

Regular Article

Threats of Deep Learning Attacks to Chaos-Based Spread-Spectrum System

Nguyen Thanh Dat, Hoang Duc Hieu, Nguyen Xuan Quyen, Ta Thi Kim Hue

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Correspondence: Ta Thi Kim Hue, hue.tathikim@hust.edu.vn

Communication: received 21 January 2026, revised 13 April 2026, accepted 30 April 2026

Online publication: 13 May 2026, Digital Object Identifier: 10.21553/rev-jec.434

Abstract– Chaos-based spread-spectrum systems exploit chaotic carriers or sequences resembling random noise to transmit and recover information, often employing chaotic spectrum spreading as a form of physical-layer encryption. Representative spread-spectrum schemes, such as Chaotic CDMA and Differential Chaos Shift Keying (DCSK), have been shown to be highly sensitive to noise, parameter mismatches, and their reliance on obfuscation makes them vulnerable to sophisticated attacks. In this paper, we examine the susceptibility of chaos-based spread-spectrum schemes to modern deep learning techniques. By integrating Gramian Angular Field (GAF) encoding with the ResNet50 architecture, we demonstrate effective prediction and reconstruction of hidden chaotic key sequences, even under noisy and fading channel conditions. Experimental results confirm that digital communication systems employing chaotic spread-spectrum modulation can be systematically compromised by advanced deep learning methods, posing new challenges for enhancing their robustness and cryptographic reliability in practical applications.

Keywords– Chaotic communication, chaotic spread-spectrum, chaos-based coherent modulation, gramian angular field.

1 INTRODUCTION

Chaotic spread-spectrum is a communication technique where a message signal is hidden by a chaotic carrier, typically by directly adding the message to a chaotic waveform [1]. In digital chaos-based coherent modulation systems, chaotic oscillators are used to encode information into noise-like signals that are difficult for eavesdroppers to decipher [2]. The appeal of chaotic spread-spectrum lies in chaos theory's hallmark properties: sensitivity to initial conditions and ergodic, wide-band behavior that makes the transmitted signal resemble random noise. This provides a form of physical-layer security, as an unauthorized receiver sees an apparently random waveform with the message submerged in chaos. Significance: Chaos-based communication is studied for secure wireless, optical, and even underwater links [3]. The inherent unpredictability of chaotic signals can make eavesdropping and interception exceedingly challenging. Chaotic spread-spectrum, in particular, has been used to transmit confidential analog and digital signals (e.g., voice, data) by hiding them under a chaotic carrier. Authorized receivers that share the correct chaotic system or key can synchronize to the chaos and recover the message, while outsiders ideally see only noise [2]. Use cases include military and government communications, fiber-optic links, wireless sensor networks, and other scenarios demanding low-probability-of-intercept transmissions. The motivation for targeting these systems is twofold: (1) to assess their proper security (since history has

shown many chaotic ciphers to be weaker than initially believed), and (2) to develop attack tools (like deep learning models) that can extract information from chaotic spread-spectrum signals, thereby identifying vulnerabilities before adversaries do. Underlying Principles: chaotic spread-spectrum relies on the existence of synchronized chaotic generators. A transmitter ("master" chaos system) produces a chaotic waveform and combines it with the plaintext signal. A receiver ("slave" chaos system) runs an identical chaotic system; if kept in sync, the same chaotic carrier can be regenerated and subtracted out, revealing the original message [4]. In digital chaotic modulation, rather than continuous addition, information can be embedded by switching chaotic states or by correlating transmitted chaotic sequences.

The core idea is that chaos provides a complex carrier that acts like an encryption key: only with the right chaotic key or synchronization can the message be extracted. This approach avoids the computational overhead of conventional encryption, but its security is based on physical chaos properties rather than proven ciphers [5]. However, chaotic spread-spectrum and related schemes face fundamental challenges. Real-world chaos-based systems can be sensitive to noise and parameter mismatch, which complicates synchronization. Moreover, the security of chaotic spread-spectrum is not absolute, as it operates by obscurity (hiding in noise) rather than by rigorous cryptographic randomness. Researchers have identified that chaotic communications often have low confidentiality and can be vulnerable to

analysis [6].

Related work - Evaluating attacks on chaos-based modulation systems that use deep learning techniques requires defining the threat model, what the attacker knows, and what the attacker can do. There are three primary threat scenarios. The first is a passive ciphertext-only attacker, who can only listen to the chaotic signals without any prior knowledge of the system parameters or plaintext. Despite being the most challenging scenario, recent advances in deep learning have demonstrated that neural networks, such as RNNs and LSTMs, can effectively learn and predict chaotic sequences. They can train models on recorded signals to generate surrogate chaos sequences, enabling correlation with received data to extract bits [7]. Studies demonstrate that neural networks can distinguish chaos from noise and forecast chaotic sequences with high accuracy, thus reducing the system's security. The second scenario involves a known plain text or pilot-assisted attacker who has partial knowledge of the transmitted data, such as headers or synchronization sequences. This known information allows the attacker to use supervised learning to recognize how specific chaotic patterns encode bits. Neural networks can be trained using these known segments to quickly adapt and decode unknown portions by estimating the system's parameters or chaos mapping, especially if the same keys or encoding schemes are reused across sessions. This approach leverages leakage points in communication protocols to significantly enhance an attacker's decoding capability. The third threat is from active or adaptive attackers. Such attackers might inject signals to probe the system or manipulate the environment, or they could have insider knowledge enabling them to train surrogate models of the chaotic system. Once trained, these models can decode messages even if the attacker loses direct access, because deep learning models can generalize and adapt over time. In some cases, attackers could insert their own signals and influence the system to glean more information or maintain synchronization in [8].

Our work - This paper examines how a known-plaintext attacker can exploit those vulnerabilities, especially using modern deep learning, to break chaotic spread-spectrum in coherent chaos-based modulation systems. Deep learning significantly elevates the threat level against chaos-based communications [9]. These models excel at sequence prediction, pattern recognition, and transfer learning, enabling attackers, even in ciphertext-only scenarios, to infer message content or system parameters with remarkable accuracy. This highlights the need for robust, unpredictable chaos design and strict key management to mitigate these vulnerabilities and ensure the security and integrity of chaos-based communication systems. Experimental results show that the proposed GAF-ResNet50 architecture is more effective than related models, such as CNN-1D [10], in this known-plaintext attack scenario, better accuracy than LSTM [11], Hybrid CNN-LSTM [12] and MLE [13], GAF-ResNet50 is adaptable in terms of performance.

2 CHAOTIC SPREAD-SPECTRUM AND DIGITAL CHAOS-BASED COHERENT MODULATION

Chaos-based digital modulation schemes come in several forms, each using chaotic signals to encode bits. Additive chaotic spread-spectrum, the earliest chaos-based security method, is the simplest scheme, in which a message $m(t)$ is directly added to a chaotic carrier signal $c(t)$. The transmitter outputs $s(t) = m(t) + c(t)$, and a synchronized receiver subtracts an internally generated chaos $c'(t)$ to recover $m(t)$ [3]. The disadvantage of this method is its susceptibility to noise and parameter mismatch. Even a slight desynchronization causes errors, and an observer with knowledge of the chaotic generator can attempt to replicate and subtract the chaos. In fact, chaotic spread-spectrum is noted to have low confidentiality and poor noise tolerance, so it is rarely used in practice for digital modulations [2]. Noise in the channel easily disrupts chaotic synchronization, making recovery difficult for the legitimate receiver (and potentially revealing message patterns to an interceptor when sync fails).

Chaotic Shift Keying (CSK): A coherent chaos modulation where binary or M-ary data symbols are mapped to different chaotic attractors or signals [2]. For example, bit "0" might be transmitted by a chaotic oscillator at parameter set A, and bit "1" by the same oscillator at parameter set B. The receiver, knowing these two chaotic signal patterns, tries to synchronize or match the received waveform to one of the chaos generators to decide the bit [4]. Essentially, the basis functions are chaotic signals instead of sinusoids. CSK requires strict synchronization of the chaotic generator at the receiver to each new chaotic segment sent. If the transmitter rapidly switches chaos parameters per bit, the receiver must reliably generate the corresponding chaotic waveform for demodulation. In theory, CSK can achieve a bit-error-rate (BER) similar to conventional BPSK under perfect conditions. In practice, however, achieving and maintaining chaotic synchronization in a noisy, time-varying environment is extremely difficult. This limits CSK's real-world use because losing sync even briefly causes data loss.

Differential Chaos Shift Keying (DCSK): A non-coherent scheme that avoids needing the receiver to generate a chaos replica [1]. In DCSK, each bit period is split into two halves: the first half carries a reference chaotic signal, and the second half carries a modulated version of that signal (for instance, the same chaotic sequence or its inverse, depending on whether the bit is 0 or 1). The receiver correlates the two halves. A high correlation might indicate the bit was "0" (same chaos sent twice), whereas an inverted correlation indicates bit "1" (second half was chaos $\times -1$). DCSK thus dispenses with explicit chaos synchronization; the chaotic waveform is sent along with the data in every bit period. Trade-offs: DCSK is much simpler to implement and more robust to sync issues than CSK [?]. However, it sacrifices bandwidth and energy efficiency, since each bit requires transmitting a long chaotic reference. Half the time carries no new information, doubling the required

bandwidth for a given data rate. This also creates a security concern: reusing the chaotic reference each bit (or sending it openly) can aid an attacker. As one study notes, transmitting the same chaotic sequence twice per bit “negatively affects security” because an eavesdropper can recognize and use the reference copy. Variations and Advanced Schemes: Over the years, many variants have been proposed to improve chaos-based modulation performance [5]. Examples include Chaos On-Off Keying (COOK), where bits are transmitted by the presence or absence of a chaotic burst; FM-DCSK (frequency-modulated DCSK), which uses frequency shifts of chaos; Quadrature Chaos Shift Keying (QCSK), which sends two chaotic signals in parallel quadrature channels to double data rate; and multi-carrier or M-ary generalizations [2]. These retain the core idea of using chaotic sequences as signals, but tweak how they are transmitted to improve reliability or throughput. Some systems use chaotic spreading sequences in a manner analogous to spread-spectrum (replacing m-sequences or Gold codes with chaos) to achieve multi-user communication with low cross-correlation. Each scheme must balance chaos synchronization requirements, data rate, noise resilience, and complexity.

Operation of chaotic spread-spectrum: In classical analog chaotic spread-spectrum, the chaotic oscillator continuously produces a high-frequency chaotic waveform that is much larger (in amplitude or power) than the message. The message (e.g. an audio signal) is added so that the composite looks like the chaos plus a tiny perturbation. The receiver, running an identical chaotic oscillator synchronized via a driving signal from the transmitter, subtracts the chaos out, recovering the message [3]. In digital chaotic spread-spectrum, the same principle applies: a chaotic sequence is added to a digital modulation (like a QAM signal) to create a chaos-based modulation signal. Only someone who can generate or estimate that chaotic sequence can subtract it and retrieve the original modulation.

Therefore, digital chaos-based modulation schemes offer unconventional ways to hide signals in plain sight, but many suffer from practical limitations. chaotic spread-spectrum is simple but fragile to noise. CSK is conceptually secure but impractical without perfect sync. DCSK is more robust but at the cost of redundancy and potential leakage of the reference. This opens the door for attackers to exploit these weaknesses, as discussed next. Modern research often favors hybrid approaches (such as chaos combined with traditional spread spectrum or chaos combined with encryption) to shore up security, but understanding the attack methodologies is crucial for any design [5].

3 DEEP LEARNING-BASED ATTACKS

Chaos-based Direct Sequence Code Division Multiple Access (DS-CDMA), often referred to as *Chaotic Shift Keying* (CSK) in the literature, represents a modulation technique that employs chaotic sequences for spreading and despreading operations. The CSK and chaos-based

DS-CDMA systems are considered equivalent when the receiver is capable of generating an exact replica of the chaotic signal used at the transmitter, achieving *perfect chaotic synchronization* [4]. This synchronization enables the receiver to despread the incoming signal and accurately recover the transmitted information. In this scheme, the data symbols $s_i = \pm 1$ with a symbol period T_s are spread by a *chaotic spreading sequence* x_k , where each chaotic chip is generated at every interval T_c such that $x_k = x(kT_c)$. As illustrated in Figure 1, the transmitter multiplies the data symbol by the locally generated chaotic sequence, producing the spread signal $e(t)$, which is then transmitted through the wireless channel. At the receiver side, the received signal $r(t)$ is first multiplied by the locally regenerated chaotic sequence synchronized with that of the transmitter. The resulting signal is then integrated over one symbol duration T_s , producing a *decision variable* D_i . The transmitted bit is finally estimated by comparing this variable to a predefined threshold. This system leverages the *non-periodic and noise-like characteristics* of chaotic sequences, which provide inherent security and improved resistance to interference and multipath fading compared to conventional periodic spreading codes. When synchronization is well maintained, chaos-based DS-CDMA exhibits performance and operational characteristics similar to traditional DS-CDMA systems, while offering enhanced privacy and robustness in multiuser communication environments.

The eavesdropping receiver can receive the transmitted signal $r(t)$ over the wireless channel and use AI for analysis. The signal $r(t)$ is the transmitted signal $e(t)$ after being affected by noise and fading of the wireless channel. Since $e(t) = S_i \times x_{k,i}$, when a bit +1 is transmitted, $S_i = 1$, thus $e(t) = x_{k,i}$. At this moment, the transmitted signal is the original chaotic sequence, which the eavesdropping receiver can intercept and analyze to recover it.

Modern deep learning (DL) techniques have proven remarkably effective at modeling and decoding complex, nonlinear patterns – precisely the kind of challenge posed by chaotic signals, shown in Figure 2. In [14, 15], researchers have applied neural networks (including Convolutional Neural Networks, recurrent networks like LSTMs, and Reservoir Computing models) to attack or demodulate chaos-based communication systems [16]. These approaches treat the chaotic transmitter or channel as a black box to be learned from data. Deep learning provides a data-driven shortcut to cracking chaotic spread-spectrum systems [7, 8]. Where traditional attacks required mathematical insight or lucky guesses about the system, a sufficiently complex neural network can learn the entire demodulation function. This includes learning to synchronize (or bypass sync), equalize channel distortions, and decode the underlying message [6].

The following equations describe the step-by-step process for transforming a time series into a GAF matrix, as outlined below.

Step 1: Robust Normalization

The time series $\phi = [\phi_1, \phi_2, \dots, \phi_n]$ is normalized to

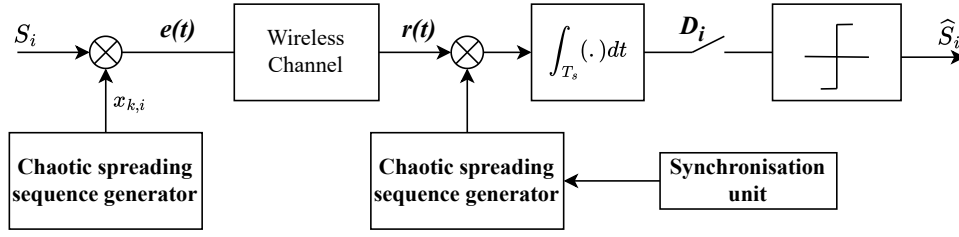


Figure 1. Chaos-based DS CDMA Transmission.

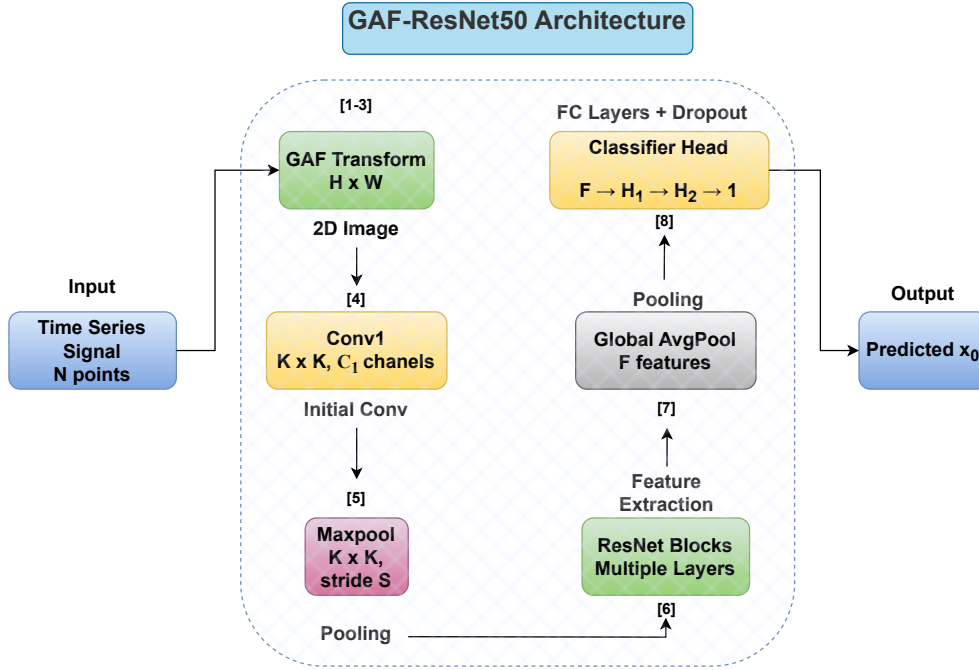


Figure 2. Deep Learning - based Known Plaintext Attacks Using GAF ResNet50 Architecture.

the range $[-1, 1]$ using the following formula

$$\phi_{\text{norm},i} = 2 \times \frac{\phi_i - \phi_{\min}}{\phi_{\max} - \phi_{\min}} - 1, \quad (1)$$

where $\phi_i, i = [1, n]$ is the amplitude in the sample i^{th} , we denoted that $\phi_{\min} = \min(\phi)$ and $\phi_{\max} = \max(\phi)$ are the minimum and maximum values of the time series, respectively.

The range of edge cases is handled to close to zero. If $\phi_{\max} - \phi_{\min} < 10^{-10}$, we have $\phi_{\text{norm}} = 0$. Consequently, the normalized values are clipped to ensure they lie within $[-1, 1]$

$$\phi_{\text{norm},i} = \text{clip}(\phi_{\text{norm},i}, -1, 1). \quad (2)$$

Step 2: Compute Polar Angles

The normalized values are converted to polar angles using the arccosine function

$$\theta_i = \arccos(\phi_{\text{norm},i}), \quad (3)$$

where θ_i lies in the range $[0, \pi]$.

Step 3: Construct GAF Matrix

The GAF matrix G is constructed using the cosine of

the sum of polar angles

$$G_{i,j} = \cos(\theta_i + \theta_j), \quad (4)$$

where $G_{i,j}$ is the element at row i and column j in the $n \times n$ GAF matrix, θ_i and θ_j is the polar angle that is the normalized value at indices i and j . As a result, G is an $n \times n$ symmetric matrix representing the 2D transformation of the time series.

Step 4: Initial Convolution (Conv1)

The GAF matrix G is treated as grayscale input X . The first convolutional layer is modified for single-channel input by averaging pretrained RGB weights

$$W_{\text{gray}}^{(1)} = \frac{1}{c_{\text{in}}} \sum_{c=1}^{c_{\text{in}}} W_c^{(1)}, \quad (5)$$

followed by convolution and batch normalization with ReLU activation:

$$Z^{(1)} = \text{ReLU}(\text{BN}(W_{\text{gray}}^{(1)} * X + b^{(1)})), \quad (6)$$

where $*$ denotes convolution with kernel size k , stride s , padding p , and c_{in} is the number of input channels (typically 3 for pretrained RGB).

Step 5: Max Pooling

The output of the initial convolution is passed through max pooling to reduce spatial dimensions

$$Z^{(1)} = \text{MaxPool}(Z^{(1)}, k_{\text{pool}}, s_{\text{pool}}), \quad (7)$$

where k_{pool} is the pooling kernel size and s_{pool} is the pooling stride.

Step 6: Residual Blocks

ResNet consists of multiple stages with bottleneck residual blocks. For input $Y_{\in}$ in a block

$$\begin{cases} Y_1 = \text{ReLU}(\text{BN}(W_1 * Y_{\in} + b_1)), & W_1 \in \mathbb{R}^{c/4 \times c_{\text{in}} \times 1 \times 1} \\ Y_2 = \text{ReLU}(\text{BN}(W_2 * Y_1 + b_2)), & W_2 \in \mathbb{R}^{c/4 \times c/4 \times k_{\text{mid}} \times k_{\text{mid}}} \\ Y_3 = \text{BN}(W_3 * Y_2 + b_3), & W_3 \in \mathbb{R}^{c \times c/4 \times 1 \times 1} \end{cases} \quad (8)$$

We obtain the output $Y_{\text{out}} = \text{ReLU}(Y_3 + P(Y_{\in}))$, where P is the identity if dimensions match, or a projection convolution $W_p \in \mathbb{R}^{c \times c_{\in} \times 1 \times 1}$ otherwise. Each stage has output channels c and a number of blocks b , with downsampling in the first block of subsequent stages.

Step 7: Global Average Pooling

After the final stage $Z^{(L)} \in \mathbb{R}^{d_f \times h' \times w'}$

$$F = \frac{1}{h' \times w'} \sum_{i=1}^{h'} \sum_{j=1}^{w'} Z_{:,i,j}^{(L)} \in \mathbb{R}^{d_f}. \quad (9)$$

Step 8: Classifier Head

The feature vector F is processed through fully connected layers with dropout and batch normalization:

$$\begin{cases} H_1 = \text{dropout}(p, \text{linear}(F; W_{h1}, b_{h1})), & W_{h1} \in \mathbb{R}^{d_1 \times d_f}, \\ H_2 = \text{BN}(\text{ReLU}(H_1)), \\ H_3 = \text{dropout}(p, \text{linear}(H_2; W_{h2}, b_{h2})), & W_{h2} \in \mathbb{R}^{d_2 \times d_1}, \\ H_4 = \text{BN}(\text{ReLU}(H_3)), \\ \hat{x}_0 = \text{dropout}(p/2, \text{linear}(H_4; W_{h3}, b_{h3})), & W_{h3} \in \mathbb{R}^{1 \times d_2}, \end{cases} \quad (10)$$

where $\hat{x}_0$ is the predicted initial condition.

We assume that a known-plaintext attack is available in this experiment, in which specific fields or preambles are known (e.g., packet headers or sync words). An attacker who knows even a small portion of the transmitted data (plaintext) and its chaotic spread spectrum form now has a training set for supervised learning. For instance, if a chaotic time series always begins with a known initial value x_0 , the attacker can use that to quickly train a neural network to learn how those orbits are represented in the chaotic signal. This is analogous to how legitimate receivers might send training sequences to adapt an equalizer; the attacker co-opts them to train a chaos-based spread-spectrum sequence. In the threat model, the attacker does not initially know the secret chaos key, also known as the initial value x_0 . Deep learning can rapidly adapt to known patterns, such as one-shot or few-shot learning. The attacker's model could transfer-learn the chaotic system parameters from a short known sequence and then apply them to decode the unknown parts. This is particularly potent if the system reuses the same chaotic key or mapping for an entire session. In this way, a network can implicitly estimate the channel and chaos given some training data. If the communications pro-

ocol exposes these "leaks," a DL attacker will exploit them to improve accuracy dramatically.

Table I
DESCRIPTION OF DATASETS

x_0	E_b/N_0	α	β	M	Samples
0.1 $\rightarrow$ 0.9	15	0.5	32	32	9
0.09 $\rightarrow$ 0.09	15	0.5	32	32	9
0.001 $\rightarrow$ 0.09	15	0.5	32	32	9
0.1 $\rightarrow$ 0.9	15	0.5	32	32	9
0.11 $\rightarrow$ 0.91	15	0.5	32	32	9
0.0011 $\rightarrow$ 0.0091	15	0.5	32	32	9
0.1 $\rightarrow$ 0.05	0 $\rightarrow$ 16	0.5	32	32	85
0.1 $\rightarrow$ 0.05	0 $\rightarrow$ 16	0.5	32	32	95
0.1 $\rightarrow$ 0.05	0 $\rightarrow$ 16	0.5	32	32	99
1e-05 $\rightarrow$ 9.e-05	0 $\rightarrow$ 16	0.5	32	32	99
1e-06 $\rightarrow$ 9.e-06	0 $\rightarrow$ 10	0.5	32	32	99

Table I presents the datasets, divided into multiple cases, and describes the key parameters for each scenario considered in this study. In Case 1, the parameter x_0 ranges from 0.1 to 0.9, with E_b/N_0 fixed at 15 dB, and the parameters $\alpha = 0.5$, $\beta = 32$, and $M = 32$ are kept constant, with each configuration generating 9 samples. Case 2 has x_0 at 0.09, with E_b/N_0 still at 15 dB, and other parameters remaining unchanged. In Case 3, x_0 is set to 0.1, while in Case 4, it is set to 0.01, both with the fixed parameters. Case 5 extends the analysis by varying x_0 from 0.0011 to 0.0091, with E_b/N_0 fixed at 15 dB. Case 6 examines the system performance as E_b/N_0 ranges from 0 to 16 dB, with x_0 varying from 0 to 0.099 and the number of samples increasing to 99. Finally, Case 7 focuses on extremely small x_0 values from 1×10^{-6} to 9×10^{-6} , also with 99 samples. These configurations help evaluate system performance under different parameter conditions.

4 EXPERIMENTAL RESULTS

4.1 Known-Plaintext Attack Efficiency

In our experiment, we predicted x_0 within the range (0.000001 $\rightarrow$ 0.000004), the histograms show that the errors are concentrated from 2 to about 4×10^{-5} . The density peaks fluctuate between 250,000 and 300,000, reflecting high precision and consistency in the predictions at these values x_0 . Predictions in this range indicate the strongest (most robust) model, with errors primarily due to low noise levels (high E_b/N_0). Next, x_0 within the range (0.00001 $\rightarrow$ 0.00004), the error distribution is wider than in the previous row, with errors mainly concentrated from 0 to 2×10^{-5} . The density peaks hover around 300,000, with a more variable distribution compared to the upper row. This range is more sensitive to noise, and errors increase at lower E_b/N_0 levels. Figures 3 and show the distribution of absolute errors in predicting initial condition values x_0 of a chaotic signal with noise and fading. The graph consists of histograms in several cases corresponding to specific values of x_0 . The horizontal axis represents

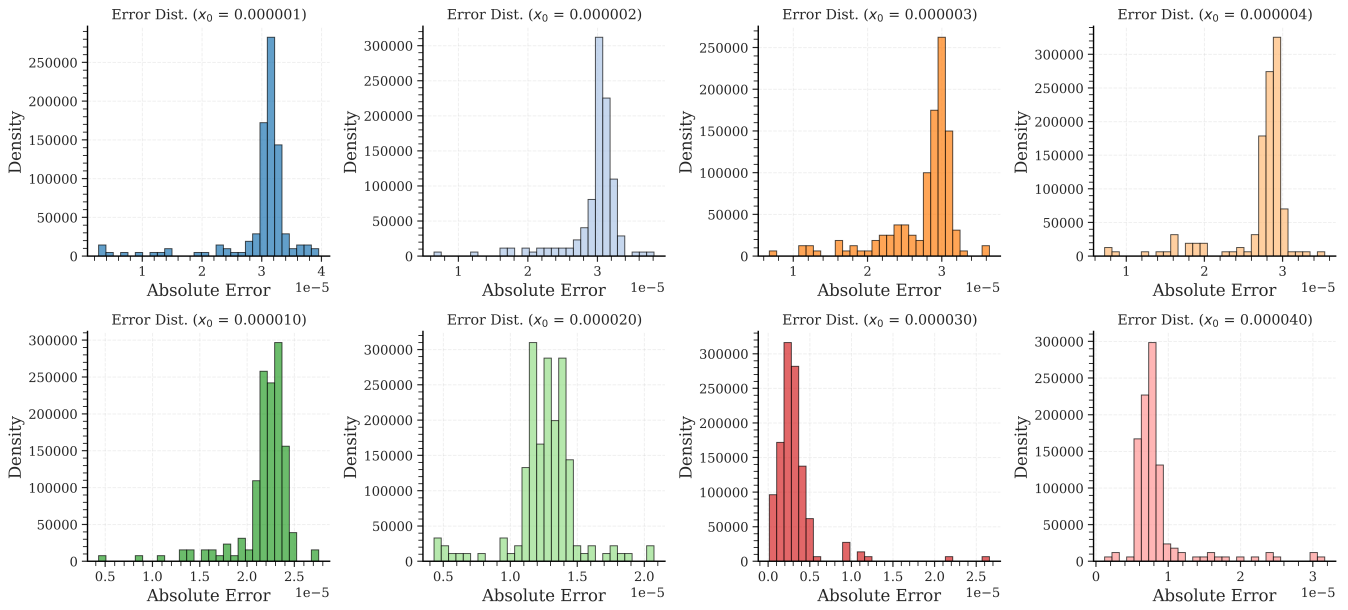


Figure 3. Error Distribution Analysis.

the absolute error (unit 1×10^{-5}), and the vertical axis indicates the density of predicted errors concentrated in certain ranges.

Figures 4 and 5 compare the original time series and the predicted series, revealing a remarkable similarity in their overall shape and distribution, indicating our model's effectiveness in capturing the essential characteristics of the data. The top graph shows the actual values of the time series, exhibiting natural fluctuations and a specific distribution pattern. In contrast, the bottom graph presents the model's predictions, which closely mirror these properties. The resemblance between the waveforms and the amplitude distributions suggests that the model has successfully learned the underlying statistical features of the original data. This similarity directly correlates with the high prediction accuracy observed for the initial value x_0 of the chaotic spreading sequences, which is critical in time series forecasting as it influences subsequent predictions. Figures 6 and 7 show that RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are low (ranging from 1×10^{-5} to 5×10^{-5}), which demonstrates that the GAF-ResNet model predicts x_0 effectively, even in cases affected by larger white noise values E_b/N_0). It is clear that increasing E_b/N_0 does not significantly affect the model's accuracy; therefore, further enhancement is needed through additional training on the dataset, even under various noise conditions. In our experiment, x_0 values within the range 0.1 – 0.9 yield lower RMSE and MAE, which is reasonable given chaotic sensitivity; nevertheless, the relative differences are slight, reflecting the model's good generalization.

Figures 3 and 8 show the distribution of absolute errors in predicting initial condition values x_0 of a chaotic signal with noise and fading. The graph consists of histograms in several cases corresponding to specific values of x_0 . The horizontal axis represents the absolute error (unit 1×10^{-5}), and the vertical axis

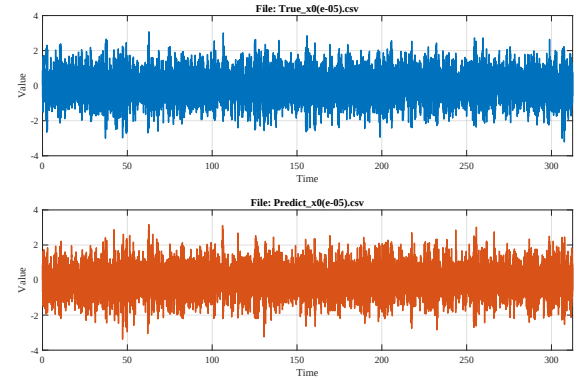
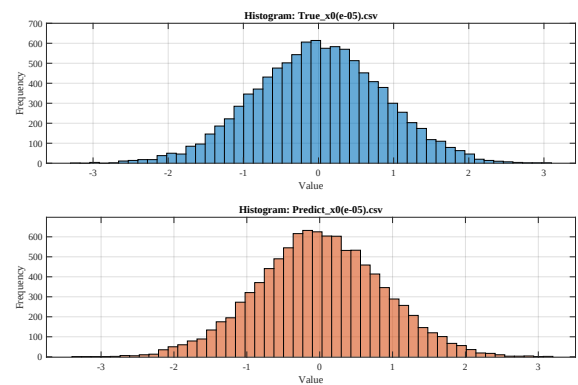
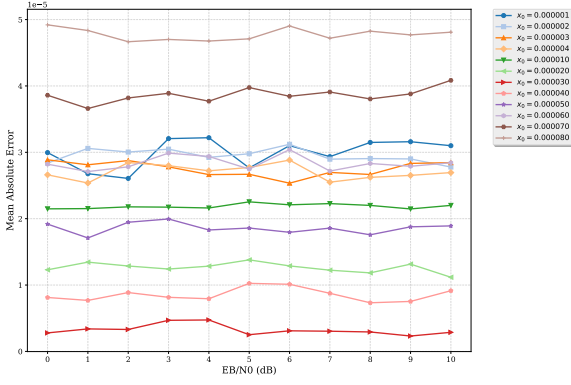
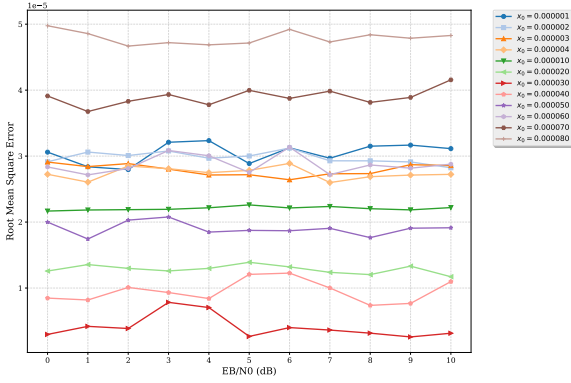
Figure 4. The predictive value x_0 for the chaotic spreading sequences.

Figure 5. Comparison of Histograms between the original and predictive chaotic spreading sequence.

indicates the density of predicted errors concentrated in certain ranges.

4.2 Accuracy Evaluation

Table II reports results for a regression task, where lower RMSE and MAE indicate better performance,

Figure 6. MAE vs. E_b/N_0 Level.Figure 7. RMSE vs. E_b/N_0 Level.

while higher R^2 and Pearson correlation indicate better fitting quality. Our proposed GAF-ResNet50 outperforms CNN-1D on all four metrics: RMSE decreases from 0.00087 to 0.00063 (27.6%), MAE decreases from 0.00059 to 0.00046 (22.0%), R^2 increases from 0.9959 to 0.9979, and Pearson correlation increases from 0.9984 to 0.9997, indicating that the proposed architecture can capture more informative temporal features and nonlinear dependencies. In Table III, the proposed method has more parameters (25.6M) and a longer inference time (173.52 ms per batch) than CNN-1D (16.08M parameters and 147.8 ms) leading to reduce efficiency. This increase is a reasonable trade-off for achieving substantially higher accuracy. Therefore, the proposed method is more suitable for accuracy-critical applications, such as offline analysis, cloud-assisted monitoring, and decision-support systems, whereas CNN-1D may be more appropriate for lightweight and real-time scenarios with strict latency or hardware constraints.

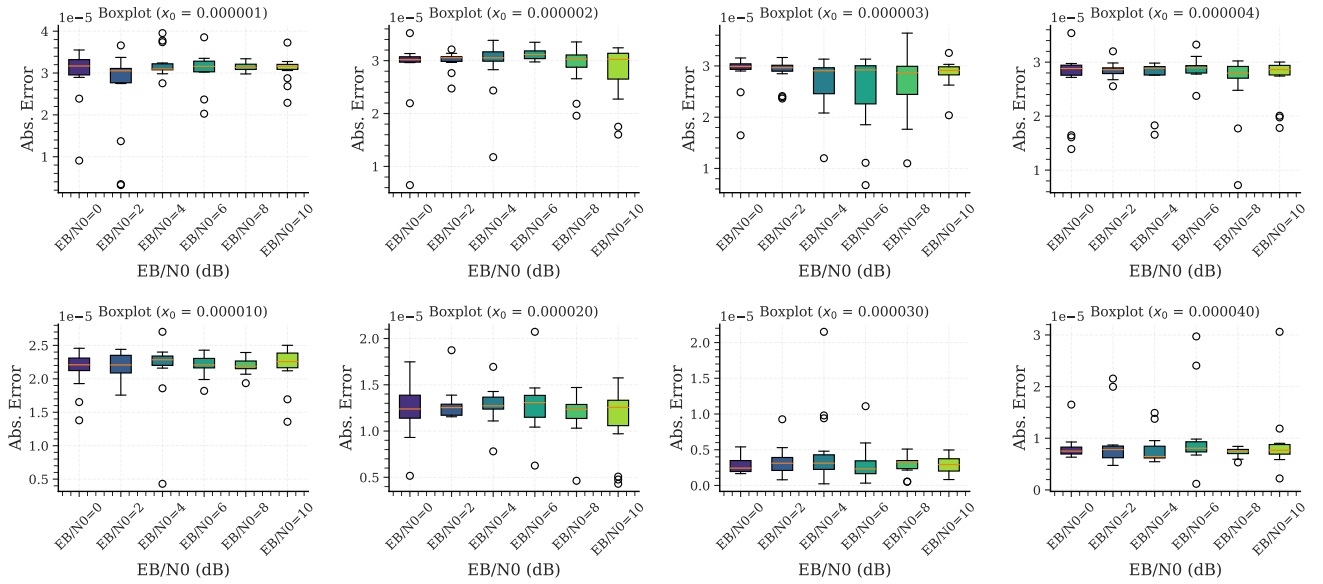
In the context of the attack scenario, CNN-1D learns features from local sliding windows, which means its receptive field is inherently limited, making it unable to capture long-range dependencies in chaotic sequences that are crucial for distinguishing different values of x_0 . In contrast, the GAF transformation converts the time series $\phi = [\phi_1, \phi_2, \dots, \phi_n]$ into a 2D matrix $G_{i,j} = \cos(\theta_i + \theta_j)$, where $\theta_i = \arccos(\phi_{\text{norm},i})$ is the angular representation of the value at time step i . Each element $G_{i,j}$ encodes the relationship between

any two time steps i and j across the entire sequence without being constrained by a local window. Consequently, the full temporal structure of the chaotic sequence is preserved in a global 2D representation, allowing ResNet50 to access global context during feature learning. Chaotic systems such as the logistic map exhibit fractal attractor structures that serve as geometric fingerprints uniquely characterizing each value of x_0 . When the chaotic sequence is represented as a GAF matrix, this attractor structure emerges as learnable 2D image patterns. In the proposed architecture, we integrate both Channel Attention and Spatial Attention through the CBAM (Convolutional Block Attention Module) mechanism: Channel Attention learns to emphasize the most informative feature maps for identifying x_0 , while Spatial Attention learns to focus on the most relevant regions in the GAF matrix. This mechanism enables the model to selectively attend to the underlying chaotic attractor structure rather than processing all regions equally, explaining why ResNet50 with attention outperforms CNN-1D, which lacks an explicit mechanism for selective feature weighting.

Moreover, model compression techniques such as INT8 quantization can potentially reduce the model size to approximately 25-30 MB without significant degradation in accuracy, making deployment on edge devices feasible when required. The inference time of 173.52 ms for a batch of 48 samples (equivalent to 3.6 ms per sample) is 17.4% slower than that of CNN-1D. However, with a throughput of 276.62 samples per second, the proposed model comfortably satisfies the real-time processing requirements of MC-DCSK systems, which typically demand 100-200 samples per second. Accepting a 17% increase in latency in exchange for a 27.6% improvement in RMSE and a 22% reduction in MAE constitutes a highly valuable trade-off. In terms of computational complexity, the proposed model requires 4.1 GFLOPs, which is higher than CNN-1D (1.27 GFLOPs) but lower than LSTM (6.88 GFLOPs), and well within the processing capabilities of modern GPUs (e.g., RTX 3090 with 35 TFLOPs). Experimental results indicate that the model can operate at 60-80 FPS on consumer-grade GPUs, sufficient for most practical applications. Compared with MLE, the proposed method is 1.56 times slower (173.52 ms vs. 111.3 ms) but achieves a 71-fold improvement in RMSE. Compared with LSTM, it is 3.02 times faster while delivering 78.1% higher accuracy. Overall, relative to CNN-1D, the proposed approach sacrifices a modest speed penalty (−17%) to achieve substantial gains in accuracy (+27.6% RMSE and +22% MAE).

5 CONCLUSION

Our paper emphasizes the vulnerability of chaos-based spread-spectrum systems, such as Chaotic CDMA and DCSK, to modern deep learning attacks. By combining Gramian Angular Field (GAF) encoding with the ResNet50 architecture, we demonstrated the ability to accurately predict and reconstruct hidden chaotic

Figure 8. Boxplot Error By E_b/N_0 per x_0 .Table II
ACCURACY COMPARISON

Model	RMSE	MAE	R2 Score	Pearson Correlation
CNN-1D [10]	0.00087	0.00059	0.9959	0.9984
LSTM [11]	0.00287	0.00180	0.9559	0.9780
Hybrid CNN-LSTM [12]	0.00213	0.00131	0.9756	0.9879
MLE [13]	0.0448	0.00985	0.9243	0.8961
Our work	0.00063	0.00046	0.9979	0.9997

Table III
PERFORMANCE COMPARISON

Model	Parameters	Model Size (MB)	Inference time per batch (ms)	Throughput (samples/second)	FLOPs (B)
CNN-1D	16.08	61.35	147.80	324.76	1.27
LSTM	4.38	78.12	523.48	91.69	6.88
Hybrid CNN-LSTM	2.96	11.29	306.27	156.72	6.10
MLE	N/A	N/A	111.30	8.98	N/A
Our work	25.60	102.01	173.52	276.62	4.10

spread-spectrum sequences from the predictive initial value x_0 , even under challenging conditions such as noise and fading channels. Our experimental results show that the proposed GAF-ResNet model achieves low error metrics, with Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values ranging from 1×10^{-5} to 5×10^{-5} . This indicates effective prediction performance across diverse parameter settings. Obviously, the system remains robust to variations in fading noise, such as α and E_b/N_0 , with higher noise levels having minimal impact on accuracy. It means that the deep learning model generalizes well across different conditions. The analysis of prediction errors, presented through error distributions and box plots, further confirms the model's precision, particularly at low to moderate noise levels. The accuracy assessments indicate that GAF-ResNet50 offers higher estimation accuracy at the expense of increased model size and inference time. This makes it more appropriate for offline forensic analysis and worst-case security vulnerability assessment, where accurate estimation of the initial condition is critical. In contrast, CNN-1D is more

suitable for real-time and resource-limited deployment. These results therefore demonstrate a practical trade-off between attack effectiveness and deployment efficiency.

REFERENCES

- [1] M. Itoh, "Spread Spectrum Communication via Chaos," *International Journal of Bifurcation and Chaos*, vol. 9, pp. 155–213, 1999.
- [2] G. Kaddoum, "Wireless Chaos-Based Communication Systems: A Comprehensive Survey," *IEEE Access*, vol. 4, pp. 2621–2648, 2016.
- [3] T.-L. Liao and S.-H. Tsai, "Adaptive synchronization of chaotic systems and its application to secure communications," *Chaos, Solitons & Fractals*, vol. 11, no. 9, pp. 1387–1396, 2000.
- [4] M. Mobini and G. Kaddoum, "Deep Chaos Synchronization," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1571–1582, 2020.
- [5] M. W. A. Ashraf, A. R. Singh, A. Pandian, M. Bajaj, I. Zaitsev, and R. S. Rathore, "Enhancing Network Security with Hybrid Feedback Systems in Chaotic Optical

- Communication," *Scientific Reports*, vol. 14, no. 1, p. 24958, 2024.
- [6] A. Anees and I. Hussain, "A Novel Method to Identify Initial Values of Chaotic Maps in Cybersecurity," *Symmetry*, vol. 11, p. 140, 2019.
 - [7] M. Mobini, M. Akhlaghi, Z. Mobini, and E. Alarcon, "Design of a SIMO Deep Learning-Based Chaos Shift Keying Demodulation Scheme," *Sensors*, vol. 22, no. 1, p. 333, 2022.
 - [8] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial Machine Learning for 5G Communications Security," *IEEE Communications Magazine*, vol. 59, pp. 102–108, 2021.
 - [9] T. T. K. Hue, N. T. Dat, N. H. Tien, and N. X. Quyen, "Deep Hybrid Learning Model for Fading Mitigation in MC-DCSK Receivers," *International Journal of Bifurcation and Chaos*, 2026.
 - [10] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, March 2016.
 - [11] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, July 2019.
 - [12] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN-LSTM Model for Gold Price Time-Series Forecasting," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 351–17 360, 2020.
 - [13] Y. An, X. Li, and F. Ding, "Maximum Likelihood Based Multi-Innovation Stochastic Gradient Identification Algorithms for Bilinear Stochastic Systems With ARMA Noise," *International Journal of Adaptive Control and Signal Processing*, vol. 37, no. 11, pp. 2690–2715, 2023.
 - [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
 - [15] Z. Wang and T. Oates, "Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks." [Online]. Available: <https://api.semanticscholar.org/CorpusID:16409971>
 - [16] J. Xia, L. Chen, and H.-F. Ma, "Enhanced Chaotic Communication with Machine Learning," *AIP Advances*, vol. 14, no. 11, 2024.



Nguyen Thanh Dat received the Bachelor's degree in Electronics and Telecommunications Engineering (Talented Program) from the School of Electrical and Electronic Engineering (SEEE), Hanoi University of Science and Technology (HUST), Vietnam, in 2026. His research interests include AI applications, deep learning, chaos-based communications, chaos theory, and AI security.



Hoang Duc Hieu received the Bachelor's degree in Electronics and Telecommunications Engineering in 2024. He is currently pursuing the Master's degree in Telecommunications Engineering. His research interests include wireless communications, chaos-based applications, signal processing, physical layer security, and artificial intelligence applications in communication systems.



Nguyen Xuan Quyen received the Diploma of Engineer, Master and PhD degrees in Electronics and Telecommunications in 2006, 2008 and 2013, respectively. He was an exchange PhD student at the Institute for Smart System Technologies, Alpen-Adria Klagenfurt University, Austria, during 2011–2012. During 2014–2015, he worked as a postdoctoral researcher at the Department of Computer Architecture, Polytechnic University of Catalonia (UPC BarcelonaTech), Spain. He was also an academic visitor at the Department of Electronic and Computer Engineering, University of Limerick, Ireland in 2015, at the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom in 2016, and at the Department of Computer and Network Engineering, The University of Electro-Communications, Tokyo-Japan in 2017. From 2017 to current, he has been serving as a senior lecturer and associate professor at the School of Electrical and Electronic Engineering (SEEE), Hanoi University of Science and Technology (HUST), Vietnam. His main research interests are microwave design and implementation, chaos-based digital communications, physical layer security, and wireless communications.



Ta Thi Kim Hue received the Diploma of Engineer, Master and PhD degrees in Electronic Engineering and Telecommunication Technologies from Hanoi University of Science and Technology, Vietnam, in 2007, 2009, and 2016, respectively. From 2015 to 2016, she was an exchange PhD student at the Department of Electronics and Informatics, Vrije Universiteit Brussel, Belgium. She has served as a lecturer at the SET of HUST since 2012. She is (co-)author of over 20 international publications. She has been a program committee member for numerous conferences and workshops (ICCE 2016, 2018, ATC 2017, 2018). She regularly reviews papers for multiple journals, such as the Journal of Science and Technology and the Journal of Military Science and Technology Vietnam. She worked for almost 15 years as a researcher at the Signal Information Processing Laboratory, Hanoi University of Science and Technology. Her current interests include complex networks, cryptography, and physical security protocols for future internet.