scientific reports

Check for updates

OPEN

Real-Time jamming detection using windowing and hybrid machine learning models for pre-saturation alerts

J. Sormayli, M. Darvishi, K. Zarrinnegar & M. R. Mosavi✉

This paper proposes a new deep learning and machine learning model for detecting deception and suppression jamming in Ublox-M8T receivers operating under GNSS interference. This solution employs XGBoost for real-time classification of jamming signals, implemented on an STM32H743 microcontroller to ensure ultra-low latency, making it suitable for navigation in various environments. This work's key contribution is integrating a windowing mechanism for pre-saturation alerts and early activation of jamming detection which enhances system reliability by distinguishing between high-credibility and low-credibility GNSS data under static and dynamic jamming conditions. To validate the model, a series of experiments were conducted using a software-defined radio transmitter to simulate jamming scenarios. Genuine GNSS and jamming signals were collected under controlled conditions, and the data were pre-processed through feature normalization, correlation analysis, and feature selection based on importance in the mentioned systems. The XGBoost classifier, trained and tested on this processed dataset, achieved a detection rate of 99.97%, a precision of 99.94%, and a Matthews correlation coefficient of 0.9992, with an average prediction time of only 20 microseconds per sample in the implemented mode, making it an excellent choice for real-time systems. Additionally, the windowing mechanism enhances system performance by proactively initiating countermeasures before reaching saturation, ensuring continuous operation during high-intensity jamming attacks.

Keywords Jamming interference, Machine learning, Deep learning, Windowing, Hybrid models, Real-Time detection

Global Navigation Satellite System (GNSS) particularly GPS (Global Positioning System) satellites orbit the Earth at an altitude of approximately 20,000 km, serving as a crucial component of modern technology by delivering precise positioning and navigation services[1]. As a result, the received power of these signals is minimal, often comparable to the noise power level. This characteristic makes the system's signals highly susceptible to various forms of intentional and unintentional attacks and interference[2]. Intentional interference can be further categorized into two types: (1) jamming, which disrupts and distorts the system's positioning capability[3–5], and (2) spoofing, which replaces the genuine signal with a counterfeit one[6–8]. Given the growing reliance on this system across various aspects of daily life, such interferences can result in significant costs and consequences. Unintentional interferences, such as multi-path effects and radio frequency disturbances, can degrade the accuracy of navigation solutions. On the other hand, intentional interferences are deliberately designed to disrupt and impair navigation performance. Jamming is one of the most prevalent forms of interference, widely accessible to the public and available in various types. In many instances, the jamming signal's power surpasses that of the navigation signal, obstructing the original signal from reaching the receiver[3].

Traditional techniques for detecting radio frequency interference in GNSS receivers primarily rely on signal processing methods. One notable approach for jamming detection is the method applied during the pre-correlation stage. The key techniques used in this method include monitoring the Automatic Gain Control (AGC), analyzing the digital samples of the received signal, and examining the time and frequency spectrum of the signal captured by the receiving antenna. In the first method, for instance, a sudden variation in the AGC can signal the presence of strong interference. Such interference could potentially lead to a loss of quantization during the analog-to-digital conversion process[9–11]. The second method involves statistical analysis of digital samples from the received signal. Techniques such as analysis of variance can be employed to identify deviations

Department of Electrical Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran. ✉email: M_Mosavi@iust.ac.ir

from the expected behavior of Global Navigation Satellite System (GNSS) signals[12,13]. The third method focuses on analyzing the time and frequency spectrum of the received signal to identify unexpected or strong signals that may interfere with GNSS operations. Tools like the Fast Fourier Transform (FFT) are commonly employed to examine the spectral content of the signal, enabling the detection of various types of interference[14]. Another approach to detect jamming interference is the post-correlation detection method. In this technique, after GNSS signals are correlated with locally generated replicas, the analysis is concentrated on the outputs of the correlation process[15]. This method employs techniques such as statistical analysis of the correlation output, which encapsulates the combined information of the received signals. Key parameters for this analysis include the Carrier-to-Noise power ratio (C/N_0) and pseudo-range measurements[16]. Measurements like pseudo-range, which represent the distance between the satellite and the receiver, are susceptible to interference[17]. Time-domain signal analysis facilitates the detection of transient interference events, such as pulses or noise bursts, which might go unnoticed in frequency-domain analysis[18]. Advanced signal processing techniques, such as the Short-Time Fourier Transform (STFT)[19] and the wavelet transform[20], enable the effective analysis of signals that exhibit variations across both time and frequency domains. This approach is particularly effective in detecting and classifying complex jamming, such as chirp signals that sweep across a broad frequency range. Additionally, the integration of signal processing techniques with antenna arrays[21] and supplementary hardware has been widely explored for mitigating, detecting, and countering jamming and spoofing attack scenarios.

Recent studies indicate that, in recent years, Machine Learning (ML) based approaches have emerged as the most commonly used methods for detecting jamming and spoofing attacks. In[22], researchers explored the classification of chirp jamming signals using K-Nearest Neighbor (KNN) techniques applied during the pre-correlation stage. Meanwhile[23], examined three types of jamming signals: continuous wave, chirp, and pulse utilizing Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks to predict and extract weak GPS signals, even under severe jamming conditions. Additionally[24], employed algorithms based on Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs) to classify receiver signals into two categories: interference present or interference-free. In[25], the authors utilized bidirectional LSTM-NN, achieving an accuracy of over 98% in detecting interference attacks. In[26], a comprehensive analysis was conducted to classify jamming and spoofing attacks using a variety of ML algorithms, including Neural Networks (NNs), SVMs, nearest neighbors, kernel approximation, decision trees, Bayesian methods, and ensemble classifiers. In[27], researchers explored interference classification using transfer learning and CNN by converting jamming signals into various image-based representations, such as power spectral density, spectrograms, and histograms. In[28], researchers utilized a CNN model to pinpoint the most likely location of an interference source, relying solely on ADS-B data gathered within a few hours from the target airspace. In[29], the authors analyzed three types of GPS interference signals by extracting four entropy features, including power spectral entropy, to create a combined entropy dataset. They then employed SVM and Random Forest (RF) algorithms to classify and identify the interference signals. The results demonstrate that the RF algorithm achieves a high-Detection Rate (DR) for interference signals, with an average accuracy exceeding 90%, significantly outperforming SVM. In[30], a LSTM NN is introduced as a detection algorithm, combined with time-frequency analysis for signal pre-processing. In[31], the authors introduced an approach called FPSDNN, which employs a Deep Neural Network (DNN) built upon CNN for fingerprint recognition. This method demonstrates significant effectiveness in accurately classifying various types of GNSS interference. In[32], the study presents a Deep Learning (DL) model that integrates Principal Component Analysis (PCA) and Bayesian Optimization (BO) for feature selection, followed by bidirectional LSTM with an Attention Mechanism (BiLSTM-A) for accurate jamming detection. The use of ML and DL algorithms to detect spoofing attacks has long been a focus of researchers, driven by the complexity of the attacks and the need for more advanced detection models[33].

Given the advancements in jamming and spoofing techniques in GNSS, there is a growing need to develop advanced ML models. However, focusing solely on model design without considering practical implementation can diminish the overall effectiveness of the system. GNSS devices typically operate under strict resource constraints, including limited processing power, memory, and power supply, especially in mobile or remote applications. While a highly complex ML model may offer excellent diagnostic capabilities in theory, it could fail in practice if it consumes excessive device resources. Detecting jamming and spoofing is a time-critical task that demands an immediate response to prevent disruptions or security breaches. If a poorly implemented ML model introduces significant processing delays, it can render the system ineffective or unusable in high-stakes scenarios. To maintain the integrity of GNSS systems under attack, it is crucial to implement models that enable real-time analysis and rapid decision-making. Adopting safe and robust implementation practices, such as on-device processing and minimizing data transmission, helps mitigate additional risks. These practices also ensure that the model's detection capabilities remain accurate, responsive, and adaptable to evolving threats.

This paper proposes a hybrid jamming detection technique designed to identify deception and suppression signals. A hybrid model leveraging ML detects GNSS jamming and is complemented by a windowing mechanism to facilitate the detection of transient modes. The primary input to the model is a time-series dataset comprising multiple features of the navigation signal captured over a single-time period using an Ublox-M8T receiver. This approach combines the established reliability of traditional techniques with the advanced, adaptive capabilities of ML, creating a more accurate, reliable, and scalable jamming detection system. By combining the output of the ML model with a windowing technique, early detection of the onset and cessation of GNSS signal disturbances becomes feasible. Moreover, accuracy, network size, and response time are carefully considered to ensure the model's suitability for real-time applications on embedded hardware.

The remaining of this paper is organized as follows. Section 2 depicts jamming pre-alert conceptual. Section 3 provides an overview of the types of jamming in GNSS systems and highlights the unique characteristics of each jamming type. Section 4 details the experimental setup employed for testing and signal collection. Section 5 focuses on developing classifiers, covering the pre-processing of collected datasets, the training process, and the

performance evaluation of ML and DL models. Section 6 introduces the concept of early warning for jamming onset and cessation, explaining a combined detection approach using a ML model integrated with a windowing mechanism to enhance detection accuracy with additional indicators. Section 7 outlines the implementation of the hybrid model on embedded hardware, including its online testing and a comparison with other methods. Finally, Sect. 8 concludes the study and discusses potential directions for future research.

Jamming pre-alert framework

This work aims to develop a hybrid detection framework combining modern ML/DL techniques and traditional signal processing methods to enable real-time detection and early warning of GNSS jamming. The study's scope includes extracting effective features from raw GNSS data provided by the ublox-M8T receiver under three conditions: two valid operational scenarios and one jamming attack scenario. Figure 1 illustrates the conceptual structure of the proposed framework, including data processing stages.

According to Fig. 1 the specific objectives of this work are as follows:

- To extract and analyze distinguishing features from raw GNSS data in the presence and absence of jamming attacks.
- To design and train an intelligent ML/DL-based model capable of detecting jamming interference.
- To implement an early warning mechanism based on a windowing approach for timely detection of jamming events.
- To evaluate and compare the performance of traditional and intelligent methods within the proposed hybrid framework.

Types of jamming in GNSS

A critical aspect of any study is providing a comprehensive classification of the phenomenon under investigation. Section 3.1 focuses on the GNSS signal. Section 3.2 provides a detailed introduction to suppression jamming and its various types, while Sect. 3.3 discusses deception jamming, including an analysis of the mechanisms and relationships associated with repeater jamming.

GNSS signal display

Satellite navigation receiver signal processing typically includes three key steps: (1) pre-processing, (2) baseband Intermediate Frequency (IF) signal processing, and (3) outputting navigation information[34]. The main components of a GNSS receiver are shown in Fig. 2.

A typical GNSS signal comprises a carrier wave modulated with a spreading code (e.g., the C/A code for GPS) and navigation data. The general form of the GNSS signal transmitted by a satellite, s(t), is expressed in Eq. (1)[35]:

$$s\left(t\right) = A_c.C\left(t - \tau\right).e^{j(2\pi f_c t + \phi)} \tag{1}$$

where A_c is the amplitude of the carrier signal, $C\left(t\right)$ is the broadcast code sequence (for example, PRN code) with chip rate R_c, τ is the time delay related to the satellite range, f_c is the carrier frequency, ϕ is the initial phase of the carrier.

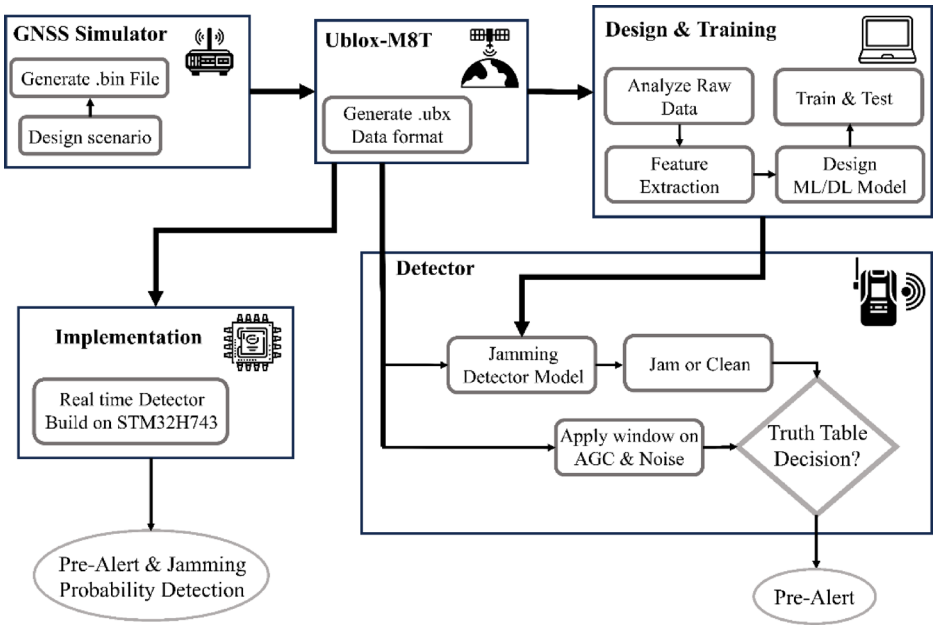

Fig. 1. Jamming pre-alert Conceptual framework.

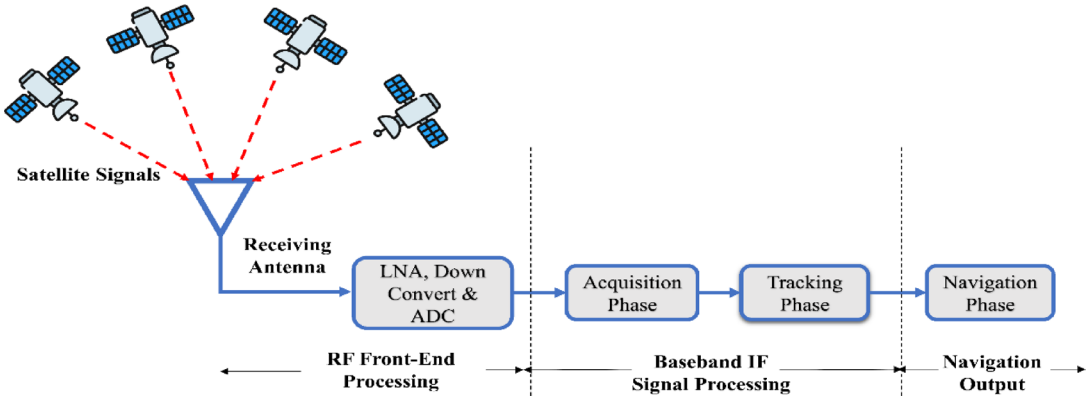

Fig. 2. The building blocks of a GNSS receiver.

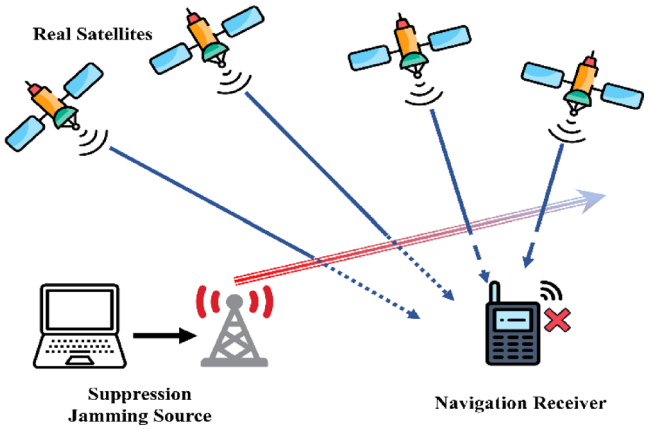

Fig. 3. Schematic diagram of suppression jamming.

The received signal r(t) from a satellite, considering multi-path and Doppler effects, can be modeled as Eq. (2)[35,36]:

$$r(t) = \sum_{i=1}^{M} \alpha_i . C(t - \tau_i) . e^{j(2\pi (f_c + f_{Di})t + \phi_i)} + n(t) \tag{2}$$

where M is the number of multi-path components, α_i is the amplitude of the i-th multi-path component, τ_i represents the time delay of the i-th multi-path component, f_{Di} denotes the Doppler shift of the i-th multi-path component, and ϕ_i indicates the phase offset of the i-th multi-path component. Additionally, $n(t)$ represents Gaussian Additive White Noise (GAWN) with a power spectral density of N_o.

Suppression jamming

Suppression jamming in GNSS involves transmitting interfering signals that increase the noise level or obscure legitimate GNSS signals, thereby hindering the receiver's ability to detect and process these signals accurately[37]. Figure 3 illustrates a schematic diagram of suppression jamming. Table 1 outlines the types of suppression jamming, but for this discussion, we will primarily focus on single-frequency and broadband jamming, as these are the most prevalent forms of suppression jamming encountered in GNSS systems.

Single-frequency suppression jamming introduces a noise signal, denoted as j(t), to the input of the GNSS receiver, thereby increasing the noise level. This jamming signal can be modeled as follows in Eq. (3)[38]:

$$j(t) = A_j . \omega(t) \tag{3}$$

where A_j is the amplitude of the jamming signal, $\omega(t)$ is the Gaussian white noise process with unit power spectral density. Broadband jamming transmits the signal at several GNSS frequencies simultaneously, which can be modeled as Eq. (4)[39].

	Type	Advantage	Disadvantage
	Single-frequency[37,38,40]	Simplicity and efficiency, energy concentration, good jamming effect, possibility of research and optimization	Narrow bandwidth, sensitivity to signal type, limited applicability, predictable
	Pulse[37,41]	High-performance, small bandwidth, adjustable frequency	Complexity of settings, different effect on different signals, limited frequency coverage, complexity of implementation
	Sweep[22,37,42,43]	Wide frequency coverage, the ability to adjust the frequency sweep range, reducing the efficiency of anti-jamming algorithms, wide influence on different signals	Complexity of implementation, high-energy consumption, high-detection capability, need for advanced equipment for production
	Matched spectrum[37,44]	Excellent jamming effect, accurate targeting, reducing receiver error rate, use for civilian code jamming	Need for detailed information, complexity of implementation, limited application, high-cost
Suppression jamming[41]	Continuous and stable[45]	Stability in jamming, predictable effect, high- controllability, wide frequency coverage	Ease of diagnosis, high-energy consumption, reduced effectiveness against some algorithms, need for advanced equipment
	Discontinuous and unstable[45]	Reducing the effectiveness of anti-jamming algorithms, adjustable parameters, high-impact on receivers, suitable for advanced research	Complexity of implementation, uncertain effectiveness, need for high-technical knowledge, influence on the performance of other receivers
	Broad band[39]	Complete or partial drowning of the target signal, reducing the efficiency of the navigation receiver, high-effectiveness	High-energy consumption, need for wide bandwidth, complexity of implementation
	Narrow band[46,47]	High-energy density, large power spectrum density, good self-correlation, low-energy consumption	Lack of complete immersion of the target signal, sensitivity to anti-jamming technologies, limitation in application

Table 1. Types of suppression jamming with a summary of advantages and disadvantages.

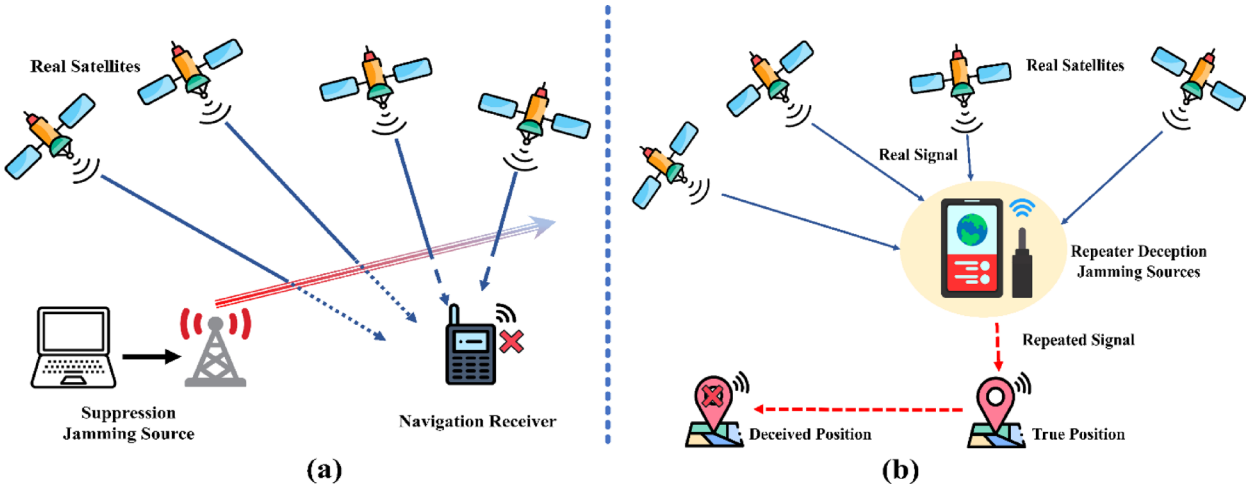

Fig. 4. Schematic diagram of deception jamming: (**a**) generated and (**b**) repeater.

$$j\left(t\right)=\sum_{k=1}^{K}A_{jk}.\omega_{k}\left(t\right).e^{j2\pi f_{c_k}t} \tag{4}$$

where A_{jk} is the amplitude of the jamming signal at the k-th GNSS frequency and $\omega_{k}\left(t\right)$ is the Gaussian white noise of independent processes for each frequency band.

In suppression jamming, the total received signal *r'(t)* at the GNSS receiver includes the legitimate GNSS signal, the jamming signal, and the inherent noise, which is given in Eq. (5).

$$r'\left(t\right)=r\left(t\right)+j\left(t\right)=\sum_{i=1}^{M}\alpha_{i}.c\left(t-\tau_{i}\right).e^{j(2\pi\left(f_{c}+f_{Di}\right)t+\phi_{i})}+j\left(t\right)+n\left(t\right) \tag{5}$$

Deception jamming

Deception jamming in GNSS is a sophisticated electronic warfare that aims to mislead GNSS receivers by manipulating or falsifying the received signals[37]. Unlike suppression jamming, which primarily raises the noise floor to degrade signal quality, deception jamming aims to deceive the receiver by providing false information, leading to incorrect navigation and timing solutions. Figure 4 shows a schematic diagram of deception jamming.

Table 2 lists the types of deception jamming, but this work primarily focuses on the repeater, as it is the most common form of deception jamming in GNSS (as shown in Fig. 4a). A repeater records legitimate GNSS signals and replays them with delays or intentional modifications, creating false positions[48,49]. The recorded signal is s(t), which is the reconstructed jamming signal, as expressed in Eq. (7).

	Type		Advantage	Disadvantage
Deception jamming[41]	Repeater[48,49]		Easy operation, usability for military codes, simple delay control, multiple implementation potential	Known as multi-path jamming, low-success rate in deception, large error in position deception, need to isolate and filter the received signal
	Generated[50,51]	Signal acquisition stage[52,53]	Forcing the receiver to lose tracking loop lock creates a high-power correlation peak	Increasing the noise floor, the possibility of detecting changes in the environment, the need for precise synchronization of the signal
		Signal tracking stage[54]	Adjustable code phase and rate, prevent receiver lock loss, stealth deception in tracking mode	The need for precise phase and code rate adjustment, the risk of detection by anti-deception algorithms, the need for precise power control
		Primary navigation[55,56]	Simple implementation and low-cost, basic technology, easy to implement	Difficulty in achieving synchronization with the real signal, easy detection by advanced detection methods, lower efficiency in deception
		Intermediate navigation[55,56,57]	Better synchronization with real signals, ability to deception military codes, good coverage and harder detection	Complex implementation, the need for external equipment and accurate synchronization, sensitivity to changes in the received signal
		Advanced navigation[57,58,59]	Highly effective, difficult to detect due to advanced simulation of real conditions, comprehensive deception	Very high-cost, limited to a small area close to the target receiver, requiring precise synchronization and technological advances

Table 2. Deception types of jamming with a summary of advantages and disadvantages.

$$j\left(t\right) = G.s\left(t - \Delta\tau\right) = G.\mathrm{A_c}.\mathrm{c}\left(\mathrm{t} - \Delta\tau - \tau\right).\mathrm{e}^{\mathrm{j}(2\pi\,\mathrm{f_c}(\mathrm{t}-\Delta\tau)+\Phi)} \tag{7}$$

where G is the gain added by the jammer and Δτ is the delay added by the jammer. The total received signal $r'\left(t\right)$ in the GNSS receiver is in the form of Eq. (8).

$$r'\left(t\right) = r\left(t\right) + j\left(t\right) = \sum_{i=1}^{M} \alpha_i.c\left(t - \tau_i\right).e^{j(2\pi\,(f_c+f_{Di})t+\phi_i)} + G.\mathrm{A_c}.\mathrm{c}\left(\mathrm{t} - \Delta\tau - \tau\right) .\mathrm{e}^{\mathrm{j}(2\pi\,\mathrm{f_c}(\mathrm{t}-\Delta\tau)+\Phi)} + n\left(t\right) \tag{8}$$

Generated jamming generates fake GNSS signals that mimic legitimate signals but carry false information about the satellite's identity, timing, or navigation data, as shown in Fig. 4b[50]. Generated jamming offers high-flexibility and strong concealment, as it can independently control various parameters. However, to generate effective jamming, the structure of the navigation signal must be known in advance. Since the structure of military navigation signals is typically unknown and difficult to crack, this presents a significant challenge for generating such jamming. Therefore, this jamming model cannot be implemented against military code signals, and its scope of use is limited.

Repeater jamming does not require knowledge of the satellite navigation signal structure to interfere with military codes. However, the receiver easily recognizes and treats it as multi-path interference, reducing its effectiveness and likelihood of success. If a jamming repeater retransmits the received satellite signal without accounting for the target receiver's actual position and velocity, it will cause a significant discrepancy between the authentic and deceived positions. Therefore, a critical step in repeater jamming is isolating and purifying the received signal, integrating information about the target receiver's position and velocity, and calculating and controlling the repeater's delay[59].

In a real-world scenario, suppression jamming is often employed initially to disrupt the satellite navigation receiver, causing it to lose lock. During recovery, a higher power jamming signal can generate a dominant correlation peak in the two-dimensional search space defined by Doppler frequency and code phase.

This peak can cause the receiver to lock onto the jamming signal during acquisition and tracking modes. However, due to the high-power signal, the noise floor of the receiver increases, and the obvious change of the receiver environment caused by the high-power signal is also possible to be detected by the receiver. Regarding the deployment of repeater jamming, it is mainly focused on how to obtain a good mapping relationship between the actual location of the target (real point) and the deceived location of the target (virtual point) and the reasonable delay of the repeater with the reasonable deployment of the jamming source location[60]. There is a mapping relationship between the actual and virtual points that can be realized physically. According to the number of jamming sources, it can be divided into single-station and multi-station jamming source deployment[61]. Single-station repeater jamming offers the advantages of simple equipment and an extensive, effective jamming range but has limited flexibility in signal source placement. In contrast, though more complex, multi-station repeater jamming allows for optimal deployment, enabling area and path mapping that enhances the effectiveness of deception jamming and extends its operational range[62].

The relationship between the numbers of repeaters n is expressed as the Eq. (9).

$$|S_iD| = |S_iR_i| + |R_iT| + ct_i \;,\; for\; i = 1,\, 2, \ldots,\, n \tag{9}$$

where S_i is the i-th satellite in the positioning area, R_i is the i-th source of repeater jamming, t_i is the signal delay for the i-th source of repeater jamming, T is the true position of the receiver, D is the deceived position of the receiver, c is the speed of light, $|S_iD|$ Distance from i-th satellite to position D, $|S_iR_i|$. The distance from the i-th satellite to the i-th source of repeater jamming and $|R_iT|$. It is the distance from the i-th source of

repeater jamming to the T position. The relationship of mapping close to the real position of T_o and the deceived position of D_o is in the form of an Eq. (10).

$$|S_iD_o| + ct_T = |S_iR_i| + |R_iT_o| + ct_i \ , \ for\ i = 1, 2, \ldots, n \tag{10}$$

where t_T is the clock difference at the receiver between the correct position T and T_o, Eq. (11) linearizes the system of equations using Newton's iterative method for n repeater jamming sources.

$$\begin{bmatrix} \frac{x_D - x_{S_1}}{|S_1D|} & \frac{y_D - y_{S_1}}{|S_1D|} & \frac{z_D - z_{s_1}}{|S_1D|} & C \\ \frac{x_D - x_{S_2}}{|S_2D|} & \frac{y_D - y_{S_2}}{|S_2D|} & \frac{z_D - z_{s_2}}{|S_2D|} & C \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_D - x_{S_n}}{|S_nD|} & \frac{y_D - y_{S_n}}{|S_nD|} & \frac{z_D - z_{s_n}}{|S_nD|} & C \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \\ \Delta_{t_T} \end{bmatrix} = \begin{bmatrix} |S_1R_1| + |R_1T_o| + ct_1 - |S_1D| \\ |S_2R_2| + |R_2T_o| + ct_2 - |S_2D| \\ \vdots \\ |S_nR_n| + |R_nT_o| + ct_n - |S_nD| \end{bmatrix} \tag{11}$$

where x_{S_i}, y_{S_i}, z_{s_i} is the position of satellite i and Δ_x, Δ_y, Δ_z are the coordinates of the deceived position and Δ_{t_T} is the recurring clock difference change.

The mapping relationship between the real position T and the deceptive position D for n repeaters can be expressed as Eq. (12).

$$\begin{bmatrix} \frac{x_{S_2} - x_{S_1}}{|S_1D| + |S_2D|} & \frac{y_{S_2} - y_{S_1}}{|S_1D| + |S_2D|} & \frac{z_{s_2} - z_{s_1}}{|S_1D| + |S_2D|} \\ \frac{x_{S_3} - x_{S_2}}{|S_2D| + |S_3D|} & \frac{y_{S_2} - y_{S_1}}{|S_2D| + |S_3D|} & \frac{z_{s_3} - z_{s_2}}{|S_2D| + |S_3D|} \\ \vdots & \vdots & \vdots \\ \frac{x_{S_n} - x_{S_{n-1}}}{|S_{n-1}D| + |S_nD|} & \frac{y_{S_n} - y_{S_{n-1}}}{|S_{n-1}D| + |S_nD|} & \frac{z_{s_n} - z_{s_{n-1}}}{|S_{n-1}D| + |S_nD|} \end{bmatrix} \begin{bmatrix} \Delta x_T \\ \Delta y_T \\ \Delta z_T \end{bmatrix} = \begin{bmatrix} \frac{x_{R_2} - x_{R_1}}{|R_1T| + |R_2T|} & \frac{y_{R_2} - y_{R_1}}{|R_1T| + |R_2T|} & \frac{y_{R_2} - y_{R_1}}{|R_1T| + |R_2T|} \\ \frac{x_{R_3} - x_{R_2}}{|R_2T| + |R_3T|} & \frac{y_{R_3} - y_{R_2}}{|R_2T| + |R_3T|} & \frac{z_{s_3} - z_{s_2}}{|R_2T| + |R_3T|} \\ \vdots & \vdots & \vdots \\ \frac{x_{R_n} - x_{R_{n-1}}}{|R_{n-1}T| + |R_nT|} & \frac{y_{S_n} - y_{S_{n-1}}}{|R_{n-1}T| + |R_nT|} & \frac{z_{s_n} - z_{s_{n-1}}}{|R_{n-1}T| + |R_nT|} \end{bmatrix} \begin{bmatrix} \Delta x_D \\ \Delta y_D \\ \Delta z_D \end{bmatrix} \tag{12}$$

where x_{R_i}, y_{R_i}, z_{R_i} position of the i-th jamming source Repeater, Δx_T, Δy_T, Δz_T are repeater position changes for the actual position T. For a proportional mapping between the real position and the deceived position, the Eq. (13) must hold.

$$\begin{bmatrix} \Delta x_T \\ \Delta y_T \\ \Delta z_T \end{bmatrix} = \begin{bmatrix} \Delta x_D \\ \Delta y_D \\ \Delta z_D \end{bmatrix} \tag{13}$$

Finally, this system solves and provides the deployment coordinates for a repeater jammer, and the mapping relationship between the real and deceived positions is ensured based on the satellite's and repeater jammer's geometry.

Test setup

After learning about the harmful phenomenon of jamming in GNSS, a laboratory platform is needed for data collection and testing. Section 4.1 discusses the laboratory substrate used. Section 4.2 mentions the number and classification of the resulting data.

Data collection

In this work, a static scenario example was implemented in the Iran University of Science and Technology (IUST) laboratory. Figure 5 shows the steps of implementing the jamming attack scenario in the presence of a signal recording and condition checking unit along with a jamming detector unit. In part (a), two B-200 GNSS Simulators connected to GNU radio on Host1 simulate the suppression and deception jamming signals similar to the original signal. Specifically, GNSS Simulator1 operates as a suppression jammer by emitting a powerful, broadband signal on the GLONASS L1 carrier frequency (1602 MHz), while GNSS Simulator2 functions as a deception jammer by broadcasting GNSS signals that mimic GPS location data. The Ublox-M8T receiver, a single-frequency device equipped with a UART serial output via the binary UBX protocol, provides various navigation parameters derived from raw data. In part (b), the system is modular, separating the signal recording and condition checking components from an independent jamming detector. This design facilitates maintenance and future upgrades while ensuring that each component can be optimized individually. A dedicated recording unit on Host2 captures the jamming signal and stores high-quality data, which is subsequently preprocessed to train machine learning or deep learning models. These models are then integrated into a real-time hardware evaluation framework, enabling efficient, independent operation in practical field deployments.

Figure 6 illustrates the setup of a GNSS jamming attack simulator and detector. In this study, a GPS deception jamming attack scenario is defined, where the GPS simulator serves as the source of the attack. The GNSS simulator functions as a suppression jammer, ensuring that positioning is carried out exclusively via GPS signals. According to this scenario, the receiver will receive wrong positioning data, which will be NoFix. This is while the receiver's real location is inside the IUST and is fixed.

A safe zone is provided for the testing area to prevent disturbance to other nearby electronic devices. This is achieved by adjusting the GNSS simulator's transmitter gain while observing the reception of GNSS signals with another receiver.

Dataset

The dataset used in this study was specifically collected and comprises 51,315 samples, representing a diverse range of operational scenarios and environmental conditions. This variety enhances the dataset's robustness,

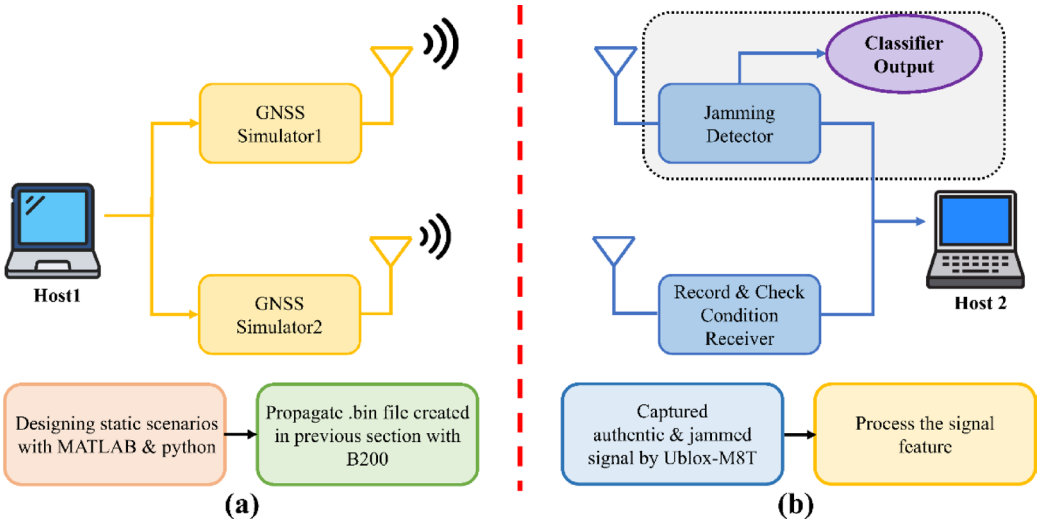

Fig. 5. Overview of the implementation process for a GNSS jamming attack scenario.

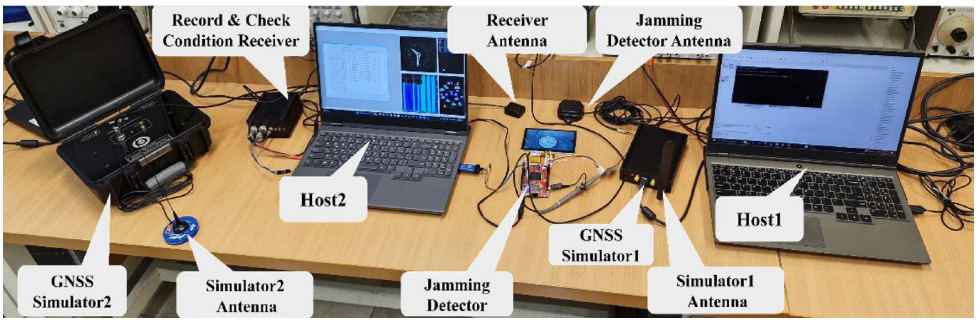

Fig. 6. Scenario generation equipment with jamming attack detector.

making it well-suited for developing reliable real-time jamming detection algorithms. The dataset is divided into two main categories: (1) clean (label 0), comprising 40,912 samples, and (2) jamming (label 1), comprising 10,403 samples. The diverse scenarios in the clean class enhance the system's adaptability to real-world challenges, reducing false positives and improving reliability in operational environments.

Classification development

After gathering the data samples required for detecting and classifying attacks, the datasets undergo pre-processing to ensure they are suitable for training and evaluating the classifiers. This step is crucial for improving the accuracy and performance of the detection models. Section 5.1 describes the pre-processing of the collected data, while Sect. 5.2 details the training process and performance evaluation of the classifiers.

Pre-processing

The Ublox-M8T receiver offers numerous effective features; however, ten of the most critical features have been selected to detect suppression and deception jamming attacks. These features are optimized for lightweight, real-time detection and are presented in Table 3. Next, the correlation among the ten selected features is evaluated using Spearman's correlation algorithm, which accounts for potential non-linear relationships between features, as illustrated in Fig. 7[63]. Finally, standard scaling is applied to the samples using the formula:

$$x'_{ij} = (x_{ij} - \mu_j)/\sigma_j \tag{14}$$

where x_{ij} is the value of the i-th sample for feature j, and μ_j and σ_j represent the mean and standard deviation of feature j, respectively.

In tabular data, each row corresponds to an observation, and each column represents an attribute or feature. In classification tasks, one of these columns typically contains the labels or classes the model aims to predict. When transforming tabular data into a time series format for classification using models like LSTM, selecting an appropriate window size is crucial. The window size determines the temporal context captured for each prediction, balancing the need to represent relevant patterns while avoiding unnecessary complexity that could

Feature	Description	Unit
Doppler	Standard deviation for the Doppler effect of satellites in view at a given moment	Hz
VACC	Estimating vertical accuracy	mm
NUMSV	The number of satellites available at any given time	-
SNR	Standard deviation for the carrier noise ratio of satellites in view at a given moment	-
SQM	An average of the signal quality of available satellites at any moment	-
Noise level	Noise level measured by the GNSS core	-
AGC	AGC monitor	-
CW jamming indictor	Indicates the occurrence of jamming	-
Clk Acc	Estimation of frequency accuracy	Ps/s
Fix mode	Receiver status to determine the position	-

Table 3. A summary of features extracted from the Ublox-M8T receiver module.

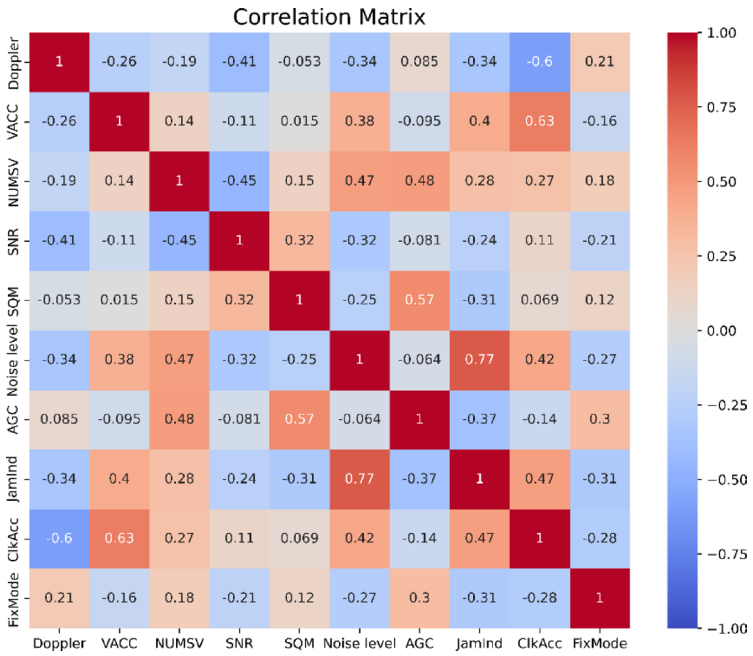

Fig. 7. Spearman's correlation of the features based on the collected data set.

degrade model performance. The window size determines the number of past observations considered when predicting the class for a given time step. Evaluating model performance using the F1 score is particularly useful, as it balances precision and recall. This is especially important for imbalanced datasets, where one class may significantly outnumber others, ensuring a fair assessment of the model's ability to identify minority classes accurately. In this case, the window size is adjusted based on the Ublox-M8T module's accessible frequency range, which spans from 1 to 10 Hz. This adjustment ensures that detection can occur quickly by accounting for data frame changes in GNSS signals. The optimal window size values determined for the LSTM, Gated Recurrent Unit (GRU), and Recurrent Neural Network (RNN) models are 8, 5, and 3, respectively, as shown in Fig. 8.

Classifiers performance assessment

The ML-based classifiers utilized in this study include Extreme Gradient Boosting (XGB), RF, KNN, Logistic Regression (LR), SVM, and Naive Bayes (NB). Each classifier represents a specific category in ML modelling: ensemble-based, instance-based, regularization-based, tree-based, Bayesian-based modelling, and DL-based classifiers, including MLP NN, LSTM, GRU, and RNN; these models are based on NNs. The hyperparameters of all classifiers are optimized using a randomized search algorithm, enabling the identification of the optimal configuration for each model customized to the given dataset. After determining the optimal hyperparameters, the classifiers are trained using the available dataset, validated for performance, and tested to ensure reliability and effectiveness. The following criteria are used to evaluate the performance of the classifier, including:

Matthews's Correlation Coefficient (MCC):

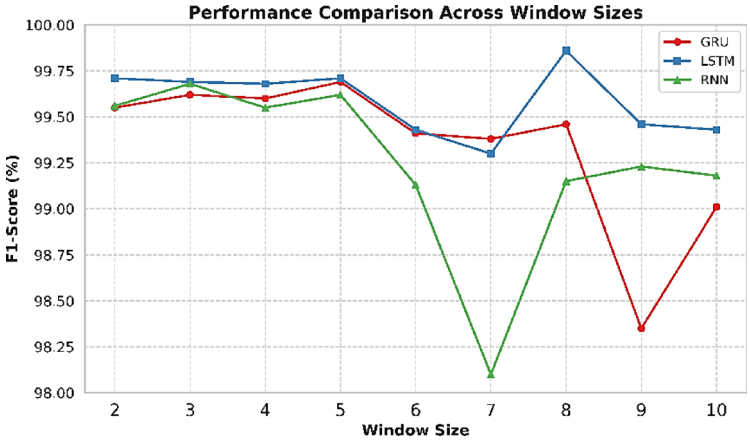

Fig. 8. Comparison of F1 score of GRU, LSTM and RNN models in different window sizes.

$$\text{MCC} = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{15}$$

Area Under the Precision-Recall Curve (PR AUC):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$\text{PR AUC} = \sum_{i=1}^{n-1} (Recall_{i+1} - Recall_i) \times \frac{Precision_i + Precision_{i+1}}{2} \tag{18}$$

Detection Rate (DR):

$$\text{DR} = \frac{TP + TN}{TP + FN + TN + FP} \tag{19}$$

F1-Score:

$$\text{F1}_{Score} = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{20}$$

TP, TN, FP, and FN are positive samples predicted as positive (i.e., true positive), negative samples predicted as negative (i.e., true negative), negative samples predicted as positive, and positive samples predicted as negative, respectively. The MCC is a robust metric that evaluates performance by accounting for TP, TN, FP, and FN. The metric is particularly effective in handling scenarios with imbalanced classes. Similarly, the PR AUC evaluates the trade-off between precision and recall. The PR AUC is calculated as the area under the precision-recall curve, which illustrates the relationship between precision and recall across varying thresholds. The PR AUC is particularly beneficial when working with unbalanced datasets, as it emphasizes the performance of the positive class. DR measures the proportion of true positive and true negative samples accurately identified by the classifier. Lastly, the F1-score, calculated as the harmonic mean of precision and recall, provides a balanced evaluation of a model's ability to correctly predict the positive class while minimizing false positives and false negatives.

In DL models, Floating Point Operations (FLOPs) quantify the number of computational operations a model performs during inference. This metric provides an estimate of a model's computational complexity and resource demands. Comparing models based on FLOPs allows for assessing their efficiency and scalability, aiding in decisions about their suitability for deployment in resource-constrained environments or real-time applications. Models with lower FLOPs typically operate more efficiently, making them well-suited for hardware with constrained processing resources like edge devices and mobile platforms. However, FLOPs alone are insufficient because they do not consider other factors affecting deployment performance, including the memory footprint, number of parameters, and bandwidth. In traditional ML models such as NB, LR, KNN, SVM, XGBoost, and RF, FLOPs are less used as a complexity measure than DL models. This is because their computational complexity strongly depends on factors such as the size of the dataset, the number of features, and the structure of the model (especially in the case of trees and ensemble models). Traditional ML models do not heavily depend on large matrix multiplications, unlike NNs, where FLOPs are more directly relevant and often expressed as

Model	DR (%)	Precision (%)	MCC	PR AUC	F1 (%)	TT (ms)	PT (ms)	TP	TSP (Byte)	FLOPs
LSTM	**99.95**	**99.73**	**0.9983**	**0.9986**	**99.86**	**260.38**	**0.14**	**891**	**3564**	**4821**
GRU	99.82	99.87	0.9944	0.9963	99.55	341.46	0.09	671	2684	3621
RNN	99.88	99.62	0.9964	0.9974	99.72	345.44	0.04	231	924	1221
MLP	99.81	99.75	0.9940	0.9959	99.52	236.87	0.03	77	308	157

Table 4. Criteria for two-class jamming DL models and classifiers.

Model	DR (%)	Precision (%)	MCC	PR AUC	F1 (%)	TT (ms)	PT (ms)	TOPs
Boosting (XGB)	**99.97**	**99.94**	**0.9992**	**0.9994**	**99.94**	**5221.36**	**0.68**	**180**
Bagging (RF)	99.92	99.81	0.9976	0.9976	99.81	3218.30	0.42	200
SVM	98.45	98.60	0.9832	0.9879	98.66	342.81	0.04	21
KNN	99.92	99.87	0.9976	0.9983	99.81	2241.25	0.29	-
LR	99.49	98.98	0.9844	0.9890	98.75	354.06	0.05	26
NB	96.38	95.87	0.9500	0.9641	96.02	520.20	0.07	143

Table 5. Comparison of the calculated criteria for two-class jamming recognition and classifier ML models.

Total Operations (TOPs). These operations encompass the computations performed by a trained model during inference and output detection. KNN algorithm necessitates storing the entire dataset in memory or ensuring offline accessibility. Its computational demand is notably high for large datasets, as its complexity increases with the number of training samples. Additionally, the hardware performance of KNN is primarily influenced by the efficiency of distance computations and sorting algorithms. Tables 4 and 5 show the evaluation scores of the resulting classifiers. The TT, PT, TP, and TSP are Total Testing Time, Average Prediction Time per Sample, Total Parameters, and Total Size of Parameters, respectively. Training, validation, and testing are performed on a 64-bit Windows 11 machine with an Intel(R) Core(TM) i7-12700 H 2.30 GHz and 16 GB of DDR5-4200 MHz memory. In this approach, the data is randomly shuffled across ten distinct iterations, with each iteration splitting the dataset into two parts: 70% is used for training the model, 15% for model validation, and 15% for final testing. This 5-fold cross-validation technique provides a more reliable model performance evaluation. In each iteration, a different subset of the data is reserved for testing, while the remaining data is used for training the model. Finally, the average scores of all these steps are reported as the model's overall accuracy.

The LSTM DL-based classifier is the top-performing model among time series-based approaches across various metrics. It achieves a 99.95% detection rate, with a precision of 99.73%. Additionally, it demonstrates a Matthews correlation coefficient of 0.9983 and a PR AUC of 0.9986. Moreover, it achieves a balanced accuracy of 99.97% and an F1-Score of 99.86%. Although the total test time is marginally lower at 260.38 ms compared to other models, the prediction time per sample is efficient at 0.14 ms. Furthermore, the model includes 891 trainable parameters, requiring a storage space of only 3564 bytes.

The results indicate that the ML-based XGB classifier stands out as the best model, achieving a recognition rate of 99.97% with a precision of 99.94%. It also demonstrates the highest MCC of 0.9992 and a PR AUC of 0.9994. Moreover, it achieves balanced accuracy of 99.96% and F1-Score of 99.94%. Despite the slightly higher total test time (5221.36 ms) compared to some other models, its prediction time per sample (0.68 ms) and the number of operations required to generate the network output are estimated to be 180 operations, which makes its overall performance has become the best choice. Figure 9 shows the confusion matrices of the optimal classifiers for XGB and LSTM. These matrices allow the evaluation of classifiers by showing the number of TP, TN, FP and FN samples. For instance, the XGB confusion matrix depicted in Fig. 9b demonstrates that 1567 jamming attack samples are correctly classified, with only one instance misclassified either a jamming attack identified as clean data or clean data identified as a jamming attack.

Hybrid model

Integrating modern and traditional approaches provides a balanced solution, leveraging the strengths of both methodologies while mitigating their weaknesses. Section 6.1 discusses the necessity of early warning mechanisms for detecting jamming phenomena. Section 6.2 introduces the advantages of the moving window method, and Sect. 6.3 proposes a hybrid model that combines ML techniques with the windowing approach for enhanced detection.

Early warning of entry and exit from jamming

In supervised learning, detecting class transitions in real-time systems, such as shifts from class zero to class one or vice versa, presents a considerable challenge. This is particularly critical in jamming detection applications, where early identification can enhance system responsiveness and prevent saturation or loss of positional accuracy. One effective way to address this challenge is to use windowing techniques to monitor certain parameters that indicate class change[64]. Analyzing the behavior of key features within a specified window of time or data points makes it possible to predict changes in classification before the transition is fully realized, enabling early detection and response. Windowing techniques employ thresholding methods by comparing the

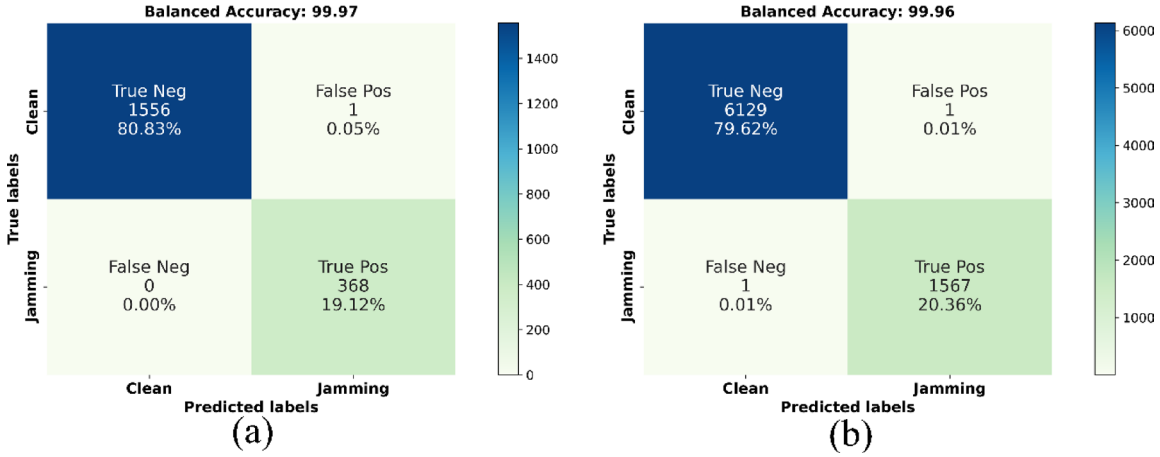

Fig. 9. Confusion matrices of optimal classifiers: (**a**) LSTM and (**b**) XGB.

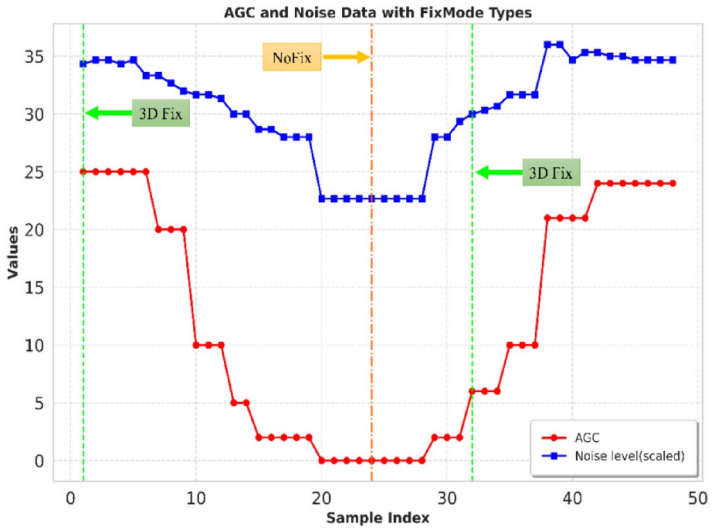

Fig. 10. Variations of AGC and noise level values in a given scenario under normal outage conditions.

aggregated results within each window to a predefined reference value. A flag is triggered to signal a potential class transition when the parameter consistently exceeds this threshold within the window[65]. This method enables proactive detection rather than reactive classification, offering the system additional time to adjust to changing conditions. Moreover, the windowing approach minimizes false positives and negatives, particularly when abrupt changes are inferred solely from single-point measurements.

Several studies[66,67] highlight the critical role of GNSS signal power levels in disrupting receiver lock status and introducing deception jamming scenarios. Consequently, monitoring parameters such as AGC and noise level, or equivalent metrics in various receivers, can provide early warnings of such occurrences before the GNSS receiver becomes fully saturated. Two critical aspects of early detection in jamming events are identifying both the onset and cessation of jamming. The primary objective is to alert the receiver prior to the jammer's full saturation, enabling a timely switch to alternative navigation solutions. On the other hand, detecting the cessation of jamming or exiting the jamming-contaminated area provides valuable insights into the effective range of the jamming signal. This awareness also accelerates the transition back to the primary navigation solution, namely the GNSS system, ensuring minimal disruption to operations. Figures 10 and 11 illustrate two real-world scenarios depicting changes in AGC and noise level and the corresponding positioning lock status in the Ublox-M8T receiver. Specifically, Fig. 10 demonstrates a transition scenario where movement occurs from an open space into a closed environment under NoFix GNSS signal conditions. In this scenario, noise level and AGC values noticeably decrease before the receiver transitions to a NoFix state. Conversely, noise level and AGC values increase upon exiting the enclosed space, enabling the receiver to regain its 3D Fix mode.

Figure 11 illustrates a laboratory scenario involving suppression jamming. Initially, the GNSS signal maintains a 3D Fix state. Upon initiating the jamming scenario, the signal transitions to a NoFix state. Eventually, the signal returns to its original 3D Fix condition with the jammer deactivated. In this scenario, the onset of jamming

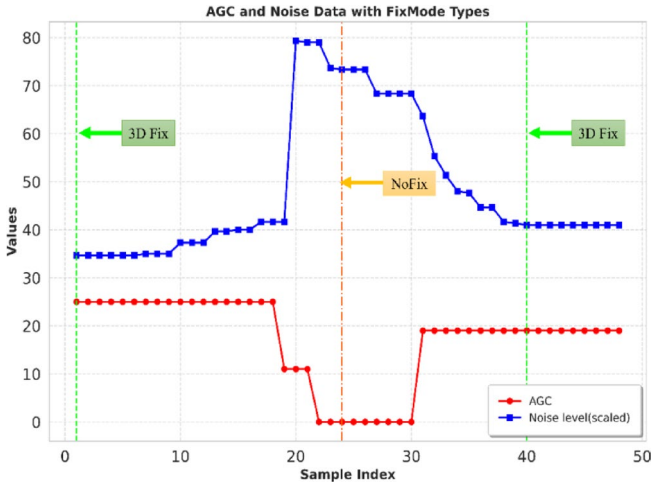

Fig. 11. Variations of AGC and noise level values in a specific scenario in jamming conditions.

causes a noticeable increase in the noise level value while the AGC value decreases. Conversely, when the jammer signal is disabled, the noise level value begins to decline, and the AGC value rises.

By monitoring the AGC and noise level parameters, it becomes feasible to detect the onset of a jamming scenario and differentiate it from low-accuracy positioning or signal interruption conditions. This enables timely switching to alternative navigation solutions before the user's system becomes saturated. Additionally, tracking these parameters can help identify when the jammer's range has been exited or the jamming scenario has concluded.

Moving window

Early detection can be achieved by employing a moving window algorithm to monitor variations in AGC and noise level values over time. This method allows for identifying trends that signal the initiation of a jamming scenario. The moving window algorithm is a time series analysis method that segments continuous data into smaller, overlapping intervals or windows. Key metrics like the mean are computed within each window to capture and summarize the signal's behavior. This approach is well-suited for analyzing AGC and noise level data, as it mitigates short-term variations while highlighting significant trends. In jamming detection, such trends manifest as sustained increases or decreases in AGC and noise level values, typically indicating the proximity or influence of a jammer on the GNSS receiver. The algorithm can identify gradual signal degradation by analyzing the mean values of consecutive windows, signaling potential interference. The use of overlapping windows ensures continuity in the analysis and enhances sensitivity to data variations while maintaining enough smoothing to reduce false positives caused by short-term fluctuations. The algorithm computes the average AGC and noise level values for each data window and compares them to the averages from the previous window. If the current average exceeds the previous one, an increase is identified, while a decrease is noted when the current average is lower. This process is applied to all windows sequentially, enabling the algorithm to monitor changes over time. A sequential thresholding mechanism is employed to reduce the likelihood of false alarms, ensuring that only consistent trends trigger alerts. For instance, if the algorithm identifies three consecutive increases in AGC or noise level values, it generates a warning signal, indicating potential interference. Similarly, a reduction alert is issued if three successive decreases are detected. This alert system minimizes false alarms by filtering out short-term fluctuations caused by environmental factors. It emphasizes sustained changes in signal quality, which are more likely to signify jamming activity.

Augmented jamming detection model and ML

Integrating ML with domain-specific signal processing techniques, such as feature extraction or pre-filtering, reduces data complexity and enhances real-time performance. This hybrid approach leverages signal processing for efficient real-time analysis while effectively utilizing ML's capability to model nonlinear and complex relationships. The hybrid model operates in two main steps: windowed signal monitoring and ML validation.

Windowed signal monitoring

The initial detection phase employs a windowing technique to analyze AGC and noise level signals. These signals are segmented into fixed-size windows, and the average values of the features are computed for each window. The model then monitors changes in these averages across successive windows. The system's core mechanism identifies patterns of three consecutive increases or decreases in AGC or noise level values, signaling potential anomalies or interference. This pattern indicates a sustained deviation from typical behavior, potentially signaling the onset of jamming. Upon detecting such a trend, the model issues an early warning highlighting the likelihood of interference.

Validation of ML with XGBoost

Although the windowing technique offers a rapid method for detecting anomalies, it is essential to validate whether these early irregularities genuinely signify a jamming event. The detection system integrates an XGBoost ML classifier specifically trained to identify jamming events. When an alarm is raised during the windowing step, XGBoost processes the current data to classify the scenario as either jamming (class 1) or clean (class 0). This multi-layered approach ensures that early warnings generated by the windowing technique are validated through a more advanced classifier, thereby minimizing the likelihood of false positives. Figure 12 illustrates the performance of the proposed hybrid model.

Flags and output

The final stage of the proposed model involves synthesizing the outcomes from both detection layers to deliver a concise and informative output. The model generates a series of flags that offer detailed insights into the detected event, including its severity, persistence, and classification status. These flags specify whether the system has identified high or low-signal reliability and detected conditions of jamming or exiting a jamming scenario. This interpretation layer facilitates more precise real-time decision-making by distinguishing between varying levels of jamming risk through the combined analysis of windowed signal monitoring and classifier output. The combination of the ML model and windowed method is based on a truth table so that:

1. When AGC and noise level values both increase within the same window, and the classifier determines no jamming is present, the system generates a HIGH VALID flag. This indicates that the positioning signal is robust and free from interference.
2. If AGC decreases while noise level increases, and the classifier confirms no jamming at the current moment, the system generates an IN-JAM flag. This suggests that jamming is likely to occur in the near future.
3. If AGC increases while noise level decreases, and the classifier detects jamming, the system generates an OUT-JAM flag. This indicates recovery and the system exiting a jamming event.
4. If AGC and noise level decrease and the classifier confirms no jamming, the system generates a LOW VALID flag, signaling a weak positioning signal.

In hybrid systems that integrate a classifier with an auxiliary technique like windowing to identify transitions between classes, the efficacy of the primary classifier predominantly determines the overall system's performance. If the classifier's performance is suboptimal, the hybrid model will propagate the errors of the primary model, resulting in inaccurate outcomes. Consequently, the overall effectiveness of the hybrid approach is fundamentally dependent on the performance of its primary classifier.

Implementation and comparison

As highlighted in Tables 4 and 5, the XGB model demonstrates the best performance among various ML and DL models for jamming detection. When deploying a pre-trained model like XGB on an STM32 microcontroller, it is crucial to ensure that evaluation parameters, such as accuracy and DR, remain consistent. Additionally, compatibility with the microcontroller's resource constraints and inference efficiency must be carefully assessed to ensure optimal performance in real-time applications. To assess the detector's performance accuracy, approximately 300 diverse scenarios both dynamic and stationary were conducted using GNSS satellite signals. Of these scenarios, 150 involved jamming attack simulations, while the remaining 150 represented valid signal conditions without interference. This study identified the timely or delayed detection of jamming attack entry and exit as key evaluation metrics for the detector receiver. Additionally, the performance of the Ublox antenna and receiver hardware plays a significant role in accurately signaling the onset and cessation of jamming attacks. However, under identical conditions utilizing a commercial antenna, the Ublox-M8T receiver

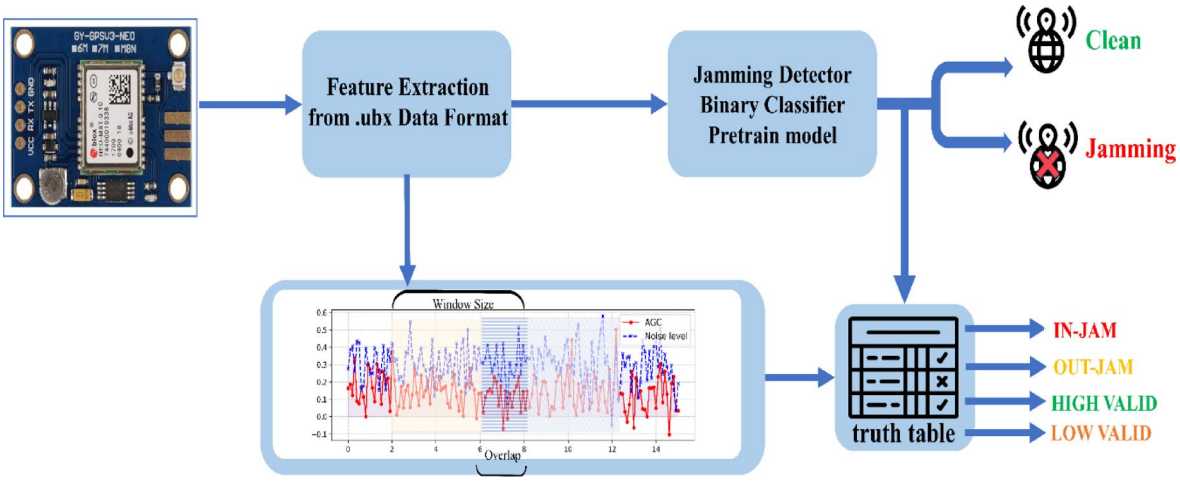

Fig. 12. Proposed hybrid model based on ML and windowing.

Model	Accuracy	Precision	Recall	Flash memory	RAM D1 memory	Average output time (us)	OS	Real-time capability	Windowing
XGB	**%98**	**%96**	**%93**	**%3.23**	**%2.60**	**20**	**FreeRTOS**	**Yes**	**Detect in/out jam**

Table 6. Classifier performance implemented in receiver detector using online test data.

Method	Investigated parameter	Equipment	Advantages	Limitations
LSTM-autoencoders[68]	Time series in the received signal	Software upgrade	Detecting all types of anomalies in the system, not needing to be labeled	Lack of attention to implementation, inability to distinguish the type of attack
BiLSTM-A[30]	Features available in the Ublox-M8T receiver	Software upgrade	High-accuracy in detection	Not paying attention to the real-time nature of the problem, need for high pre-processing
Federated learning[69]	Spectrogram image of GNSS signal	Software upgrade	Privacy and security, variety of detection	Communication delay, complexity in the model
ResNet[70]	Spectrogram image of GNSS signal	Software upgrade	Adaptability to new scenarios, comprehensive analysis	Computational complexity, extensive pre-processing, the need for dedicated receiver hardware, and implementation challenges
MobileNet-V2[27]	Spectrogram image of GNSS signal	Software upgrade, additional hardware	Comprehensive coverage of signal disturbance, scalogram-based feature extraction, transfer learning to reduce training time	Heavy computational requirements, dependence on data pre-processing, focus on power without considering environmental factors
RF[71]	Features available in the Ublox-M8 receiver	GNSS receiver (M8T), flight controller, raspberry Pi	Hardware implementation, effective model evaluation, minimal performance impact	Reliance on MAVLink protocol, lack of hardware and memory benchmarks, high-weight and dimensions
This work	**Doppler, VACC, NUMSV, SNR, SQM, noise level, AGC, jamming inductor, Clk Acc, fix mode**	**GNSS receiver (M8T), dual frequency active antenna, ARM processor**	**Real-time detection, low-cost, high-detection accuracy, high-reliability in detection and early warning**	**Expensive equipment if you need to add more frames**

Table 7. Qualitative comparison between the methods presented in previous works and those presented in this work.

and an STM32H7 microcontroller the overall performance of the jamming attack detector can be thoroughly evaluated. Furthermore, key complexity parameters, including FLASH memory usage, D1 RAM consumption, and microcontroller execution time, are considered for performance assessment. Additionally, quantization has been excluded to maintain the trained model's efficiency. The implementation results are summarized in Table 6.

Implementing XGBoost on the STM32H743 for jamming detection demonstrates robust model performance, achieving 98% accuracy, 96% precision, and 93% recall. These metrics highlight the model's reliability in detecting jamming events while maintaining a low-false positive rate. This model's missed events are minimal, and it demonstrates high efficiency in resource utilization, occupying only 3.23% of the available flash memory and 2.60% of RAM_D1. With an average output time of 20 µs, the model ensures real-time capability within the FreeRTOS environment, providing sufficient reaction time for the receiver operating at a maximum frequency of 10 Hz. Furthermore, the model incorporates a windowing mechanism to detect the onset and cessation of jamming events efficiently. Its robust performance makes it well-suited for deployment in real-time embedded systems. Table 7 compares the proposed method against referenced techniques in terms of equipment, advantages, and limitations. The results highlight that jamming detection using the XGBoost algorithm outperforms other methods' accuracy and efficiency.

Conclusions

This study explored ML and DL approaches for detecting jamming attacks. Ten key features were selected to identify suppression and deception jamming scenarios, requiring only minimal pre-processing. In addition to diagnostic analysis, a hybrid model was proposed to enhance jamming detection accuracy by integrating monitoring with traditional windowing features. This approach improves the system's capability to generate specific output flags and situational awareness, enabling it to respond appropriately to various signal states. The system's real-time capabilities ensure early detection of jamming events, allowing for rapid responses and minimizing disruptions to GNSS-dependent applications. Hardware was implemented using the STM32H7 series ARM processor and the Ublox-M8T receiver. Through practical and diverse testing, the performance of the jamming attack detector was evaluated and refined to detect various types of jamming attacks with acceptable accuracy. Additionally, when the GNSS jamming signal generator was activated or deactivated, the system performed satisfactorily in announcing both the entry into and exit from the jamming scenario. Finally, the qualitative comparison between this work's results and previous studies demonstrated that tree-based algorithms offer efficient performance in jamming attack detection. These algorithms provide greater reliability than DL methods, particularly for lightweight and real-time implementations.

Data availability

The dataset used and/or analyzed during the study available from the corresponding author on reasonable request.

Received: 18 January 2025; Accepted: 4 July 2025
Published online: 09 July 2025

References

1. Hegarty, C. J. & Chatre, E. Evolution of the Global Navigation SatelliteSystem (GNSS). *Proceedings of the IEEE* **96**(12), 1902–1917 (2008).
2. Zarrinnegar, K., Tohidi, S., Mosavi, M. R., Sadr, A. & De Andrés, D. M. M. Improving Cross Ambiguity Function using Image Processing Approach to Detect GPS Spoofing Attacks. *Iranian Journal of Electrical and Electronic Engineering* **19**(1), 1–12 (2023).
3. Gao, G. X., Sgammini, M., Lu, M. & Kubo, N. Protecting GNSS Receivers from Jamming and Interference. *Proceedings of the IEEE* **104**(6), 1327–1338 (2016).
4. Hu, Y., Bian, S., Li, B. & Zhou, L. A Novel Array-based Spoofing and Jamming Suppression Method for GNSS Receiver. *IEEE Sensors Journal* **18**(7), 2952–2958 (2018).
5. Kim, H., Jin, G. & Won, J. GNSS Cloud-data Processing Technique for Jamming Detection, Identification, and Localisation. *IET Radar, Sonar & Navigation* **14**(8), 1143–1149 (2020).
6. Zhou, W., Lv, Z., Wu, W., Shang, X. & Ke, Y. Anti-spoofing Technique Based on Vector Tracking Loop. *IEEE Transactions on Instrumentation and Measurement* **72**, 1–16 (2023).
7. Xie, J., Liu, Q., Wang, L., Gong, Y. & Zhang, Z. Localizing GNSS Spoofing Attacks using Direct Position Determination. *IEEE Sensors Journal* **22**(15), 15323–15333 (2022).
8. Tohidi, S. & Mosavi, M. R. GNSS Spoofing Detection using a Fuzzy Classifier based on Time-Frequency Analysis of the Autocorrelation Function. *GPS Solutions* **28**(3), 1–23 (2024).
9. Akos, D. M. Who's Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC). *Navigation* **59**(4), 281–290 (2012).
10. Broumandan, A., Siddakatte, R. & Lachapelle, G. An Approach to Detect GNSS Spoofing. *IEEE Aerospace and Electronic Systems Magazine* **32**(8), 64–75 (2017).
11. Miralles, D., Levigne, N., Akos, D. M., Blanch, J. and Lo, S. Android Raw GNSS Measurements as the New Anti-Spoofing and Anti-Jamming Solution, The International Technical Meeting of the Satellite Division of The Institute of Navigation, 334–344 (2018).
12. Wang, P., Cetin, E., Dempster, A. G., Wang, Y. & Wu, S. Time Frequency and Statistical Inference Based Interference Detection Technique for GNSS Receivers. *IEEE Transactions on Aerospace and Electronic Systems* **53**(6), 2865–2876 (2017).
13. Borio, D. Panova Tests and Their Application to GNSS Spoofing Detection. *IEEE Transactions on Aerospace and Electronic Systems* **49**(1), 381–394 (2013).
14. Yousif, T. & Blunt, P. Interference Mitigation for GNSS Receivers using FFT Excision Filtering Implemented on an FPGA. *Engineering* **3**(4), 439–466 (2022).
15. Sheridan, K., Ying, Y., and Whitworth, T. Pre- and Post-Correlation GNSS Interference Detection within Software Defined Radio, Proceedings of the ION GNSS Conference, 3542–3548 (2012).
16. Bek, M. K., Shaheen, E. M. & Elgamel, S. A. Mathematical Analyses of Pulse Interference Signal on Post-correlation Carrier-to-Noise Ratio for the Global Positioning System Receivers. *IET Radar, Sonar & Navigation* **9**(3), 266–275 (2015).
17. Liu, K., Wu, W., Wu, Z., He, L. & Tang, K. Spoofing Detection Algorithm Based on Pseudorange Differences. *Sensors* **18**(10), 3197 (2018).
18. Sun, K., and Zhang, J. GNSS Interference Detection Test by using Fractional Fourier Transform, IEEE Conference on Computer and Communications, 837–842 (2019).
19. Zhu, X., Zhang, Z., Gao, J. & Li, W. Two Robust Approaches to Multicomponent Signal Reconstruction from STFT Ridges. *Mechanical Systems and Signal Processing* **115**, 720–735 (2019).
20. Sun, K. & Zhang, T. A New GNSS Interference Detection Method Based on Rearranged Wavelet-Hough Transform. *Sensors* **21**(5), 1714 (2021).
21. Arribas, J., Fernandez-Prades, C. & Closas, P. Antenna Array Based GNSS Signal Acquisition for Interference Mitigation. *IEEE Transactions on Aerospace and Electronic Systems* **49**(1), 223–243 (2013).
22. Qin, W. & Dovis, F. Situational Awareness of Chirp Jamming Threats to GNSS Based on Supervised Machine Learning. *IEEE Transactions on Aerospace and Electronic Systems* **58**(3), 1707–1720 (2022).
23. Wang, C. Z., Kong, L. W., Jiang, J., and Lai, Y. C. Machine Learning-Based Approach to GPS Anti-jamming, GPS Solutions, **25**(3) (2021).
24. Morales Ferre, R., de la Fuente, A., and Lohan, E. S. Jammer Classification in GNSS Bands Via Machine Learning Algorithms, Sensors, **19**(22), 4841 (2019).
25. Yang, B. et al. Research on GNSS Interference Recognition based on Roi of Correlation Peaks. *International Journal of Satellite Communications and Networking* **40**(5), 330–342 (2022).
26. Yakkati, R. R., Pardhasaradhi, B., Zhou, J., and Cenkeramaddi, L. R. A Machine Learning based GNSS Signal Classification, IEEE Symposium on Smart Electronic Systems, 532–535 (2022).
27. Elango, A., Ujan, S., and Ruotsalainen, L. Disruptive GNSS Signal Detection and Classification at Different Power Levels using Advanced Deep-Learning Approach, International Conference on Localization and GNSS (ICL-GNSS), 1–7, (2022).
28. Liu, Z., Lo, S., and Walter, T. GNSS Interference Detection using Machine Learning Algorithms on ADS-B Data, ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation, Oct. (2021).
29. Xu, J., Ying, S. & Li, H. GPS Interference Signal Recognition based on Machine Learning. *Mobile Networks and Applications* **25**(6), 2336–2350 (2020).
30. Fu, D., Li, X., Mou, W., Ma, M. & Ou, G. Navigation Jamming Signal Recognition Based on Long Short-Term Memory Neural Networks. *Journal of Systems Engineering and Electronics* **33**(4), 835–844 (2022).
31. Chen, X., He, D., Yan, X., Yu, W. & Truong, T. K. GNSS Interference Type Recognition with Fingerprint Spectrum DNN Method. *IEEE Transactions on Aerospace and Electronic Systems* **58**(5), 4745–4760 (2022).
32. Reda, A., Mekkawy, T., Tsiftsis, T. A. & Mahran, A. Deep Learning Approach for GNSS Jamming Detection based PCA and Bayesian Optimization Feature Selection Algorithm. *IEEE Transactions on Aerospace and Electronic Systems* **60**(6), 8349–8363 (2024).
33. Ding, Y., and Pham, K. 1 GNSS Interference Identification beyond Jammer Classification, IEEE Aerospace Conference, 1–8 (2023).
34. Engel, F., Heiser, G., Mumford, P., Parkinson, K. & Rizos, C. An Open GNSS Receiver Platform Architecture. *Journal of Global Positioning Systems* **3**(1 & 2), 63–69 (2004).
35. Pascual, D., Park, H., Camps, A., Arroyo, A. A. & Onrubia, R. Simulation and Analysis of GNSS-R Composite Waveforms using GPS and Galileo Signals. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **7**(5), 1461–1468 (2014).
36. Sadrieh, S. N., Broumandan, A. & Lachapelle, G. Doppler Characterization of a Mobile GNSS Receiver in Multipath Fading Channels. *Journal of Navigation* **65**(3), 477–494 (2012).

37. Silva Lorraine, K. J. & Ramarakula, M. A Comprehensive Survey on GNSS Interferences and the Application of Neural Networks for Anti-jamming. *IETE Journal of Research* **69**(7), 4286–4305 (2021).
38. Karsi, M. F. & Lindsey, W. C. Effects of CW Interference on Phase-Locked Loop Performance. *IEEE Transactions on Communications* **48**(5), 886–896 (2000).
39. Xu, H., Cheng, Y. & Wang, P. Jamming Detection in Broadband Frequency Hopping Systems based on Multi-segment Signals Spectrum Custering. *IEEE Access* **9**, 29980–29992 (2021).
40. Islam, S., Bhuiyan, M. Z., Thombre, S. & Kaasalainen, S. Combating Single-frequency Jamming through a Multi-frequency, Multi-constellation Software Receiver: A Case Study for Maritime Navigation in the Gulf of Finland. *Sensors* **22**(6), 2294 (2022).
41. Zhou, Q., Li, Y. & Niu, Y. A Countermeasure against Random Pulse Jamming in Time Domain based on Reinforcement Learning. *IEEE Access* **8**, 97164–97174 (2020).
42. Li, B., et al., Influence of Sweep Interference on Satellite Navigation Time-domain Anti-jamming, Frontiers in Physics, **10** (2023).
43. Zhao, X., Huang, X., Tang, X., Feng, X. & Sun, G. Chirp Pseudo-noise Signal and Its Receiving Scheme for Leo Enhanced GNSS. *IET Radar, Sonar & Navigation* **16**(1), 34–50 (2021).
44. Baek, J., Seungsoo, Y., and Sun, Y., Jamming Effect Analysis of Two Chinese GNSS BeiDou-II Civil Signals, Int. J. Electr. Comput. Eng., 840–845, (2012).
45. Konovaltsev, A., Lorenzo, D., Hornbostel, A., and Enge, P. Mitigation of Continuous and Pulsed Radio Jamming with GNSS Antenna Arrays, Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2008), 2786–2795 (2008).
46. Huang, L. et al. Research on Detection Technology of Spoofing under the Mixed Narrowband and Spoofing Interference. *Remote Sensing* **14**(10), 2506 (2022).
47. Moussa, M. M., Osman, A., Tamazin, M., Korenberg, M. & Noureldin, A. Enhanced GPS Narrowband Jamming Detection using High-resolution Spectral Estimation. *GPS Solutions* **21**(2), 475–485 (2016).
48. He, X., Liao, K., Peng, S., Tian, Z. & Huang, J. Interrupted-sampling Repeater Jamming-suppression Method based on a Multi-stages Multi-domains Joint Anti-jamming Depth Network. *Remote Sensing* **14**(14), 3445 (2022).
49. Tang, C., Ding, J., Qi, H. & Zhang, L. Smart Forwarding Deceptive Jamming Distribution Optimal Algorithm. *IET Radar, Sonar & Navigation* **18**(6), 953–964 (2024).
50. Schmidt, D., Radke, K., Camtepe, S., Foo, E. & Ren, M. A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures. *ACM Computing Surveys* **48**(4), 1–31 (2016).
51. Meng, Q., Hsu, L. T., Xu, B., Luo, X. & El-Mowafy, A. A GPS Spoofing Generator using an Open Sourced Vector Tracking-based Receiver. *Sensors* **19**(18), 3993 (2019).
52. Kerns, A. J., Shepard, D. P., Bhatti, J. A. & Humphreys, T. E. Unmanned Aircraft Capture and control via GPS Spoofing. *Journal of Field Robotics* **31**(4), 617–636 (2014).
53. Gao, Y. & Li, G. A New GNSS Spoofing Signal Power Control Algorithm for Receiver Sensors in Acquisition Phase and Subsequent Control. *Sensors* **22**(17), 6588 (2022).
54. Psiaki, M. L., O'Hanlon, B. W., Bhatti, J. A., Shepard, D. P. & Humphreys, T. E. GPS Spoofing Detection via Dual-receiver Correlation of Military Signals. *IEEE Transactions on Aerospace and Electronic Systems* **49**(4), 2250–2267 (2013).
55. Xie Xiaogang, X. X., Lu Mingquan, L. M., and Zeng Dazhi, Z. D. Research on GNSS Generating Spoofing Jamming Technology, IET International Radar Conference, (2015).
56. Iudice, I., Pascarella, D., Corraro, G., and Cuciniello, G. A Real/fast-time Simulator for Impact Assessment of Spoofing & Jamming Attacks on GNSS Receivers, International Workshop on Metrology for AeroSpace (MetroAeroSpace), 309–314, (2024).
57. Pardhasaradhi, B., Srihari, P. & Aparna, P. Navigation in GPS Spoofed Environment using M-best Positioning Algorithm and Data Association. *IEEE Access* **9**, 51536–51549 (2021).
58. Meng, L., Yang, L., Yang, W. & Zhang, L. A Survey of GNSS Spoofing and Anti-spoofing Technology. *Remote Sensing* **14**(19), 4826 (2022).
59. Li, X. et al. Overview of Jamming Technology for Satellite Navigation. *Machines* **11**(7), 768 (2023).
60. Song, Z., Jingshu, Y., Gaofeng, P., and Jiabao, J. GPS Area-mapping Deceiving Unites Region Navigation Integrative System, International Conference on Computer Science and Information Technology, 189–191, (2010).
61. Priyadarshani, R., Park, K. H., Ata, Y., and Alouini, M. S. Jamming Intrusions in Extreme Bandwidth Communication: A Comprehensive Overview, arXiv, vol. 2403.19868, (2024).
62. Larsen, S. S., Jensen, A. B. & Olesen, D. H. Characterization of Carrier Phase-based Positioning in Real-world Jamming Conditions. *Remote Sensing* **13**(14), 2680 (2021).
63. Hauke, J. & Kossowski, T. Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *QUAGEO* **30**(2), 87–93 (2011).
64. Zhang, L., Zhao, J. & Li, W. Online and Unsupervised Anomaly Detection for Streaming Data using an Array of Sliding Windows and Pdds. *IEEE Transactions on Cybernetics* **51**(4), 2284–2289 (2021).
65. Morong, T., Puričer, P. & Kovář, P. Study of the GNSS Jamming in Real Environment. *International Journal of Electronics and Telecommunications* **65**(1), 65–70 (2019).
66. Borio, D., O'Driscoll, C., and Fortuny, J. GNSS Jammers: Effects and Countermeasures, ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing, (2012).
67. Xia, H. H. A Simplified Analytical Model for Predicting Path Loss in Urban and Suburban Environments. *IEEE Transactions on Vehicular Technology* **46**(4), 1040–1046 (1997).
68. Savolainen, O., Elango, A., Morrison, A., Sokolova, N., and Ruotsalainen, L. GNSS Anomaly Detection with Complex-valued LSTM Networks, International Conference on Localization and GNSS (ICL-GNSS), 1–7, (2024).
69. Wu, P., Calatrava, H., Imbiriba, T. & Closas, P. *Jammer Classification with Federated Learning* 228–234 (IEEE/ION Position, 2023).
70. Mehr, I. E., and Dovis, F. A Deep Neural Network Approach for Classification of GNSS Interference and Jammer, *TechRxiv*, (2023).
71. Nayfeh, M., et al., A Real-time Machine Learning-based GPS Spoofing Solution for Location-dependent UAV Applications, IEEE Conference on Electro Information Technology, 289–293, (2023).

Author contributions
All authors equally contributed in the article.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to M.R.M.