

Article

Benchmarking Frequency Hopping Algorithms for Resilient Wireless Communications Under Multiple Jamming Environments

Ivan Laktionov 

Department of Software of Computer Systems, Faculty of Information Technologies, Dnipro University of Technology, UA49005 Dnipro, Ukraine; laktionov.i.s@nmu.one

Abstract

Frequency-hopping spread spectrum (FHSS) information and communication systems are widely used to enhance the resilience of wireless networks to interference. The effectiveness of traditional algorithms is significantly reduced in the presence of active and intelligent jammers. The aim of this manuscript is to develop an adaptive FHSS algorithm based on statistical learning of frequency-channel quality and to analyze its effectiveness compared with existing algorithms. A software simulator has been developed that implements six FHSS algorithms, four intentional jamming models, and one baseline noise condition, and provides statistical evaluation based on sequential packet-level simulation, including bit error rate, error vector magnitude, normalized throughput, computational complexity, and resource assessment. The software-implemented adaptive algorithm delivered the best results among all those studied, achieving the highest overall performance metric. Compared with a system without FHSS, a 64.43% reduction in the bit error rate, 22.80% increase in throughput and 60.71% reduction in the error vector magnitude were achieved. The results obtained confirm the feasibility of using adaptive statistical frequency channel selection to improve the interference resilience of FHSS systems. Promising areas for further research include the modeling of intelligent jammers and the experimental verification of the implemented algorithm on computing platforms with limited resources.

Keywords: frequency hopping; anti-jamming communications; wireless system; statistical channel learning; software; spectrum adaptation

1. Introduction

At present, there is dynamic growth in the number of wireless communication systems. These systems form the information and communications infrastructure for a significant number of technical devices, such as autonomous robotic platforms, unmanned aerial vehicles, Internet of Things systems, and others. This, in turn, significantly increases the demands placed on the resilience of information transmission channels in the face of deliberate electronic jamming. One of the most widespread and dangerous threats to wireless information and communication networks is active jamming. Indeed, such radio-electronic jamming can significantly reduce throughput, increase the probability of packet loss, and disrupt the functioning of systems as a whole [1–3]. Consequently, the development and implementation of effective methods to counter deliberate jamming are among the main areas of current research in protecting information and communication systems and networks [4].



Academic Editor: Andreas Kassler

Received: 22 July 2026

Revised: 10 September 2026

Accepted: 15 September 2026

Published: 16 September 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

A method for ensuring interference-resistant communication that has become widely used in modern practice is FHSS [4,5]. This method is based on periodically varying the operating frequency according to a defined hopping sequence. However, despite extensive global experience in the application of FHSS, modern electronic warfare systems are increasingly utilizing adaptive and intelligent algorithms capable of analyzing hopping patterns and predicting future transmission frequencies [4,6]. Recent scientific and applied research shows that a significant number of modifications to the FHSS have been proposed to date. In particular, relevant scientific studies have focused on the use of multi-sequential frequency hopping [7], the optimization of FHSS index modulation [8], the increase in throughput of frequency-hopping systems [9], and the application of reinforcement [10,11] and deep [12] learning algorithms. A separate area of research involves the use of quantum random number generators (QRNGs). This approach allows the generation of practically unpredictable frequency-hopping sequences, thereby enhancing the cryptographic security of FHSS systems [13].

Along with the development of FHSS, models of intentional jamming are also being actively modernized. In addition to traditional (narrowband and broadband) jamming, reactive jamming is attracting considerable attention from researchers. Devices that generate reactive interference begin emitting it almost instantly upon detecting the useful signal. Such models are considerably more dangerous because they consume virtually no energy in standby mode and can precisely focus their interference on active transmission channels [14,15]. It is also worth noting that intelligent jamming systems have been the subject of intensive research in recent times. Such systems predominantly use deep learning and reinforcement learning methods to adapt their own strategy for suppressing wireless networks [16,17].

Much current scientific research and practical development also focuses on using artificial intelligence algorithms to improve the interference resilience of wireless systems. The application of reinforcement learning enables the dynamic adaptation of transmission parameters based on the current state of the radio channel, the history of interference sources, and accumulated statistics on frequency usage [6,11,12]. Similar approaches to optimizing technical and functional modes are actively used in satellite-ground information and communication systems [18] and reconfigurable intelligent surface-assisted telecommunication systems [19].

Recent studies, therefore, increasingly emphasize adaptive decision-making, intelligent spectrum management, and environment-aware communication strategies as important directions for improving the resilience of wireless communication. These developments highlight the transition from predefined transmission strategies toward methods that can exploit information about the current communication and interference environment.

A particular area of research that warrants attention in this article is the comparison of FHSS algorithms and the evaluation of their effectiveness in various scenarios involving intentional interference, using computer modeling techniques. Such research enables an objective analysis of the advantages of different FHSS strategies in the early stages of designing wireless information and communication systems, and allows for repeated replication of the computerized experiment. For example, in [9], the authors propose a method to increase the throughput of FHSS systems by statistically analyzing the bit error rate (BER) and interference resilience. Such an approach has confirmed the validity of using modeling as the primary tool for evaluating the effectiveness of FHSS algorithms. At the same time, recent research increasingly uses adaptive methods for selecting frequency channels. In particular, the scientific work [20] proposes a frequency-hopping algorithm utilizing reinforcement learning. This research applies modeling methods to formulate a channel selection strategy based on the current state of the interference environment. A

similar line of research is presented in [21], where adaptive frequency hopping is implemented using an algorithm that estimates the interference environment, thereby enhancing the effectiveness of FHSS under variable operating conditions. Furthermore, a significant number of contemporary scientific studies consider FHSS not only as a means of countering radio-electronic interference, but also as a subject of adaptive analysis and classification [22]. This indicates that a promising direction for the development of FHSS is the combination of classical frequency-hopping algorithms with adaptive decision-making mechanisms and software-oriented modeling [5,23]. This allows their effectiveness to be assessed across a wide range of scenarios involving intentional interference.

Recent studies increasingly employ reinforcement and deep learning to adapt frequency-hopping strategies to changing interference conditions. The authors of [24] formulated radar anti-jamming frequency hopping as a Markov decision process and used reinforcement learning to adapt the frequency-hopping interval according to the jamming environment. The authors of [25] further considered a two-dimensional action space involving transmission frequency and power and developed a deep reinforcement learning approach for anti-jamming frequency-hopping communications. These approaches demonstrate the effectiveness of learning-based adaptation, particularly in dynamic and partially unknown interference environments. However, they generally introduce additional requirements for model training, state representation, reward design, neural network inference, or iterative learning, which may increase implementation complexity and computational cost.

Alternative approaches focus on increasing the unpredictability and security of frequency-hopping sequences. Chaotic frequency-hopping methods have been investigated for anti-jamming wireless and UAV communications, exploiting the sensitivity and pseudo-randomness of chaotic maps to make hopping sequences less predictable [26]. More broadly, biologically inspired and machine-learning-based methods have been investigated to improve the covert and secure characteristics of wireless communication systems [27,28]. These studies demonstrate that increasing sequence unpredictability or incorporating intelligent signal-processing mechanisms can provide important security benefits. Nevertheless, sequence-generation mechanisms alone do not necessarily provide adaptive responses to the current interference state of individual frequency channels.

Considering the significant number of studies devoted to FHSS systems, the analysis of relevant scientific publications confirms the need for further research on developing approaches for the comprehensive evaluation of various FHSS algorithms across different operational scenarios. This conclusion is based on the fact that most authors analyze a limited number of algorithms for a specific type of interference within a restricted range of signal-to-noise ratios (SNR). This makes it difficult to objectively compare the effectiveness of different frequency-hopping methods under reproducible operating scenarios. However, a unified and reproducible comparison of conventional, stochastic, deterministic, chaotic, quantum-like, and lightweight adaptive FHSS mechanisms under the same set of interference conditions remains insufficiently addressed.

Models of electronic warfare jamming require further attention. This is because most published works use additive white Gaussian noise (AWGN) or a single type of deliberate jamming. Most real-world electronic warfare systems combine broadband, narrowband, and reactive jamming mechanisms. Consequently, the known results require further investigation when implemented across a wide range of practical application scenarios.

Particular attention should be paid to adaptive FHSS algorithms. The vast majority of modern approaches utilize deep learning or reinforcement learning, which are characterized by high computational complexity and lengthy training. For real-world wireless devices with limited computational resources, such solutions can be difficult to integrate into software. Therefore, the search for algorithms that combine ease of implementation,

reduced implementation and training requirements, and adaptability to changes in the interference environment remains a priority.

An analysis of the existing literature also shows that FHSS algorithms are most often evaluated solely on traditional metrics such as BER or throughput. At the same time, error vector magnitude (EVM) characteristics and frequency channel utilization statistics are considered far less frequently. It is precisely the comprehensive use of such criteria that allows for a more objective assessment of the effectiveness of FHSS algorithms, not only in terms of noise immunity but also in terms of the predictability of their operation.

The results of the comparative analysis of representative FHSS and anti-jamming approaches, along with the identification of research gaps, considering the need for a comprehensive comparison of existing approaches and the proposed adaptive algorithm within a unified software environment, taking into account reduced computational requirements for model training and deployment, are presented in Table 1.

Table 1. Comparative analysis of representative FHSS and anti-jamming approaches and identification of research gaps in relation to the aim and objectives of this article.

Main Methodology	Learning and/or Training Requirements	Main Metrics	Identified Research Gap	Approach to Addressing	Reference
Reinforcement learning against reactive jammers	Reinforcement learning training and reward	Throughput and anti-jamming performance	Focused on only reactive jammer and learning-based strategy	Multiple jammer types evaluated under identical conditions	[6]
Deep and reinforcement learning-based hopping strategy	Reinforcement learning and deep Q-learning	Anti-jamming communication performance	Requires training and neural-network inference	Online statistical adaptation without model training	[20]
Interference sensing for adaptive FHSS	Spectrum/interference estimation	Detection and adaptive FHSS performance	Focus on interference sensing rather than unified comparison of FHSS algorithms	Joint channel-quality adaptation and comparative evaluation	[21]
Reinforcement learning-based adaptive FHSS interval	Reinforcement training and reward formulation	Anti-jamming performance, interception and integration efficiency	Training-based adaptation and radar-specific formulation	Lightweight online statistical channel-quality learning without offline training	[24]
Deep learning with a dual-action network	Deep learning training and inference	Throughput, energy, switching overhead	Higher model and computational complexity	Statistical channel adaptation with a substantially simpler learning architecture	[25]
Chaotic frequency hopping	No machine learning training	Communication and security performance	Improves sequence unpredictability but does not explicitly learn channel quality	Adaptive statistical channel-quality estimation	[26]
Review of biologically inspired covert communication	Not applicable	Covert communication characteristics	Focuses on a review of covert acoustic communication	Unified FHSS anti-jamming simulation for wireless communication	[27]

The comparative analysis presented in Table 1 reveals several unresolved issues regarding the article's aim and objectives. First, learning-based adaptive FHSS approaches provide powerful mechanisms for responding to dynamic interference but generally require model training, reward design, or neural-network inference. Second, chaotic and quantum-based hopping algorithms improve the unpredictability of the hopping sequence but do not necessarily provide channel-state adaptation based on transmission feedback. Third, many existing studies evaluate a specific FHSS mechanism against a limited set of jammers, making direct comparisons between different hopping strategies difficult. Finally, computational complexity and signal-quality indicators, such as EVM, are less frequently considered alongside BER and throughput. These identified research gaps motivate the development of a lightweight adaptive mechanism and its evaluation within a unified simulation framework.

A distinctive feature of the approach proposed and implemented in this article is the formalization and software implementation of six different FHSS algorithms: conventional fixed-frequency transmission (NoFHSS), random frequency-hopping spread spectrum (RandomFHSS), linear feedback shift register-based frequency-hopping spread spectrum (LFSRFHSS), chaotic frequency-hopping spread spectrum based on the tent map (ChaoticFHSS), quantum random number generator-based frequency hopping spread spectrum (QRNGFHSS) and adaptive frequency hopping spread spectrum with statistical channel learning (AdaptiveFHSS).

Another important contribution of this work is the formalization and software implementation of a comprehensive set of jamming models, which includes AWGN, narrowband, broadband, reactive and combined jammers. The combined interference model integrates broadband, narrowband and reactive interference mechanisms, enabling the simulation of more realistic operational scenarios for electronic warfare systems.

Unlike most well-known studies, the performance of the algorithms is evaluated through a comprehensive statistical experiment that covers various types of interference, several SNR levels, and 200 sequential packet-level observations for each parameter combination. In addition to the traditional metrics of BER, throughput, and EVM, the statistical characteristics of frequency-hopping algorithms are also analyzed, in particular, computational and memory requirements and the ranking of algorithms according to integral performance metrics.

Accordingly, the main contributions of this study are a unified software-based simulation framework for the reproducible comparison of six FHSS algorithms; an online AdaptiveFHSS mechanism based on statistical channel-quality learning without offline training; a unified set of AWGN, narrowband, broadband, reactive and combined jamming models; a comprehensive evaluation based on BER, throughput, EVM and frequency-channel utilization under identical simulation conditions.

So, the main aim of this article is to conduct a comprehensive comparative analysis of FHSS algorithms to ensure interference-resistant wireless communication across various intentional interference models, using a unified sequential packet-level computer simulation environment. The article compares the NoFHSS, RandomFHSS, LFSRFHSS, ChaoticFHSS, QRNGFHSS and AdaptiveFHSS algorithms under identical operating conditions, and their performance is evaluated in terms of BER, throughput and EVM for AWGN, narrowband, broadband, reactive and combined jammer models.

In contrast to training-based adaptive FHSS approaches, the proposed AdaptiveFHSS does not rely on an offline dataset, a neural network model, or a separately trained predictive mechanism. Instead, it performs online statistical estimation of frequency-channel quality from transmission feedback and combines the exploitation of high-quality channels with controlled random exploration.

The remainder of the article is organized as follows. Section 2 describes the FHSS algorithms considered, mathematical models, jamming scenarios, and the evaluation methodology. Section 3 presents the simulation results and comparative analysis. Section 4 discusses the main findings, limitations, and directions for further research. Finally, Section 5 summarizes the main conclusions of the study.

2. Materials and Methods

2.1. Simulation Framework

The computer experiments conducted as part of the research were performed within a single modeling software environment written in Python 3.12.13. The following libraries were used in the software's implementation: math, random, numpy==2.0.2, pandas==2.2.2, matplotlib==3.10.0. The model architecture is designed to ensure uniform evaluation

conditions across all investigated FHSS algorithms, regardless of interference type. Each experiment simulates the complete transmission cycle of a single data packet from the transmitter to the receiver, passing through all stages of the wireless communication system sequentially.

The structure of a single computer experiment is shown in Figure 1. At the beginning of each iteration, a new data packet consisting of random bits of a fixed length (4096 bits by default) is generated. The data packet is then modulated using quadrature phase-shift keying (QPSK). Next, a signal propagation environment is established, in which information is accumulated regarding the utilization of frequency channels, transmission results and the impact of intentional interference. Once the frequency-hopping sequence has been generated by the FHSS algorithm, the transmitted signal is formed and then the corresponding jamming model is applied. At the receiving side, the signal is demodulated and transmission quality metrics are calculated. The results are then used to update the statistics of the adaptive algorithm and the channel model.

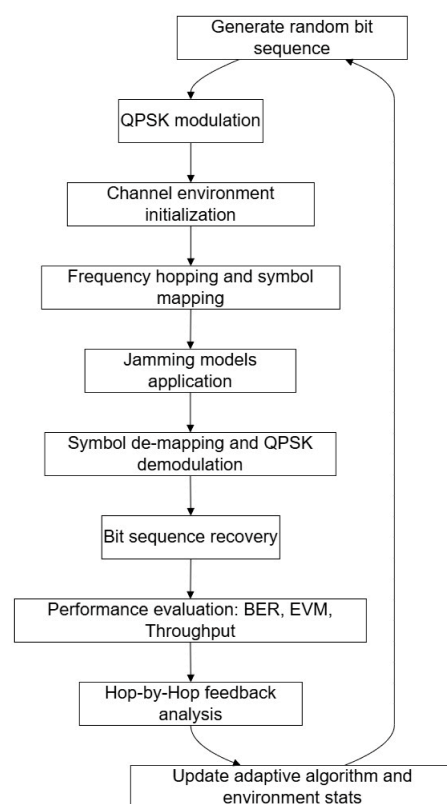


Figure 1. A detailed flowchart of a single computer experiment.

To ensure systematic evaluation of the investigated scenarios, a sequential packet-level simulation approach was used [29]. The space of computer experiments was defined taking into account the possibility of generating all possible combinations of the FHSS algorithm, the type of jammer, the SNR and the number of sequential packet-level observations. The study utilized six frequency-hopping algorithms (NoFHSS, RandomFHSS, LFSRFHSS, ChaoticFHSS, QRNGFHSS, and AdaptiveFHSS), four intentional jamming models and one baseline noise condition (AWGN, narrowband, broadband, reactive and combined jammer), and six SNR levels ranging from -5 dB to 20 dB in 5 dB steps. The SNR values were used as channel-model parameters for determining the variance of the AWGN component added to the received complex signal. Thus, SNR represents the signal-to-noise ratio assumed for the communication channel and is applied consistently across all investigated FHSS algorithms. For each parameter combination, 200 sequential packet-level runs were performed, with

the adaptive algorithm and channel environment retaining their state between successive packets. Thus, the total number of packet-level simulation runs amounted to 36,000.

Unlike a conventional Monte Carlo scheme based on statistically independent repetitions, the present model employs a persistent learning mechanism. For each series of experiments, a single instance of the FHSS algorithm and a single instance of the environment model are created, which retain the accumulated statistics between individual packets [30]. Therefore, the 200 runs within each parameter combination should be interpreted as sequential observations from a persistent simulation trajectory rather than as fully independent statistical replicates. As a result, the adaptive algorithm gradually forms estimates of the quality of the frequency channels and utilizes them when generating subsequent frequency-hopping sequences, as illustrated in Figure 2.

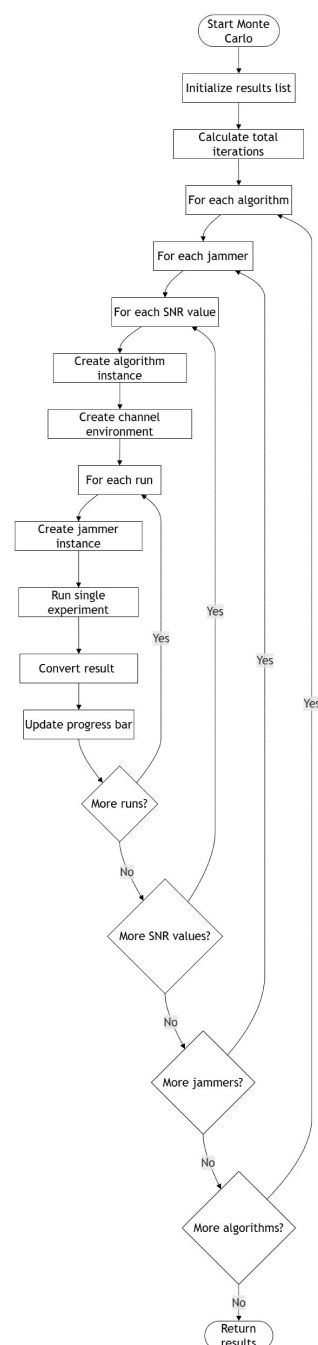


Figure 2. A flowchart of the sequential packet-level simulation cycle incorporating a persistent learning approach.

Thus, once the transmission of each packet is complete, the reception results are used to update the medium statistics and the adaptive algorithm in accordance with (1):

$$S_{k+1} = f(S_k, R_k), \quad (1)$$

where S_{k+1} is the updated state of the modeled environment; S_k is the current state of the modeled environment; R_k is the result of transmitting the k th data packet; $f(\cdot)$ is the statistics update function.

For AdaptiveFHSS, the update function $f(S_k, R_k)$ corresponds to the channel-quality update procedure described in Section 2.3, where the transmission outcome R_k is used to update the quality estimate of the selected channel. The resulting quality estimates are then used by the adaptive selection procedure in subsequent hopping cycles.

So, the approach employed, as set out in (1), enables the accumulation of experience by an adaptive algorithm to be modeled during the system's long-term operation.

2.2. Infocommunication System Model

A model of a digital wireless communication system using FHSS was proposed and implemented in this article. The overall system architecture comprises a data sequence generator, a digital modulator, an FHSS transmitter, a transmission channel model, a noise generator, an FHSS receiver, a demodulator and a transmission quality assessment unit. A generalized view of the software-implemented system model is shown in Figure 3.

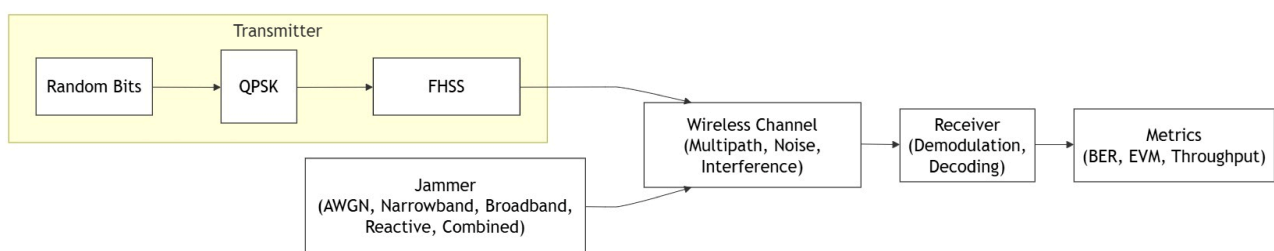


Figure 3. A block diagram of a software-implemented infocommunication system.

When performing computer experiments, data packets of equal length were used, which ensured the accuracy of the comparative evaluation of FHSS algorithms. In accordance with the model's basic configuration, each packet contains 4096 data bits. Following QPSK modulation, each complex symbol corresponds to two bits of information (the order of quadrature phase shift keying is 4). Thus, the number of complex symbols in a packet was 2048. Information is transmitted using frequency hopping. Based on the adopted model parameters, each frequency hop corresponds to 16 QPSK symbols. Therefore, the total number of frequency hops per packet is 128. These parameters remained constant for all investigated algorithms and all simulation scenarios. However, they can be adapted to the conditions of specific practical applications by adjusting the numerical values of relevant variables.

QPSK was used for digital data transmission, enabling two information bits to be represented by a single complex symbol. To verify the correct implementation of the physical layer, both the time-domain waveforms of the in-phase and quadrature components and the frequency-domain spectrum of the QPSK baseband signal were analyzed. The corresponding results are presented in Figures 4 and 5, respectively.

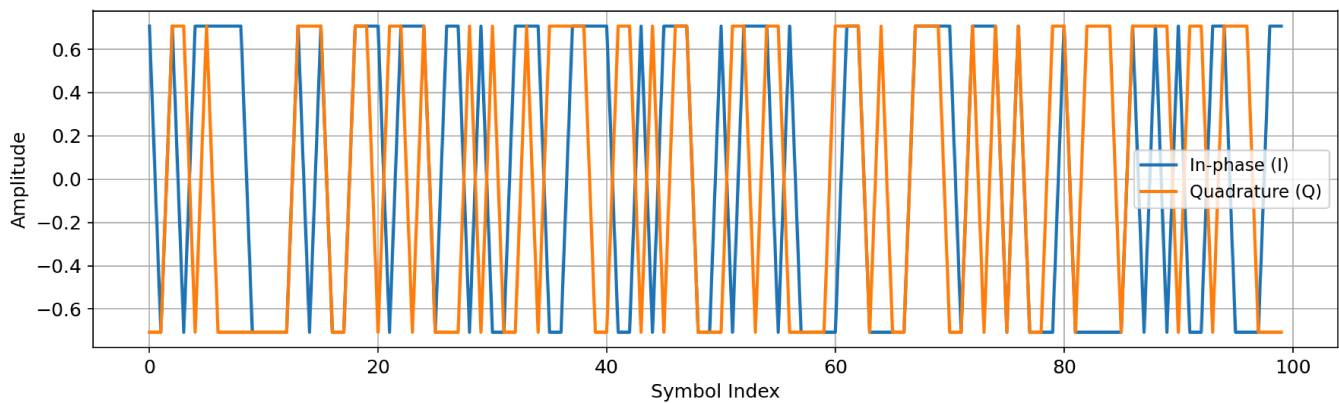


Figure 4. A fragment (the first 100 symbols of a data packet) of a QPSK baseband signal.

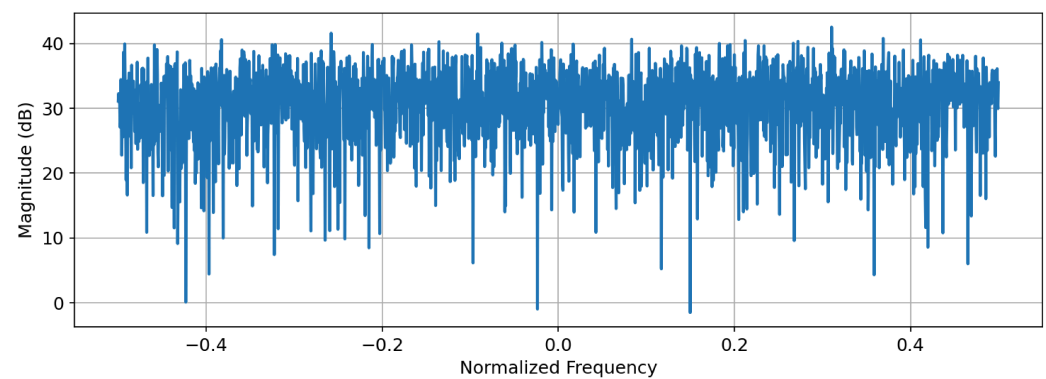


Figure 5. Frequency-domain spectrum of the QPSK baseband signal.

A preliminary spectral analysis confirms that the composite reference signal has been correctly formed and that there are no unwanted spectral components that could distort the results of the jamming simulator modeling.

Following digital modulation, the resulting sequence of symbols is transmitted using a frequency-hopping approach. In this case, the software-implemented structure of a single packet is described by the following logical sequence:

$$s_i = \{s_1, s_2, \dots, s_{2048}\} \rightarrow H_j = \{h_1, h_2, \dots, h_{128}\} \rightarrow \text{Segments}_k = \{\text{Segm}_1, \text{Segm}_2, \dots, \text{Segm}_{128}\}, \quad (2)$$

where s_i is a formed sequence of complex symbols; H_j is a generated hopping sequence; Segments_k are individual signal segments transmitted at the corresponding frequencies.

A specialized software class has been created to model the transmission process. This class implements the interface between the FHSS algorithm and the jamming models. The main functions are accumulating usage statistics for each frequency channel; storing information about channels affected by interference; generating a quality estimate for each channel; and providing an adaptive channel-quality estimation mechanism for the adaptive FHSS algorithm. A quality estimate Q_i is maintained for each frequency channel and updated based on transmission feedback. The specific update and channel-selection procedures implemented in AdaptiveFHSS are described in Section 2.3.

Consequently, the block diagram shown in Figure 3 was translated into a software model based on the following formalized description. The general form of the signal transmission model is as follows:

$$RS = ISM(H(M(b))), \quad (3)$$

where RS is the received symbol; ISM is the formalized model of the jammer; H is the formalized frequency-hopping procedure in accordance with the selected FHSS algorithm; M is the formalized QPSK modulation procedure; b is the information bit sequence.

After reception, the inverse transformation is carried out:

$$\hat{b} = D(RS), \quad (4)$$

where $\hat{b}$ is the result of the inverse transformation of the bit sequence; D is the formalized procedure for demodulation and reconstruction; RS is the received symbol.

Thus, the results of executing the formalized procedures (3) and (4) provide the mathematical basis for the software implementation of the information and communication system's functional principles under investigation. The resulting bit sequences are used to compute metrics that comprehensively characterize the quality of the FHSS algorithms.

2.3. Implemented FHSS Algorithms

This article is devoted to a comparative analysis of six frequency-hopping algorithms (NoFHSS, RandomFHSS, LFSRFHSS, ChaoticFHSS, QRNGFHSS and AdaptiveFHSS), which differ in the way they generate the hopping sequence. All algorithms use the same digital communication system model, the same QPSK modulation, the same packet size and operate under the same conditions of intentional interference (see Figure 3). This ensures an objective comparison of their performance. Figure 6 shows examples of frequency-hopping sequences generated by all investigated FHSS algorithms.

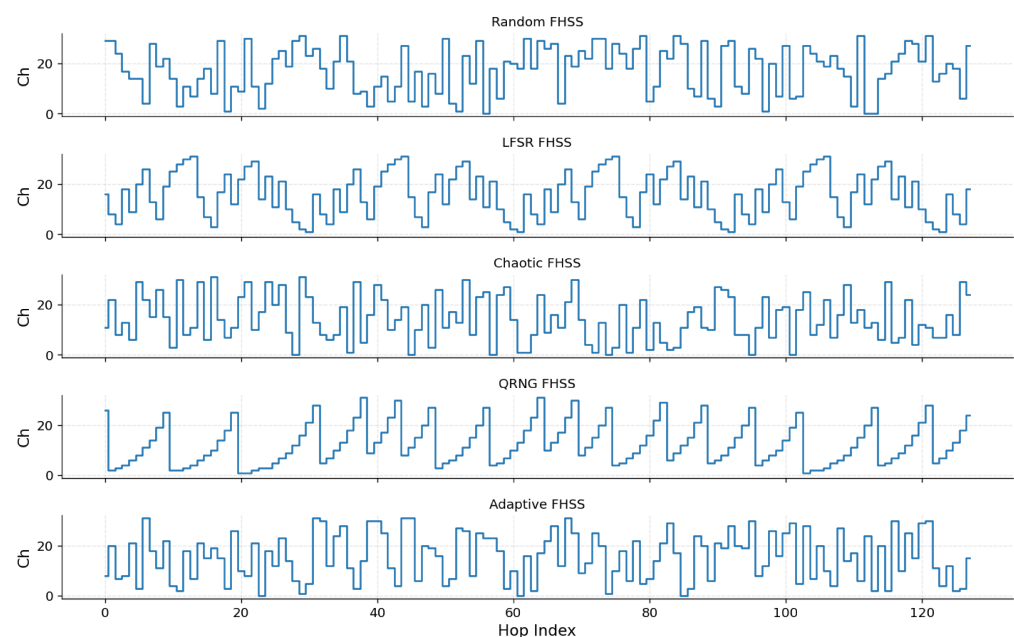


Figure 6. Frequency hopping sequences of implemented FHSS algorithms.

It can be seen that RandomFHSS, QRNGFHSS and AdaptiveFHSS ensure an almost uniform utilization of the available channels. LFSRFHSS generates a deterministic pseudo-random sequence, whilst ChaoticFHSS uses the tent map chaotic mapping.

The NoFHSS algorithm is used as a baseline model for comparison. The entire data packet is transmitted over a single fixed-frequency channel without any hopping. In this case, the formalized description of the NoFHSS algorithm is as follows:

$$h_i = h_0, \forall i, \quad (5)$$

where h_i is the frequency-hopping sequence; h_0 is the constant data transmission channel.

This NoFHSS scheme, described by (5), is most vulnerable to narrowband and reactive jammers. This is because the entire data packet is transmitted on a single fixed frequency.

The implemented RandomFHSS algorithm generates a hopping sequence by independently selecting each subsequent frequency channel at random. In this case, the formal description of the RandomFHSS algorithm is as follows:

$$h_i \sim U(0, N_c - 1), \quad (6)$$

where h_i is the frequency hopping sequence; U is the uniform discrete distribution function; N_c is the number of available frequency channels.

This implementation of the RandomFHSS algorithm, as described in (6), does not use information from previous transmissions and does not accumulate statistics on channel usage.

The LFSRFHSS algorithm, which was implemented in software in this study, is formalized by the following equations:

$$\begin{aligned} x_{k+n} &= \bigoplus_{i \in T} x_{k+i}, \\ h_i &= x_i \bmod N_c \end{aligned} \quad (7)$$

where x_{k+n} is the new register state; T is the feedback taps; $\oplus$ is the XOR logical operation; h_i is the frequency-hopping sequence; N_c is the number of available frequency channels.

This approach, based on (7), ensures a high rate of pseudo-random sequence generation. However, because they are deterministic, they can be partially predicted.

The implemented ChaoticFHSS algorithm uses a chaotic tent map. This article employs an approach based on iterative comparison:

$$x_{k+1} = \begin{cases} 2x_k, & x_k < 0.5 \\ 2(1 - x_k), & x_k \geq 0.5 \end{cases} \quad (8)$$

where x_k is the current value of the system state; x_{k+1} is the next value of the system state; k is the iteration number.

Consequently, the frequency channel number is defined as:

$$h_i = x_i N_c, \quad (9)$$

where h_i is the sequence of frequency hopping; x_i is the sequence of system states; N_c is the number of available frequency channels.

Thus, the chaotic mapping using (8) and (9) ensures sufficient sensitivity to the initial conditions and a complex structure of the hopping sequence.

In implementing the QRNGFHSS algorithm, a quantum random number generator simulation was used to generate independent, equally probable channel numbers. In the software simulator, QRNGFHSS is implemented as a deterministic quantum-like entropy model rather than a physical quantum random number generator. The generated sequence is used solely as a source of hopping indices and does not represent measurements from a physical quantum random source. No dedicated statistical randomness testing was performed on the generated sequence, as the purpose of QRNGFHSS in this study is to provide a software-level reference for evaluating the effect of the hopping sequence on anti-interference performance. This approach made it possible to simulate the statistical properties of quantum random sequences without the need to use external quantum

services. In this case, the formalized description of the channel number determination process is as follows:

$$h_i = q_i \bmod N_c, \quad (10)$$

where h_i is the frequency-hopping sequence; q_i is an independent random number generated by the quantum generator emulator; N_c is the number of available frequency channels.

Unlike the LFSRFHSS implementation, sequence (10) lacks a deterministic structure and is difficult to predict.

The AdaptiveFHSS algorithm implemented in this article utilizes statistical estimation of frequency channel quality. A similar principle of adaptive channel selection, based on accumulated information about the spectrum state, is widely used in modern intelligent anti-jamming methods [31,32]. In contrast to these, the proposed implementation employs a simplified statistical mechanism for channel estimation that does not require offline model training or a separately trained predictive model. This approach has been chosen based on the potential for applying the research results to devices with limited computational resources. In the proposed implementation, upon completion of the transmission of each data packet, estimates (Q_i) are generated that characterize the reliability of each frequency channel. When generating the next hopping sequence, priority is given to channels with higher indicator values (Q_i). The channel-quality estimates are subsequently used in the adaptive channel-selection procedure described by (11)–(14).

After each packet is transmitted, the channel estimates are updated based on the transmission success statistics, thereby implementing a mechanism for continuous statistical learning.

The main AdaptiveFHSS parameters were held constant across all experiments to ensure a consistent comparison between the investigated hopping algorithms. The initial channel-quality value was set to 1, ensuring identical initial conditions across all channels and preventing any channel from being preferred before transmission feedback accumulates. The successful-hop reward was set to 0.15, providing a moderate increase in the estimated quality after a successful transmission. The unsuccessful-hop penalty was set to 0.55, introducing a substantially stronger response to unsuccessful transmissions and enabling rapid suppression of degraded channels. The forgetting factor was set to 0.998, allowing previously accumulated channel information to be retained while remaining responsive to gradual changes in channel conditions. The exploration probability was set to 0.05, maintaining occasional random channel selection while keeping the main selection process driven by the learned channel-quality estimates. The quality-score bounds were defined as $Q_{min} = 0.05$ and $Q_{max} = 10$ to prevent the estimates from becoming numerically negligible or excessively dominant. The environment-based and internal quality estimates were weighted by 0.7 and 0.3, respectively, giving greater importance to accumulated transmission feedback while retaining the contribution of the algorithm's internal channel score. After a channel is selected, its temporary weight is reduced by a factor of 0.85 to limit repeated selection of the same channel within the same hopping sequence and to maintain frequency diversity. The value 0.85 was empirically selected as a moderate penalty that discourages consecutive channel reuse without excessively suppressing channels with persistently high quality.

A hop is considered successful if all bits transmitted during the hop are correctly recovered, i.e., if no bit errors occur within that hop; otherwise, the hop is considered unsuccessful. For channel i , the internal quality score is updated according to:

$$Q_i^{k+1} = \begin{cases} \min(Q_i^k + r, Q_{max}), & \text{if the hop is successful,} \\ \max((1 - p)Q_i^k, Q_{min}), & \text{otherwise;} \end{cases}, \quad (11)$$

where r is the successful-hop reward; p is the unsuccessful-hop penalty; Q_{min} , Q_{max} are the quality-score bounds.

In addition to the internal algorithm score, the selection procedure uses an environment-based channel-quality estimate. For channel i , this estimate is calculated from accumulated successful and unsuccessful transmissions as:

$$E_i = \frac{S_i}{S_i + J_i + \varepsilon}, \quad (12)$$

where S_i is the number of successful observations; J_i is the number of unsuccessful observations; $\varepsilon = 10^{-9}$ is a constant to prevent division by zero.

The final selection score is obtained by combining the environment and internal channel-quality estimates:

$$W_i = 0.7E_i + 0.3Q_i, \quad (13)$$

The selection of numerical values for the weighting coefficients is described above. However, these may be modified in the program code depending on the specific conditions of practical applications.

With probability 0.05, the algorithm performs random exploration by selecting a channel uniformly from the available set. Otherwise, the next channel is sampled probabilistically according to:

$$P_i = \frac{W_i}{\sum_{j=1}^{N_c} W_j}, \quad (14)$$

where P_i is the probability of selecting the i th frequency channel for the next hop; W_i is the current weighted quality score of the i th frequency channel; W_j is the weighted quality score of the j th frequency channel; i is the index of the channel for which the selection probability is being calculated; j is the channel index used in the summation; N_c is the number of available frequency channels.

Therefore, AdaptiveFHSS implements an explicit exploration–exploitation mechanism rather than deterministic selection of the channel with the maximum score. When several channels have identical scores, no deterministic tie-breaking rule is applied. Their equal weights yield equal selection probabilities under the probabilistic sampling procedure, while the 5% exploration mechanism provides an additional opportunity to select any available channel. So, the channel-quality statistics are updated after each transmitted packet, using the hop-level feedback collected during that packet.

The source code that implements the investigated FHSS algorithms, in accordance with the mathematical formulations presented in this subsection, is publicly available [33].

2.4. Implemented Jamming Models

To evaluate the effectiveness of the investigated FHSS algorithms, four intentional jamming models and one baseline noise condition were implemented: AWGN, narrowband, broadband, reactive, and combined jammers. These models differ in how they affect the communication system and in how they reflect the most common scenarios for deliberate radio-electronic jamming of wireless networks [5,15,34–37]. All models are implemented as separate classes within the software model and share a common interface to interact with the environment, ensuring consistent and reproducible test conditions across all frequency-hopping algorithms.

All jamming signals are generated in the complex baseband domain and added to the corresponding QPSK signal according to the selected jamming model. The jamming power is controlled relative to the signal power within the simulated channel, while the bandwidth and affected frequency channels are determined by the corresponding jammer

configuration. Narrowband jamming is restricted to selected frequency channels, whereas broadband jamming affects the complete simulated frequency band. The same channel spacing and frequency-channel configuration are used for all FHSS algorithms to ensure a consistent comparison. All model parameters and implementation-specific settings are provided in the corresponding software implementation available at [33] to ensure the reproducibility of the simulation experiments.

The AWGN model is used as a baseline channel-noise condition rather than as an intentional jammer, providing a reference scenario without frequency-selective interference. The noise variance is determined from the specified SNR, which is used as a channel-model parameter and is kept identical for all compared FHSS algorithms. The software implementation of this model in this case is as follows:

$$\begin{aligned} r_k &= s_k + n_k; \\ n_k &\sim N(0, \sigma^2), \end{aligned} \quad (15)$$

where r_k is the received complex symbol; s_k is the k th complex QPSK symbol; n_k is complex Gaussian noise; N is the normal distribution; 0 is the mean of AWGN; σ^2 is the variance of AWGN, which is determined on the basis of a given SNR in the range from -5 dB to 20 dB in 5 dB steps.

The implemented narrowband jammer model affects only a specific set of frequency channels in accordance with the following formalized description:

$$J(c) = \begin{cases} 1, & c \in C_J \wedge \text{Rand}(0,1) < p_{jam}; \\ 0, & \text{otherwise}, \end{cases} \quad (16)$$

where $J(c)$ is the jammer function; c is the index of the available frequency channel; C_J is the set of channels subject to deliberate jamming; Rand is a random variable with a uniform distribution; p_{jam} is the probability of narrowband jamming being activated.

The implemented broadband jammer model generates interference simultaneously across the entire operating frequency range. In the model, each hop can be suppressed with a given probability in accordance with the following formalized description:

$$P_{jam}(J = (0,1)) = p, \quad (17)$$

where P_{jam} is the probability of interference from the jammer; J is a random variable describing the state of the jammer ($J = 1$ —jammer active, $J = 0$ —jammer inactive); p is the probability of the jammer being activated.

The software implementation of the reactive jammer uses a mechanism to store the history of frequency channel usage. To this end, after each hopping event, the number of the channel used is recorded in the history buffer. To limit the amount of information stored, a parameter specifies the maximum number of recent hopping events to be considered during analysis. In this way, the jammer analyses only the most recent time interval of the FHSS system's operation, enabling it to respond quickly to changes in the hopping sequence. Once the history has been accumulated, the usage frequency for each channel is determined. Based on the frequencies obtained, a set of the most frequently used channels is formed. This set is used by the jammer to suppress the most likely subsequent frequency hops, subject to the implemented reaction delay. Another feature of the implementation is that it accounts for the jammer's reaction time. Therefore, the implemented model represents a delayed reactive jammer: the interference decision is based on previously observed frequency-hopping events rather than on instantaneous access to the current transmission channel. This design reflects a practical scenario in which the jammer first

observes the hopping behavior and then adapts its interference strategy. A generalized formalized description of the software implementation of the reactive jammer model is as follows:

$$\begin{aligned} T &= \text{TopN}(f(c)); \\ f(c) &= \sum_{h_i \in H} 1(h_i = c), \end{aligned} \quad (18)$$

where T is the set of channels to be suppressed; TopN is the set of channels with the highest number of uses; $f(c)$ is the number of times channel c , for which the usage history is being tracked, has been used out of the total set of channels; H is the history of the set of used channels; h_i is the number of the channel used during the i th hopping; $1(h_i = c)$ is an indicator function equal to 1 if the condition that the channel under analysis matches a channel in the usage history is met, and equal to 0 if this condition is not met.

To simulate the most challenging operating conditions, a combined jammer has been implemented, which integrates reactive, narrowband and broadband signal suppression mechanisms. The jamming mechanisms are mutually exclusive within each frequency hop. Only the highest-priority applicable mechanism is activated, and the corresponding interference component is added to the transmitted signal. Unlike the independent use of individual models, the combined jammer employs a hierarchical decision-making mechanism in which different types of interference are assigned varying priorities. When processing each frequency hop, the possibility of reactive jamming is checked first. If the current channel belongs to the set of predicted channels generated by the reactive jammer, AWGN is added to the signal. If reactive jamming is not applied, a check is performed to determine whether the channel belongs to the set of narrowband interference channels. For such channels, jamming is performed with probability p_1 . Only in the absence of the effects of the first two models is wideband jamming performed with probability p_2 . Consequently, the generalized formalized description of the combined jammer is as follows:

$$J_{\text{combined}}(c_i) = \begin{cases} J_{\text{reactive}}(c_i), & c_i \in T, \\ J_{\text{narrowband}}(c_i), & c_i \in C_J \wedge u_1 < p_1, \\ J_{\text{broadband}}(c_i), & u_2 < p_2, \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

where J_{combined} , J_{reactive} , $J_{\text{narrowband}}$, $J_{\text{broadband}}$ are the functions of the combined, reactive, narrowband and broadband jammers, respectively; c_i is the frequency channel number used during the i th hopping; T is the set of channels defined by the reactive jammer model; C_J is the set of narrowband jammer channels; u_1 , u_2 are independent random variables with a uniform distribution; p_1 , p_2 are the probabilities of activating the narrowband and broadband jammers, taken in this study to be 0.7 and 0.45, respectively.

Thus, the final received signal contains the QPSK signal combined with the interference component generated by the selected jamming mechanism; no simultaneous superposition of reactive, narrowband, and broadband jamming components is applied within the same hop.

It is also worth noting that, once the processing of each frequency hop has been completed, the current channel number is added to the reactive jammer's history. This ensures that the jammer adapts to the behavior of the FHSS system and makes it possible to predict the most probable subsequent frequency hops.

A graphical representation of the effect of the specified models on the investigated signal, in accordance with (15)–(19), is shown in Figure 7.

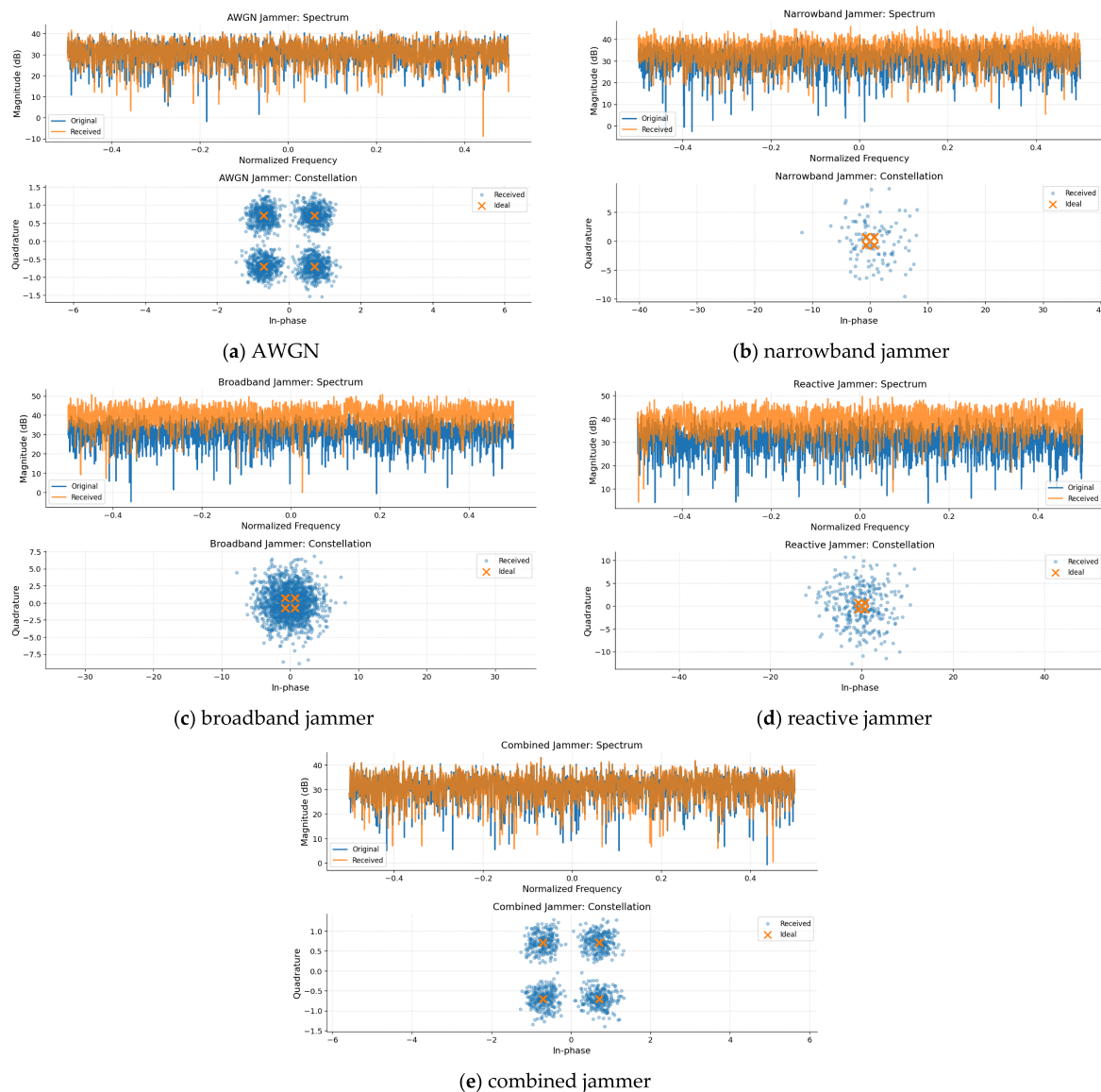


Figure 7. Visualization of the implemented jamming models on the analyzed signal.

2.5. Performance Evaluation Metrics

One of the key quality characteristics of digital transmission is the BER, which quantifies the fraction of bits received in error [38,39]. In a software implementation, the BER is calculated by directly comparing the transmitted and received bit sequences:

$$BER = \frac{1}{N} \sum_{i=1}^N 1(b_i^{tx} \neq b_i^{rx}), \quad (20)$$

where BER is the bit error rate metric; N is the number of bits in the data packet; i is the ordinal number of the bit; b_i^{tx} , b_i^{rx} are the transmitted and received bits, respectively; $1(\cdot)$ is the indicator function.

In this article, the quality of demodulation is assessed using the EVM metric. This metric characterizes the average deviation of the received complex symbols from their ideal positions within the signal constellation [40]. The EVM is calculated after QPSK demodulation and frequency de-hopping, using the transmitted QPSK symbols as the reference constellation. The QPSK symbols are normalized to unit average power before EVM calculation, so that the metric represents the relative constellation error independently

of the absolute signal amplitude. In the software implementation, the EVM is calculated based on the Euclidean distance between the corresponding complex symbols using the following formula:

$$EVM = \sqrt{\frac{1}{N} \sum_{i=1}^N |s_i^{rx} - s_i^{tx}|^2}, \quad (21)$$

where EVM is the error vector magnitude metric; N is the number of complex QPSK symbols; i is the symbol index; s_i^{tx} , s_i^{rx} denote the normalized transmitted and received QPSK symbols after frequency de-hopping, respectively, with s_i^{tx} serving as the reference symbol.

The throughput is used to assess the efficiency of a communication channel. In the proposed software implementation, it is defined as:

$$Throughput = \frac{N_{total} - N_{error}}{N_{total}}, \quad (22)$$

where $Throughput$ is a metric for throughput; N_{total} is the total number of bits transmitted; N_{error} is the total number of bits received in error.

It is worth noting that, in addition to analyzing individual data transmission quality metrics, this study employs a comprehensive methodology for comparing FHSS algorithms. This approach enables a comprehensive assessment of their effectiveness across all the investigated scenarios. Once the sequential packet-level simulation has been completed, the mean values of metrics (20)–(22) are calculated for each algorithm. A summary table of the mean results is compiled based on these statistical characteristics.

To ensure a fair comparison of metrics with different units of measurement and ranges, all criteria are first normalized to the interval from 0 to 1. For metrics where a decrease indicates an improvement in system quality (BER and EVM), normalization is performed using (23). Since Throughput is mathematically derived from BER in the adopted simulation model, it was excluded from the composite score to avoid double-counting of the same error-related performance factor.

$$Metric_i^{norm} = \frac{Metric_{max} - Metric_i}{Metric_{max} - Metric_{min}}, \quad (23)$$

where is the normalized value of the relevant metric for the i th algorithm; $Metric_i$ is the current value of the relevant metric for the i th algorithm; $Metric_{max}$, $Metric_{min}$ are the maximum and minimum values of the relevant metric across all algorithms.

Once normalization has been performed, an integral performance index (IS) is calculated for each analyzed algorithm using the following formula:

$$IS_i = a_1 BER_i^{norm} + a_2 EVM_i^{norm}, \quad (24)$$

where IS_i is the integral quality assessment of the i th algorithm; BER_i^{norm} , EVM_i^{norm} are the normalized metrics for the i th algorithm; a_1 , a_2 are the weighting coefficients for the metrics.

As can be seen from the analysis of (24), the software implementation allows for the adjustment of the weighting of metrics on the overall quality of the assessment, depending on the specific objectives of the analysis. In the current implementation, equal weights of 0.5 were assigned to BER and EVM to provide a balanced assessment of bit-level reliability and signal quality.

To additionally characterize the computational requirements of the implemented FHSS algorithms, an execution-time and memory-consumption benchmark was performed under identical simulation conditions. The benchmark was conducted using the same software environment and system configuration as the main sequential packet-level simulations. For

each algorithm, 1000 packets were processed, with 32 available frequency channels, 4096 information bits per packet, and 128 hopping operations per packet. The execution time was measured both per packet and per hopping operation, while peak Python memory allocation was recorded using the tracemalloc module.

3. Results

3.1. Performance Analysis of FHSS Algorithms

This subsection presents the results of a comprehensive comparative analysis of implemented FHSS algorithms under identical computer simulation conditions. The aim of the study was to evaluate the performance of FHSS algorithms using a single set of quality metrics (BER, Throughput and EVM). The integrated dashboards in Figure 8 summarize the results of the sequential packet-level simulations for each FHSS algorithm as a function of SNR. For each SNR value, the plotted BER, Throughput, and EVM values are obtained by averaging the corresponding results over the sequential packet-level observations and the considered jamming scenarios. This approach provides a comprehensive view of interference resilience, communication channel utilization efficiency and received signal quality. This enhances the objectivity of the comparison between the investigated algorithms and enables identification of their strengths and weaknesses before a detailed analysis of individual jamming models is conducted. A graphical interpretation of the performance analysis results for FHSS algorithms is shown in Figure 8.

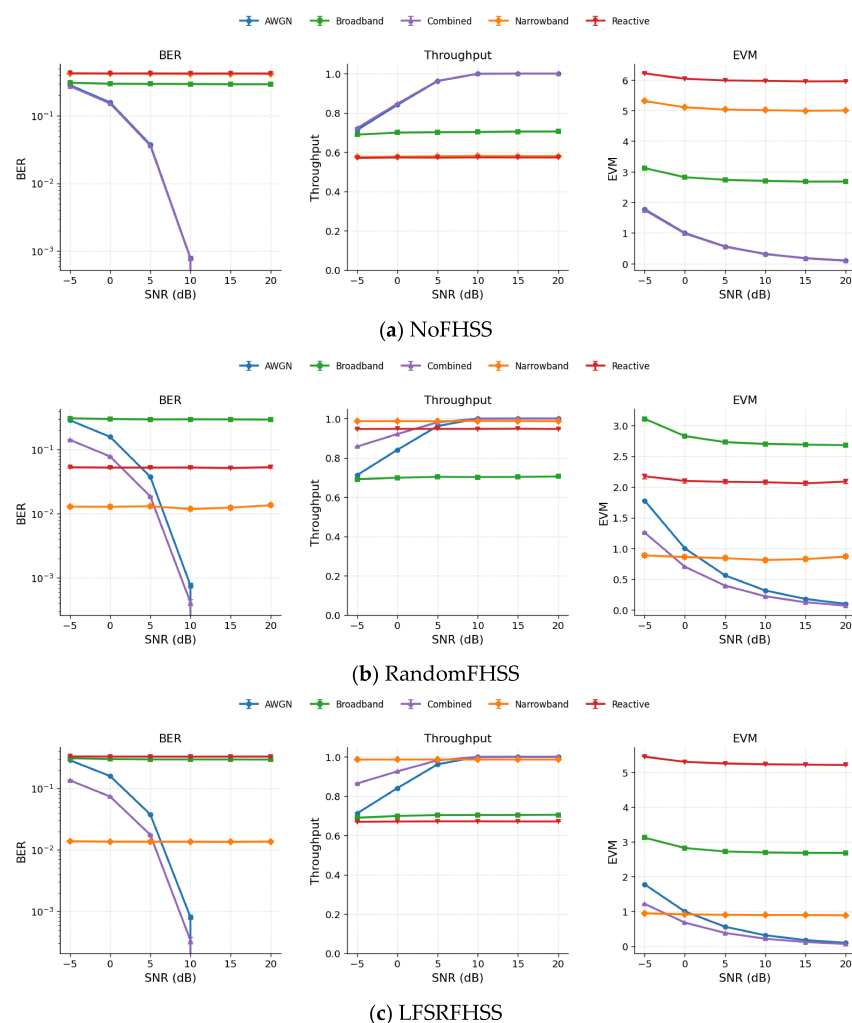


Figure 8. Cont.

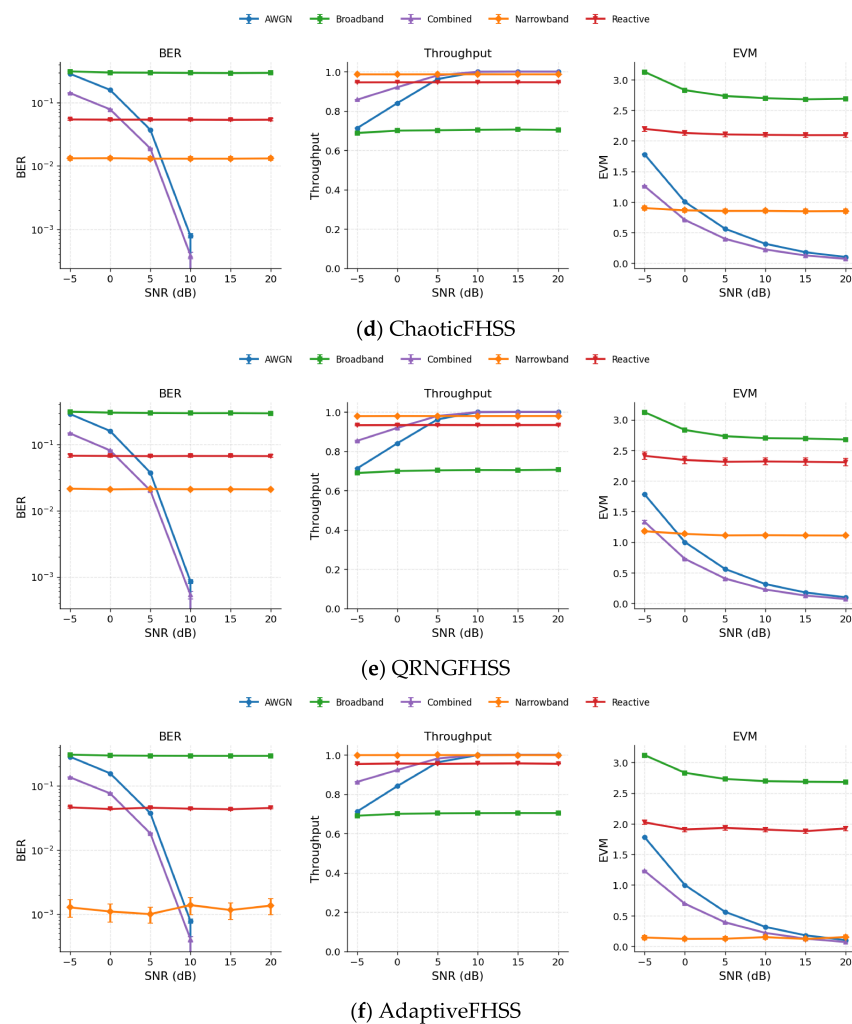


Figure 8. Visualization of the results of the performance analysis of FHSS algorithms.

A comparison of the results shown in Figure 8 reveals that RandomFHSS, QRNGFHSS, and ChaoticFHSS exhibit similar behavior across all investigated metrics. For these algorithms, with integrated consideration of jamming models, as the SNR increases from -5 dB to 10 dB, the BER decreases from approximately 0.2 to 0.001 , the Throughput increases from 0.70 to almost 1.0 , whilst the EVM decreases from 3.1 to 0.3 . The almost complete overlap of the corresponding curves indicates the following. Under the modeling conditions used, the type of hopping sequence generator is not a decisive factor in improving the system's resilience to interference. The results obtained suggest that the main potential for improving FHSS efficiency lies in the use of adaptive control mechanisms.

The LFSRFHSS algorithm also demonstrates high transmission-quality metrics across most of the scenarios studied. However, it is characterized by the highest EVM values, excluding NoFHSS. These values reach 5.5 , which may be due to the deterministic nature of the frequency hopping pattern.

In contrast, the AdaptiveFHSS implemented in this study delivers the best results across virtually all the metrics examined. Even at an SNR of 10 dB, the BER falls to approximately 0.0001 , the Throughput reaches a value close to 1.0 , and the EVM varies from 0.2 to 2.6 across different types of jammers, demonstrating the high efficiency of the adaptive frequency channel selection mechanism.

As expected, the NoFHSS model, which does not use frequency hopping, demonstrates the lowest efficiency. It is characterized by the highest BER and EVM, with maximum EVM exceeding 6 , which is more than an order of magnitude higher than that of AdaptiveFHSS.

This confirms that frequency hopping is a prerequisite for the interference-resistant operation of modern wireless communication systems.

Thus, when all investigated metrics are taken together, the AdaptiveFHSS algorithm demonstrates the best overall performance. It is worth noting that the RandomFHSS, QRNGFHSS and QRNGFHSS algorithms form a group of algorithms with similar characteristics. The LFSRFHSS algorithm occupies an intermediate position, whilst NoFHSS provides the lowest transmission quality.

The research findings have confirmed the need for a more detailed analysis of how the specific characteristics of individual jamming models affect the performance of each algorithm. These findings are presented in the following subsection.

3.2. Performance Analysis Under Different Jamming Environments

To assess the resilience of the investigated FHSS algorithms to deliberate electronic jamming, simulations were conducted across five different jamming scenarios. Figure 9 shows the dependence of the analyzed quality metrics on the SNR value for each jamming scenario.

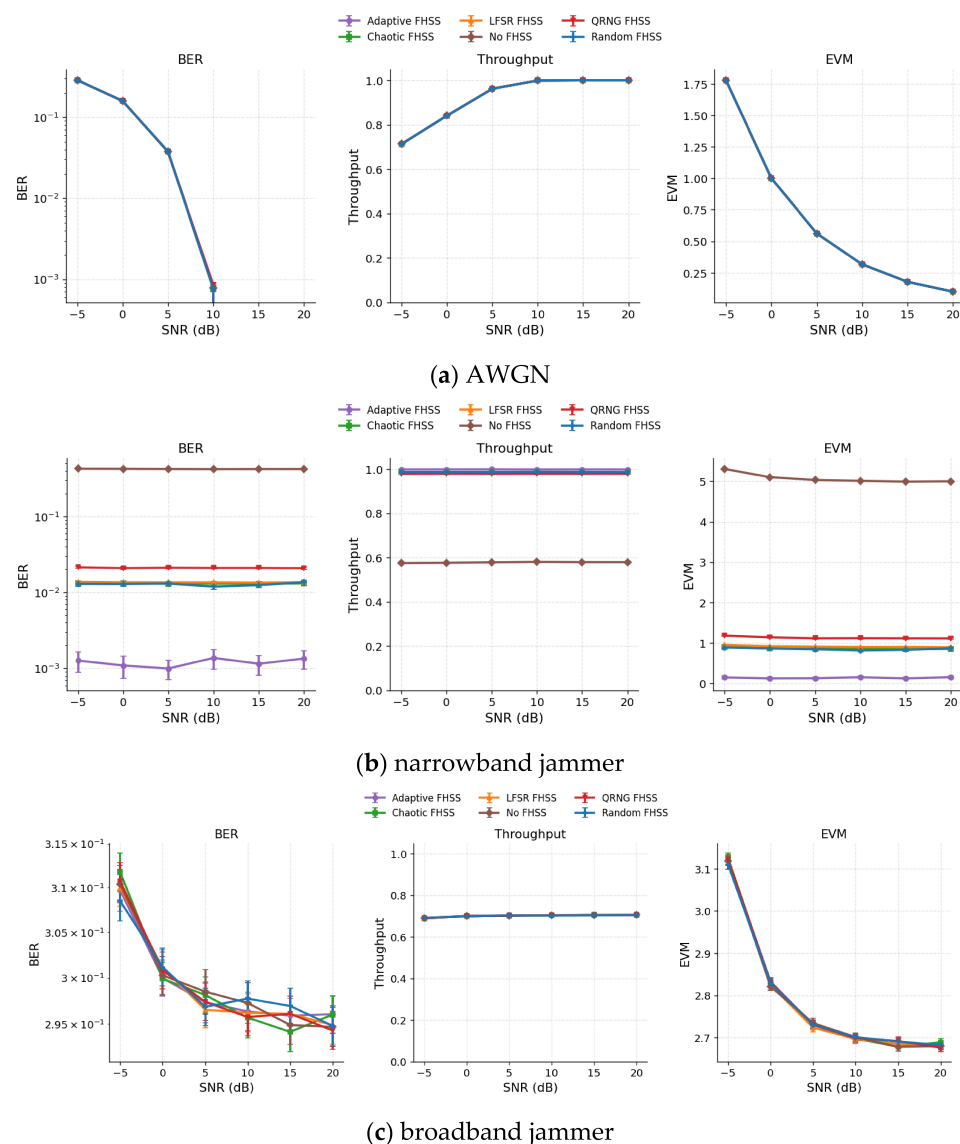


Figure 9. Cont.

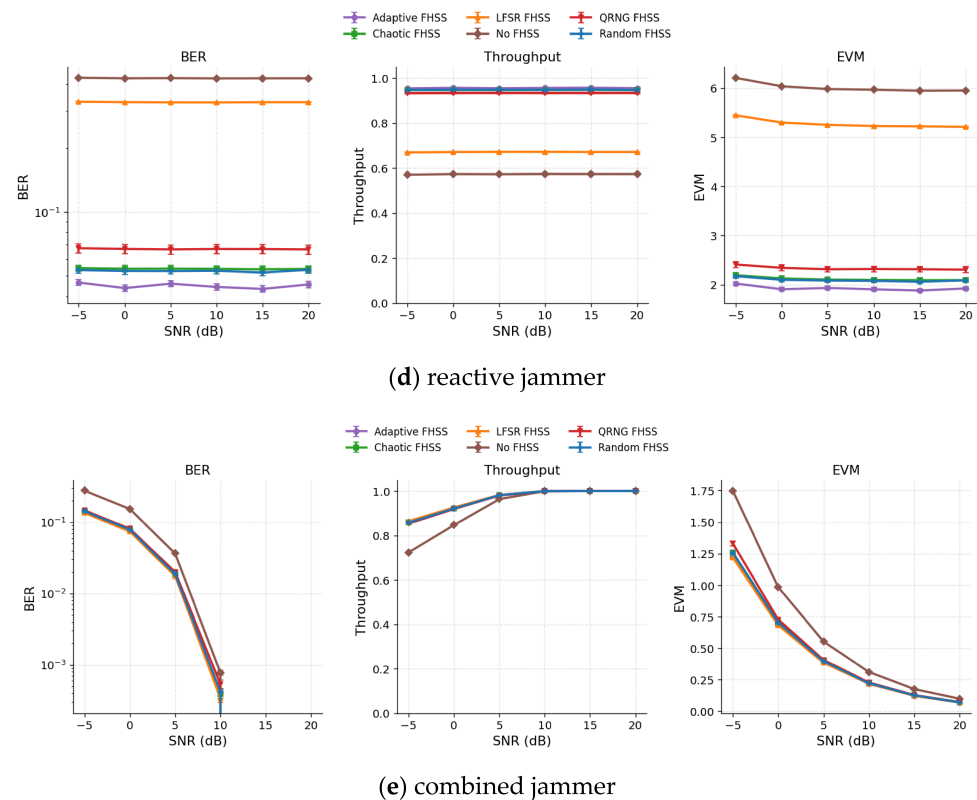


Figure 9. Performance comparison of FHSS algorithms under different jamming environments.

An analysis of the results shown in Figure 9 demonstrates that the effectiveness of FHSS algorithms depends significantly on the type of jammer. For the AWGN and combined jammer models, all algorithms demonstrate the expected improvement as the SNR increases. The BER metric decreases to around 0.001, the Throughput approaches 1 and the EVM is reduced by approximately 11 times.

The most significant negative impact is observed for the reactive jammer model. In this scenario, the BER is almost independent of the SNR, the Throughput remains virtually constant, and the EVM exceeds 6 even at high SNR values. The NoFHSS algorithm is particularly vulnerable, whilst AdaptiveFHSS delivers the lowest BER and EVM values of all the algorithms studied.

The narrowband jammer model proved to be the least critical. Most algorithms maintain the Throughput close to 1, whilst the BER does not exceed a few percentage points. AdaptiveFHSS once again demonstrates the best interference resilience, effectively avoiding channels with high levels of interference.

At the same time, the RandomFHSS, QRNGFHSS and ChaoticFHSS algorithms exhibit very similar performance across all scenarios. This suggests that, with the model used, the method for generating the hopping sequence offers no significant advantages in the absence of adaptive mechanisms. This effect, in turn, is of interest for further research. The main advantage of AdaptiveFHSS is its use of statistical learning of channel quality, which enables it to deliver the best results across virtually all the metrics considered.

3.3. Comparative Performance Analysis Under Different SNR Levels

To provide a more detailed comparison of the algorithms, taking into account the corresponding jamming models at different SNR levels, Figure 10 shows bar charts of the metrics under analysis.

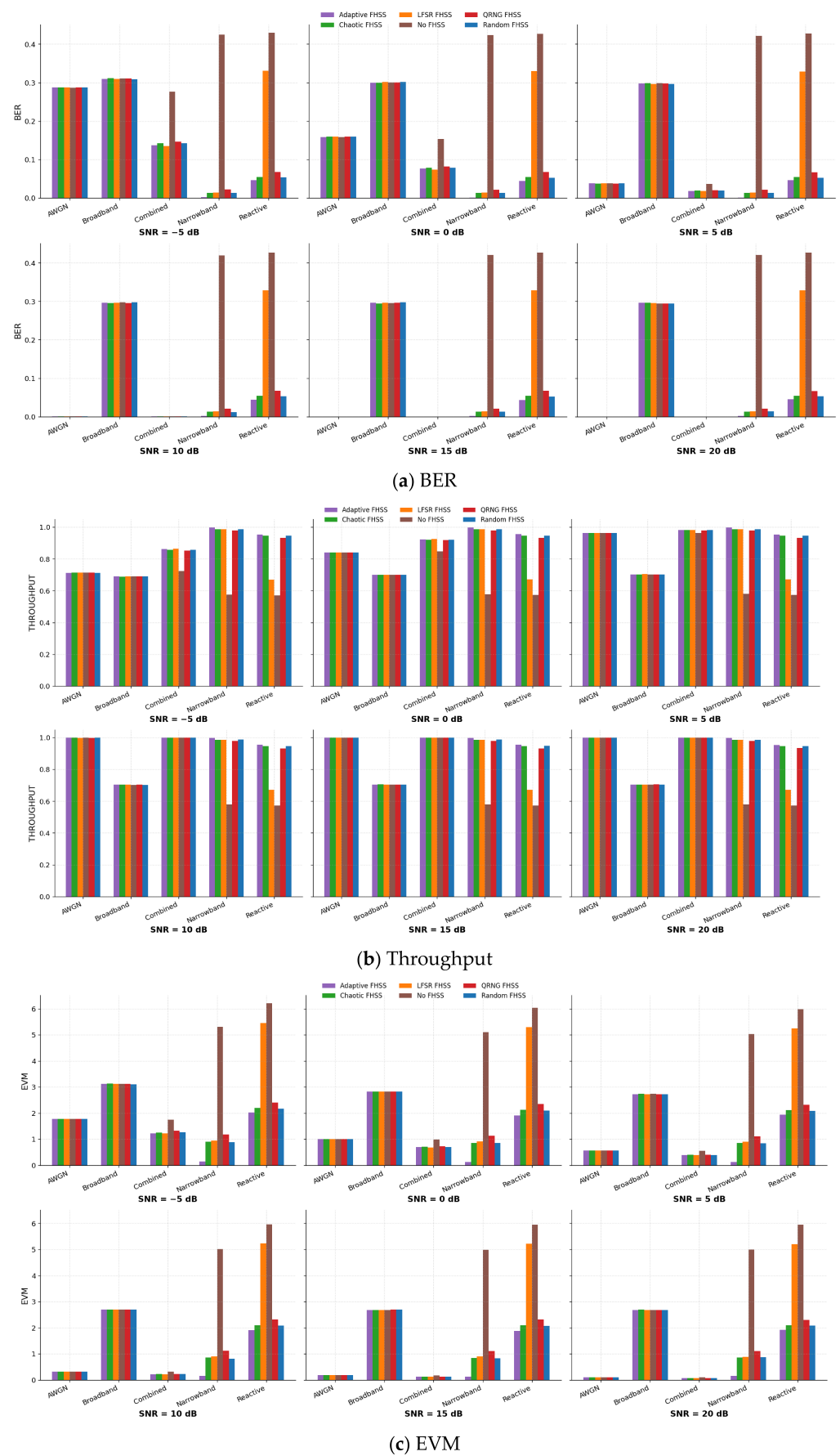


Figure 10. Results of a comparison of FHSS algorithms based on the metrics analyzed for different jamming models at fixed SNR values.

As shown in Figure 10, AdaptiveFHSS consistently achieves the lowest BER values for narrowband and reactive jammers, whilst NoFHSS exhibits the highest error rates across virtually all SNR levels. For AWGN, broadband and combined jammers, the differences between the algorithms are negligible, which confirms the results of the previous analysis.

An analysis of the Throughput shows that, in the absence of active interference (AWGN model) and under combined jammer conditions, all algorithms achieve a near-maximum value of this metric. The biggest differences are observed for narrowband and reactive jammers, where AdaptiveFHSS maintains the highest Throughput values, whilst NoFHSS exhibits the greatest degradation.

A comparison based on the EVM confirms similar patterns. The lowest EVM values are observed for AdaptiveFHSS, particularly under narrowband and reactive jammer conditions, whereas NoFHSS has the highest EVM values, indicating the highest level of distortion in the received signal. For a broadband jammer, all algorithms exhibit similar EVM values, indicating that the effect of broadband interference is practically identical regardless of the method used to generate the hopping sequence.

Overall, the results obtained confirm the outcomes of the preliminary analysis. The advantages of AdaptiveFHSS are particularly evident in scenarios involving targeted jammers (narrowband and reactive), whilst in the case of AWGN, broadband and combined jammers, the performance of most algorithms is similar. This demonstrates the effectiveness of adaptive frequency channel selection, particularly in the face of adaptive and narrowband attacks.

3.4. Computational Complexity and Resource Assessment

To complement the communication-performance analysis, the computational requirements of the implemented FHSS algorithms were additionally evaluated under identical simulation conditions. The obtained results are summarized in Table 2.

Table 2. Computational and memory requirements of the implemented FHSS algorithms.

Algorithm	Time/Packet, ms	Time/Hop, μ s	Peak Memory, MB
NoFHSS	1.9736	15.4187	0.0062
RandomFHSS	1.0629	8.3037	0.0062
LFSRFHSS	0.9173	7.1666	0.0068
ChaoticFHSS	5.0912	39.7751	0.0075
QRNGFHSS	1.1561	9.0323	0.0066
AdaptiveFHSS	27.7191	216.5557	0.0099

The results demonstrate that the proposed AdaptiveFHSS introduces a higher computational overhead than the non-adaptive algorithms, requiring 27.72 ms per packet and 216.56 μ s per hopping operation. For comparison, execution times range from 0.92 ms for LFSRFHSS to 5.09 ms for ChaoticFHSS. At the same time, the memory requirements remain comparable across all implementations, with peak allocation ranging from 0.0062 MB for NoFHSS and RandomFHSS to 0.0099 MB for AdaptiveFHSS. Thus, the main computational cost of AdaptiveFHSS is associated with execution time, whereas its memory requirements remain low.

To complement the asymptotic analysis, numerical scaling estimates were calculated for representative values of the number of channels (C) and hopping operations (H). The estimated computational load (HC) ranges from 128 to 32,768 relative operation units, while the corresponding space requirement ($C+H$) ranges from 24 to 384 units. These estimates confirm the $O(HC)$ time and $O(C+H)$ space complexity of AdaptiveFHSS. The corresponding code for the calculation is provided in [33].

3.5. Comprehensive Performance Assessment

To summarize the results, a comprehensive comparison of the investigated FHSS algorithms was conducted across all experiments. The comparison was based on the average normalized values of metrics (20)–(22). Subsequently, an integral performance index (24) was calculated for each algorithm, taking the two selected criteria, BER and EVM, into account simultaneously. The results of this analysis are presented in Table 3.

Table 3. Average algorithm performance.

Rank	Algorithm	BER	Throughput	EVM	IS
1	AdaptiveFHSS	1.000	1.000	1.000	1.000
2	RandomFHSS	0.975	0.975	0.903	0.939
3	ChaoticFHSS	0.974	0.974	0.899	0.936
4	QRNGFHSS	0.947	0.947	0.845	0.896
5	LFSRFHSS	0.649	0.649	0.553	0.601
6	NoFHSS	0.000	0.000	0.000	0.000

As shown in Table 3, the implemented AdaptiveFHSS algorithm delivered the best results among all the algorithms studied, based on a combination of criteria including transmission reliability, channel utilization efficiency, and received signal quality.

The RandomFHSS and ChaoticFHSS algorithms demonstrated fairly high performance, with an integral performance index (IS) of 0.939 and 0.936, respectively. Both algorithms demonstrated very similar normalized values for BER and Throughput (0.975 and 0.974, respectively), but were outperformed by AdaptiveFHSS in terms of EVM. The QRNGFHSS algorithm ranked fourth with an IS of 0.896, indicating high efficiency but lower performance compared with AdaptiveFHSS. The similar performance of RandomFHSS, QRNGFHSS, and ChaoticFHSS can be attributed to the common statistical property of their hopping sequences, which distribute channel selections over the available frequency set without using channel-quality feedback. Thus, although the sequence generation mechanisms differ, their comparable channel-selection behavior yields only a limited difference in anti-interference performance under the conditions considered. Consequently, investigating the potential integration of different frequency-hopping sequence generation methods with adaptive strategies is a promising area for further research.

The LFSRFHSS received a significantly lower integral performance index (IS = 0.601). Despite satisfactory results in individual scenarios, its normalized values for BER, Throughput, and EVM (0.649, 0.649, and 0.553, respectively) were significantly lower than those of the best-performing group of algorithms. The performance gap between LFSRFHSS and ChaoticFHSS, QRNGFHSS, and RandomFHSS is more pronounced. This result is consistent with the deterministic structure of LFSR-generated sequences, which may lead to less diversity in channel selection under the interference conditions considered. However, this interpretation should be regarded as a possible explanation rather than a definitive causal relationship and requires dedicated sequence-level analysis in future experiments. As expected, the NoFHSS system achieved the lowest results, reflecting its lack of interference resilience in the absence of a frequency-hopping mechanism.

The practical effectiveness of the implemented AdaptiveFHSS is further confirmed by the results of comparisons with other algorithms, as shown in Table 4. It is worth noting that, in order to ensure greater methodological objectivity, the average absolute values of the BER, Throughput and EVM metrics were used when calculating the relative improvement, rather than their normalized values. The improvement values in Table 4 are calculated from the original, non-normalized performance metrics and therefore should

not be directly compared with the normalized values reported in Table 3. For BER and EVM, the improvement is calculated as the relative reduction, whereas for Throughput, it is calculated as the relative increase with respect to the corresponding baseline algorithm.

Table 4. Relative performance improvement of AdaptiveFHSS over baseline algorithms.

Compared with	BER Improvement, %	Throughput Improvement, %	EVM Improvement, %
NoFHSS	64.43	22.80	60.71
LFSRFHSS	38.89	6.98	40.84
QRNGFHSS	8.71	0.99	19.30
ChaoticFHSS	4.56	0.49	13.47
RandomFHSS	4.30	0.46	13.02

Compared with the baseline NoFHSS algorithm, the BER was reduced by 64.43%, the Throughput was increased by 22.80% and the EVM was reduced by 60.71%. Compared with LFSRFHSS, the implemented AdaptiveFHSS algorithm delivered a 38.89% improvement in BER, a 6.98% increase in Throughput and a 40.84% reduction in EVM. Even when compared with the best competing algorithms, AdaptiveFHSS consistently demonstrates an advantage. In particular, compared with QRNGFHSS, the improvements are 8.71% in BER, 0.99% in Throughput and 19.30% in EVM; compared with ChaoticFHSS—4.56%, 0.49% and 13.47%, respectively; compared with RandomFHSS—4.30%, 0.46% and 13.02%, respectively.

To further assess the stability of AdaptiveFHSS during online adaptation, the cumulative average values of BER, Throughput, and EVM were analyzed as a function of the number of transmitted packets. The experiment was conducted under reactive jamming at SNR = 10 dB, as this represents a challenging dynamic interference scenario with moderate signal quality, making it suitable for evaluating the stabilization of the adaptive mechanism. The results are shown in Figure 11.

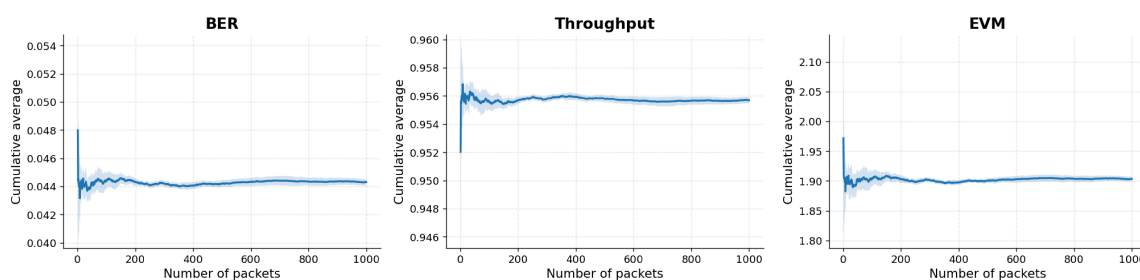


Figure 11. Convergence of cumulative performance metrics of AdaptiveFHSS under reactive jamming at SNR = 10 dB.

The results indicate a short initial transient period, after which all three metrics stabilize around quasi-stationary levels. After approximately 100–200 packets, the cumulative values remain close to $\text{BER} \approx 0.044\text{--}0.045$, $\text{Throughput} \approx 0.956$, and $\text{EVM} \approx 1.9$, with only minor subsequent variations. This behavior indicates that AdaptiveFHSS rapidly establishes stable channel-quality estimates and maintains consistent communication performance during further operation. The results demonstrate that using a persistent learning mechanism, which accumulates statistics on the quality of frequency channels and accounts for them when generating subsequent hopping sequences, ensures sustained improvements in the efficiency of wireless communication systems. Unlike FHSS algorithms, which use exclusively random or deterministic frequency selection, AdaptiveFHSS combines the

properties of random hopping with adaptive channel state estimation, enabling the best results to be achieved across all the metrics studied.

4. Discussion

4.1. Scientific Novelty and Practical Significance

The research results obtained have demonstrated that the software-implemented AdaptiveFHSS algorithm consistently delivers better performance in terms of BER, Throughput and EVM metrics compared with other FHSS implementations across a wide range of jammer models. Unlike most recent studies, which focus on improving the hopping sequence generator or on using machine learning methods to predict the spectral environment, this article employs an approach based on adaptive statistical learning of frequency channel quality, without the need for prior model training. The proposed approach avoids the need for a preliminary model-training stage and therefore provides a simpler learning architecture than training-based machine learning and/or deep learning approaches. However, the computational benchmark shows that the adaptive channel-selection mechanism introduces additional execution-time overhead in the current software implementation. This trade-off should be considered when deploying the algorithm on resource-constrained devices.

Some recent studies have also considered adaptive and learning-based anti-jamming approaches, including adaptive frequency selection and reinforcement-learning-based spectrum management [20,21]. However, direct quantitative comparison with these methods is beyond the scope of the present study, as their implementations and experimental configurations cannot be fully reproduced from the available information. Therefore, the present work focuses on a unified comparison of the implemented FHSS algorithms under identical simulation conditions.

The obtained results should be considered in the context of recent learning-based anti-jamming approaches. The authors of [24] formulated adaptive frequency hopping as a reinforcement learning problem and optimized the hopping interval based on interactions with the jamming environment. The authors of [25] employed a deep learning architecture with joint frequency and power decisions, while the authors of [20] used deep learning to derive hopping strategies in a dynamic wideband jamming environment. These approaches provide substantially richer learning capabilities than the statistical mechanism implemented here, but they also require model training and additional inference procedures. The proposed AdaptiveFHSS occupies a different point in this design space. It sacrifices the ability to learn complex long-term interference patterns in favor of a simpler online mechanism that directly accumulates channel-quality information from transmission feedback. This makes the proposed method potentially more suitable as a lightweight adaptive baseline for systems where offline training or neural network inference is undesirable.

Studies on the use of chaotic frequency hopping have shown that chaotic generators enhance the cryptographic strength and unpredictability of hopping sequences. However, they have virtually no effect on interference resilience in the presence of active jammers. Similar conclusions were reached in this study, in which ChaoticFHSS exhibited performance characteristics comparable to those of RandomFHSS and QRNGFHSS. This outcome indicates that increasing the generator's entropy does not, in itself, guarantee an improvement in link quality under active electronic jamming.

The practical value of the proposed approach also lies in the fact that the entire algorithm has been implemented as a software simulator, which allows the modeling of various FHSS algorithms and several jamming scenarios, as well as statistical evaluation based on sequential packet-level simulation observations. Unlike most published works, which limit their analysis to individual scenarios or specific types of interference, the

software developed provides a unified environment for the comprehensive comparison of FHSS algorithms.

Overall, the comparison with the existing literature indicates that the proposed approach is not intended to replace reinforcement learning or deep-learning anti-jamming methods in scenarios that require predicting complex or long-term interference patterns. Its main distinction is the use of online statistical channel-quality estimation without an offline training stage, combined with probabilistic exploration–exploitation and persistent transmission feedback. The experimental contribution is complementary: rather than optimizing a single adaptive strategy for a specific jammer, the proposed framework evaluates both adaptive and non-adaptive FHSS mechanisms under identical communication, SNR, and interference conditions. The resulting comparison shows that the main performance gain of AdaptiveFHSS occurs under targeted interference, whereas its advantage diminishes under broadband interference.

4.2. Study Limitations

In the research carried out for this article, several assumptions were made, enabling a comparative analysis of FHSS algorithms under identical simulation conditions.

In particular, the simulations were carried out for a communication channel with AWGN and for realized jammer models. This approach allows for a fair comparison of algorithms under identical conditions, although it does not account for certain characteristics of a real radio channel, such as multipath propagation, fading or Doppler effects.

No hardware implementation of the proposed AdaptiveFHSS algorithm was performed in this study. Therefore, the reported execution time and memory consumption results characterize the Python-based software simulator and should not be interpreted as direct hardware resource or real-time performance requirements. Practical deployment on embedded or software-defined radio platforms remains a subject of future experimental validation.

The implemented AdaptiveFHSS model utilizes statistical accumulation of information on the quality of individual frequency channels, which enables adaptation without complex pre-training procedures. At the same time, this article did not consider more complex approaches to predicting the spectral situation, which take into account temporal or spatial patterns of spectrum usage.

The comparison with learning-based methods also highlights an important limitation of the present approach. AdaptiveFHSS does not explicitly model long-term temporal dependencies in spectrum occupancy and therefore cannot be expected to exploit complex patterns as effectively as recurrent or deep reinforcement learning architectures. Its advantage lies instead in avoiding the training stage and maintaining a comparatively simple online adaptation mechanism. Consequently, the proposed method should be regarded as a lightweight adaptive alternative rather than as a direct replacement for more sophisticated learning-based anti-jamming algorithms.

To ensure the reproducibility of the experiments, all interference generator models used a fixed set of parameters. The selected values made it possible to generate representative simulation scenarios and carry out a valid comparison of the algorithms. Investigating the impact of individual parameters on the system's performance may, however, be the subject of future research.

The algorithms were evaluated using sequential packet-level simulations, which enabled the statistical analysis of a large number of transmission scenarios. The practical implementation of the proposed algorithm on physical computing platforms or other radio equipment is regarded as the logical next stage of the research.

4.3. Future Research Directions

The results obtained open up several promising directions for further research. The first area is the extension of the communication channel model to take into account multi-path propagation and the spatial and temporal characteristics of the wireless medium. This will enable an assessment of the effectiveness of the implemented AdaptiveFHSS algorithm under conditions as close as possible to those of real-world communication networks.

Another promising area is the integration of more sophisticated artificial intelligence methods, in particular reinforcement learning or transformer models, for predicting the state of the spectrum and generating hopping sequences that take into account long-term patterns in the use of frequency channels.

Further development is also required to model more complex jamming systems that use machine learning algorithms for adaptive prediction of subsequent frequency hopping. This will enable the creation of more realistic scenarios for the confrontation between FHSS systems and intelligent electronic warfare systems.

Of particular interest is the investigation into the reasons for the virtually identical performance of RandomFHSS, QRNGFHSS and ChaoticFHSS, as revealed during the modeling. The results indicate that, for the noise models used, it is not the method of generating the random hopping sequence that decisively influences noise immunity, but rather the presence of a mechanism to adapt it to the current state of the spectrum. Further research could focus on determining the conditions under which quantum or chaotic random number generators would provide statistically significant advantages. Future work will additionally investigate hopping-sequence characteristics, including channel-selection diversity, repetition patterns, and temporal correlations, to establish their quantitative relationship with anti-interference performance. A dedicated analysis of deterministic sequence properties will also be considered to clarify the observed performance degradation of LFSRFHSS under specific jamming conditions.

Finally, an important step in further research is the practical implementation of AdaptiveFHSS on resource-constrained computing platforms and the experimental verification of the results in a real-world wireless environment. This will enable an assessment of the proposed algorithm's performance in the presence of hardware constraints, time delays and real-world radio frequency interference.

5. Conclusions

A software simulator for FHSS systems has been developed and studied in this article. It implements six frequency-hopping algorithms, five interference source models, and statistical evaluation based on sequential packet-level simulation observations and the BER, Throughput and EVM metrics. The proposed simulator provides a unified environment for comprehensive research on the interference resilience of FHSS algorithms.

In the research, the AdaptiveFHSS algorithm was implemented in software. This algorithm utilizes the statistical accumulation of information on the quality of frequency channels during the generation of hopping sequences. Unlike the implemented RandomFHSS, LFSRFHSS, ChaoticFHSS and QRNGFHSS algorithms, the AdaptiveFHSS-based approach enables adaptation to the characteristics of the radio channel without the use of pre-training or complex machine learning models.

According to comprehensive modeling results, AdaptiveFHSS demonstrated the best overall performance among the algorithms studied, achieving a maximum integral performance index of 1.0, confirming its consistent superiority across the criteria examined.

Compared with the baseline implementation, the proposed AdaptiveFHSS achieved a 64.43% reduction in BER, 22.80% improvement in Throughput and 60.71% reduction in EVM relative to a system without frequency hopping.

Computational benchmarking demonstrated that AdaptiveFHSS provides its superior communication performance at the cost of increased execution time. The average processing time was 27.72 ms per packet, compared with 0.92–5.09 ms per packet for the non-adaptive algorithms. At the same time, the peak memory consumption of AdaptiveFHSS was only 0.0099 MB, remaining comparable to the other implementations, which ranged from 0.0062 MB to 0.0075 MB.

The results also showed that RandomFHSS, QRNGFHSS and ChaoticFHSS exhibit very similar characteristics, whilst the main improvement in noise immunity is achieved precisely through the use of an adaptive frequency channel selection mechanism. This confirms the potential of using statistical channel quality learning to design modern adaptive FHSS systems in the face of active electronic countermeasures.

Funding: The article was prepared within the framework of the project 2025.06/0047 ‘Information technologies of cryptographic protection and data authentication for mobile and satellite communication systems’. This project received funding from the National Research Foundation of Ukraine.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and source code presented in this study are available in Zenodo at <https://doi.org/10.5281/zenodo.22011999>.

Acknowledgments: During the preparation of this work, the author used GPT 5.3 (Generative AI-assisted language model) and Grammarly 1.2.287.1943 to improve language clarity, grammar, and overall readability of the article. No generative AI tools were used for data analysis, result fabrication, or interpretation of findings. The author has reviewed and edited the output and takes full responsibility for the content of this publication.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AdaptiveFHSS	Adaptive frequency hopping spread spectrum with statistical channel learning
AWGN	Additive white Gaussian noise
BER	Bit error rate
ChaoticFHSS	Chaotic frequency-hopping spread spectrum based on the tent map
EVM	Error vector magnitude
FHSS	Frequency-hopping spread spectrum
LFSRFHSS	Linear feedback shift register-based frequency-hopping spread spectrum
NoFHSS	Conventional fixed-frequency transmission
QPSK	Quadrature phase-shift keying
QRNGFHSS	Quantum random number generator-based frequency hopping spread spectrum
QRNGs	Quantum random number generators
RandomFHSS	Random frequency-hopping spread spectrum
SNR	Signal-to-noise ratio
UAV	Unmanned aerial vehicle

References

1. Laktionov, I.; Diachenko, G.; Moroz, D.; Getman, I. A Comprehensive Review of Cybersecurity Threats to Wireless Infocommunications in the Quantum-Age Cryptography. *Internet Things* **2025**, *6*, 61. [CrossRef]
2. Yu, C.; Chen, S.; Wang, F.; Wei, Z. Improving 4G/5G air interface security: A survey of existing attacks on different LTE layers. *Comput. Netw.* **2021**, *201*, 108532. [CrossRef]
3. Harvanek, M.; Bolcek, J.; Kufa, J.; Polak, L.; Simka, M.; Marsalek, R. Survey on 5G Physical Layer Security Threats and Countermeasures. *Sensors* **2024**, *24*, 5523. [CrossRef] [PubMed]

4. Pirayesh, H.; Zeng, H. Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. *arXiv* **2021**, arXiv:2101.00292. [[CrossRef](#)]
5. Hnatushenko, V.V.; Laktionov, I.S.; Udoviyk, I.M. Software-based evaluation of pseudorandom frequency-hopping for wireless infocommunication cybersecurity. *Nauk. Visnyk Natsionalnoho Hirnychoho Universytetu* **2026**, *2*, 141–159. [[CrossRef](#)]
6. Pourranjbar, A.; Kaddoum, G.; Ferdowsi, A.; Saad, W. Reinforcement Learning for Deceiving Reactive Jammers in Wireless Networks. *IEEE Trans. Commun.* **2021**, *69*, 3682–3697. [[CrossRef](#)]
7. Huang, T.; Liu, Y.; Liu, X.; Wang, M. A New Improved Multi-Sequence Frequency-Hopping Communication Anti-Jamming System. *Electronics* **2025**, *14*, 523. [[CrossRef](#)]
8. Tao, J.; Shi, Y.; Li, Y.; Wan, L.; Zhang, Y. Multi-slot optimized index modulation based on FHSS for efficient anti-jamming. *Phys. Commun.* **2025**, *71*, 102657. [[CrossRef](#)]
9. Yu, Z.; Hao, Z.; Yao, W.; Jia, M. A Capacity Enhancement Method for Frequency-Hopping Anti-Jamming Communication Systems. *Electronics* **2023**, *12*, 4457. [[CrossRef](#)]
10. Rao, N.; Xu, H.; Qi, Z.; Wang, D.; Peng, X.; Jiang, L. Adaptive Jamming Decision-Making Against FHSS Communications via Inexpert Demonstrations Assisted Meta Reinforcement Learning. *IEEE Commun. Lett.* **2025**, *29*, 105–109. [[CrossRef](#)]
11. Yang, H.; Xiong, Z.; Zhao, J.; Niyato, D.; Wu, Q.; Tornatore, M.; Secci, S. Intelligent Reflecting Surface Assisted Anti-Jamming Communications Based on Reinforcement Learning. *arXiv* **2020**, arXiv:2012.12761. [[CrossRef](#)]
12. Wang, Y.; Liu, X.; Wang, M.; Yu, Y. A hidden anti-jamming method based on deep reinforcement learning. *arXiv* **2020**, arXiv:2012.12448. [[CrossRef](#)]
13. de Curtò, J.; de Zarzà, I.; Cano, J.-C.; Calafate, C.T. Enhancing Communication Security in Drones Using QRNG in Frequency Hopping Spread Spectrum. *Future Internet* **2024**, *16*, 412. [[CrossRef](#)]
14. Perković, T.; Rudeš, H.; Damjanović, S.; Nakić, A. Low-Cost Implementation of Reactive Jammer on LoRaWAN Network. *Electronics* **2021**, *10*, 864. [[CrossRef](#)]
15. Šabić, J.; Perković, T.; Begušić, D.; Šolić, P. Practical Realization of Reactive Jamming Attack on Long-Range Wide-Area Network. *Sensors* **2025**, *25*, 2383. [[CrossRef](#)] [[PubMed](#)]
16. Pourranjbar, A.; Kaddoum, G.; Saad, W. Recurrent Neural Network-based Anti-jamming Framework for Defense Against Multiple Jamming Policies. *arXiv* **2022**, arXiv:2208.09518. [[CrossRef](#)]
17. Lin, X.; Liu, A.; Han, C.; Liang, X.; Sun, Y.; Ding, G. Intelligent Adaptive MIMO Transmission for Nonstationary Communication Environment: A Deep Reinforcement Learning Approach. *IEEE Trans. Commun.* **2025**, *73*, 5965–5979. [[CrossRef](#)]
18. Li, D.; Sun, Y.; Peng, J.; Cheng, S.; Yin, Z.; Cheng, N. Dual Network Computation Offloading Based on DRL for Satellite-Terrestrial Integrated Networks. *IEEE Trans. Mob. Comput.* **2025**, *24*, 2270–2284. [[CrossRef](#)]
19. Sun, Y.; Lin, Z.; An, K.; Li, D.; Li, C.; Zhu, Y. Multi-Functional RIS-Assisted Semantic Anti-Jamming Communication and Computing in Integrated Aerial-Ground Networks. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 3597–3617. [[CrossRef](#)]
20. Qi, J.; Zhang, H.; Qi, X.; Peng, M. Deep Reinforcement Learning Based Hopping Strategy for Wideband Anti-Jamming Wireless Communications. *IEEE Trans. Veh. Technol.* **2024**, *73*, 3568–3579. [[CrossRef](#)]
21. Wang, D.; Wang, J.; Zhong, Y. An Interference Sensing Algorithm Based on Duration Units of Frequency Points for Adaptive Frequency-Hopping System. *IEEE Access* **2023**, *11*, 115403–115414. [[CrossRef](#)]
22. Lan, M.; Luo, Z.; Jiang, M. Intelligent Modulation Recognition of Frequency-Hopping Communications: Theory, Methods, and Challenges. *Big Data Cogn. Comput.* **2025**, *9*, 318. [[CrossRef](#)]
23. Laktionov, I.S.; Hnatushenko, V.V.; Udoviyk, I.M.; Olevskiy, V.I. Simulation-driven assessment of cryptographic algorithms for resource-constrained infocommunication networks. *Nauk. Visnyk Natsionalnoho Hirnychoho Universytetu* **2025**, *6*, 148–156. [[CrossRef](#)]
24. Ailiya; Yi, W.; Varshney, P.K. Adaptation of Frequency Hopping Interval for Radar Anti-Jamming Based on Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2022**, *71*, 12434–12449. [[CrossRef](#)]
25. Li, X.; Chen, J.; Ling, X.; Wu, T. Deep Reinforcement Learning-Based Anti-Jamming Algorithm Using Dual Action Network. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 4625–4637. [[CrossRef](#)]
26. Atheeq, C.; Gulzar, Z.; Al Reshan, M.S.; Alshahrani, H.; Sulaiman, A.; Shaikh, A. Securing UAV Networks: A Lightweight Chaotic-Frequency Hopping Approach to Counter Jamming Attacks. *IEEE Access* **2024**, *12*, 38685–38699. [[CrossRef](#)]
27. Qiao, G.; Bilal, M.; Liu, S.; Babar, Z.; Ma, T. Biologically inspired covert underwater acoustic communication—A review. *Phys. Commun.* **2018**, *30*, 107–114. [[CrossRef](#)]
28. Bilal, M.; Zuberi, H.H.; Jaffar, A.; Riaz, W.; Khan, M.A.; Alharbi, A.; Miyajan, A.; Liu, S. Covert underwater communication through cepstrum modulation mimicking Pseudorca crassidens whistles using machine learning. *Sci. Rep.* **2026**, *16*, 4155. [[CrossRef](#)] [[PubMed](#)]
29. Celis, S.; Farhat, M.; Zhang, L.; Bagci, H.; Eltawil, A.M.; Salama, K.N. On Coding and Decoding Reconfigurable Radiation Pattern Modulation Symbols. *Electronics* **2021**, *10*, 614. [[CrossRef](#)]

30. Hasan, C.; Agapitos, A.; Lynch, D.; Castagna, A.; Cruciata, G.; Wang, H.; Milenovic, A. Continual Model-based Reinforcement Learning for Data Efficient Wireless Network Optimisation. *arXiv* **2024**, arXiv:2404.19462. [[CrossRef](#)]
31. Xu, Y.; Ren, G.; Chen, J.; Zhang, X.; Jia, L.; Kong, L. Interference-Aware Cooperative Anti-Jamming Distributed Channel Selection in UAV Communication Networks. *Appl. Sci.* **2018**, *8*, 1911. [[CrossRef](#)]
32. Niu, Y.; Zhou, Z.; Pu, Z.; Wan, B. Anti-Jamming Communication Using Slotted Cross Q Learning. *Electronics* **2023**, *12*, 2879. [[CrossRef](#)]
33. Laktionov, I. Python project for Benchmarking Frequency Hopping Algorithms for Resilient Wireless Communications [Computer software]. *Zenodo* **2026**. [[CrossRef](#)]
34. Aygur, M.; Kandeepan, S.; Giorgetti, A.; Al-Hourani, A.; Arbon, E.; Bowyer, M. Narrowband Interference Mitigation Techniques: A Survey. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3455–3482. [[CrossRef](#)]
35. Xu, H.; Cheng, Y.; Wang, P. Jamming Detection in Broadband Frequency Hopping Systems Based on Multi-Segment Signals Spectrum Clustering. *IEEE Access* **2021**, *9*, 29980–29992. [[CrossRef](#)]
36. Tang, Z.; Wang, Y.; Cao, Y.; Ren, K.; Tang, X. Wideband jamming suppression for DSSS system based on low-rank and sparse optimization. *Phys. Commun.* **2026**, *76*, 103083. [[CrossRef](#)]
37. Singh, P.; Salameh, H.B.; Bohara, V.A.; Srivastava, A.; Ayyash, M. On mitigating reactive jamming with dynamic resource allocation in optical IRS and UAV-assisted FSO-based networks. *Phys. Commun.* **2024**, *67*, 102520. [[CrossRef](#)]
38. Kulaç, S.; Şahin, M. Bit Error Rate Performance Improvement for Orthogonal Time Frequency Space Modulation with a Selective Decode-and-Forward Cooperative Communication Scenario in an Internet of Vehicles System. *Sensors* **2024**, *24*, 5324. [[CrossRef](#)] [[PubMed](#)]
39. Ferreira Dias, C.; Rodrigues de Lima, E.; Fraidenraich, G. Bit Error Rate Closed-Form Expressions for LoRa Systems under Nakagami and Rice Fading Channels. *Sensors* **2019**, *19*, 4412. [[CrossRef](#)] [[PubMed](#)]
40. Fatadin, I. Estimation of BER from Error Vector Magnitude for Optical Coherent Systems. *Photonics* **2016**, *3*, 21. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.