Dual-Stage Deep Learning Approach for Efficient Interference Detection and Classification in GNSS

Dual-Stage Deep Learning Approach for Efficient Interference Detection and Classification in GNSS

Iman Ebrahimi Mehr, Outi Savolainen, Laura Ruotsalainen, Fabio Dovis
Politecnico di Torino, Department of Electronics and Telecommunications, Turin, Italy
University of Helsinki, Department of Computer Science, Helsinki, Finland

BIOGRAPHY

Iman Ebrahimi Mehr received the B.Sc. in electronic engineering from Azad University, Tehran, Iran, and his M.Sc. Degree in ICT for smart society from Politecnico di Torino, Turin, Italy, in 2012 and 2021, respectively. He is currently working on his Ph.D. in 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.

Outi Savolainen received the B.Sc. in computer science and M.Sc. in data science from the University of Helsinki, in 2021 and in 2023, respectively. Since autumn 2023, she has been a doctoral researcher at Spatiotemporal Data Analysis research group at the Department of Computer Science at the University of Helsinki, Finland. Her current research interests are Bayesian deep learning, explainable AI, and GNSS signal anomaly detection and classification.

Laura Ruotsalainen is a Professor of Spatiotemporal Data Analysis for Sustainability Science at the Department of Computer Science at the University of Helsinki, Finland. She leads a research group in spatiotemporal data analysis for sustainability science (SDA) which performs research on GNSS techniques, estimation, machine learning and computer vision methods using spatiotemporal data for sustainable smart cities especially via smart mobility. She is a member of the steering group of the Finnish Center for AI (FCAI).

Fabio Dovis received the M.Sc. and Ph.D. degrees 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.

ABSTRACT

Radio-frequency interference (RFI) can degrade performance or completely prevent the estimation of position, velocity, and time by the receiver. Recently, there has been an increasing trend of jamming events and at the same time, our society is relying heavily on GNSS-based solutions in a wide range of different application domains. To ensure the reliability of these solutions, it is crucial to recognize the possible interference even in low power levels and in the early phase of the signal processing. This can lead to applying the mitigation techniques correctly, by identifying the type and power of the interference. In this article, we introduce a novel approach that operates at the raw signal level (In-phase/Quadrature signal samples) and combines the strengths of two network architectures. The proposed dual-stage architecture consists of a Long Short-Term Memory based autoencoder for GNSS interference detection and Convolutional Neural Networks for classification. This selective approach minimizes the inference and processing time in the real-time monitoring system by computing a spectrogram and extracting the statistical signal features for classification only after detecting the interference in the first stage. Our proposed monitoring system is designed to recognize 16 types of RFI and further classify the power of received interference. Our detection and classification models achieved overall accuracies of 100% and 98.26%, respectively.

I. INTRODUCTION

In Global Navigation Satellite Systems (GNSS), where precise positioning and timing are fundamental to various applications ranging from navigation and transportation to telecommunications and emergency services, interference poses a significant threat. It is well known that GNSS signals are vulnerable to various forms of interference, including intentional and unintentional radio frequency emissions. Radio Frequency Interference (RFI) has the potential to introduce errors during the signal

processing phases of receivers, leading to a degradation in Position, Velocity, and Time (PVT) estimation or even resulting in the complete loss of signal tracking. Hence, interference detection and classification are crucial to uphold reliability and initiate interference mitigation activities.

Traditional methods of managing GNSS interference depend on basic signal processing methods and predefined thresholds for detection. However, the introduction of advanced machine learning (ML) presents a transformative possibility in the field where ML's capacity to determine patterns and correlations within data holds great potential for revolutionizing the detection and even the classification of GNSS interference. Despite its potential benefits, it is important to note that interference classification through machine learning can demand significant time and resources, even when no interference is present. Moreover, the timeliness of each ML model's detection or inference time becomes a critical factor, particularly in real-time applications. Additionally, in our previous work Mehr and Dovis (2022), the GNSS signal without interference was treated as a class in the classification task which required running the entire classification process even when there was no interference, making it time-consuming. To overcome these challenges, our proposed methodology aims to implement a dual-stage approach for detection and classification utilizing Deep Neural Networks (DNN). Instead of uniformly classifying every input signal segment, we prioritize classification only when interference is detected in the first stage. This streamlined approach reduces processing time, saves storage space, and allows for a larger volume of samples to be analyzed in real-time signal streams, enhancing the system's overall efficiency and responsiveness. The novelty of our method lies in its ability to combine the strengths of Long Short-Term Memory (LSTM) networks for detection and Convolutional Neural Networks (CNN) for classification within the context of GNSS interference. This methodology operates at the raw signal level (In-phase/Quadrature (IQ) signal samples), which is crucial, especially if mitigation methods must be applied directly to raw signals. The proposed methodology addresses the time constraints of real-time interference classification by implementing a selective approach, aiming to minimize the inference and processing time of real-time streams.

The remainder of this paper is organized as follows: Section II discusses the background of interference detection and classification. Section III discusses the proposed dual-stage monitoring system methodology. Section IV details the data gathering process and the practical implementation of the proposed architecture. Section V presents the evaluation of the model's performance in both stages, highlighting the accuracy and inference time in real-time monitoring. Finally, Section VI concludes the paper, summarizing our findings and offering insights into future lines of research and improvement.

II. BACKGROUND

In the literature, numerous studies have explored the detection of interferences in GNSS using conventional signal processing methods. These techniques are typically categorized into two groups: pre-correlation and post-correlation. In the pre-correlation stage, the I/Q samples of the received GNSS signal in various domains, or certain features of the front-end such as Automatic Gain Control, are analyzed to detect interferences. On the other hand, post-correlation methods rely on analyzing the output of the tracking loop for interference detection. These methods typically involve analysis of the correlation output or monitoring the carrier-to-noise ratio (C/N_0), where a decrease in C/N_0 can be caused by the presence of interference. For a comprehensive analysis of these techniques, refer to Dovis (2015); Borio et al. (2016); Kaplan and Hegarty (2017); Qiao et al. (2023); Spens et al. (2022). However, such post-correlation techniques may prove inadequate in scenarios with high interference power, as the tracking loop may lose lock on the acquired signal.

These conventional interference detection approaches often rely on predefined rules or adaptive thresholds. However, adapting these thresholds is not always straightforward, as it may require accurate estimation of the noise level, and they may still fail to dynamically adapt to evolving interference scenarios or accurately distinguish between genuine signals and interference. A significant drawback of these techniques is their limited adaptability to specific interference characteristics. As a result, to achieve a realistic interference detection in receivers, multiple techniques must be implemented to create an interference detection alert. Given the expanding potential of ML, many recent studies have focused on leveraging ML for interference detection and further classification. Indeed, considering the challenges posed by conventional methods, leveraging machine learning algorithms can facilitate the development of a unified model capable of learning intricate patterns of interference signals and adapting to dynamic interference scenarios. ML methods can also be categorized based on their input into pre and post-correlation stages and Table 1 provides an overview of these techniques.

Paper	Receiver Stage	Input	ML model
Li et al. (2016)	Pre and Post correlation	Power spectrum, C/N_0, and correlator outputs	SVM and Twin-SVM
Wu et al. (2017)	Pre-correlation	IQ samples	CNN and fully connected networks
Ferre et al. (2019)	Pre-correlation	Spectrogram of IQ samples	SVM and CNN

Continued on next page

Table 1 – continued from previous page

Paper	Receiver Stage	Input	ML model
Xu et al. (2020)	Pre-correlation	Features of IQ samples	SVM and random forest
Shu et al. (2021)	Pre-correlation	Features of IQ samples	SVM optimized by improved genetic algorithm
Chen et al. (2022)	Pre-correlation	Spectrogram of IQ samples	CNN
Swinney and Woods (2021)	Pre-correlation	Concatenation image of the spectrogram, PSD, histogram, and raw constellation of IQ samples	CNN, SVM, random forest, and logistic regression
Yang et al. (2022)	Post correlation	Tracking loop output	BiLSTM
Qin and Dovis (2022)	Pre-correlation	Features of IQ samples	k-nearest
Nicola et al. (2020)	Pre-correlation	Features from the IQ sample spectrogram	SVM
Elango et al. (2022)	Pre-correlation	Scalogram of IQ samples	CNN
Nasser et al. (2022)	Post correlation	Mean C/N_0, number of tracked signals, and satellite elevation	SVM
van der Merwe et al. (2022)	Pre and Post correlation	Features of IQ samples and C/N_0	Autoencoder and random forest
Raichur et al. (2022)	Post correlation	Satellite elevation and azimuth, C/N_0, AGC value, pseudo rate uncertainty	Neural network based on U-Net
Brieger et al. (2022)	Pre and Post correlation	Spectrogram, PSD and kurtosis of IQ samples, number of visible satellites and C/N_0	CNN and TS-Transformer
Fu et al. (2022)	Pre-correlation	Time-series of signal entropy and instantaneous frequency of received signal	LSTM
Jdidi et al. (2022)	Pre-correlation	Features and spectrogram of IQ samples	K-means and random forest
Ding and Pham (2023)	Pre-correlation	Correlation of the received signals, energy distribution, and Spectrogram	SVM
Wu et al. (2023)	Pre-correlation	Spectrogram of IQ samples	CNN
Ivanov et al. (2024)	Post correlation	C/N_0	Recurrent neural network
Ott et al. (2024)	Pre-correlation	Spectrogram of IQ samples	CNN
Zhong et al. (2024)	Pre-correlation	Spectrogram of IQ samples	CNN

Table 1: overview of ML techniques for interference classification

III. METHODOLOGY

The proposed methodology is illustrated in Figure 1, providing an overview of the approach where the idea is to detect possible interference by using a simple data representation processed from the input stream. The more sophisticated signal representation is extracted only in the detected presence of the interference for the classifier. In the first stage of prediction pipeline, an LSTM-based autoencoder is trained with the sequences of spectral presentations determined from the GNSS clean signals (without interference). An autoencoder is an unsupervised neural network architecture which efficiently compresses (encodes) input data to its core features and then reconstructs (decodes) the original input from this compressed representation (Goodfellow et al., 2016). Based on this nature of the autoencoder, our approach performs binary classification to determine either the presence of the interference or clean signal input by evaluating the reconstruction error. In order to set a threshold that allows the model to decide if the reconstruction error is not within the expected range, a decision threshold is implemented based on the probability of false alarm. The detection model architecture will be detailed in Section III.1. If no interference is detected in the first stage, the detector continues for the next sample input taken from the stream. Upon detecting interference, the process advances to a second stage for classification by a CNN architecture classifier. Here, the same input signal segment undergoes classification using the dedicated model. Before classification, relevant information must be extracted through preprocessing which will be explained in Section III.2, involving analysis of the input signal across various domains.

1. Detection Model

Before delving into the proposed architecture of LSTM-based autoencoder, it is pertinent to review the foundational components of LSTMs. Recurrent networks, such as LSTM, are designed to capture temporal dependencies from the input sequence (Goodfellow et al., 2016). An LSTM layer consists of cells and each cell contains components called input i, forget f and output o gates. The input gate controls how much of the input information is stored in the current cell state, the forget gate determines the retention of information from the previous cell state in the current cell state, and the output gate manages the amount of memory content included in the output. The sigmoid (σ) activation function determines the degree to which the

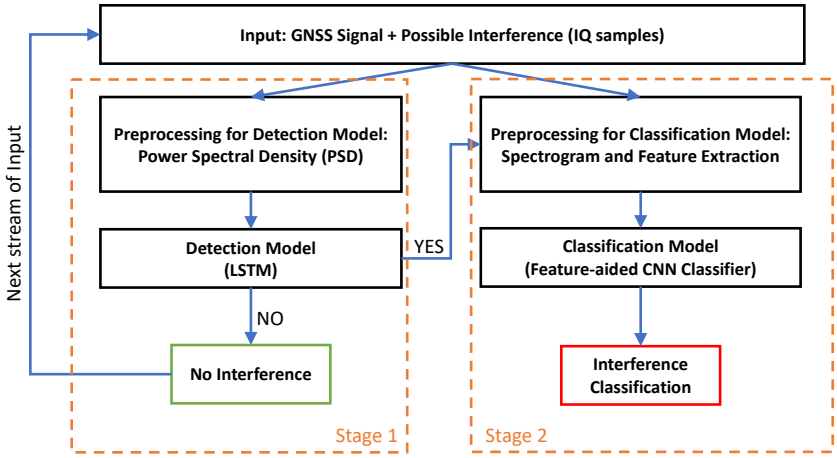


Figure 1: Proposed methodology

input data and the previous hidden state should be accumulated or retained. The computation of cell activation $\hat{c}$ includes the tanh activation function. The cell state c is updated using the output values of the forget and input gates, while the hidden state h is computed using the cell state and output gate values. The corresponding formulas for each time step t are formulated as follows (Hochreiter and Schmidhuber, 1997):

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + b_{xi} + W_{hi}h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{xf}x_t + b_{xf} + W_{hf}h_{t-1} + b_{hf}) \\ \hat{c}_t &= \tanh(W_{xc}x_t + b_{xc} + W_{hc}h_{t-1} + b_{hc}) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \hat{c}_t \\ o_t &= \sigma(W_{xo}x_t + b_{xo} + W_{ho}h_{t-1} + b_{ho}) \\ h_t &= o_t \circ \tanh(c_t) \end{aligned} \tag{1}$$

where b_* represents bias vectors, and W_* denotes weight matrices for input and hidden state across different gates and cell activation, and finally $\circ$ is the hadamard product.

The architecture of the LSTM autoencoder is partially based on Savolainen et al. (2024) and is illustrated in Figure 2a where both encoder and decoder units include two LSTM layers of which each contains two layers of LSTM cells. As shown in Figure 2b, between the cell layers, one dropout layer is included as regularization to reduce overfitting (Goodfellow et al., 2016). The decoder unit has one additional linear layer which is aimed to help with the prediction capability of the autoencoder. The reconstruction error is computed as Mean Squared Error (MSE) loss

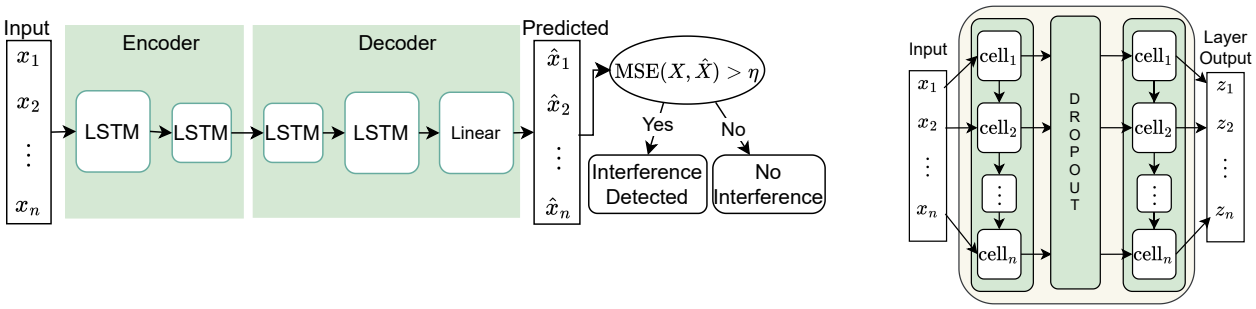
$$\mathbf{MSE}(X, \hat{X}) = \frac{1}{T}\sum_{1}^{T}\frac{|x_t - \hat{x}_t|^2}{NFFT} \tag{2}$$

where x_t is input, $\hat{x}_t$ is the prediction for time step t, $|\cdot|$ determines the vector norm, and T is the length of the sample sequence. MSE is first computed for each t, averaged with the number of FFT (NFFT) bins and finally the sample specific error is obtained by averaging the sum of errors per time step.

As previously explained, a decision threshold is set based on the probability of false alarm to determine whether the MSE falls within the expected probability range. To achieve this, the trained LSTM model is first evaluated on the validation dataset, which consists entirely of samples from GNSS signals without interference. Then, a normal distribution is fitted to these points to extract the mean μ and standard deviation σ of the distribution. Finally, the decision threshold η is derived using the following formula (Parzen, 1960):

$$p\left(\frac{e-\mu}{\sigma} > \eta\right) = p_{fa} \tag{3}$$

where p_{fa} is is the desired probability of false alarm and set to 0.001, and e represents the reconstruction error.



(a) Detection pipeline

(b) LSTM layer architecture

Figure 2: LSTM -based autoencoder architecture.

2. Classification Model

The classification model utilized in this pipeline is a CNN-based approach previously introduced in Mehr and Dovis (2023, 2022). Named the feature-aided CNN classifier, this model is designed to classify different kinds of interferences in the GNSS band. In this methodology, the raw samples of received signals, which may contain various types of interference, are first transformed into a Time-Frequency Representation (TFR) image. Additionally, statistical features extracted from the time and frequency domains of the raw signal samples are incorporated to aid the model in classifying low-power interferences. The image generated by the TFR is processed through the CNN to extract the most significant and informative features from the raw pixel data of the images. These extracted features are subsequently combined with statistical features. This fusion of diverse features creates a combined feature vector, which is then fed into the three-depth fully connected layers for classification. For this work, the CNN architecture employed is ResNet50, known for its effectiveness in deep learning tasks.

IV. DATA GATHERING AND PRACTICAL IMPLEMENTATION

1. Data Flow Architecture

This section outlines the data flow architecture used to evaluate the effectiveness of a proposed methodology for detecting and classifying interference in GNSS signals. The architecture is depicted in Figure 3, illustrating how GNSS signals are managed through distinct processing pathways. This structured approach is pivotal in assessing the methodology's performance in real-world scenarios. It dictates how GNSS signals are processed and analyzed, simulating real scenarios of interfered signals in the laboratory to ensure robust detection and classification of interference. Subsequent sections will provide a detailed overview of each component within this architecture, explaining the systematic handling of GNSS signals and the methodology's application.

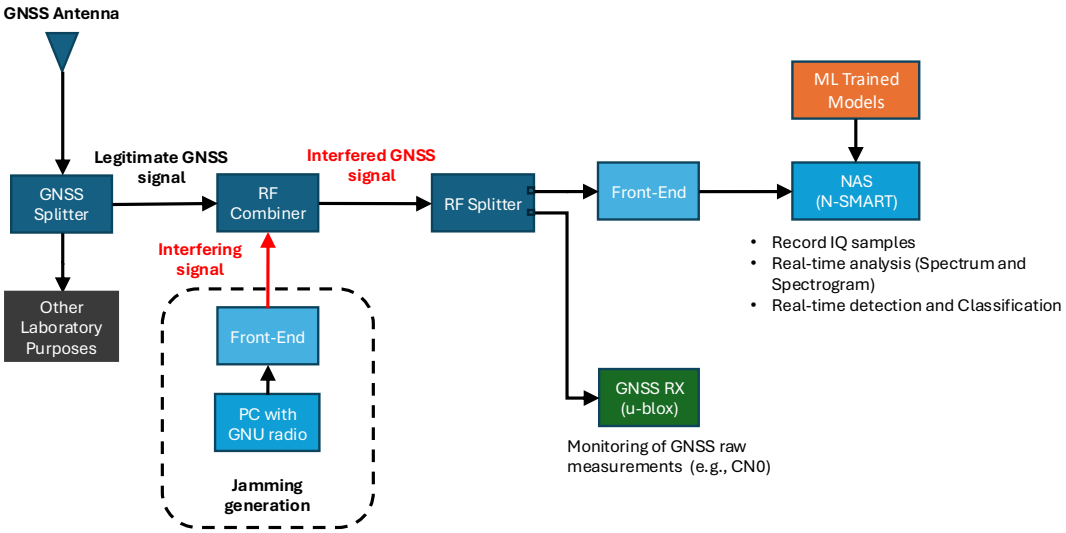


Figure 3: Data flow architecture

a) Legitimate GNSS Signals

The GNSS signal path includes an antenna, a 60m low-attenuation coaxial cable and an active power splitter. The antenna is a reference 3D choke ring ground plane that supports multiple GNSS constellations and bands with remarkable multipath rejection performance. It incorporates a built-in Low-Noise-Amplifier (LNA) with a gain of 29 dB. The active power splitter distributes the received GNSS signals to four RF outputs, with each output additionally amplified by 10 dB. This results in a total amplification of the received signal of 39 dB from the antenna to the testbed. The received signal power at the spectrum analyzer, measured over a bandwidth of 25 MHz is -116 dBW, including RF connectors and excess cable path loss.

b) Interfering signal

The interfering signal branch includes equipment for generating and injecting interfering signals, such as jamming signals. The SDR (Software-Defined Radio) paradigm is utilized to generate various types of interference.

Firstly, different types of interference, as explained and mathematically modeled in Mehr and Dovis (2023), are simulated using a MATLAB script. The simulated interference signals are then stored separately as binary files with a sampling frequency of 25 MHz and 16-bit quantization for both real and imaginary components.

Secondly, the RF interference signals are transmitted using a front-end controlled by GNU Radio, with the generated binary files serving as the input. The front-end used in this experiment is a Universal Software Radio Peripheral (USRP) N210, a flexible and versatile hardware device for SDR. The bandwidth and bit quantization of this front-end match the parameters of the binary files. It is worth noting that the output power of the USRP is very high, around -75 dBW. To mitigate this, a series of attenuators totaling 60 dB is used to reduce the output power to zero. The output power of the USRP is then controlled through the SDR, adjustable from -130 to -105 dBW over a bandwidth of 25 MHz in steps of 5 dB. Figure 4 shows the laboratory testbed for generating interference signals.

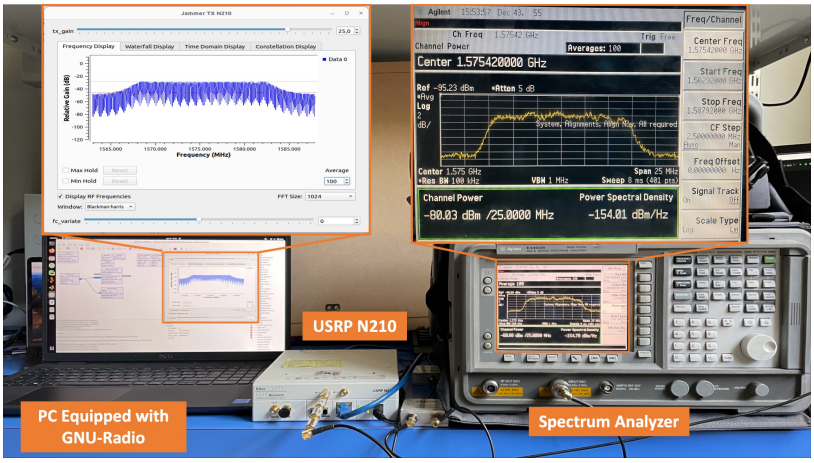


Figure 4: Interference generation testbed

2. Practical Implementation and Deployment of ML Models

This section details the practical implementation and deployment of the machine learning (ML) models used for detecting and classifying interference in GNSS signals. The deployment process involves several stages, including the integration of the trained models into the data processing pipeline, the real-time analysis of incoming signals, and the mechanisms for monitoring the model's performance based on ongoing data collection.

As illustrated in Figure 3, the main component for data processing is the Network Attached Storage (NAS). A software platform called N-SMART, designed by the NavSAS group (Mehr et al., 2023), is installed on top of the NAS. The primary objectives of N-SMART are to record IQ signal samples, run the ML pipeline, and provide real-time alerts in case of interference detection. This platform receives IQ signal samples with a sampling frequency of 25 MHz and 16-bit quantization from the USRP front-end via a LAN connection as the input. These IQ data undergo a preprocessing stage to provide the appropriate input for each ML model. Another input to this platform is the trained models for detection and classification, which take the preprocessed data to make decisions. It is worth noting that the detection model is developed using PyTorch (Ansel et al., 2024), while the classification model is developed using TensorFlow Keras library (Abadi et al., 2015).

3. Dataset for Model Training

This section presents the dataset used for training the machine learning models. The dataset includes GNSS signals recorded both in the presence of interference and without. Each element of this IQ dataset lasts for one minute.
A a first step, GNSS signals are recorded in the presence of different interference events at various power levels. In total, there are 16 different types of interference with 6 different power levels, as explained in Section IV.1 b). This power range includes low power level (-130 dBW), where the performance of the GNSS receiver is degraded, to high power levels (-105 dBW), where the receiver is completely blinded. The GNSS signals in this experiment belong to the L1 band with a center frequency of 1.57542 GHz.
Additionally, another class, representing GNSS signals without interference, is recorded throughout the day at hourly intervals. This ensures the dataset includes samples with different satellite geometries, providing a comprehensive set of conditions for training.

a) Data Preprocessing and Training

This section explains the preprocessing procedure, which is essential for preparing the appropriate input for the ML models.

Detection model: The input sequence for the detection model consists of eight consecutive PSD vectors which computed using the Welch method with a Hann windowing function applied to 100 μs long snapshots. Therefore, the total data duration for one sample is 800 μs. The number of FFT points (NFFT) is set to 256 for each PSD estimation. Each PSD is scaled by the value of NFFT. Using the same constant for scaling each PSD value vector enhances the visibility of changes in input values for the trained network. This approach promotes faster model convergence and improves the detection capability of interfering signals, even those with lower transmission power.
The detection model is trained exclusively on recorded GNSS signals without any interference. During the testing phase, the performance of the model is evaluated using both interfered and legitimate GNSS signals. For the interfered GNSS signals, only those with the lowest interference power from the dataset, specifically -130 dBW, are selected. This choice is made because if the model can detect interference at this lower power level, which is less prominent in the PSD, it is expected to also detect interference at higher power levels effectively. The training and validation datasets consist of 3112 and 778 samples, respectively, comprising only GNSS signals (Clean). The test set includes 1795 samples of legitimate clean GNSS signals and 2250 samples of interfered GNSS signals.
For the loss optimization, Adam optimizer Kingma and Ba (2017) with a learning rate of 0.001 is used. To obtain a faster convergence and therefore reduction in the computational time during the training, a batch size of 16 is chosen. The number of epochs was set to 20 of which only eleven were executed due to the early stop implementation, where the validation loss failed to improve four epochs. The hidden size of each layer is set as 400, 200, 400, 256, 256, where the sequence is starting from the first layer.

Classification model: As explained in Section III.2, the inputs to the classification model are the TFR and statistical features. Initially, the input signal (a window of 100 μs) is examined within the time-frequency domain using the Short-Time Fourier Transform (STFT), which is chosen for its computational efficiency and accuracy in representing signal characteristics. Concurrently, additional relevant features are extracted from both the time and frequency domains of the same time series of input signals (a window of 200 μs) as numerical features.
Finally, following the explained procedure of preprocessing, the image dataset is created from the IQ dataset. This dataset includes 98 classes, one of which refers to GNSS signals with no interference and a total of 96 classes that correspond to specific types of interference at a certain power level. During the recording of IQ signals with the USRP, when there is high-power interference, the USRP experiences brief malfunctions lasting a few milliseconds. The dataset also includes a class for these occurrences, labeled as "USRP Malfunction". The number of samples in each dataset class is uniformly distributed around 5800, except for the "USRP Malfunction" class, which has 7879 samples.

V. EXPERIMENTAL FINDINGS AND ANALYSIS

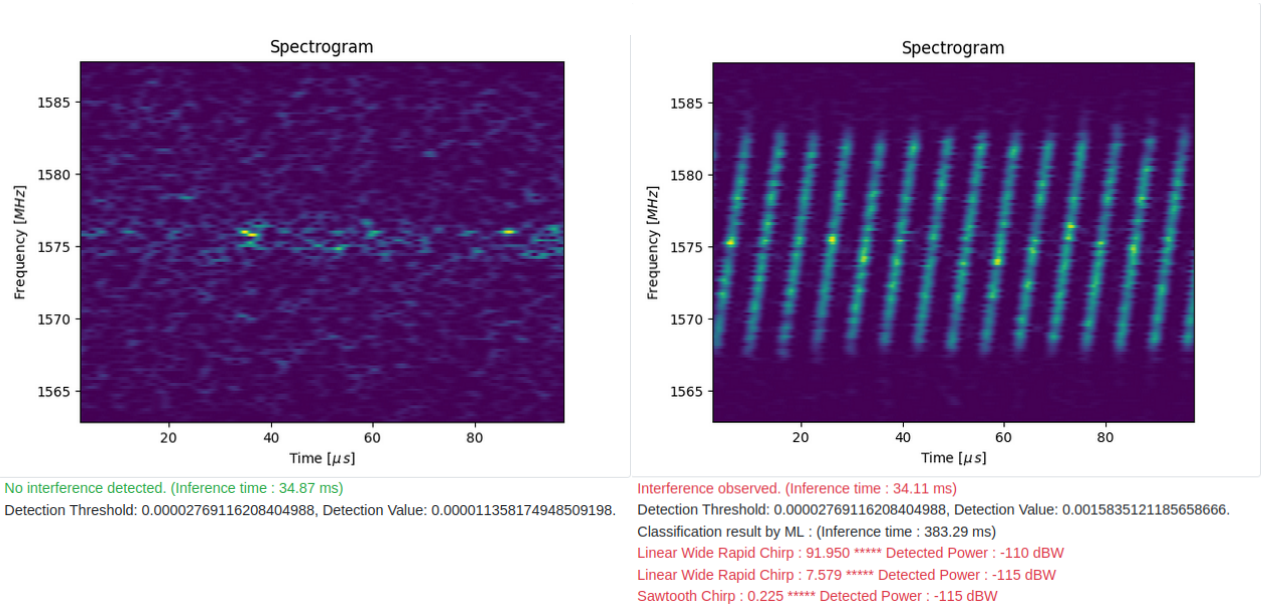
This section presents the results of the detection and classification performance. The evaluation metrics used include accuracy, precision, recall, and F1-score, which provide a comprehensive understanding of each model's effectiveness in detecting and further classifying various types and power of interference in GNSS signals. Furthermore, the task completion time is measured for each model, defined as the total time taken to make a decision, including the preprocessing steps. These metrics are first examined for the overall performance of the detection and classification models and are summarized in Table 2. Then the following sections detail performance of each model.

The proposed methodology prioritizes classification only when interference is detected, where this approach has led to a substantial decrease in detection time, allowing for a larger volume of samples to be analyzed in real-time signal streams. This streamlined approach not only optimizes efficiency without compromising accuracy but also enhances system responsiveness in processing signal streams, making it highly suitable for applications requiring rapid decision-making based on real-time data

Table 2: Overall performance of proposed methodology

Model	Accuracy [%]	Precision [%]	Recall [%]	F1-Score [%]	Task Completion Time [ms]
Detection Model	100	100	100	100	34
Classification Model	98.26	98.63	98.40	98.46	383

analysis. Figure 5 illustrates the deployment of the detection and classification models within the User Interface Monitoring system in real time.



(a) Detection model output in case of No Interference

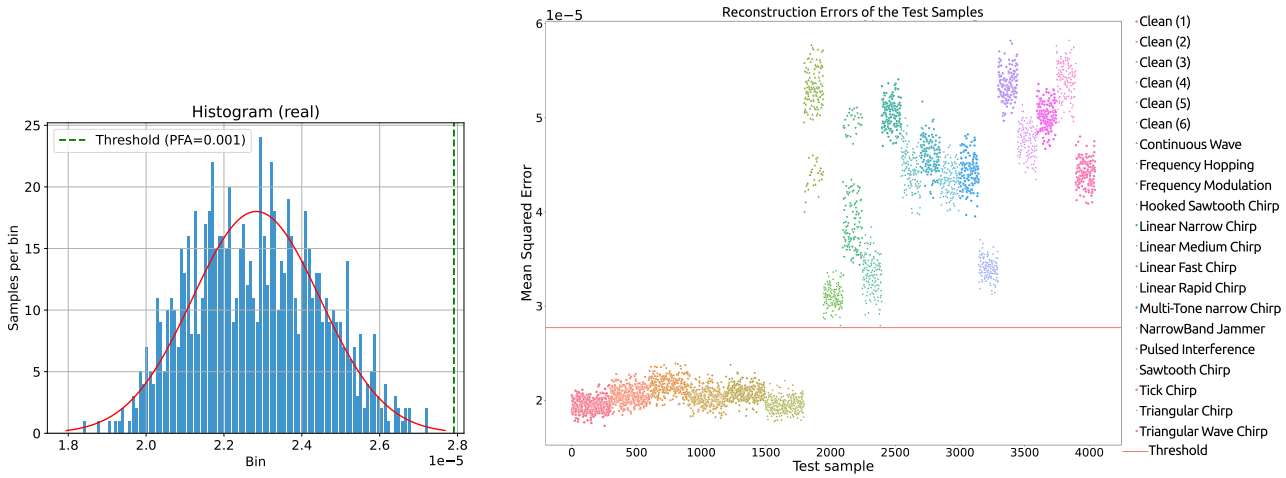
(b) ML model output accurately classifying the shape and power of the interference during the transmission of the jammer. This output was captured at the same moment as Figure 4, showing a Linear Wide Rapid chirp signal with a power of -80 dBm (-110 dBW).

Figure 5: Real-time interference alerting user interface with integrated ML models

1. Detection Performance

In Figure 6a, we can observe that the distribution of reconstruction error values (blue bins) resembles the normal distribution (red curve). The used threshold determination method ensures that the threshold can adapt to the natural fluctuation range of the received legitimate signal power. This yields a reduction of the number of false alarms invoking the classification model.

It was observed that increasing the interference power results in higher loss values compared to lower power levels, as stronger interference more clearly deviates from the normal condition of GNSS signals without interference. Therefore, for the evaluation of the detection model, we have focused on the scenario where the interference has low power (-130 dBW), making it more challenging for the model to detect. The detection model achieved an accuracy of 100% (Table 2), which indicates that the new scaling approach and the threshold computation increased the model's ability to distinguish the legitimate GNSS signal sample from the interfered one. The deviation of the loss values of the test samples are seen in Figure 6b. Interferences with small bandwidth, such as frequency hopping (FH), linear narrow (LN) chirp, and narrowband (NB) jammers, were more challenging to detect than the other tested interference types by having the loss value close to the determined threshold. In general, increasing the threshold value to reduce the number of false alarms may result in the misclassification of low-power interference samples, particularly for these three types of interference. This occurs because the resulting low-power PSD does not significantly differ from the PSD of the legitimate signal.



(a) Threshold determination. The rounded reconstruction error of the samples are presented on x-axis (Bin).

(b) Reconstruction errors computed for each test sample.

Figure 6: Detection model's performance

2. Classification Performance

According to Table 2, the classification model demonstrates high reliability and overall performance with an accuracy of 98.26%, correctly classifying the vast majority of input signals. Its precision score signifies that when the model predicts an interference type and power level, it is correct 98.63% of the time, indicating very few false positives. The recall shows the model's effectiveness in identifying true positive cases by detecting 98.40% of all actual instances of interference. The F1-score of 98.46% confirms a strong balance between precision and recall, underscoring the model's capability to make accurate predictions while successfully identifying almost all true instances of interference. Furthermore, the classification model's performance is evaluated across various interference types and power levels, as shown in Table 3. The overall accuracy for each power level and class-specific accuracy results provide insights into the model's effectiveness.

As shown in Table 3, as the jamming power increases, the models' accuracy improves because stronger interference becomes more distinguishable in the time-frequency representation of the received signals, which we used as input images for our model. As a specific example, in the worst case with respect to the other type of interferences in this article, pulsed interference with a power of -130 dBW achieved an accuracy of 47%. This is because each pulse lasts only a few microseconds, making it very difficult to detect within a 100-microsecond window. Furthermore, when the power is extremely low and nearly buried under the noise floor, it becomes even more challenging to recognize its presence.

VI. CONCLUSION

In this work, we introduced a dual-stage monitoring system that leveraged the strength of both LSTM and CNN networks for detecting and classifying GNSS interference in real-time. A detection model placed before a classification unit can reduce computational resources as the time consuming data pre-processing methods for classification are only executed in presence of detected inference. We show that the proposed architecture is robust against different types of RFI with varying transmission power levels by achieving a detection accuracy of 100% followed by the overall classification accuracy of 98.26%. Future work could focus on optimizing the model and exploring new architectures for LSTM and CNN networks to improve accuracy and reduce processing time. Additionally, expanding the model's application to handle a broader range of interference types and scenarios will be pursued.

ACKNOWLEDGEMENTS

The Ph.D. work of I. Ebrahimi Mehr is 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. The Ph.D. work of O. Savolainen is supported by the Academy of Finland Flagship program: Finnish Center for Artificial Intelligence (FCAI) and the Academy of Finland project 338043 Resilience and security of geospatial data for critical infrastructures (REASON), and the Department of Computer Science, University of Helsinki. This

Table 3: Class-wise accuracy results of classification model

Interference Class	Accuracy [%]					
Clean GNSS (No Interfernece)	92.72					
USRP Malfunction	96.01					
Interference Power	-130 dBW	-125 dBW	-120 dBW	-115 dBW	-110 dBW	-105 dBW
Continuous Wave Interference	100	100	100	99.95	99.95	96.27
Frequency Hopping (FH) Jammers	87.26	95.98	98.32	99.65	100	100
Frequency Modulation (FM) Interference	100	100	97.15	100	99.95	99.81
Hooked Sawtooth Chirp	100	100	100	100	100	100
Linear Narrow Chirp	100	100	100	100	100	100
Linear Wide Slow Chirp	100	100	100	100	100	100
Linear Wide Medium Chirp	100	99.81	96.82	100	100	100
Linear Wide Fast Chirp	99.95	100	100	100	100	100
Linear Wide Rapid Chirp	100	100	100	100	99.11	100
Multi-Tone Narrow Chirp	99.56	100	100	100	99.76	100
Narrowband Jammer	89.15	97.13	95.06	97.43	100	100
Pulsed Interference	47.63	75.7	89.86	99.17	99.02	99.56
Sawtooth Chirp	100	100	100	100	100	100
Tick Chirp	100	100	100	100	99.61	100
Triangular Chirp	99.59	100	100	100	100	100
Triangular Wave Chirp	100	100	100	100	96.46	100
Overall Accuracy of Each Power Level	95.19	98.03	98.57	99.76	99.61	99.72

paper is part of the project NODES which has received funding from the MUR – M4C2 1.5 of PNRR funded by the European Union - NextGenerationEU (Grant agreement no.ECS00000036)

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., and Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Borio, D., Dovis, F., Kuusniemi, H., and Lo Presti, L. (2016). Impact and detection of GNSS jammers on consumer grade satellite navigation receivers. *Proceedings of the IEEE*, 104(6):1233–1245.

Brieger, T., Raichur, N., Jdidi, D., Ott, F., Feigl, T., van der Merwe, J., Rügamer, A., and Felber, W. (2022). Multimodal learning for reliable interference classification in GNSS signals. *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, Colorado,*, pages 3210–3234.

Chen, X., He, D., Yan, X., Yu, W., and Truong, T.-K. (2022). GNSS interference type recognition with fingerprint spectrum DNN method. *IEEE Transactions on Aerospace and Electronic Systems*, 58(5):4745–4760.

Ding, Y. and Pham, K. (2023). 1 GNSS interference identification beyond jammer classification. In *2023 IEEE Aerospace Conference*, pages 1–8.

Dovis, F. (2015). *GNSS Interference Threats and Countermeasures*. The GNSS technology and application series. Artech House.

Elango, A., Ujan, S., and Ruotsalainen, L. (2022). Disruptive GNSS signal detection and classification at different power levels

using advanced deep-learning approach. In *2022 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–7.

Ferre, R., Fuente, A., and Lohan, E. S. (2019). Jammer classification in GNSS bands via machine learning algorithms. *Sensors*, 19:4841.

Fu, D., Li, X., Mou, W., Ma, M., and Ou, G. (2022). Navigation jamming signal recognition based on long short-term memory neural networks. *Journal of Systems Engineering and Electronics*, 33(4):835–844.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Ivanov, V., Scaramuzza, M., and Wilson, R. C. (2024). Deep temporal semi-supervised one-class classification for GNSS radio frequency interference detection. *Journal of Navigation*, page 1–23.

Jdidi, D., Brieger, T., Feigl, T., Contreras Franco, D., van der Merwe, J., Rügamer, A., Seitz, J., and Felber, W. (2022). Unsupervised disentanglement for post- identification of GNSS interference in the wild. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, Colorado,*.

Kaplan, E. and Hegarty, C. (2017). *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. *ArXiv*, abs/1412.6980.

Li, W., Huang, Z., Lang, R., Qin, H., Zhou, K., and Cao, Y. (2016). A real-time interference monitoring technique for GNSS based on a twin support vector machine method. *Sensors*, 16(3).

Mehr, I. E. and Dovis, F. (2022). Detection and classification of GNSS jammers using convolutional neural networks. In *2022 International Conference on Localization and GNSS (ICL-GNSS)*, pages 01–06.

Mehr, I. E. and Dovis, F. (2023). A deep neural network approach for detection and classification of GNSS interference and jammer. doi: `10.36227/techrxiv.22212121`.

Mehr, I. E., Minetto, A., Dovis, F., Pica, E., Cesaroni, C., and Romano, V. (2023). An open architecture for signal monitoring and recording based on SDR and docker containers: A GNSS use case. In *IEEE EUROCON 2023 - 20th International Conference on Smart Technologies*, pages 66–71.

Nasser, H., Berz, G., Gómez, M., Fuente, A., Fidalgo, J., Li, W., Pattinson, M., Truffer, P., and Troller, M. (2022). GNSS interference detection and geolocalization for aviation applications. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, pages 192–216.

Nicola, M., Falco, G., Morales Ferre, R., Lohan, E.-S., de la Fuente, A., and Falletti, E. (2020). Collaborative solutions for interference management in GNSS-based aircraft navigation. *Sensors*, 20(15).

Ott, F., Heublein, L., Raichur, N. L., Feigl, T., Hansen, J., Rügamer, A., and Mutschler, C. (2024). Few-shot learning with uncertainty-based quadruplet selection for interference classification in GNSS data. *ArXiv*, abs/2402.09466.

Parzen, E. (1960). *Modern Probability Theory and Its Applications*. Wiley.

Qiao, J., Lu, Z., Lin, B., Song, J., Xiao, Z., Wang, Z., and Li, B. (2023). A survey of GNSS interference monitoring technologies. *Frontiers in Physics*, 11.

Qin, W. and Dovis, F. (2022). Situational awareness of chirp jamming threats to GNSS based on supervised machine learning. *IEEE Transactions on Aerospace and Electronic Systems*, 58(3):1707–1720.

Raichur, N., Brieger, T., Jdidi, D., Feigl, T., van der Merwe, J., Ghimire, B., Ott, F., Rügamer, A., and Felber, W. (2022). Machine learning-assisted GNSS interference monitoring through crowdsourcing. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*.

Savolainen, O., Elango, A., Morrison, A., Sokolova, N., and Ruotsalainen, L. (2024). Gnss anomaly detection with complex-valued lstm networks. In *2024 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–7.

Shu, J., Liao, Y., and Luan, X. (2021). An interference recognition method based on improved genetic algorithm. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 496–500.

Spens, N., Lee, D.-K., Nedelkov, F., and Akos, D. (2022). Detecting GNSS jamming and spoofing on android devices. *NAVIGATION: Journal of the Institute of Navigation*, 69(3).

Swinney, C. J. and Woods, J. C. (2021). GNSS jamming classification via CNN, transfer learning & the novel concatenation of signal representations. In *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–9.

van der Merwe, J. R., Franco, D. C., Jdidi, D., Feigl, T., Rügamer, A., and Felber, W. (2022). Low-cost COTS GNSS interference detection and classification platform: Initial results. In *2022 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–8.

Wu, P., Calatrava, H., Imbiriba, T., and Closas, P. (2023). Jammer classification with federated learning. In *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 228–234.

Wu, Z., Zhao, Y., Yin, Z., and Luo, H. (2017). Jamming signals classification using convolutional neural network. In *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 062–067.

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

Yang, B., Dong, C., Gao, B., Liu, Y., Cui, W., and Gao, F. (2022). Research on GNSS interference recognition based on ROI of correlation peaks. *International Journal of Satellite Communications and Networking*, 40(5):330–342.

Zhong, W., Xiong, H., Hua, Y., Shah, D. H., Liao, Z., and Xu, Y. (2024). TSFANet: Temporal-spatial feature aggregation network for GNSS jamming recognition. *IEEE Transactions on Instrumentation and Measurement*, 73:1–13.