s K B^2$	$T_s K B^2$	$T_s K B^2$
κ_c	0	0	$\frac{-A^4}{B^4}$	$\frac{2\overline{D}A^4}{DB^4}$	$\frac{\left(\overline{D}-D\right)A^4}{DB^4}$
s_h	$\log_b\left(2\pi\sigma_n^2\right)$ $+\frac{1}{\ln b}$	$\log_b\left(2\pi B^2\right)$ $+\frac{1}{\ln b}$	$\log_b\left(2\pi\sigma_n^2\right)$ $+\frac{A^2+B^2}{2\sigma_n^2\ln b}+\Lambda(A)$	$\log_b\left(2\pi\sigma_n^2\right)+\frac{1}{\ln b}$ $+D\log_b\left(\frac{A^2}{D\sigma_n^2}+1\right)$	$\log_b\left(2\pi\sigma_n^2\right)+D\Lambda\left(\frac{A}{D}\right)+$ $+\frac{2A^2+\left(1+\overline{D}\right)\sigma_n^2}{2\sigma_n^2\ln b}$

[1] To simplify the Table, $B^2 = A^2 + \sigma_n^2$. [2] To simplify the Table, $C^2 = A^2 \cdot \sigma_n^2$.

However, these are not popular for the radio-frequency interference signal use case, as the data are primarily complex-valued.

An interesting approach to improve the transform domain is to clean it. For example, singular-value-decomposition-based filtering can suppress noise components in the transform and improve later ML training [65].

Several transform approaches are presented in this section. Any of these could be used for preprocessing the data. However, the STFT is selected for the remainder of the article, as it is the most processing-efficient approach. This property makes it especially suitable for low SWAP applications. More advanced approaches such as the KLT or CWD could improve classification accuracy, but the additional overhead they need makes them impractical.

IV. SIMPLIFIED SIGNAL MODEL

In this section, five simplified signal models are presented. Later in the complex signal model, the approximation is made that complex signals are composites of these simplified signals. This approach provides a simplified approach to building complexity with simple building blocks and provides an intuitive explanation of the expected outcomes.

Appendix A derives the statistical properties of the five signal models and Table I shows a summary of them. It helps to compare the expected results of the five signals directly. The subsequent sections define the signals and comment on the possibility to distinguish them based on the values in Table I.

A. No-Interference Model

The no-interference model $x_\omega[n]$ assumes a complex Gaussian noise signal, i.e., the GNSS signal is below the noise floor

$$x_\omega[n] = \omega[n] \tag{19}$$

$$\{\Re\{\omega[n]\}, \Im\{\omega[n]\}\} \sim \mathcal{N}\left(0, \frac{\sigma_n}{\sqrt{2}}\right) \tag{20}$$

where $x_\omega[n]$ is the no-interference signal that contains the noise signal $\omega[n]$, $\Re\{\cdot\}$ is the real operator, $\Im\{\cdot\}$ is the imaginary operator, and $\mathcal{N}(0, \frac{\sigma_n}{\sqrt{2}})$ denotes a zero-mean Normal distribution with standard deviation $\frac{\sigma_n}{\sqrt{2}}$.

B. Gaussian Envelope Constant Transmission

The Gaussian envelope constant transmission (GC) signal model for a Gaussian envelope (i.e., the signal is noiselike) with constant transmission x_{GC} (i.e., it is always transmitting or "ON") is

$$x_{GC}[n] = A s_{GC}[n] + \omega[n] \tag{21}$$

$$\{\Re\{s_{GC}\}, \Im\{s_{GC}\}\} \sim \mathcal{N}\left(0, \frac{1}{\sqrt{2}}\right). \tag{22}$$

It is clear from the statistics in Table I that the energy $s_e(x_{GC}[n])$ and the entropy $s_h(x_\omega[n])$ can separate the GC signal from the no-interference signal. The kurtosis $\kappa_c(x_{GC}[n])$ gives the same output as the no-interference signal and is not helpful for this signal case.

C. Constant Envelope Constant Transmission

The constant envelope constant transmission (CC) signal model for a constant envelope (i.e., the signal has a constant gain) with constant transmission (i.e., always ON) x_{CC} is

$$x_{CC}[n] = A s_{CC}[n] + \omega[n] \tag{23}$$

$$s_{CC}[n] = e^{j\phi} \qquad \phi \sim \mathcal{U}(0, 2\pi) \tag{24}$$

where the phase ϕ uniformly distributed $\mathcal{U}(0, 2\pi)$. This model represents frequency-modulated continuous-wave, binary phase-shift keying, binary offset carrier, quadrature phase modulation, and four quadrature amplitude modulation signals and is a popular choice for communication and ranging applications.

As the complex kurtosis is no longer zero-valued, it is an excellent metric to distinguish the CC signal from the no-interference and GC signals. Similarly, the Shannon entropy differs from the no-interference and the GC signals. It shows the potential to be used as an applicable metric. However, as it differs from the other two, a multithreshold approach would be needed.

D. Gaussian Envelope Partial Transmission

The Gaussian envelope partial transmission (GP) is only active for a limited time. It could be, for example, a pulsed signal or a signal fading due to propagation effects. The time when the signal is "ON," i.e., transmitting, can be modeled with a duty cycle D. The duty cycle D relates to the likelihood that the signal is present in any given sample. The signal model for a Gaussian envelope with partial transmission x_{GP} is

$$x_{\text{GP}}[n] = \begin{cases} \frac{A}{\sqrt{D}} s_{\text{GC}}[n] + \omega[n] & \text{with likelihood } D \\ \omega[n] & \text{with likelihood } \overline{D} \end{cases} \quad (25)$$

where $\overline{D}$ is the inverse duty cycle, i.e., $\overline{D} = 1 - D$. The likelihood $\overline{D}$ is enacted for each sample. Note that the signal power A is scaled with the square root of the duty cycle D, allowing the signal to have the same average signal power.

The signal has the same power as the previous signals and is not a useful indicator to distinguish the interference type. The expected value of the kurtosis is positive, i.e., $\kappa_{\text{GP}} \in [0, \infty)$. It is in contrast to the CC model, where $\kappa_{\text{CC}} \in [-1, 0]$. Therefore, an initial assumption shows that by simply considering the sign of the kurtosis, CC and GP signals can be distinguished.

E. Constant Envelope Partial Transmission

The constant envelope partial transmission (CP) model transmits for short periods, similar to the GP. The signal model for a constant envelope with partial transmission x_{GP} is

$$x_{\text{CP}}[n] = \begin{cases} \frac{A}{\sqrt{D}} s_{\text{CC}}[n] + \omega[n] & \text{with likelihood } D \\ \omega[n] & \text{with likelihood } \overline{D}. \end{cases} \quad (26)$$

In Table I, it is clear that the kurtosis is similar to the GP model. Therefore, the partial models are separable from the constant models using the kurtosis. It is expected, as the kurtosis is a good indication of a mixture model. However, as highlighted in the approximations, they are similar, which indicates that the GP and CP will be challenging to separate from each other.

F. Summary of the Five Signal Models

The five signal models represent a simplified classification case. One model has no interference. Two contain continuous interference signals, and two are partially transmitting interference signals. These basic models can be used to create more advanced models. For example, a chirp signal is a CC signal, but if it is deinterleaved through an STFT, it results in several channels with CP characteristics. Furthermore, it allows the bandwidth and chirp rates of the signal to be deduced. This example shows that using simple statistical models of Section III in conjunction with the spectral approaches in Section III-D, advanced classification opportunities are created.

In the next section, these simplified models are easily separable using basic supervised ML techniques. Later in Section VI, advanced processing models are presented and tested in Section VII with more complex ML for superior classification performance.

V. SINGLE-CHANNEL PERFORMANCE EVALUATION

A Monte Carlo simulation verifies the achievable performance with the statistical metrics of Section III and the signal models in Section IV. The simulation generates the five signals with varied INRs and duty cycles D. The INR sweep is crucial, as the received interference power is not constant, yet many studies keep this value static resulting in unrealistic classification scenarios [18], [19]. Although the INR (or commonly the signal-to-noise ratio) is a standard approach to characterize noise statistics, the energy-to-noise ratio (ENR) is more appropriate as it considers integration gain (i.e., how long the signal was observed for). Therefore, the ENR is provided as a reference for more context.

Each signal has $K = 2500$ samples and is generated at a sample rate of $f_s = 2.56$ MHz (i.e., the equivalent integration time is 977 μs). This samplerate is achievable with a low-cost NeSDR SMArt v4 using an RTL2832U digital video broadcasting radio-frequency front-end (RFFE) as used in a previous low-cost GNSS interference detection station [2]. The Monte Carlo does 10^6 runs with each of these configurations.

The metrics are stored, and their pdfs are calculated from the 10^6 values. Next, the pdf of the no-interference signal is compared with the interference signal pdfs to generate the receiver operating characteristic (ROC) for every interference signal. The ROC plots the probability of false alarm P_{FA} versus the probability of detection P_{D} [28] with varied threshold settings (6). It measures the capability of the metric for detection without tuning a threshold. Finally, the area under the curve (AUC) of an ROC curve provides a total summarized performance of a detector. It is defined as

$$\text{AUC} = \int_0^1 P_{\text{D}}\left(\lambda(P_{\text{FA}})\right) \, dP_{\text{FA}} \quad (27)$$

where $\lambda(P_{\text{FA}})$ is the threshold as a function of the probability of false alarm P_{FA}. The AUCs value is ideal for condensing a massive search space to highlight opportunities. An AUC of 1 is the ideal detector (i.e., 100% correct detection), and an AUC of 0 is the inverse of the ideal detector where the sign is inverted (i.e., the pdfs are swapped). These extreme values are desired, but the worst value is an AUC of 0.5, indicating that the performance is not better than just randomly selected outcomes.

Fig. 2 shows the AUC for some selected metrics. The energy detector in Fig. 2(a) shows that an AUC of 1 is reached with an INR above -14 dB (ENR $\geq$ 20 dB). This plot is identical for all four interference signals. Fig. 2(b)–(d) shows the kurtosis for various values. These figures provide exciting insights into how kurtosis is affected by the duty cycle D, the ENR, and the signal type. In particular, the CP signal tends to AUC $= 1$ with small duty cycles and AUC $= 0$ with large duty cycles. Finally, Fig. 2(e) and (f) shows the Shannon entropy metrics. Note that the GC signal cannot detect as the entire plane is white. The CC signal

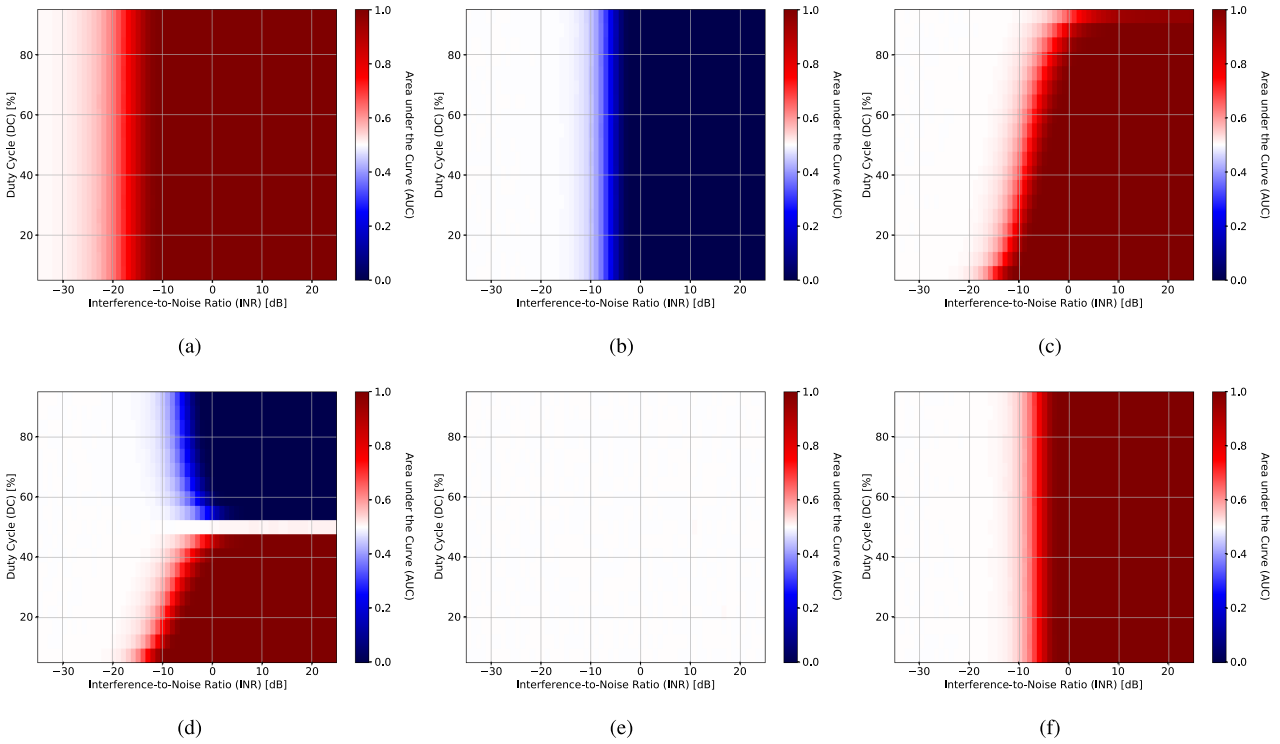


Fig. 2. AUC for different metrics compared to the no-interference case. (a) GC energy. (b) CC kurtosis. (c) GP kurtosis. (d) CP kurtosis. (e) GC entropy. (e) CC entropy.

shows ideal detection INR above -4 dB (ENR $\geq$ 30 dB), indicating entropy is less sensitive than energy.

These examples showed that the metrics could provide classification capabilities if sufficient ENR is available. However, the INR values are negative for the chosen integration time. Furthermore, only interference signals with significantly positive INR affect GNSS signals due to the suppression capabilities of the underlying code-division multiple access structure. Therefore, the potential classification capabilities are more sensitive than what is needed for receiver protection.

In the next stage, a subset of the metrics of the Monte Carlo simulation is used for a supervised ML approach to verify classification capabilities. A decision tree (DT) [66] with 75% of the dataset as training and the remaining 25% for testing is used. The Gini impurity criterion is selected for decision-making (splitting) [66]. A max depth of 4 nodes (i.e., limiting the DT complexity) and without depth restriction (i.e., use as many nodes as necessary, but risk overfitting). A depth of 4 nodes is the minimal complexity needed to separate the five selected classes and is the simplest DT. A DT classifies the data by following a tree of threshold comparisons. The node it ends up on is the final class it selects. Fig. 3 shows an example of the DT applied to the data for the first four decisions.

Two scenarios are considered. First, a simple case with INR values of only 0 dB and +10 dB (ENR of 34 dB and 44 dB) and a 50% duty cycle for pulsed signals with a total of 1.2×10^6 samples. This scenario presents the ideal case

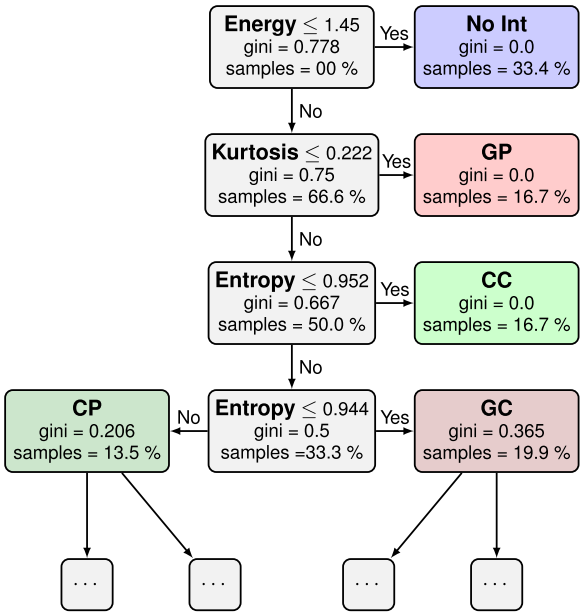
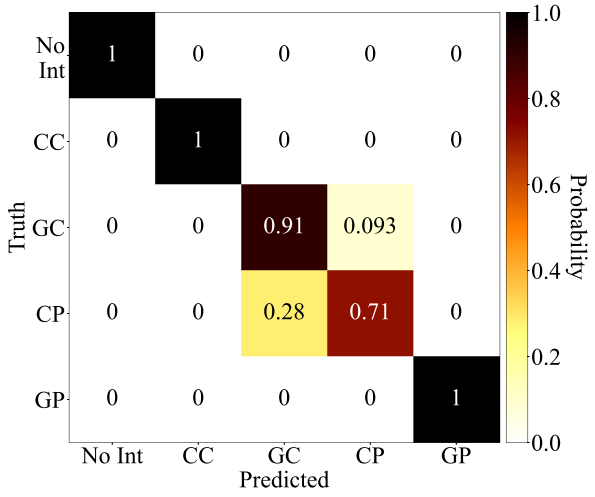


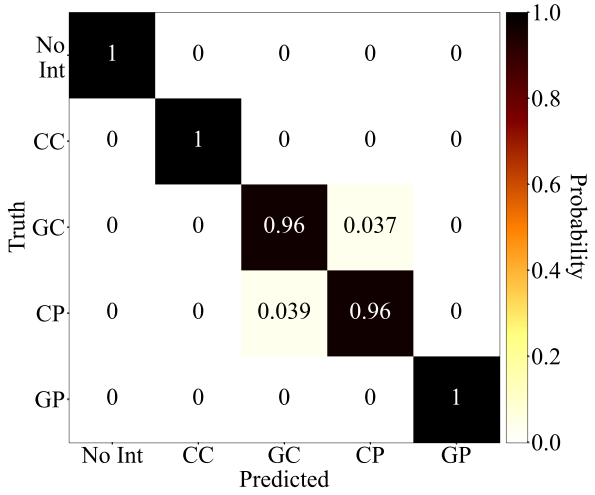
Fig. 3. DT logic for a simple tree with example data.

with limited INR values. All classes have the same number of samples, resulting in a balanced classifier process.

Fig. 4 shows the confusion matrix for the simple case. The max depth four and no depth restriction perfectly classify the no-interference, CC, and GP. However, the GC and CP have reduced classification accuracy. The node depth

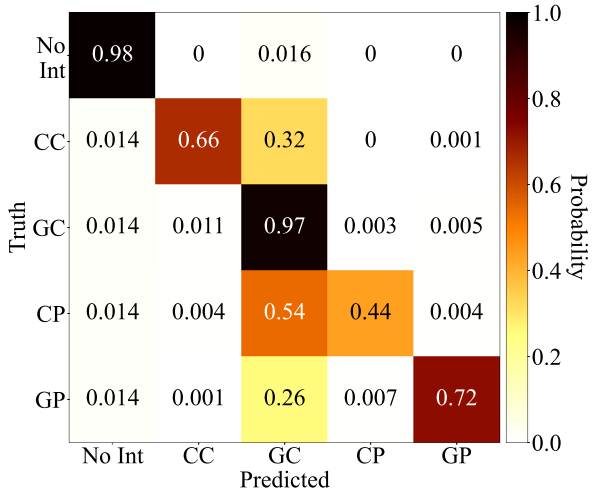


(a)

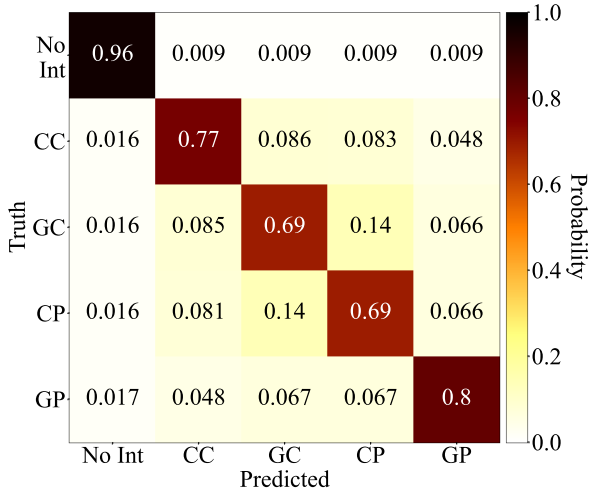


(b)

Fig. 4. Confusion matrix for the simple DT. (a) Max depth 4. (b) No depth restriction.



(a)



(b)

Fig. 5. Confusion matrix for the complete DT. (a) Max depth 4. (b) No depth restriction.

restriction performs better with 96% correct classification and an overall F1-Score of 0.987, compared to an overall F1-Score of 0.937 for the max depth four. Both classifiers demonstrated excellent performance, but the data only contains high INR, which is easier to classify.

Second, the complete case considered the even INRs from -12 dB to 12 dB (ENR of 22 dB and 46 dB) and a 50% duty cycle for pulsed signals with a total of 7.8×10^6 samples. In this case, the DT is trained over an extensive range of data and represents a more realistic case for GNSS interference monitoring deployments. Furthermore, it also included signals that are weak and challenging to classify.

Fig. 5 shows the confusion matrices for the complete case. The classification is significantly worse compared to the simple case. The no-depth restriction does better with an F1-Score of 0.813, compared to the max depth 4, with a score of 0.792. In both cases, the no-interference class does excellent, with a correct classification exceeding 96%, indicating that interference detection is much better than classification. Similarly, incorrect classification is more common between the four interference classes.

These two DT cases indicated that the simple metrics of Section III are sufficient for classifying the classes of Section IV. However, the considerable uncertainty is the required ENR for reliable classification. Therefore, a third test with the DTs is done. A DT is trained for each INR and duty-cycle D separately. The classifier has the INR and duty cycle as prior information, so this is considered a practical performance upper bound. The harmonic F1-Score for each DT is calculated and is defined as [67]

$$F_1(c) = \frac{2P_t}{2P_t + N_t + N_f} \tag{28}$$

$$\mu_H = N_c \cdot \left(\sum_{c=0}^{N_c} F_1^{-1}(c) \right)^{-1} \tag{29}$$

where $F_1(c)$ is the harmonic F1-Score of the cth class, P_t is the true positives (correct classification), N_t is the true

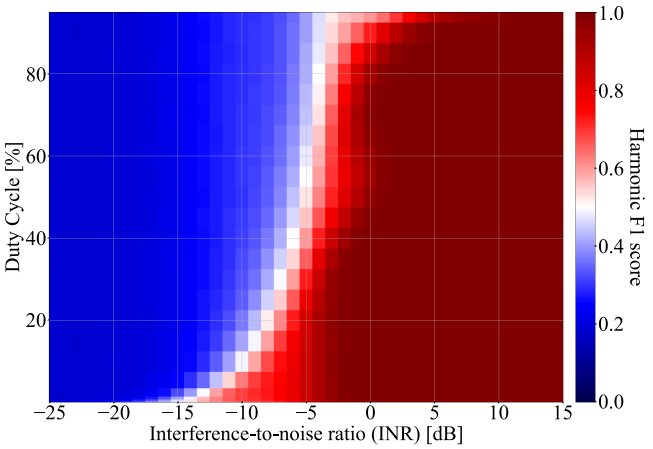


Fig. 6. Harmonic F1-Score of DTs trained under multiple INR and duty cycle conditions.

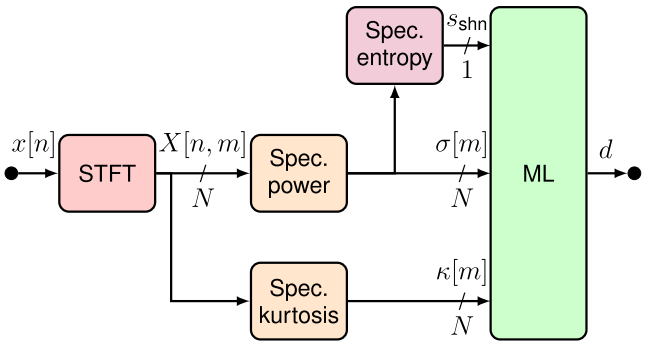


Fig. 7. Block diagram for the signal pre-processing for the advanced processing model.

negatives (correct assigning to a different class), N_f is the false negatives (incorrect assigning to a different class), and μ_H is the harmonic F1-Score over all classes N_c.

Fig. 6 shows the harmonic F1-Score for each DT. This graph is handy for analyzing the classification performance. First, it can be seen that an INR exceeding 0 dB (ENR $\geq$ 34 dB) is required to achieve F1-Scores near 1. Second, low-duty cycle signals, e.g., $D < 20\%$, are much easier to classify at low INR. For example, with $D = 1\%$, only -10-dB INR is needed to achieve an F1-Score of 0.7. Similarly, as the duty cycle approaches 100%, classification performance deteriorates. For example, with $D = 99\%$, at least 5-dB INR is needed to achieve an F1-Score exceeding 0.99. Comparing these two examples shows a difference of 15 dB in signal power is needed depending on the signal type.

Fig. 6 is a fantastic benchmark for the expectation of classification performance for simple signals. The results show that an INR of 0 dB in most cases is sufficient to classify the interference signals accurately. However, the integration gain should be correctly considered, as the number of samples K and the sample time T_s impact these results. Furthermore, basic supervised ML approaches are deployed for this evaluation. In particular, DTs are known to overfit the data and perform inferior to more advanced methods such as RF [68]. Nevertheless, they present a minimum viable proof of concept.

In the next section, more complex signal models and processing chains are considered based on the observations up until now.

VI. PROCESSING MODEL

Only a single channel is characterized, and the base performance expectations are shown so far. In this section, an advanced architecture that uses spectral processing is presented. This architecture amplifies the capabilities of the single channels and allows for superior classification, as will be demonstrated in Section VII. Furthermore, this architecture only saves the statistics of the data, resulting in considerably fewer metrics (features) that are necessary and used by the ML.

Fig. 7 shows the processing model. Each signal has $K = 51200$ samples and is generated at a sample rate of $f_s = 2.56$ MHz (i.e., the equivalent integration time is 20 ms), and the samples are passed through an STFT of size $N = 64$.

The resultant sizes for each channel are

$$f_s' = \frac{f_s}{N} = 40\ \text{kHz} \qquad K' = \left\lfloor \frac{K}{N} \right\rfloor = 800. \tag{30}$$

The spectral power and spectral kurtosis are determined for each channel. As the Shannon entropy showed less usefulness in the previous tests (see Section V) and requires more complex processing by determining pdfs, it is omitted. However, the spectral entropy over all the $N = 64$ channels is determined.

In total, 129 metrics (features) are extracted: 64 for spectral energy, 64 for spectral kurtosis, and 1 for spectral entropy. This architecture is efficient and ideal for low SWAP applications and has been proven successful in running in real time on an SBC [2], [11].

Appendix B derives the computational complexity for the algorithms and applies this to the data sizes. A total of 1.980 MOps/Snapshot is required for the preprocessing and inference (i.e., for classification). As a comparison a CNN using a ResNet-50 architecture [19], would require at least 2104.63 MOps/Snapshot. Therefore, the presented architecture uses three orders of magnitude fewer operations than a CNN. It demonstrates the efficiency of this method and it explains why the classification can efficiently run in real time on an SBC without hardware acceleration, as opposed to CNN, which often requires full computer setups, graphics processing unit (GPU) acceleration, or ML accelerators. Furthermore, the presented method uses 129 real-valued features for ML, compared to the $K = 51\,200$ real values of the STFT spectrogram that an equivalent CNN would use, resulting in two orders of magnitude fewer features used for ML. It significantly reduces the ML complexity. Considering these reductions in the number of operations, the capability to run on smaller computing platforms, and the reduction in the feature space, explain why the presented methods are considered as low SWAP methods [2].

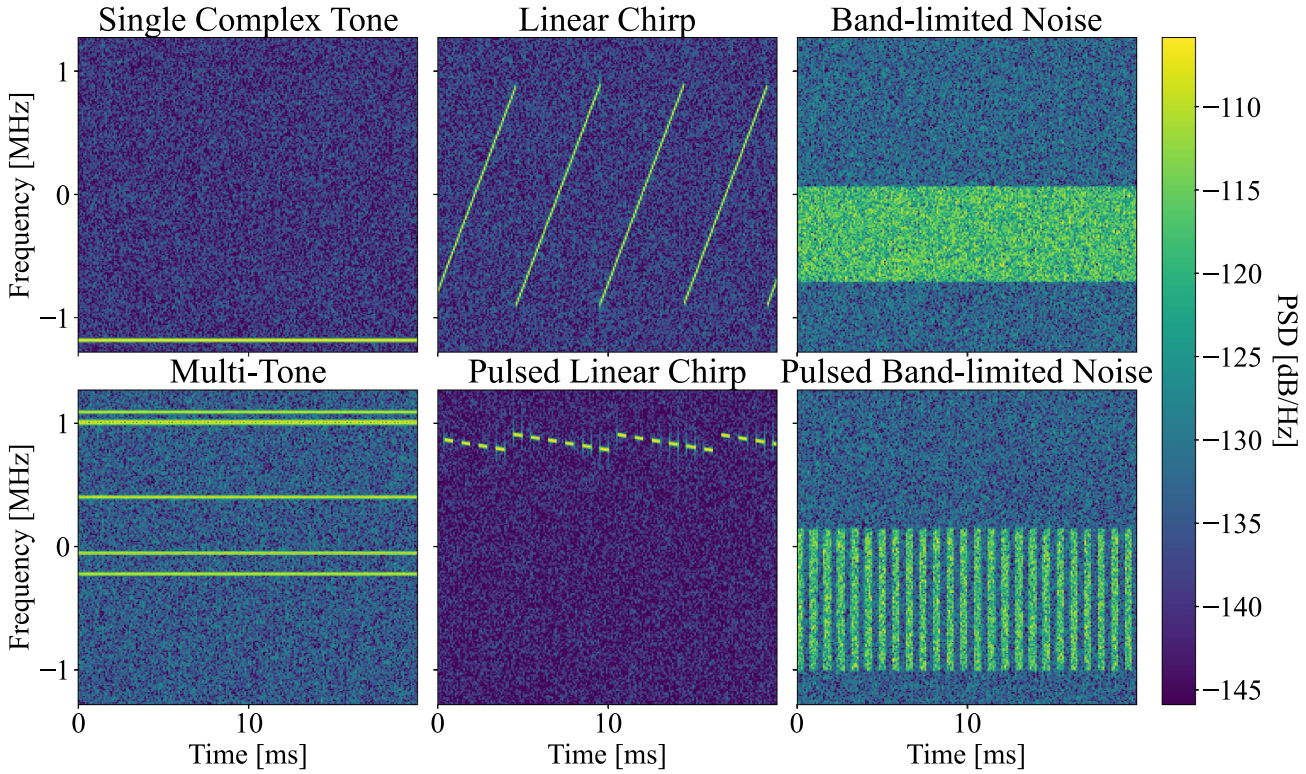


Fig. 8. Example spectrograms of six interference signals generated with an INR of 10 dB and with a 50% duty cycle for the pulsed signals (bottom center and bottom right).

The framework could be extended to include the Shannon entropy per channel. However, this is considered future work. Furthermore, the metrics do not consider interchannel relations, which results in a significant loss of context. Therefore, a CNN architecture outperforms this approach in the ideal training case. Therefore, another suggestion is to add interchannel features such as correlation or mutual information [69]. Nevertheless, this will increase complexity but is an exciting topic for future research. The tradeoff between performance and low-SWAP is an important practical consideration for developing real-world ML deployments, which is often overlooked in theoretical research. Finally, multimodel approaches to use both the statistical features and the CNN [19], [20] could further improve performance but with even more processing overhead.

VII. COMPLEX SIGNAL PERFORMANCE EVALUATION

A dataset of complex signals is generated for more realistic signal conditions and made available [70]. The dataset comprises 73.2×10^6 signals with the same data considerations of $K = 51\,200$ and $f_s = 2.56$ MHz. The INR is generated from -35 dB to 25 dB in 1 dB steps (61 parameter values). Eight equally distributed interference signals are generated (i.e., 7.32×10^6 per class), and twice as many noninterference values (14.64×10^6 in the class). The interference signals are generated with random properties (start phase, start frequency, etc.) to facilitate signal diversity for ML training. The following interference classes are generated.

1) Single complex tone with each signal having a random in-band center frequency.
2) Multitone signal, a composite between two to six random in-band tones.
3) Linear chirp with a random center frequency and bandwidth, but limited to always being in-band and avoiding aliasing effects. The bandwidths are generated with uniform distribution and range between 80 kHz and 2.56 MHz, and the chirp repetition times vary between 16.4 and 330 μs.
4) In-band band-limited noise with bandwidths uniformly selected to be between 640 kHz and 1.6 MHz. The center frequencies are also randomized.

5)–8)Pulsed signals. A combination of any of the previous signals (single tone, multitone, chirp, and noise) and a pulse mask with a random pulsing period. The pulse widths vary between 1.17 μs (equivalent to at least three samples per pulse) and 2 ms. The duty cycles range from 0.1% to 95% (24 parameter values).

Therefore, nine classes exist: four continuous interference classes, four pulsed interference classes, and the no-interference class. Each INR and duty cycle set has 5×10^3 data points per interference class and 10^4 for the noninterference class. Fig. 8 shows the spectrograms of six example signals from the dataset (the spectrogram is the

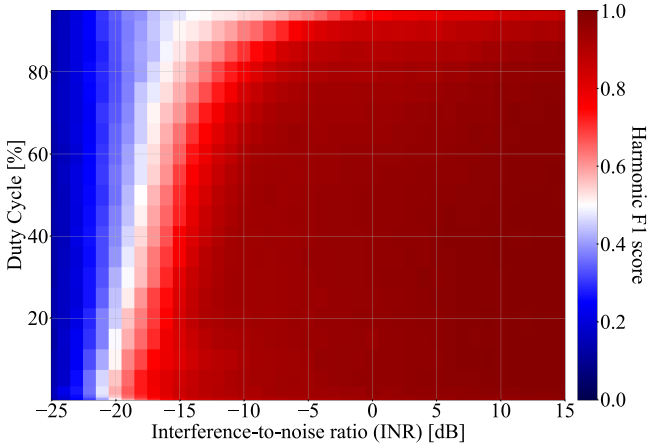


Fig. 9. Harmonic F1-Score of XGBoost Trees trained under multiple INR and duty cycle conditions.

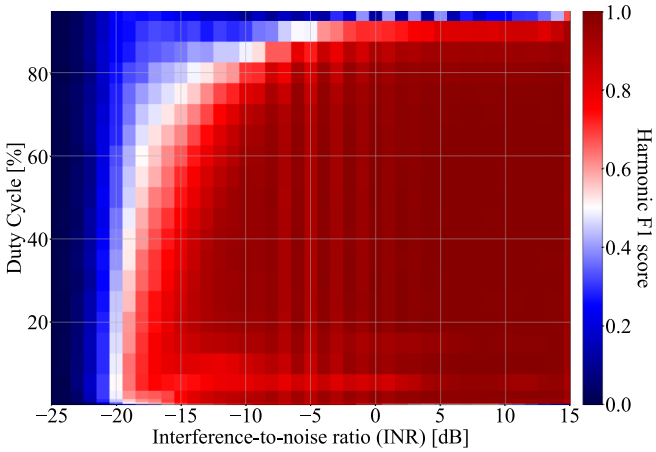


Fig. 10. Harmonic F1-Score of a single XGBoost tree trained under a selection of INR and duty cycle conditions but evaluated over the entire dataset.

absolute squared values of the STFT that removes phase information).

The generated signals are processed with the architecture described in Section VI, and the 129 features are extracted and used for a supervised ML. The data are split into 75% for training and 25% for testing.

A more sophisticated ML algorithm than the DT was used to improve performance due to the increased feature space. Using several parallel DTs with an ensemble improves performance and limits overfitting. The most popular DT ensemble method is an RF. An extreme gradient boosting (XGBoost) further extends the RF and is used as it has previously shown improved performance compared to traditional RFs [71]. XGBoost leverages an ensemble of modified DTs known as classification and regression trees (CARTs) [72]. In contrast to traditional DTs, CARTs include a score within their decision nodes that enhances the optimization and overall decision-making of the ensemble. Additionally, the learning is done via *Boosting*, where the objective function is optimized given the weighted sum of the CARTs, i.e., the ensemble decision. Finally, the max depth of each tree is set at $D_T = 129$ to allow all the features to be part of the decision process.

A. Single INR and Duty Cycle Training

An XGBoost classifier is trained for each INR and duty cycle value. Fig. 9 shows the harmonic F1-Score for each XGBoost classifier, similar to Section V.

Even though more classes with more variety exist, the XGBoost classifier significantly improves the simple DT in Section V. This improvement is first contributed to the segregation capabilities of the FFT, which is particularly effective for distinguishing narrowband signals from wideband signals. Additionally, the FFT segregates the signal, which in the ideal case, where only one bin is affected by the interference, results in $10 \times \log_{10} 32 = 15$ dB improvement of the INR. Second, the XGBoost is known to be significantly better than basic DTs or other RF approaches.

Fig. 9 shows the harmonic F1-Score; it can be seen that an INR exceeding -10 dB (ENR $\geq$ 24 dB) is required to achieve F1-Scores near 1. Most uncertainty comes from distinguishing between the interference signal and its pulsed version (e.g., between a single tone and a pulsed single tone), as with the single channel result that low-duty cycle signals are easier to classify.

B. Multiple INR and Duty Cycle Training

The previous section assumed prior knowledge of the INR and duty cycle as a unique XGBoost classifier is trained for each setting. In this section, a selection of INR and duty cycle values is extracted to train a model (this was mostly done due to memory limitations on the target training platform). However, the entire INR and duty cycle parameter space is used for testing. A grid consisting of 11 INR values by 18 duty cycle values is extracted to use 198 parameter pairs of the total 1032 pairs (i.e., 19.2% of the INR and duty cycle values).

Fig. 10 shows the results for this classifier. The horizontal and vertical stripes in the diagram show where the generalization of the network is underperforming to INR and duty cycle values outside the training set. It demonstrates that generalization is not trivial, as performance rapidly decreases with INR and duty cycle values outside the training dataset. Therefore, it is crucial to have realistic INR and duty cycle values in a training dataset for a practical system.

Furthermore, as a single XGBoost tree is trained for all data, and no prior information on the INR and the duty cycle is used, the results are inferior to Fig. 9. It is expected. Nevertheless, decent classification results are achieved with F1-Scores exceeding 90% with negative INR values.

VIII. DISCUSSION OF RESULTS

These results show that combining statistical spectral features and XGBoost results in excellent performance. Furthermore, the XGBoost trains significantly faster and more stable than typical NNs, making the classifier more

robust and practical for rapid prototyping. The entire classifier only requires 129 features, which is several orders of magnitude fewer than what a CNN on the same data would require, resulting in a reduced feature space. However, a CNN comparison in terms of performance is suggested as future work to verify and quantify performance and operational complexity. The reduced feature space simplifies ML training and allows for low transmission needs for a remote receiver in the case of data gathering. These observations emphasize that robust and low SWAP-C systems can be developed using signal processing principles, as introduced in Fig. 1.

Negative INR values are easily classifiable. On the surface, this looks like excellent results, especially as such low INR levels would barely affect the GNSS signals. However, if the ENR is considered, then ENR levels below 14 dB are extremely difficult to work with. This demonstrates first that the Cramer–Rao bound [28] is still quite a distance away. Second, the effect of the integration gain greatly misleads the perception of the achievable performance. Therefore, it emphasizes the need to present both the INR and ENR for a meaningful performance comparison. Furthermore, it highlights that superior performance is possible with larger snapshots at the cost of processing resources.

The most prominent limitation of this study is that only simulated data were used. Therefore, several doubts about the real-world representation of the simulation environment are raised, such as multipath effects on the interference signals, receiver front-end degradation, quantization effects, dynamic range limitations, and the presence of the GNSS signals. However, a simulated environment allows for systematic testing of the algorithms to show baseline performance in the ideal scenario. Furthermore, previous studies have shown that the same processing pipeline, including GNSS receiver outputs, can accurately perform interference classification in real-world scenarios with multipath effects on the interference signals [2]. Other applications, such as classifying unknown interferences [21], enhanced detection with MTL [20], and collaborative localization [2], [11], are also possible with the same architecture. However, these studies focus on the ML and system aspects and do not present the underlying theoretical signal processing concepts that this article introduces.

A second limitation is that only unaliased data was considered. This limitation is particularly challenging, as a relatively narrowband RFFE is assumed. Most chirp signals used by PPDs will exceed the 2.56-MHz receiver bandwidth and appear pulsed. A detailed investigation of aliasing effects on classification is suggested as future research. Furthermore, if a wideband receiver is assumed [16], the same concepts could be applied with a low likelihood of aliasing.

Finally, this study only considered interference signals that result in denial of service of the GNSS signals. Spoofing [73], which purposefully misleads a GNSS receiver, is more complex and considered outside the scope of this article.

IX. CONCLUSION

This article introduces ML approaches for low SWAP and rapid prototyping interference classification. It suggests a framework for evaluating the tradeoff between machine and human effort. This tradeoff highlights that low SWAP approaches could benefit from using conventional statistical signal processing to limit the ML complexity and design space.

The single-channel results proved that statistical metrics are sufficient for basic classification. These results provided benchmarks for classification, including which INR values are needed for accurate classification and how the duty cycles affect classification accuracy. These insights help develop improved pipelines.

Next, a more advanced pipeline that includes spectral methods is introduced. It builds on the single-channel approach. The results of the more advanced pipeline indicate that complex interference signal types, such as band-limited noise and chirp signals, can accurately be classified. The results show that with well-designed statistical signal processing, significantly fewer features are required for good classification capability. It unburdens the training task for ML and could use significantly simpler ML architectures. It paves the way for low SWAP-C architectures with stricter constraints and cannot facilitate more heave approaches, such as a CNN.

This article focused on signal preprocessing before ML. Therefore, more emphasis on ML optimization is suggested for future research. Furthermore, the existing pipeline does not provide interchannel context, which may improve classification capability. Such approaches are also suggested for future research.

APPENDIX

A. Statistical Derivations

This appendix derives the statistical features for complex signals. They are additionally summarized in Table I.

1) *No-Interference Model:* It is trivial to show that the signal has zero mean and σ_n^2 variance [28], as shown in the second column of Table I

$$\mathrm{E}\left[x_\omega[n]\right] = 0 \tag{31}$$

$$\mathrm{VAR}\left[x_\omega[n]\right] = \mathrm{E}\left[x_\omega[n]x_\omega^*[n]\right] = \sigma_n^2. \tag{32}$$

The instantaneous power $p_\omega[n] = x_\omega[n]x_\omega^*[n]$ of the signal follows a central chi-squared distributed variable with $\nu = 2$ degrees of freedom, i.e., $p_\omega[n] \sim \chi_2^2$. Similarly, the mean power $s_\mathrm{p}(x_\omega[n])$ also follows a central chi-squared distributed variable, but with $\nu = 2K$ degrees of freedom χ_{2K}^2 and is scaled by the signal variance σ_n^2

$$s_\mathrm{e}(x_\omega[n]) = \sum_{k=1}^{K-1} p_\omega[k] \tag{33}$$

$$= \sum_{k=1}^{K-1} x_\omega[k]x_\omega^*[k] \sim \frac{\sigma_n^2}{2} \cdot \chi_{2K}^2 \tag{34}$$

$$s_{\mathrm{e}}(x_\omega[n]) = T_{\mathrm{s}}K \cdot \mathrm{VAR}\left[x_\omega[n]\right] = T_{\mathrm{s}}K\sigma_n^2 \tag{35}$$

where $p_\omega[k]$ is the instantaneous power of the signal. The fourth statistical moment can be related to the power signal $p_\omega[k]$ and the applicable chi-squared nature and shows that the complex excess kurtosis is zero

$$\mathrm{E}\left[|x_\omega[n]|^4\right] = \mathrm{E}\left[p_\omega^2[n]\right] = 2\sigma_n^4 \tag{36}$$

$$\kappa_{\mathrm{c}}(x_\omega[n]) = \frac{\mathrm{E}\left[|x_\omega[n]|^4\right]}{\mathrm{E}\left[|x_\omega[n]|^2\right]^2} - 2 = \frac{2\sigma_n^4}{\sigma_n^4} - 2 = 0. \tag{37}$$

Finally, the Shannon entropy is

$$s_{\mathrm{h}}(x_\omega[n]) = \mathrm{E}\left[-\log_b P(X_\omega)\right] = \log_b\left(2\pi\sigma_n^2\right) + \frac{1}{\ln b}. \tag{38}$$

These metrics provide a benchmark for the statistical expectations for the interference signals.

2) *Gaussian Envelope With Constant Transmission:* The signal statistics are trivial to derive, as adding two Gaussian signals results in a Gaussian signal. Therefore, the derivations of the previous section can be applied.

3) *Constant Envelope With Constant Transmission:* The statistics are also part of standard models

$$\begin{aligned}\mathrm{VAR}\left[x_{\mathrm{CC}}[n]\right] &= \mathrm{E}\left[A^2 + \omega[n]\omega^*[n]\right. \\ &\quad \left. +2A \cdot \Re\left\{s_{\mathrm{CC}}[n]\omega^*[n]\right\}\right] \qquad (39)\\ &= A^2 + \sigma_n^2. \qquad (40)\end{aligned}$$

The power signal $p_{\mathrm{CC}}[k]$ follows a noncentral chi-squared distribution, which affects the fourth-order moment and the kurtosis

$$s_{\mathrm{e}}(x_{\mathrm{CC}}[n]) = \sum_{k=1}^{K-1} p_{\mathrm{CC}}[k] \sim \frac{\sigma_n^2}{2} \cdot \chi^2_{2K}\left(\lambda_{\mathrm{CC}}\right) \tag{41}$$

$$\lambda_{\mathrm{CC}} = \frac{A^2}{\sigma_n^2/2}\cos^2(\phi) + \frac{A^2}{\sigma_n^2/2}\sin^2(\phi) = \frac{2A^2}{\sigma_n^2} \tag{42}$$

$$\mathrm{E}\left[|x_{\mathrm{CC}}[n]|^4\right] = \mathrm{E}\left[p_{\mathrm{CC}}^2[n]\right] = A^4 + 4A^2\sigma_n^2 + 2\sigma_n^4 \tag{43}$$

where λ_{CC} is the noncentrality parameter. It shows that the complex excess kurtosis is no longer zero-valued

$$\kappa_{\mathrm{c}}(x_{\mathrm{CC}}[n]) = \frac{A^4 + 4A^2\sigma_n^2 + 2\sigma_n^4}{\left(A^2 + \sigma_n^2\right)^2} - 2 \tag{44}$$

$$= \frac{-A^4}{\left(A^2 + \sigma_n^2\right)^2}. \tag{45}$$

Finally, the Shannon entropy can be shown to be

$$s_{\mathrm{h}}(x_{\mathrm{CC}}[n]) = \log_b\left(2\pi\sigma_n^2\right) + \frac{2A^2 + \sigma_n^2}{2\sigma_n^2 \ln b} + \Lambda(A) \tag{46}$$

$$\Lambda(A) = \mathrm{E}\left[\log_b\left(I_0\left(\frac{A}{\sigma_n^2}\left|x_{\mathrm{CC}}[n]\right|\right)\right)\right] \tag{47}$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind and order 0, and $\Lambda(A)$ is, to the authors' knowledge, an unsolved expectation. The determination of $\Lambda(A)$ is proposed as a future problem to solve.

4) *Gaussian Envelope With Partial Transmission:* This signal is a Gaussian mixture model, which makes some analyses more difficult. However, it can be defined as the combination of the other models

$$\mathrm{E}\left[x_{\mathrm{GP}}[n]\right] = D\,\mathrm{E}\left[x_{\mathrm{GC}}[n]\right] + \overline{D}\,\mathrm{E}\left[x_\omega[n]\right] = 0 \tag{48}$$

$$\mathrm{E}\left[|x_{\mathrm{GP}}[n]|^2\right] = D\,\mathrm{E}\left[|x_{\mathrm{GC}}[n]|^2\right] + \overline{D}\,\mathrm{E}\left[|x_\omega[n]|^2\right] \tag{49}$$

$$= A^2 + \sigma_n^2. \tag{50}$$

As the "ON" portion has more instantaneous power but is "ON" for a shorter time, it shows that this signal will have the same mean power or expected power as the previous interference examples.

Similarly, the fourth statistical moment can also be determined as a linear combination of the separate values

$$\mathrm{E}\left[|x_{\mathrm{GP}}[n]|^4\right] = D\,\mathrm{E}\left[|x_{\mathrm{GC}}[n]|^4\right] + \overline{D}\,\mathrm{E}\left[|x_\omega[n]|^4\right] \tag{51}$$

$$= \frac{2A^4}{D} + 4A^2\sigma_n^2 + 2\sigma_n^4. \tag{52}$$

It shows that the kurtosis κ_{GP} is also composite

$$\kappa_{\mathrm{GP}} = \frac{\mathrm{E}\left[|x_{\mathrm{GP}}[n]|^4\right]}{\mathrm{E}\left[|x_{\mathrm{GP}}[n]|^2\right]^2} - 2 \tag{53}$$

$$= \frac{\frac{2A^4}{D} + 4A^2\sigma_n^2 + 2\sigma_n^4}{\left(A^2 + \sigma_n^2\right)^2} - 2 \tag{54}$$

$$= \frac{2\overline{D}A^4}{D\left(A^2 + \sigma_n^2\right)^2}. \tag{55}$$

Here, are some approximations for different conditions that highlight the underlying nature of the kurtosis for this signal

$$\kappa_{\mathrm{GP}} \approx \begin{cases} 2\overline{D}D^{-1} & \text{if } A \gg \sigma_n \\ 0 & \text{if } A \ll \sigma_n \\ 0 & \text{if } D = 1 \\ \infty & \text{if } D = 0 \end{cases}. \tag{56}$$

Finally, the Shannon entropy is also derived using a composite model

$$P(X_{\mathrm{GP}}) = D \cdot P(X_{\mathrm{GC}}) + \overline{D} \cdot P(X_\omega) \tag{57}$$

$$s_{\mathrm{h}}(x_{\mathrm{GP}}[n]) = D \cdot s_{\mathrm{h}}(x_{\mathrm{GC}}[n]) + \overline{D} \cdot s_{\mathrm{h}}(x_\omega[n]) \tag{58}$$

$$\begin{aligned}&= \log_b\left(2\pi\sigma_n^2\right) + \frac{1}{\ln b} \\ &\quad + D\log_b\left(\frac{A^2 + \sigma_n^2}{\sigma_n^2}\right). \qquad (59)\end{aligned}$$

5) *Constant Envelope With Partial Transmission:* This signal is a composite signal and allows the statistical derivations based on the previous models

$$\mathrm{E}\left[x_{\mathrm{CP}}[n]\right] = D\,\mathrm{E}\left[x_{\mathrm{CC}}[n]\right] + \overline{D}\,\mathrm{E}\left[x_\omega[n]\right] = 0 \tag{60}$$

$$\mathrm{E}\left[|x_{\mathrm{CP}}[n]|^2\right] = D\,\mathrm{E}\left[|x_{\mathrm{CC}}[n]|^2\right] + \overline{D}\mathrm{E}\left[|x_\omega[n]|^2\right] \tag{61}$$

$$= A^2 + \sigma_n^2. \tag{62}$$

Similarly, the fourth statistical moment can also be determined as a linear combination of the separate values

$$\mathrm{E}\left[|x_{\mathrm{CP}}[n]|^4\right] = D\,\mathrm{E}\left[|x_{\mathrm{CC}}[n]|^4\right] + \overline{D}\,\mathrm{E}\left[|x_\omega[n]|^4\right] \tag{63}$$

$$= \frac{A^4}{D} + 4A^2\sigma_n^2 + 2\sigma_n^4. \tag{64}$$

It shows that the kurtosis κ_{GP} is determined

$$\kappa_{\text{CP}} = \frac{\frac{A^4}{D} + 4A^2\sigma_n^2 + 2\sigma_n^4}{\left(A^2 + \sigma_n^2\right)^2} - 2 \tag{65}$$

$$= \frac{\left(\overline{D} - D\right)A^4}{D\left(A^2 + \sigma_n^2\right)^2}. \tag{66}$$

As before, here are the approximations for different conditions

$$\kappa_{\text{CP}} \approx \begin{cases} (1-2D)D^{-1} & \text{if } A \gg \sigma_n \\ 0 & \text{if } A \ll \sigma_n \\ \kappa_{\text{CC}} & \text{if } D = 1 \\ \infty & \text{if } D = 0 \end{cases}. \tag{67}$$

Finally, the Shannon entropy is also composite

$$P(X_{\text{CP}}) = D \cdot P(X_{\text{CC}}) + \overline{D} \cdot P(X_\omega) \tag{68}$$

$$s_{\text{h}}(x_{\text{CP}}[n]) = D \cdot s_{\text{h}}(x_{\text{CC}}[n]) + \overline{D} \cdot s_{\text{h}}(x_\omega[n]) \tag{69}$$

$$= \log_b\left(2\pi\sigma_n^2\right) + D\Lambda\left(\frac{A}{\sqrt{D}}\right) + \frac{2A^2 + \left(1+\overline{D}\right)\sigma_n^2}{2\sigma_n^2 \ln b}. \tag{70}$$

B. Algorithm Complexity

This appendix derives the theoretical computational complexity for the algorithms. An STFT consisting of N time bins and M frequency bins is assumed. Both big-O notation and the theoretical number of operations are shown. The theoretical number of operations considers multiplication, addition, division, and table look-up equally complex. Note that a complex multiplication is equivalent to four real multiplications.

The STFT scales with the number of FFTs N. Each FFT is well known to have $\mathcal{O}_{\text{FFT}}(M \log_2 M)$ complexity

$$\mathcal{O}_{\text{STFT}}(NM \log_2 M) \qquad \mathcal{N}_{\text{STFT}}^{\text{Obs}} = 4NM \log_2 M. \tag{71}$$

The energy scales linearly with the time bins N, and is repeated for each frequency bin M

$$\mathcal{O}_{\text{Energy}}(NM) \qquad \mathcal{N}_{\text{Energy}}^{\text{Obs}} = 3NM. \tag{72}$$

The variance is similar in complexity to the energy, but it must first determine and calculate the mean of the signal, resulting in more operations

$$\mathcal{O}_{\text{Var}}(NM) \qquad \mathcal{N}_{\text{Var}}^{\text{Obs}} = (7N - 1)M. \tag{73}$$

The kurtosis requires the mean and the variance, which makes it even more complex. However, it still scales linearly

$$\mathcal{O}_{\text{Kur}}(NM) \qquad \mathcal{N}_{\text{Kur}}^{\text{Obs}} = 13NM. \tag{74}$$

The spectral entropy is calculated on the spectral power values and only scales linearly with M

$$\mathcal{O}_{\text{En}}(M) \qquad \mathcal{N}_{\text{En}}^{\text{Obs}} = 4M - 1. \tag{75}$$

During an evaluation, XGBOOST, in the worst case, requires checking all the nodes for all the trees. Then, compares the results of all the trees for a final decision (inference). As such, the upper bound of the complexity of XGBOOST depends on the maximum tree depth D_T and the maximum number of trees N_T

$$\mathcal{O}_{\text{XGB}}(D_T N_T) \tag{76}$$

$$\mathcal{N}_{\text{XGB}}^{\text{Obs}} \leq D_T N_T + \sum_{k=1}^{\log_2 N_T} \frac{N_T}{2^k} = (D_T + 1)N_T - 1. \tag{77}$$

It is important to note that N_T is, in most cases, significantly less than the maximum (it converges without needing the maximum number of trees), and each tree does not necessarily reach maximum depth D_T. Furthermore, a decision node may be reached before maximum depth during testing. Therefore, the defined upper bound is pessimistic. For the selected input $D_T = 129$ and $N_T = 9000$, resulting in

$$\mathcal{N}_{\text{XGB}}^{\text{Obs}} \leq 1.169 \text{ kOps/Snapshot}. \tag{78}$$

The total preprocessing complexity, including the STFT, the spectral power, the spectral kurtosis, and the spectral entropy can then be combined. Note that the dominant effect is the STFT

$$\mathcal{N}_{\text{Pre}}^{\text{Obs}} = NM\left(13 + 4\log_2 M\right) + 4M - 1. \tag{79}$$

Using $M = 64$ and $N = 800$, as in the system setup, this then

$$\mathcal{N}_{\text{Pre}}^{\text{Obs}} = 1.896 \text{ MOps/Snapshot}. \tag{80}$$

The big-O complexity remains the same as for the STFT

$$\mathcal{O}_{\text{Pre}}(NM \log_2 M). \tag{81}$$

The combined processing for the preprocessing and XGBOOST is

$$\mathcal{N}_{\text{Total}}^{\text{Obs}} = \mathcal{N}_{\text{Pre}}^{\text{Obs}} + \mathcal{N}_{\text{XGB}}^{\text{Obs}} = 1.895 \text{ MOps/Snapshot}. \tag{82}$$

A single layer in a CNN is determined by the size of the output map $M' \times N'$, where $M' = M - K + 1$ and $N'N - K + 1$ accommodate dimension reduction from the edges in the correlation (i.e., assuming no zero padding or striding). The size of the kernel K and the number of parallel kernels N_k further scale the complexity

$$\mathcal{N}_{\text{Cov}}^{\text{Obs}} = M'N'K^2N_k \quad \mathcal{O}_{\text{Cov}}(MNK^2N_k). \tag{83}$$

Note that it adds significant complexity to each layer. However, the processing can be easily parallelized, making it ideal for parallel processing architectures such as GPUs.

The ResNet-50 CNN used in [19] is added for comparison. It consists of 50 CNN layers [74]. It has many layers with a kernel size of $K = 1$ to optimize its speed. Determining the complexity of each correlation layer results in

$$\mathcal{N}_{\text{ResNet-50}}^{\text{Obs}} = 2,104.63 \text{ MOps/Snapshot}. \tag{84}$$

This calculation omits the complexity of the final NN after the convolutional layers. However, it is already clear that

this architecture is significantly more complex than the proposed approach in this article.

REFERENCES

[1] F. Dovis, *GNSS Interference Threats and Countermeasures* (Ser. Artech House GNSS Technology and Applications Series GNSS Interference, Threats, and Countermeasures). Norwood, MA, USA: Artech House, 2015.

[2] J. R. van der Merwe et al., "Low-cost COTS GNSS interference monitoring, detection, and classification system," *Sensors*, vol. 23, no. 7, Mar. 2023, Art. no. 3452, doi: 10.3390/s23073452.

[3] J. R. Van der Merwe, A. Rügamer, F. Garzia, W. Felber, and J. Wendel, "Evaluation of mitigation methods against COTS PPDs," in *Proc. IEEE/ION Position, Location Navigation Symp.*, 2018, pp. 920–930.

[4] C. J. Hegarty, D. Bobyn, J. Grabowski, and A. Van Dierendonck, "An overview of the effects of out-of-band interference on GNSS receivers," *NAVIGATION: J. Inst. Navigation*, vol. 67, no. 1, pp. 143–161, 2020.

[5] G. X. Gao, "DME/TACAN interference and its mitigation in L5/E5 bands," in *Proc. 20th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2007, pp. 1191–1200.

[6] A. Schütz, T. Kraus, C. A. Lichtenberger, and T. Pany, "A case study for potential implications on the reception of Galileo E6 by amateur radio interference on German highways considering various transmitter-receiver-signal combinations," in *Proc. 34th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2021, pp. 1687–1696.

[7] J. Wendel, C. Kurzhals, M. Houdek, and J. Samson, "An interference monitoring system for GNSS reference stations," in *Proc. 6th ESA Workshop Satell. Navigation Technol.*, 2012, pp. 1–7.

[8] A. Morrison, N. Sokolova, and N. Ahmed, "RFI monitoring in support of safety-critical multi-band GNSS-based systems," in *Proc. Eur. Navigation Conf.*, 2019, pp. 1–6.

[9] S. Bartl, M. Kadletz, P. Berglez, and T. Dusa, "Findings from interference monitoring at a European airport," in *Proc. 34th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2021, pp. 3683–2697.

[10] H. Nasser et al., "GNSS interference detection and geolocalization for aviation applications," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 192–216.

[11] J. Hansen et al., "Initial results of a low-cost GNSS interference monitoring network," in *Proc. Conf.: Positioning Navigation Intell. Transport Syst.*, 2022, pp. 4–13.

[12] J. R. van der Merwe, F. Garzia, A. Rügamer, and W. Felber, "Advanced and versatile signal conditioning for GNSS receivers using the high-rate DFT-based data manipulator (HDDM)," *NAVIGATION: J. Inst. Navigation*, vol. 68, no. 4, pp. 779–797, 2021. [Online]. Available: https://navi.ion.org/content/68/4/779

[13] D. Borio and P. Closas, "Robust transform domain signal processing for GNSS," *NAVIGATION: J. Inst. Navigation*, vol. 66, no. 2, pp. 305–323, 2019.

[14] H. Griffiths et al., "Radar spectrum engineering and management: Technical and regulatory issues," *Proc. IEEE*, vol. 103, no. 1, pp. 85–102, Jan. 2015.

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

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

[17] J. R. van der Merwe, F. Garzia, A. Rügamer, I. C. Vidal, and W. Felber, "Adaptive notch filtering against complex interference scenarios," in *Proc. Eur. Navigation Conf.*, 2020, pp. 1–10.

[18] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, no. 4841, 2019, Art. no. 4841. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4841

[19] I. A. E. Mehr and F. Dovis, "A deep neural network approach for detection and classification of GNSS interference and jammer," Mar. 14, 2023, doi: 10.36227/techrxiv.22212121.v1.

[20] 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. 19–23.

[21] D. Jdidi et al., "Unsupervised disentanglement for post-identification of GNSS interference in the wild," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 1176–1208.

[22] K. Dmitriev, J. Schumann, and F. Holzapfel, "Toward certification of machine-learning systems for low criticality airborne applications," in *Proc. IEEE/AIAA 40th Digit. Avionics Syst. Conf.*, 2021, pp. 1–7.

[23] Z. Ke and H. Vikalo, "Real-time radio modulation classification with an LSTM auto-encoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 4935–4939.

[24] R. Calvo-Palomino, A. Bhattacharya, G. Bovet, and D. Giustiniano, "Short: LSTM-based GNSS spoofing detection using low-cost spectrum sensors," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile, Multimedia Netw.*, 2020, pp. 273–276.

[25] Y. Shin, C. Lee, E. Kim, and T. Walter, "Adopting neural networks in GNSS-IMU integration: A preliminary study," in *Proc. IEEE/AIAA 40th Digit. Avionics Syst. Conf.*, 2021, pp. 1–7.

[26] S. Semanjski, I. Semanjski, W. De Wilde, and A. Muls, "Use of supervised machine learning for GNSS signal spoofing detection with validation on real-world meaconing and spoofing data: Part I," *Sensors*, vol. 20, no. 4, Feb. 2020, Art. no. 1171, doi: 10.3390/s20041171.

[27] B. Pardhasaradhi, R. R. Yakkati, and L. R. Cenkeramaddi, "Machine learning-based screening and measurement to measurement association for navigation in GNSS spoofing environment," *IEEE Sensors J.*, vol. 22, no. 23, pp. 23423–23435, Dec. 2022.

[28] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory* (Ser. Prentice-Hall Signal Processing Series), vol. 2. Hoboken, NJ, USA: Prentice-Hall, 1998.

[29] N. Spens, D.-K. Lee, F. Nedelkov, and D. Akos, "Detecting GNSS jamming and spoofing on Android devices," *NAVIGATION: J. Inst. Navigation*, vol. 69, no. 3, 2022, Art. no. 537. [Online]. Available: https://navi.ion.org/content/69/3/navi.537

[30] R. Bauernfeind and B. Eissfeller, "Software-defined radio based roadside jammer detector: Architecture and results," in *Proc. IEEE/ION Position, Location, Navigation Symp.*, 2014, pp. 1293–1300.

[31] J. N. Gross, C. Kilic, and T. E. Humphreys, "Maximum-likelihood power-distortion monitoring for GNSS-signal authentication," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 1, pp. 469–475, Feb. 2019.

[32] H. Dahrouj et al., "An overview of machine learning-based techniques for solving optimization problems in communications and signal processing," *IEEE Access*, vol. 9, pp. 74908–74938, 2021.

[33] S. Prince, *Computer Vision: Models, Learning, and Inference* (Ser. Computer Vision: Models, Learning, and Inference). Cambridge, U.K.: Cambridge Univ. Press, 2012. [Online]. Available: https://books.google.co.uk/books?id=PmrICLzHutgC

[34] S. M. Kay, *Modern Spectral Estimation: Theory and Application* (Ser. Prentice-Hall Signal Processing Series). Hoboken, NJ, USA: Prentice-Hall, 1999.

[35] M. Nicola, G. Falco, R. M. 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. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4085

[36] K. Najim, E. Ikonen, and A.-K. Daoud, "Chapter 2: Estimation of probability densities," in *Stochastic Processes*. Oxford, U.K.: Kogan Page Science, 2004, pp. 93–166. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781903996553500116

[37] R. D. De Roo, S. Misra, and C. S. Ruf, "Sensitivity of the kurtosis statistic as a detector of pulsed sinusoidal RFI," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 7, pp. 1938–1946, Jul. 2007.

[38] A. J. Schoenwald, D. C. Bradley, P. N. Mohammed, J. R. Piepmeier, and M. Wong, "Performance analysis of a hardware implemented complex signal kurtosis radio-frequency interference detector," in *Proc. 14th Specialist Meeting Microw. Radiometry Remote Sens. Environ.*, 2016, pp. 71–75.

[39] R. Sharma and R. Bhandari, "Skewness, kurtosis and Newton's inequality," *Rocky Mountain J. Math.*, vol. 45, no. 5, pp. 1639–1643, 2015.
[40] J. Antoni, "The spectral kurtosis: A useful tool for characterising non-stationary signals," *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 282–307, 2006.
[41] V. Vrabie, P. Granjon, and C. Servière, "Spectral kurtosis: From definition to application," in *Proc. IEEE 6th Int. Workshop Nonlinear Signal Image Process.*, 2003, pp. 1–6.
[42] S. S. Udmale and S. K. Singh, "Application of spectral kurtosis and improved extreme learning machine for bearing fault classification," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 11, pp. 4222–4233, Nov. 2019.
[43] J. N. Kapur and H. K. Kesavan, *Entropy Optimization Principles and Their Applications*. Dordrecht, The Netherlands:Springer, 1992, doi: 10.1007/978-94-011-2430-0_1.
[44] M. Borowska, "Entropy-based algorithms in the analysis of biomedical signals," *Stud. Logic, Grammar, Rhetoric*, vol. 43, no. 1, pp. 21–32, 2016, doi: 10.1515/slgr-2015-0039.
[45] H. Wang, L. Guo, Z. Dou, and Y. Lin, "A new method of cognitive signal recognition based on hybrid information entropy and D-S evidence theory," *Mobile Netw. Appl.*, vol. 23, pp. 677–685, 2018.
[46] E. C. Muñoz, L. F. P. Martínez, and C. A. Hernandez, "Rényi entropy-based spectrum sensing in mobile cognitive radio networks using software defined radio," *Entropy*, vol. 22, no. 6, 2020, Art. no. 626. [Online]. Available: https://www.mdpi.com/1099-4300/22/6/626
[47] J. Gibson, "What is the interpretation of spectral entropy?," in *Proc. IEEE Int. Symp. Inf. Theory*, 1994, pp. 440–457. [Online]. Available: https://ieeexplore.ieee.org/document/395055
[48] F. Ghaderi, H. R. Mohseni, and S. Sanei, "Localizing heart sounds in respiratory signals using singular spectrum analysis," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 12, pp. 3360–3367, Dec. 2011.
[49] Z. Liu, Z. Han, Y. Zhang, and Q. Zhang, "Multiwavelet packet entropy and its application in transmission line fault recognition and classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2043–2052, Nov. 2014.
[50] A. T. Balaei and A. G. Dempster, "A statistical inference technique for GPS interference detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 4, pp. 1499–1511, Oct. 2009.
[51] W. Zhang, M. Feng, M. Krunz, and A. H. Y. Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
[52] C. Edussooriya, L. Bruton, and P. Agathoklis, "Under-decimated 3D FIR space-time cone filters using DFT polyphase filter banks for attenuation of radio frequency interference," in *Proc. IEEE 54th Int. Midwest Symp. Circuits Syst.*, 2011, pp. 1–4.
[53] R. Wu, W. Wang, L. Li, D. Lu, and L. Wang, "Distance measuring equipment interference suppression based on parametric estimation and wavelet-packet transformation for global navigation satellite systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1607–1617, Aug. 2016.
[54] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru, and A. Yahya, "Image compression techniques in wireless sensor networks: A survey and comparison," *IEEE Access*, vol. 10, pp. 82511–82530, 2022.
[55] A. Vlachaki, I. Nikolaidis, and J. Harms, "Wavelet-based analysis of interference in WSNs," in *Proc. IEEE 41st Conf. Local Comput. Netw.*, 2016, pp. 639–642.
[56] K. Sun, M. Zhang, and D. Yang, "A new interference detection method based on joint hybrid time–frequency distribution for GNSS receivers," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9057–9071, Nov. 2016.
[57] S. Haykin and T. Bhattacharya, "Modular learning strategy for signal detection in a nonstationary environment," *IEEE Trans. Signal Process.*, vol. 45, no. 6, pp. 1619–1637, Jun. 1997.
[58] M. Liu, G. Liao, N. Zhao, H. Song, and F. Gong, "Data-driven deep learning for signal classification in industrial cognitive radio networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3412–3421, May 2021.
[59] D. Vakman, "On the analytic signal, the Teager-Kaiser energy algorithm, and other methods for defining amplitude and frequency," *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 791–797, Apr. 1996.
[60] K. G. Gopan, A. Harsha, L. A. Joseph, and E. S. Kollialil, "Adaptive neuro-fuzzy classifier for 'Petit Mal' epilepsy detection using mean Teager energy," in *Proc. Int. Conf. Adv. Comput., Commun., Inform.*, 2013, pp. 752–757.
[61] D. Wenzel, H. Borsi, and E. Gockenbach, "A new method of partial discharge localization on transformers via the Karhunen-Loeve-transform," in *Proc. 1994 4th Int. Conf. Properties Appl. Dielectric Mater.*, 1994, vol. 2, pp. 622–625.
[62] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4256–4268, Sep. 2012.
[63] W. Jiang, T. Li, and B. Shi, "Classification of surface defects based on improved Gabor filter," in *Proc. 5th Int. Conf. Control, Robot., Cybern.*, 2020, pp. 151–155.
[64] M. S. Manikandan, S. R. Samantaray, and I. Kamwa, "Detection and classification of power quality disturbances using sparse signal decomposition on hybrid dictionaries," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 27–38, Jan. 2015.
[65] L. L. Maynard, "GNSS spoofing detection using machine learning and truncated singular value decomposition," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 1137–1150.
[66] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed., Belmont, CA, USA: Routledge, 1984.
[67] D. T. Murphy, E. Ioup, M. T. Hoque, and M. Abdelguerfi, "Residual learning for marine mammal classification," *IEEE Access*, vol. 10, pp. 118409–118418, 2022.
[68] R. G. Leiva, A. F. Anta, V. Mancuso, and P. Casari, "A novel hyperparameter-free approach to decision tree construction that avoids overfitting by design," *IEEE Access*, vol. 7, pp. 99978–99987, 2019.
[69] B. Guo, S. Gunn, R. Damper, and J. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 522–526, Oct. 2006.
[70] D. Contreras Franco, J. R. van der Merwe, and A. Rügamer, "GNSS processed interference features," Feb. 28, 2023, doi: 10.21227/xxfh-qp91.
[71] J. R. van der Merwe, A. Roas, D. C. Franco, K. Dietmayer, A. Rügamer, and W. Felber, "Exploiting acquisition measurements and spatial processing for improved GNSS spoofing detection and classification with snapshot receivers," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 3497–3527.
[72] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[73] J. R. Van der Merwe, X. Zubizarreta, I. Lukčin, A. Rügamer, and W. Felber, "Classification of spoofing attack types," in *Proc. Eur. Navigation Conf.*, 2018, pp. 91–99.
[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 770–778.