JaX: Detecting and Cancelling High-power Jammers Using Convolutional Neural Network

Hai N. Nguyen
nguyen.hai@northeastern.edu
Khoury College of Computer Sciences
Northeastern University, USA

Guevara Noubir
g.noubir@northeastern.edu
Khoury College of Computer Sciences
Northeastern University, USA

ABSTRACT

In this paper, we present JaX, a novel approach for detecting and cancelling high-power jammers in the scenarios when the traditional spread spectrum techniques and other jammer avoidance approaches are not sufficient. JaX does not require explicit probes, sounding, training sequences, channel estimation, or the cooperation of the transmitter. We identify and address multiple challenges, resulting in a convolutional neural network for a multi-antenna system to infer the existence of interference, the number of interfering emissions and their respective phases. This information is continuously fed into an algorithm that cancels the interfering signal. We develop a two-antenna prototype system and evaluate our approach in various environment settings and modulation schemes using SDR platforms. We demonstrate that the receiving node equipped with our approach can detect a jammer with over 99% of accuracy and achieve a Bit Error Rate as low as 10^{-6} even when the jammer power is nearly two orders of magnitude (19 dB) higher than the legitimate signal, and without modifying the link modulation. JaX is resilient against various jammers with different characteristics of jamming signals, jamming power, and timing pattern.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Computing methodologies → Neural networks**.

KEYWORDS

Jamming detection; jamming cancellation; deep learning

ACM Reference Format:
Hai N. Nguyen and Guevara Noubir. 2023. JaX: Detecting and Cancelling High-power Jammers Using Convolutional Neural Network. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/0.1145/3558482.3590178

1 INTRODUCTION

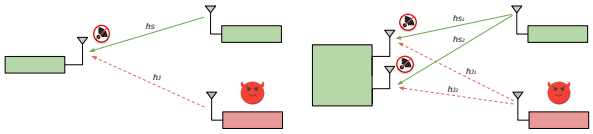
Jamming remains one of the most serious threats to wireless communications. A jamming attack targeting the PHY layer can significantly degrade the Signal-to-Noise Ratio (SNR) of wireless links. This is especially critical to systems with high SNR requirements, e.g., Wi-Fi manufacturers recommend a minimum SNR of 20 dB for data applications [6]. Furthermore, jammers do not require sophisticated RF Front-Ends design and basic jamming hardware against Wi-Fi, cellular networks, and GPS devices is a commodity that can be found on the Internet for a few dozens of dollars. Due to a series of incidents, the FCC routinely releases customer advisories cautioning against the import and use of jamming devices [10], rolled out a jammer tip line (1-855-55NOJAM), and issued several fines [11]. Despite the regulation, preventing jamming remains difficult to enforce. Furthermore, wireless softwarization is making jamming potentially a ubiquitous threat, as demonstrated by the nexmon framework [36], Google Project Zero [34], and recognized in numerous work [31, 44, 47]. Finally, jamming can also be the prelude to more sophisticated attacks such as rogue infrastructure (Wi-Fi and Cellular) and hijacking of physical assets (GPS) [19, 42].

Traditional anti-jamming relies on spread spectrum techniques including Frequency-Hopping Spread Spectrum (FHSS) and Direct-Sequence Spread Spectrum (DSSS). These techniques aim to minimize the chance that the legitimate signal is interfered by the jamming signal (*jamming avoidance*). Nowadays, with the rapid development of RF hardware and the increasing prevalence of Software-Defined Radios (SDR) [9], high-power jammers that significantly degrade communications in the wide spectrum are easier to build. As *jamming avoidance* approaches become less effective against such jammers, the ability to remove/cancel the jamming component in the received signal (*jamming cancellation*) becomes important to maintain the SNR of the wireless link.

It is also important for the receiver to ***early*** detect the jammer, before the communication link is impacted. Most of prior work typically use statistics such as Signal-to-Noise Ratio or Packet Delivery Ratio [38, 47] which can only be acquired after decoding the samples, instead of using PHY layer information for early detection. In this paper, we are concerned with the design of a *unified jamming detection-cancellation* framework (JaX) to counter high-power jammers. This framework detects the jammer and infers the necessary *jamming cancellation* parameters directly from the raw RF samples in the PHY layer. It does not require *explicit probes, sounding, training sequences, channel estimation*, or the *cooperation* of the transmitter, and can promptly react to a wide variety of jammers (e.g., AWGN, intermittent, variable power) by *cancelling* their interference. Motivated by the recent success of Deep Convolutional Neural Network (CNN) in the RF domain [24, 26], we developed a multi-antenna DL-based system to infer the existence of interference, the number of interfering emissions and their respective phases. This information is continuously fed into an algorithm that cancels the interfering signal. We evaluate JaX in a variety of scenarios with multiple modulation schemes, demonstrating good

WiSec'23, May 29-June 1, 2023, Guildford, United Kingdom

ACM ISBN 978-1-4503-9859-6/23/05…$15.00
https://doi.org/0.1145/3558482.3590178



(a) Single-antenna receiver. (b) Two-antenna receiver.

Figure 1: Illustration of a jammer interfering with the communications between two nodes.

performance for a receiver exposed to a jammer that is two orders of magnitude higher than the legitimate signal. For instance, we demonstrate that a SDR-based receiver integrating our approach can detect the jammer with over 99% of accuracy and achieve high jamming-resistance with a Bit Error Rate (BER) as low as 10^{-6} while the jammer power is 19 dB higher than the legitimate signal. Moreover, our DL model was trained using only continuous jammer transmitting over cables and is also effective for intermittent and power-variable jammers in over-the-air indoor environments (discussed in Section 5). To the best of our knowledge, our work is the first in the literature that leverages DL for *unified jamming detection-cancellation*. Our contributions are summarized as follows:

- A novel anti-jamming approach JaX for *unified jamming detection-cancellation* using Deep Convolutional Neural Network.
- A neural network architecture, model, and supporting algorithms for high-accuracy multi-antenna jamming detection and cancellation using direct PHY layer I&Q RF samples.
- A two-antenna SDR prototype leveraging JaX's proposed CNN model and anti-jamming algorithms that are agnostic to the modulations (demonstrated with BPSK, QPSK, 8-PSK, 16-QAM).
- JaX is as effective as the optimal pilot-based approach [50] in time-varying channels with 50% less transmission overhead and without requiring link modifications for pilot support.
- JaX is evaluated for various types of jammers (i.e., continuous, intermittent, constant/variable power, AWGN, modulated) in different environments, demonstrating the jammer detection accuracy of over 99% and achieving a BER as low as 10^{-6} even against jammers that are 19 dB more powerful than the user.

2 PROBLEM STATEMENT AND APPROACH

2.1 Models

Sender and Receiver Model. We made a standard assumption that the legitimate nodes are communicating over a pre-agreed channel and link parameters, including the center frequency, bandwidth, modulation (Figure 1). Further jamming resilience can be achieved by randomizing such parameters (e.g., frequency hopping) but is not the focus of this work. The sender uses a single TX antenna, and the receiver uses two identical RX antennas. We assume that the nodes are *neither* aware of each other's location, nor the location of the jammer. We consider a slow-fading channel, therefore a low-mobility for the involved parties. This concretely means that the channel do not change abruptly within a packet (e.g., 1 ms).

Adversary Model. We consider an attacker (jammer) using a single antenna transmitting signals on the same channel as the users, interfering with the legitimate communications (Figure 1). We allow a powerful adversary that already knows the link parameters such as center frequency, bandwidth and potentially other settings. The jammer is allowed to transmit either random samples or modulated packets, with a continuous or intermittent pattern. We also assume a similar low-mobility pattern for the jammer so that the channel characteristics do not change abruptly within a packet.

Communication Channel Model. The communicating nodes can use arbitrary modulation and coding schemes and are exposed to the typical additive Gaussian noise (AWGN) in addition to the jammer interference. We assume the channel gains to be fairly stable throughout the considered bandwidth. In our evaluations, we consider differential BPSK, QPSK, 8-PSK, and 16-QAM modulations for the communicating nodes. We evaluate our system in both over-the-air setups, and using RF coax cable. The cable setup is for reproducibility and in order to systematically and extensively assess the performance of the approach over a range of three orders of magnitude of powers (35 dB), and multiple phase offsets.

2.2 Theoretical Foundations and Approach

Jamming Fundamentals. We model a received signal R that comprises the transmitting signal S, adjusted to account for the channel gain h, and additive white Gaussian noise N:

$$R = hS + N \tag{1}$$

In the absence of interference, the quality/capacity of such link is determined by the Signal-to-Noise Ratio (SNR), which is proportional to $\frac{|h|^2}{|N|^2}$ (where $|.|$ denotes the complex norm). Assuming that additive noise is constant over time, the SNR only depends on $|h|$. The receiver achieves better Bit Error Rate (BER) when the channel gain $|h|$ is higher, which is reflected in a higher SNR.

Now assume the presence of a jammer, who knows the frequency channel that the legitimate nodes are operating over. The adversary transmits a "jamming" signal J that interferes with the legitimate signal S (as shown in Figure 1a). The received signal now becomes:

$$R = h_S S + h_J J + N \tag{2}$$

where h_S and h_J are the channel gains corresponding to the sender and the jammer, respectively. The decodability of the legitimate signal is now dependent on the Signal-to-Interference-and-Noise Ratio (SINR) which is proportional to $\frac{|h_S|^2}{|h_J J|^2+|N|^2}$. When the jamming power is considerably high relatively to the channel noise, $|h_J J| \gg |N|$, the SINR can be approximated as proportional to the ratio $\frac{|h_S|^2}{|h_J J|^2}$. As the interference becomes stronger, $\frac{|h_S|}{|h_J J|}$ is subsequently smaller and the legitimate signal S becomes undecodable.

Approach. To remove the jamming component from the received signal R, in a single-antenna, is challenging without having control over the jammer, knowing the jamming signal, or resorting to other dimensions to evade the jammer (e.g., as in spread spectrum). Our approach instead relies on two receiving antennas, each collects a copy of the transmitted signals (subject to different channel gains):

$$\begin{aligned} R_1 &= h_{S_1} S + h_{J_1} J + N_1 \\ R_2 &= h_{S_2} S + h_{J_2} J + N_2 \end{aligned} \tag{3}$$

Table 1: Comparison with existing work on Deep Learning-based jamming detection and jamming cancellation. JaX is the first work in the literature that practically addresses unified jamming detection-cancellation leveraging Deep Learning.

	Jamming Detection	Jamming Cancellation		Deep Learning-based	Real Emission Evaluation
		Pilots/Reference Signals Exemption	Mechanical Jamming Dampening Exemption		
JaX	✓	✓	✓	✓	✓
BJM [50]	✓		✓		✓
Yan et al. [48]	✓		✓		✓
Vo-Huu et al. [43]		✓			✓
Zhang et al. [51]	✓	Not Applicable	Not Applicable	✓	
Li et al. [21]	✓	Not Applicable	Not Applicable	✓	✓

Considering a jammer significantly above the noise, the cancellation is achieved using the formula:

$$R_1 - p_1 R_2 = p_2 S \tag{4}$$

where $p_1 = \frac{h_{J_1}}{h_{J_2}}$, and $p_2 = h_{S_1} - p_1 h_{S_2}$. If the new gain p_2 of signal S is sufficiently large, we can decode S and achieve a good BER.

The main challenge is how to estimate parameter p_1 correctly. Here, we emphasize that traditional techniques used in MIMO systems estimate such parameter relying on probing, training sequences and sounding procedures (cooperatively between the transmitter and receiver). In the following, we will show that our approach can address the problem without requiring those cooperative/explicit mechanisms. We first reformulate p_1 in the polar representation $\frac{|h_{J_1}|}{|h_{J_2}|} e^{j(\phi_{J_1} - \phi_{J_2})}$. To find p_1, we are required to estimate the *amplitude ratio* $A_J = \frac{|h_{J_1}|}{|h_{J_2}|}$ and the *phase shift* Δ_{ϕ_J}:

$$\Delta_{\phi_J} = \phi_{J_1} - \phi_{J_2} \tag{5}$$

We use two different approaches to estimate A_J and Δ_{ϕ_J}. For the *amplitude ratio*, we rely on the fact that the parameter is proportional to the square root of the ratio of jamming power received at the antennas. On that account, we estimate using the measured power in the periods before and during the collision (Section 4.2). To estimate the *phase shift*, JaX uses a lightweight, yet powerful Convolutional Neural Network (CNN) that directly estimates from the I/Q RF samples. The ability of CNN to analyze and infer diverse complex data has been investigated and utilized in various areas [2, 18, 29, 37]. Estimating phase shift involves extracting and synthesizing low-level patterns of the original jamming signal embedded in the RF samples. In conventional wireless systems design, such patterns are extracted through signal processing filters, which makes the convolutional filters of CNN an ideal candidate for the task. Our CNN not only can disentangle the collision and estimate the phase shifts, but also can infer if the estimations correspond to transmitted signals or just noise. The latter allows us to detect the presence of jammer and distinguish between the three possible states of the channel: (1) When the channel is clear, (2) when only the jammer or the sender is transmitting, and (3) when the user nodes are being interfered with by the jammer. The workflow of a JaX-enabled RF receiver is illustrated in Figure 2. To the best of our knowledge, our work is the first that considers CNN as a multi-functional approach for detecting and cancelling jammers. More details of the approach are presented in Section 3 and Section 4.

JaX can also perform well under the impact of multi-path in indoor environments (see Section 5). This does not contradict the cancellation theory, as we can explain as follows. Under the impact of multi-path effects, each receiving antenna collects multiple copies of the legitimate and jamming signals:

$$\begin{aligned} R_1 &= \sum_i h^i_{S_1} S + \sum_i h^i_{J_1} J + N_1 \\ R_2 &= \sum_i h^i_{S_2} S + \sum_i h^i_{J_2} J + N_2 \end{aligned} \tag{6}$$

where $h^i_{S_k}$ and $h^i_{J_k}$ are the channel gains of the i^{th} path from the sender and the jammer to the antenna k of the receiver, respectively. We can see that by taking $h_{S_k} = \sum_i h^i_{S_k}$ and $h_{J_k} = \sum_i h^i_{J_k}$, Equation (6) becomes equivalent to Equation (3). As also explained in [43], the sum of the channel gains of all the paths from the sender/jammer to the receiver can be viewed as a new channel gain of the line-of-sight path between the receiver and the sender/jammer being put in a different location. Therefore, JaX is still effective to counter jammers in this scenario.

However, we note that multi-antenna jamming cancellation has intrinsic limitations. Even with accurate estimates of the amplitude ratio and phase shift, it is not guaranteed that jamming cancellation can fully recover the original signal. As we mentioned earlier, removing J, results in signal S being subject to an update gain value $h_{S_1} - p_1 h_{S_2}$ where $p_1 = \frac{h_{J_1}}{h_{J_2}}$. This gain becomes small when $\frac{h_{S_1}}{h_{S_2}} \approx \frac{h_{J_1}}{h_{J_2}}$, equivalently $\Delta_{\phi_S} \approx \Delta_{\phi_J}$ i.e. the separation between the phase shifts corresponding to channel gains of signals S and J is small:

$$Sep_{\Delta_\phi} = \left| \Delta_{\phi_S} - \Delta_{\phi_J} \right| \approx 0 \tag{7}$$

This derives from the intrinsic limitation of multi-antenna system to not be able to distinguish between two emitters that are aligned with the receiver. In the later sections, we will show the impact of parameter Sep_{Δ_ϕ} to the jamming cancellation approach through extensive experiments for Bit Error Rate evaluation.

3 JAMMING DETECTION AND PHASE SHIFT ESTIMATION

3.1 Challenges and Goals of The Design

Challenges. While developing the CNN model for phase shift estimation, we encountered two main challenges. ***First,*** with a single estimation, it is hard to guess whether the phase shift associates

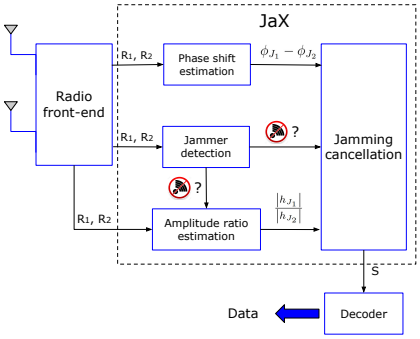


Figure 2: The workflow of a JaX-enabled RF receiver.

with the legitimate or the jamming signal. This is especially more confusing when an adversary tries to mimic the legitimate communication, e.g., by using the same modulation. As a result, in the worst case scenario, we can instead inadvertently cancel the legitimate signal. ***Second***, the receiver does not know the current state of the communication channel, i.e., how many transmitters are concurrently using it. This could lead to another catastrophic scenario when the sender is transmitting without being interfered, while the receiver still believes that a collision is happening and unintentionally removes the signal using the estimated phase shift.

Goals. We defined two goals to address the above challenges. ***First***, the phase shift estimations for both the legitimate signal S and the jamming signal J are required instead of only for the latter. This is intuitively possible to achieve with a CNN since signals S and J are typically non-coherent, therefore the unique features of both signals can be extracted by the convolutional filters. ***Second***, for each estimated phase shift, we also infer whether it is the estimation of a transmitted signal (S or J) or of noise. As such, the neural network outputs a confidence value along each inferred phase. This indicates a signal if the confidence is larger than 0.5, and indicates noise otherwise. These capabilities allow us to ***detect the jammer*** (i.e. *when both signal detection outputs indicate a signal*), and to avoid accidentally removing the legitimate signal.

3.2 Neural Network Architecture

Our development of the CNN started with defining the input layer. Naturally, we want to avoid feeding a very long stream of RF samples to the CNN at once due to the heavy computational cost. To address this, we divided the stream of I/Q samples into blocks of a fixed length M (In our implementation, $M = 128$ samples). Then, we transformed each block into a $2 \times M$ matrix where the first row comprises the In-phase (I) values and the second row comprises the Quadrature (Q) values of M RF samples (shown in Figure 3). Finally, we stacked the matrices of the two antennas to form the $2 \times M \times N$ real-valued tensor as the input of our CNN.

We have considered several possible architecture designs of the CNN and converged on an optimized CNN structure that achieves good performance in terms of processing speed and estimation correctness (see discussion in Section 5). The architecture of our CNN is illustrated in Figure 3, in which a stack of three convolutional layers with kernel size of 3×3 is followed by a 2×1 convolutional

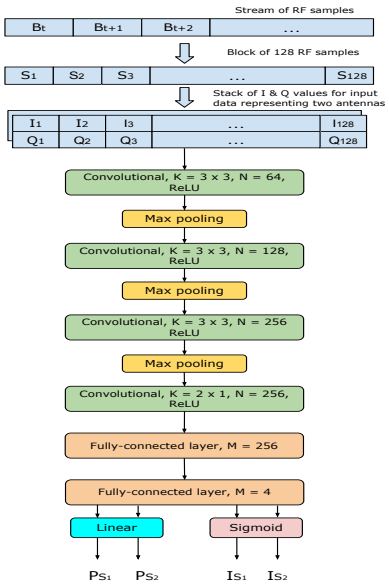


Figure 3: RF data representation and the CNN architecture for phase shift estimation and signal detection. K is the filter size and N is the number of filters in convolutional layers. M is the number of neurons in fully-connected layers.

layer and two fully-connected layers. 3×3 convolutional layer is a popular Deep Learning technique and has been used as the building block for state-of-the-art Deep Learning (DL) architecture such as VGG [37] and ResNet [15]. As 3×3 convolution is especially effective for extracting low-level features in different local regions of the data, we expected the neural network to find robust, unique features and patterns of the colliding signals from the variation in the amplitude and phase of contiguous RF samples. We added Max Pooling layers between convolutional layers to extract the important sharp features corresponding to the local max-values and to reduce the computation cost. On the other hand, we used the 2×1 convolutional layer with the sample-wise combining of I & Q channels to gain the high-level semantics of angular distance for estimating the phase shifts. Rectified Linear Unit (ReLU) activation was used for the convolutional layers because it is computationally efficient and more effective against the vanishing gradient problem [13].

The fully connected layers synthesize the features extracted from the previous convolutional layers for making predictions. The outputs P_{S_1} and P_{S_2} estimate the phase shifts for legitimate and interference signals, while I_{S_1} and I_{S_2} detect whether the corresponding phase shift estimations come from a signal or noise. (Again, 1 implies real signal while 0 implies noise). Sigmoid activation is used for I_{S_1} and I_{S_2} to limit the values in the range $[0, 1]$, while linear activation is used for P_{S_1} and P_{S_2}. We emphasize that as P_{S_1} and P_{S_2} cannot be used interchangeably, we distinguish them by having P_{S_1} learn the smaller phase shift while P_{S_2} learns the bigger one.

3.3 Data Collection

Having a large and carefully labeled dataset is a critical requirement to train a good neural network model. Unfortunately, data labeling is

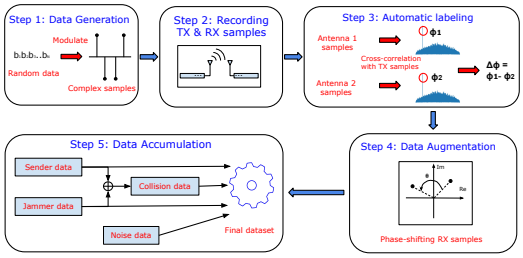


Figure 4: The procedure of building dataset for jamming detection and cancellation.

normally done by manual labor, which requires domain knowledge and takes significant amount of time and efforts. To address this challenge, we devised an efficient multi-stage approach to build a large training dataset for our CNN model, as depicted in Figure 4. In our setup, we have a single-antenna transmitter (TX) and a two-antenna receiver (RX). The TX can either act as a legitimate sender or a jammer. First, we generated random samples and saved in memory both at the TX and RX. Then, the TX transmits using the saved data samples and the RX records the received RF samples from each of the two antennas to files. Due to the channel effects, the received samples are rotated by some unknown phase shift (ϕ_1 and ϕ_2 for the two antennas as shown in Figure 4). To determine these values, we chunked the received samples and cross-correlate with the data samples already saved in the RX's memory from the first step. ϕ_1 and ϕ_2 were then computed from the angular values (argument) of the correlation outputs with the highest energy (peak). Next, we determined the phase shift Δ_{ϕ_S} (for the legitimate signal, or Δ_{ϕ_J} for the jamming signal) by taking the difference of the two angles. When the channel is static, these phase shifts will experience very little variance. Such dataset would negatively affect the training and bias the resulting model to a small range of output values. To address this, we shifted the phase of RF samples by a random value between $[-\pi, \pi]$ and adjust the labels accordingly. This resulted in a more diverse dataset. This process was performed for both the sender and the jammer. After that, we generated the data for collisions by adding the samples recorded for the sender and the jammer together, with the phase shifts Δ_{ϕ_S} and Δ_{ϕ_J} acquired from the previous process as the labels for phase shift estimation. We also collected data representing noise with the TX turned off.

The training dataset. Using the above data collection techniques, we built a dataset containing $5,450,312$ real-valued data tensors of size $2 \times 128 \times 2$ reflecting I/Q values of 128 RF samples collected by two antennas of the receiver. The receiver was connected to the sender and the jammer through coaxial cables. The transmitters and receiver were implemented on Ettus USRP B210 software-defined radios using GNURadio [1]. While the sender only transmitted modulated signals, the jammer could transmit either modulated signals or AWGN signals. The transmitting power is adjusted for varying Signal-to-Jamming ratio (SJR) between −25dB and 25dB. We note that while our model was trained on the dataset where the jammer emissions in each recording are continuous and have constant power, it can also perform very well against the intermittent and variable jammers (discussed in Section 5).

3.4 Loss Function and Training

Loss Function. During the training, our CNN aims to minimize a loss function which represents the errors of phase shift estimations and signal detections. For the phase shift estimations, we used a modified Mean Squared Error function:

$$\mathcal{L}_\phi = 1_{S_1}(\Delta_{\phi_1} - P_{S_1})^2 + 1_{S_2}(\Delta_{\phi_2} - P_{S_2})^2 \tag{8}$$

where $\Delta_{\phi_1}, \Delta_{\phi_2}$ are the ground truth values and P_{S_1}, P_{S_2} are the output estimations (shown in Figure 3). 1_{S_1} (resp. 1_{S_2}) is 1 if Δ_{ϕ_1} (resp. Δ_{ϕ_2}) associates with a signal, otherwise 0. For the signals detection, we used Binary Cross-Entropy loss function:

$$\begin{aligned}\mathcal{L}_S = -((1_{S_1}\log(I_{S_1}) + 0_{S_1}\log(1 - I_{S_1})) + \\ (1_{S_2}\log(I_{S_2}) + 0_{S_2}\log(1 - I_{S_2})))\end{aligned} \tag{9}$$

where I_{S_1}, I_{S_2} are the detection outputs associated respectively with P_{S_1}, P_{S_2}, while 0_{S_1} (resp. 0_{S_2}) is the complement of 1_{S_1} (resp. 1_{S_2}). The final loss function is the weighted sum of two loss components:

$$\mathcal{L} = \alpha\mathcal{L}_\phi + (1 - \alpha)\mathcal{L}_S \tag{10}$$

where α is the weighting parameter that balances the values of the components. Through the model validation process, we determined that $\alpha = 0.1$ provides the best results.

Training. After a large number of iterations for validation, our CNN is finalized for the training. We used PyTorch library [30] to develop the CNN model. To improve the training convergence and eliminate the needs for regularization, we utilized Batch Normalization [16] on the outputs of the convolutional layers. Furthermore, we minimized the possibility of overfitting by using a Learning Rate Decay technique [12] in which we lowered the learning rate when the validation error does not improve over a period of time, e.g., a few training epochs. We trained the CNN for 100 epochs, and chose the best model with the lowest validation loss. We emphasize that the phase shift estimations P_{S_1}, P_{S_2} are made distinguishable during training by assigning them to learn the smaller and bigger phase shifts, respectively.

4 JAMMING CANCELLATION

4.1 Analyze CNN Outputs

As described previously, at time period T the receiver collects a block of RF samples, which is fed to the CNN model to get $P^T_{S_1}, P^T_{S_2}, I^T_{S_1}, I^T_{S_2}$. $P^T_{S_i}$ represents the phase shift estimation for the current signal, while $I^T_{S_i}$ classifies the type of the corresponding phase shift, i.e., real signal or noise, with $i \in \{1, 2\}$. We remind the reader that we distinguish the estimations by the ordering $P^T_{S_1} < P^T_{S_2}$ as learned during the training. We define the *signal detection* indicator $1^T_{S_i}$ for corresponding estimation $P^T_{S_i}$ using the output $E^T_{S_i}$:

$$1^T_{S_i} = \begin{cases} 1 & E^T_{S_i} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \{1, 2\} \tag{11}$$

$1^T_{S_i}$ being equal to 1 or 0 indicates that S_i (whose phase shift is estimated by $P^T_{S_i}$) is a real signal or noise, respectively. We can therefore recognize the current state of the communication channel and subsequently acquire the correct phase shift for cancelling the jamming signal when collisions happen (as shown in Algorithm 1).

When only $1^T_{S_1}$ ***or*** $1^T_{S_2}$ ***is 1:*** This indicates that the channel is currently used by a single transmitter, which can be either the legitimate sender or the jammer. We identify the jammer by checking if the RF samples are decodable. In this case the capability of the jammer is limited to degrading the communications between the nodes by occupying the channel. If we identify the jammer's presence (samples are not decodable), the estimation $P^T_{S_i}$ where $1^T_{S_i} = 1$ signifies the jamming signal. In the case where the adversary transmits data that mimicks legitimate communications, a solution can consist of duplicating the receiver chain continuously tracking and decoding both inferred signals (at the expense of doubling the receiver cost). Smarter approaches are possible by tracking the phases of the transmitters of interest and canceling other ones.

Algorithm 1: CNN-based Jamming Cancellation

Data: $P^T_{S_1}, P^T_{S_2}, 1^T_{S_1}, 1^T_{S_2}, \Delta^{cur}_{\phi_J}$, RF samples at time period T
Result: $I^{last}_J, E_{J_i}, E_{S_i}$ with $i \in \{1,2\}$, RF samples
if $1^T_{S_1} \bigoplus 1^T_{S_2} = 1$ **then**
  Decode the RF samples;
  if *decodable* **then**
    Measure the signal power E^i_S with $i \in \{1,2\}$;
    $I^{last}_J \leftarrow 0$;
  else
    Measure the jamming power E^i_J with $i \in \{1,2\}$;
    Calculate Δ_{ϕ_J} using Equation (14) and update $\Delta^{cur}_{\phi_J}$;
    $I^{last}_J \leftarrow 1$;
  end
else if $1^T_{S_1} = 1$ ***and*** $1^T_{S_2} = 1$ **then**
  if $\left|P^T_{S_1} - \Delta^{cur}_{\phi_J}\right| < \left|P^T_{S_2} - \Delta^{cur}_{\phi_J}\right|$ **then**
    $\Delta^T_{\phi_J} \leftarrow P^t_{S_1}$;
  else
    $\Delta^T_{\phi_J} \leftarrow P^t_{S_2}$;
  end
  Calculate Δ_{ϕ_J} using Equation (14) and update $\Delta^{cur}_{\phi_J}$;
  Calculate amplitude ratio using Algorithm 2;
  Removing jamming signal by Equation (4);
else
  Skip the current period;
end

When both $1^T_{S_1}$ ***and*** $1^T_{S_2}$ ***are 1:*** In this case, we detect a collision indicating that the legitimate signal is being interfered with by a jammer. We identify the jamming phase shift $\Delta^T_{\phi_J}$ out of the two estimation outputs $P^T_{S_1}, P^T_{S_2}$ by calculating the distance to the existing estimation $\Delta^{cur}_{\phi_J}$ and picking the one with the closest distance. This is based on the prior assumption of slow-fading channel for our setup, where the phase shift varies slowly over time. Δ^{cur}_J is calculated and updated by a smoothing process on the history data of jamming phase shift estimation (Section 4.3).

We also need to estimate the amplitude ratio $A_J = \frac{|h_{J_1}|}{|h_{J_2}|}$, as discussed in Section 2.2. We introduce the intuition behind our approach. We analyze the power variation of RF samples in the periods before and during the collision, presented in details in Section 4.2. With the phase shift and amplitude ratio estimated, the receiver can solve Equation (4) with $p_1 = \frac{h_{J_1}}{h_{J_2}} = A_J e^{j\Delta_{\phi_J}}$ to null the jamming component in the received signal. The legitimate signal now has a new gain p_2 and can be used to decode the data. While p_2 is not necessary to estimate, as mentioned in Section 2.2, we note that the phase shift separation Sep_{Δ_ϕ} between signals S and J has a direct impact on the gain p_2 and subsequently the quality of the final signal. This impact will be evaluated in Section 5.

When both $1^T_{S_1}$ ***and*** $1^T_{S_2}$ ***are 0:*** This informs the receiver that neither communication nor jamming is happening in the channel and we can skip this period to avoid corrupting the estimation.

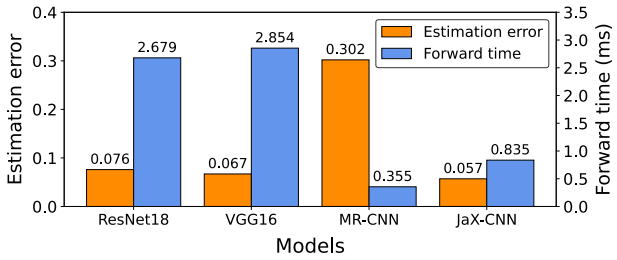


Figure 5: Comparison of CNN models for estimation error and network forward time.

4.2 Amplitude Ratio Estimation

Our amplitude ratio estimation algorithm is described in Algorithm 2. We note that Algorithm 2 is only triggered when a collision is detected and the amplitude ratio needs to be estimated for cancellation. Our approach is inspired by the observation that the signal power during a collision comprises two independent, separable components for the legitimate signal S and the jamming signal J. Suppose during time period T, the receiver collects N digital RF samples from the analog input of antenna i, the received power E^T_i can be written as:

$$E^T_i = \frac{1}{N}\sum_{t=1}^{N} |h^t_{S_i} S^t + h^t_{J_i} J^t|^2 \tag{12}$$

Given that the channel is slow-fading and not dependent on the instant time t, and because the sender's signal S and the jammer's signal J are uncorrelated, i.e. $\sum_{t=1}^N h^t_{S_i} S^t (h^t_{J_i} J^t)^* = 0$ and $\sum_{t=1}^N (h^t_{S_i} S^t)^* h^t_{J_i} J^t = 0$, Equation (12) becomes:

$$E^T_i = \frac{1}{N}(|h_{S_i}|^2 \sum_{t=1}^{N} |S^t|^2) + \frac{1}{N}(|h_{J_i}|^2 \sum_{t=1}^{N} |J^t|^2) = E_{S_i} + E_{J_i} \tag{13}$$

To estimate $A_J = \frac{|h_{J_1}|}{|h_{J_2}|} = \sqrt{\frac{E_{J_1}}{E_{J_2}}}$ (where $E_{J_i} = \frac{1}{N}(|h_{J_i}|^2 \sum_{t=1}^N |J^t|^2)$), we need to measure $E_{S_i} = \frac{1}{N}(|h_{S_i}|^2 \sum_{t=1}^N |S^t|^2)$ for two antennas $i \in \{1,2\}$. To do this, we first determine whether the legitimate sender or the jammer transmits first, right before the collision (using the detection capability described in Section 4.1), then calculate the power accordingly. Here, we assume the sender's power is stable during the transmission of a packet (slow fading channel), so the measurement of E_{S_i} at the beginning of the collision can be used until the end of that collision. Our algorithm looks at period $T-1$ right before the collision, and identifies the transmitter in that

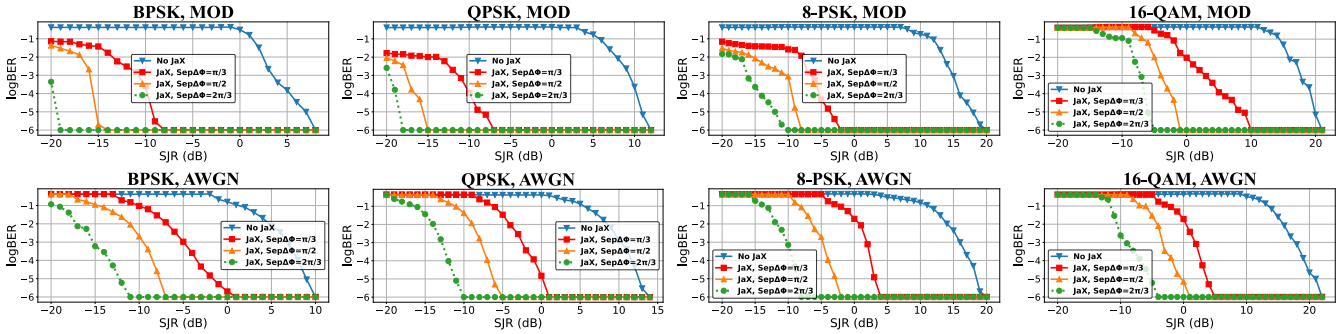


Figure 6: Bit Error Rate evaluation for over-the-cables experiments with MOD jammer (1^{st} row) and AWGN jammer (2^{nd} row).

period with parameter I_J^{last} (which is updated in Algorithm 1 and utilized in Algorithm 2). If the sender appears in period $T-1$ (where $I_J^{last} = 0$), we can measure E_{S_i} and calculate A_J. Otherwise, we know that the jammer appears in period $T-1$, and we can measure E_{J_i} and update E_{S_i} with the current power E_i. It is noted that in the latter case, the new E_{S_i} is used until the end of the collision and is not updated again if we detect a collision in the previous period.

Algorithm 2: Amplitude Ratio Estimation

Data: $E_{J_i}, E_{S_i}, 1_{S_i}^{T-1}$ with $i \in \{1,2\}$, I_J^{last}, RF samples at T
Result: A_J
Measure the current power E_i^T with $i \in \{1,2\}$;
if $I_J^{last} = 1$ ***and*** $1_{S_1}^{T-1} \bigoplus 1_{S_2}^{T-1} = 1$ **then**
  $E_{S_1} \leftarrow E_1^T - E_{J_1}$;
  $E_{S_2} \leftarrow E_2^T - E_{J_2}$;
end
$A_J \leftarrow \sqrt{\frac{E_1^T - E_{S_1}}{E_2^T - E_{S_2}}}$

4.3 Estimation Smoothing

Unlike the estimations $I_{S_1}^T, I_{S_2}^T$ which are discretized to the values of 0 and 1 for the signal detections, the real-valued $P_{S_1}^T, P_{S_2}^T$ are used directly to solve the jamming cancellation equation. This makes the cancellation process susceptible to the neural network's estimation variations and outliers [4]. We improve the robustness of the phase estimation and the subsequent jamming cancellation by stabilizing the estimations with the exponential smoothing function:

$$\Delta_{\phi_J} = \Delta_{\phi_J}^T \lambda + \Delta_{\phi_J}^{cur} (1 - \lambda) \tag{14}$$

where λ controls the smoothness of the output. We note that after performing the cancellation for the current period, $\Delta_{\phi_J}^{cur}$ is updated to the current value of Δ_{ϕ_J}. The effectiveness of the smoothing algorithm and λ parameter is discussed in Section 5.

5 EVALUATION

We split the dataset acquired in Section 3.3 into three parts used for training, validation, and testing with the ratio 0.64 : 0.16 : 0.2, respectively. Despite being trained on a single setting of jammer (continuous and constant power) and environment (over-the-cables), JaX is demonstrated to be resilient against other types of jammers in over-the-cables and over-the-air indoor environments.

5.1 Comparison with Other Neural Networks

We designed the CNN architecture with the goal of achieving good performance for both estimation correctness and processing speed. We validated our design by comparing with existing CNN models using the estimation error (defined by Equation (10)) and network forward time metrics (i.e., the elapsed time from when the network receives data to when it outputs estimations). In our setup, we trained our CNN model (called JaX-CNN), VGG16 [37], ResNet [15], and MR-CNN [29] and evaluated on our developed jamming detection-cancellation dataset. The models were developed using the Pytorch library [30] and CUDA [28] Version 10.2 running on a NVIDIA GeForce GTX 1080 GPU. We used Stochastic Gradient Descent optimizer with momentum [41] and `ReduceLROnPlateau` learning rate scheduler [49] with initial learning rate $lr = 0.005$ to train the models. To benchmark the forward time, we used Pytorch's `torch.cuda.synchronize` wrapping around neural network's forward propagation function to synchronize CUDA operations for accurate timing measurement. The final test loss and forward time were achieved by averaging over 20,000 iterated measurements, and illustrated in Figure 5. It is clear that our JaX-CNN model outperforms the other models with the test error of 0.057, lower than both ResNet18 (0.076) and VGG16 (0.067). Furthermore, *JaX-CNN is* 3.2 *times faster than ResNet and* 3.4 *times faster than VGG.* MR-CNN is faster than the other models, however, it suffers from a very high estimation error of 0.302, over 5 times higher than JaX-CNN. JaX-CNN achieves very good performance both in terms of speed and accuracy for this task and is more suitable than the other models to deploy for real-time and embedded applications.

5.2 Comparison with Existing Work

The comprehensive comparison between JaX and existing work for jamming detection and cancellation abilities is summarized in Table 1. In [48, 50], pilot signals known by the receiver are inserted into the transmitted signals to estimate channel gains and cancel jamming for multi-antenna receiver. However, this approach introduces several limitations: First, it causes additional overhead to the communications for transmitting the pilots. Second, the estimation

accuracy typically degrades in time-varying channels. Third, it requires compatibility between the transmitter and receiver to agree on when, what, and how the pilots are transmitted. In contrast, JaX does not require pilot signals, and achieves high jamming resilience against high-power jammers. To highlight those advantages, we evaluated and compared JaX with the BJM approach [50] which is the closest existing work in the literature. BJM leverages pilot signals to provide optimal data decoding at the receiver with minimal Mean Square Error. Figure 7 shows the performance of these approaches over a realistic time-varying channel (Rayleigh) with multipath fading and uncontrolled phase alignment between the jammer and sender. We simulated the Rayleigh channel using MATLAB software and compared JaX to BJM with different levels of pilot utilization, up to 50% of total transmitted data (equivalent to 50% of communication overhead). Our results demonstrate the advantages of JaX over pilot-based systems. JaX outperforms BJM with 20% pilot overhead and below, and is as good as BJM using 50% transmitted signals for pilots.

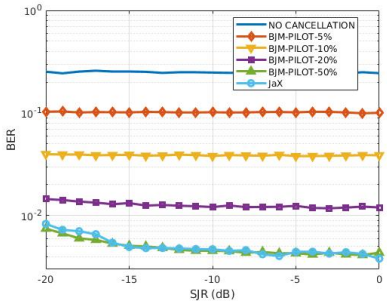


Figure 7: Comparison of JaX and BJM [50] with different levels of pilot overhead over a time-varying channel (Rayleigh).

In [43], the authors propose a hybrid mechanical-software jamming cancellation approach that does not require pilot signals. Nonetheless, the effectiveness of this approach heavily relies on the sophisticated mechanical antenna steering mechanism to dampen the jamming signal before the cancellation can take place. In contrast, JaX can perform jamming cancellation without the need for mechanical jamming dampening, even when the jamming signal is strong. Deep Learning is recently utilized in several works for **jamming detection** in simulation settings [51] and with real emissions in [21]. Nonetheless, Deep Learning for **jamming cancellation** remains unexplored. To the best of our knowledge, JaX is the first work that addresses the *unified jamming detection-cancellation* problem using Deep Learning.

5.3 Over-The-Cables Evaluation

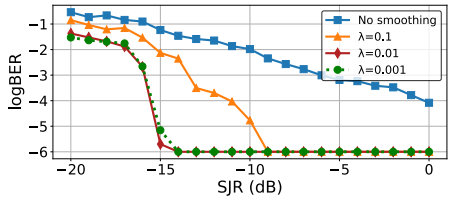
First, we evaluate the efficiency of JaX's jamming cancellation approach in a relatively idealistic environment where RF signals propagate through coaxial cables, thus multi-path and other fading effects are absent. However, this setup is reproducible and enables a fine grain experimental extensive evaluation of performance. Our setup comprises a sender, a receiver and a jammer, where the sender transmits modulated signals using differential BPSK, QPSK, 8-PSK and 16-QAM. For this experiment, we used two types of jammer: **MOD jammer** transmits modulated signals (with the same modulations as the sender), and **AWGN jammer** transmits Additive White Gaussian Noise signals. Both sender and jammer have continuous transmissions. The efficiency of jamming cancellation is measured by the Bit Error Rate metric, which we calculated by comparing and counting the error bits between the sent and received signals.

In Section 2.2 we showed that the phase shift separation $Sep_{\Delta_\phi} = \left|\Delta_{\phi_S} - \Delta_{\phi_J}\right|$ being very small can cause negative effects to the legitimate signal even when the jamming signal is completely removed. In our experiment, the transmitters were connected to the sender by identical coaxial cables, in which $Sep_{\Delta_\phi} \approx 0$. To get different values of Sep_{Δ_ϕ}, we introduce an artificial channel effect by shifting the phases of RF samples. Depending on the shifting, Sep_{Δ_ϕ} will receive a different value. We discuss the impact of Sep_{Δ_ϕ} on the efficiency of the jamming cancellation in the evaluation below.

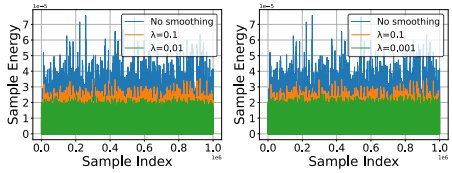
5.3.1 Impact of Phase Separation and Jammer Type. We evaluate JaX against two types of jammer:

MOD jammer. The first row of Figure 6 shows the Bit Error Rate (BER) evaluation considering four cases: No jamming cancellation (No JaX) is applied, and jamming cancellation is applied with three values of Sep_{Δ_ϕ} : $\frac{\pi}{3}$, $\frac{\pi}{2}$, and $\frac{2\pi}{3}$. In this evaluation, the jamming cancellation algorithm uses the estimation smoothing with parameter $\lambda = 0.01$. First, it is clear to see that our cancellation approach can achieve very high jamming resistance against the MOD jammer: It allows the receiver to operate at BER of 10^{-6} with the Signal-to-Jamming Ratio (SJR) of −19dB (i.e., the jammer is 79 times more powerful than the legitimate signal) for BPSK with $Sep_{\Delta_\phi} = \frac{2\pi}{3}$. The cancellation also achieves BER= 10^{-6} for QPSK under SJR= −18dB and $Sep_{\Delta_\phi} = \frac{2\pi}{3}$. Interestingly, *when compared with the case of no cancellation, our approach achieves up to* 30*dB gain when operating at a BER of under* 10^{-4} (the best result is 30dB for QPSK and 8-PSK, while for BPSK and 16-QAM, we achieve 25dB gain). In addition, we also see that the jamming cancellation performs better as Sep_{Δ_ϕ} gets bigger. For instance, the jamming resistance when operating at a BER of 10^{-6} with BPSK modulation drops by 5dB when Sep_{Δ_ϕ} decreases from $\frac{2\pi}{3}$ to $\frac{\pi}{2}$, and by 11dB when it decreases to $\frac{\pi}{3}$. It is easy to see the same trend for the other modulations. This limitation is intrinsic to multi-antenna jamming cancellation as the receiver cannot resolve two transmitter that are aligned with it. Also, our jamming efficiency when using 8-PSK and 16-QAM is lower compared to BPSK and QPSK, which is expected because they have smaller distance between the constellation points and thus are more prone to bit errors [33].

AWGN jammer. The performance of JaX against AWGN jammer is shown in the second row of Figure 6 with four settings: No JaX, and JaX with $Sep_{\Delta_\phi} = \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}$. JaX reduces the BER down to 10^{-6} under a SJR= −11dB for BPSK and −10dB for QPSK (with $Sep_{\Delta_\phi} = \frac{2\pi}{3}$). The efficiency of JaX for BPSK and QPSK reduces by 8dB compared to the case of MOD jammer. Meanwhile, the performance for 8-PSK and 16-QAM has minimal changes: JaX decreases the BER to 10^{-6} under a $SJR = -8$dB for 8-PSK and −4dB for 16-QAM, 2dB and 1dB less efficient compared to the MOD jammer case, respectively. Furthermore, it is clear that JaX's jamming resistance declines by 5dB as Sep_{Δ_ϕ} decreases to $\frac{\pi}{2}$ and by 10 − 12dB when it



(a) Bit Error Rate.



(b) Energy variation.

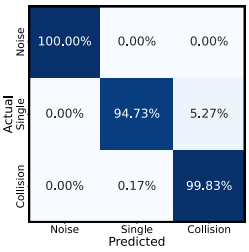
Figure 8: Estimation smoothing reduces energy fluctuation from cancellation and improves Bit Error Rate. Decreasing λ from 0.01 to 0.001 does not further improve the energy variation (Plot (b): $\lambda = 0.01$ (left) has similar variation to $\lambda = 0.001$ (right)). Similarly, for the Bit Error Rate (Plot (a)).

decreases to $\frac{\pi}{3}$. This is similar to JaX's performance against MOD jammer. Moreover, JaX still achieves a gain of up to 27dB (as observed in 8-PSK) at BER of below 10^{-4}, compared to no-cancellation. The fact that AWGN is better at jamming than modulated signals is consistent with information-theoretic results [7].

5.3.2 Impact of Estimation Smoothing. We investigate the impact of phase shift estimation smoothing in JaX with the Bit Error Rate evaluation shown in Figure 8a. In this case, we use BPSK signals and $Sep_{\Delta_\phi} = \frac{\pi}{2}$. It is clear that the smoothing significantly improves our system: We achieve 11 and 15 dB gain with $\lambda = 0.1$ and 0.01 for BER below 10^{-4}, respectively. This effect can also be seen in Figure 8b, where the estimation smoothing helps stabilizing the energy of the samples and reduce both the degree and the frequency of energy variation, resulting in lower BER for the same level of SJR. Finally, setting λ to 0.01 makes the energy more stable and yields better performance (-14 dB of SJR compared to -9dB for $\lambda = 0.1$ at BER=10^{-6}, while decreasing λ to 0.001 does not improve further. Therefore, we selected $\lambda = 0.01$ for all later evaluations.

5.4 Over-The-Air Evaluation

5.4.1 Jammer Detection. We conducted over-the-air experiments to assess JaX's ability in a more practical environment that is different from the one used for training, i.e., model trained on data recorded through cables is evaluated for over-the-air without retraining. Similar to over-the-cables experiments, our setup consists of a sender, a receiver, and a jammer. The sender transmits modulated signals, while the jammer transmits either modulated signals (MOD jammer) or AWGN signals (AWGN jammer). The testbed was positioned in an indoor environment, where there were common RF-blocking and reflecting objects such as computer, monitor, walls, and desks. To evaluate the detection capability, we focus on the classification of three channel states: (1) When there is no transmission and the channel is clear, (2) when there is a single transmission,



Accuracy	98.19%
Precision	99.02%
Recall	98.19%
F1-Score	98.58%

Figure 9: Channel classification results with confusion matrix (left) and various other metrics (right).

and (3) when there are two transmissions (from the sender and the jammer) causing collisions. We note that in the second case, the transmitter being the sender or the jammer is decided by the decoding check in Algorithm 1. Figure 9 depicts the classification results, in which the CNN classifies three states: Noise (no transmission), Single (one transmission) and Collision (two transmissions). Our CNN model achieves 98.19% accuracy, where the prediction accuracy is over 99% for Collision state and Noise state and is 94.73% for Single state. We also get high scores for other metrics, over 98% for both Precision, Recall and F1-Score. The results justify the capability of our CNN model to identify the current channel state, and the presence of jammer (by recognizing collisions with 99.83% accuracy) in the realistic environment without the needs to retrain the model trained in the idealistic environment (i.e. coaxial cables).

5.4.2 Jamming Resilience. We designed testbed configurations in the indoor over-the-air environment for different types of jammers. The distinctive features of wireless channels for each configuration was created by the random positioning of the jammer. The locations of sender and receiver were fixed in all configurations. It is noted that our focus in the over-the-air experiments is to evaluate how JaX is effective in cancelling different types of jammer operating in different configurations, compared to when it is disabled.

Over-the-air wireless channels can introduce undesirable artifacts for reproducible evaluations, such as fading or other third-party interference. A short burst of errors may have significant impact to the resulted BER especially when the target is very low (e.g., a burst of only 100 bit-errors can raise the BER to 10^{-4} when we evaluate a reception of one million bits). Therefore, we used Packet Loss Rate (PLR) as the over-the-air evaluation metric for jamming resilience to eliminate such flakiness. The data is chunked into 16-bytes packets, and a packet is lost when it has at least one bit received incorrectly (complete wireless links incorporate error correction codes to address such errors).

MOD and AWGN jammers. We designed four testbed configurations for MOD jammer (MOD_1 to MOD_4) and four configurations for AWGN jammer ($AWGN_1$ to $AWGN_4$). In the experiments, both legitimate and jamming signals were transmitted continuously with a constant power over time. Figure 10 shows the over-the-air PLR when JaX is enabled compared to when it is disabled, in the presence of a MOD or AWGN jammer. The transmitted packets are modulated with a mix of the considered modulations (BPSK, QPSK, 8-PSK, and 16-QAM). It is clear that JaX significantly reduces the PLR against the high-power jammers. In most configurations, JaX

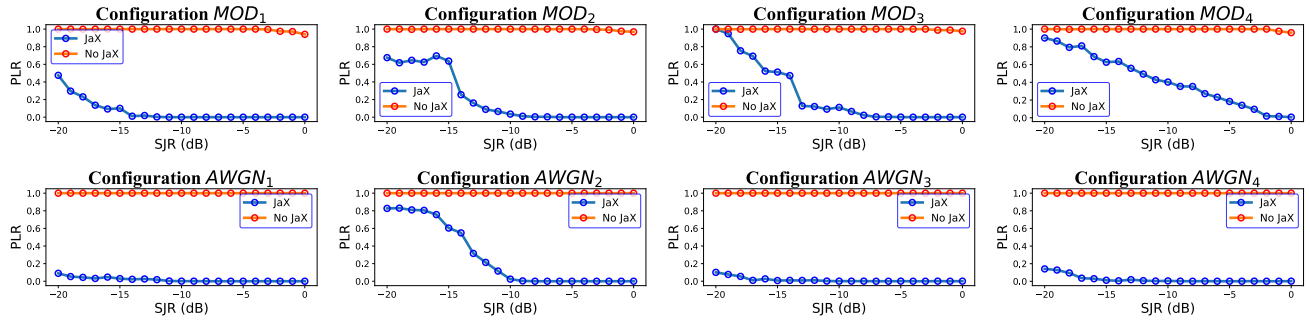
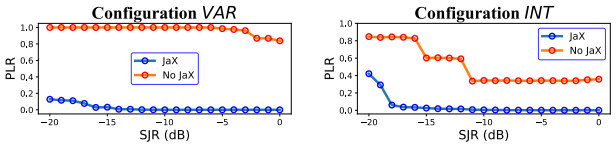


Figure 10: Performance of JaX in over-the-air settings against MOD (first row) and AWGN (second row) jammers. There are four sets of experiments for each jammer type, each set uses one configuration of positioning the sender, receiver, and jammer.



(a) Variable jammer **(b) Intermittent jammer**

Figure 11: Performance of JaX in over-the-air settings against variable and intermittent jammers.

can maintain a PLR below 0.1 when the Signal-to-Jamming Ratio is less than −10dB, i.e., the jamming power is at least 10 times higher than the TX power of user. Especially in $AWGN_1, AWGN_3$ and $AWGN_4$, JaX achieves such resilience under SJR= −17dB. Without JaX, the received PLR cannot get to below 0.9 even when SJR= 0dB (i.e., when the sender and the jammer use equal power).

Power-variable jammer. To see how JaX performs against power-variable jammer, we modified the jammer in the previous experiments to change the transmitting power randomly within a 10dB range around the averaged jamming power for every 1000 bytes transmitted in each run. It is noted that the power of the sender remained constant. The evaluation result in Figure 11a shows that JaX is able to adapt to jamming power variation and cancel up to 17dB of average jamming power to maintain a PLR below 0.1.

Intermittent jammer. We implemented an intermittent jammer that transmits signals periodically and rests for 0.1 seconds for every 2000 bytes transmitted. JaX counters intermittent jammers with the ability to classify the current state of the spectrum and identify the time slots where the jammer is present. Thanks to that, JaX can maintain very low PLR against a jammer 63 times stronger than the legitimate sender (with SJR=−18dB) while the receiver without JaX suffers much higher PLR against such jammer.

6 DISCUSSION AND RELATED WORK

One limitation of JaX, as is the case for any multi-antenna jamming cancellation systems, is that its performance degrades when the emitters are phase-aligned. The next challenge is how to ensure a desirable phase shift separation to distinguish between two emitters and cancel the unwanted signal. To address this, one can consider a larger antenna array, potentially distributed, to exploit the diversity of multi-antenna and enhance the robustness of cancellation. Exploring DL techniques for distributed antenna arrays for robust anti-jamming would be an interesting future direction.

Traditional anti-jamming at the physical layer has been relying on spread spectrum techniques, which require the coordinating nodes to pre-share a secret key. Recent research efforts have addressed that limitation for FHSS [20, 39, 40, 45], or DSSS [23, 32], or both [17]. Nonetheless, these approaches are designed with the specific goal to remove the pre-shared secret for spread spectrum and not to counter powerful jammers, i.e., a few orders stronger than the sending node while maintaining high spectral efficiency.

Significant research efforts focused on mitigating jammers at higher layers such as MAC [3, 35], network layer [8], cross-layer [5] or timing channel between datalink and network layers [46]. Nonetheless, the need for an efficient, resilient anti-jamming technique for physical layer security is still very important because of the fact that high-power jammers are increasingly easy to build nowadays.

Advances in Deep Learning have been utilized in some recent anti-jamming research. Most of those works are inspired by Deep Reinforcement Learning to find an optimal sequence of actions, such as changing spread spectrum parameters (e.g., frequency hopping) or coding schemes, to minimize the probability that the communication is impacted by the jammer [14, 22]. These approaches are designed with the main goal of avoiding jamming emissions (*jamming avoidance*). However, they do not counter high-power jammers successfully interfering with the communications. In such scenario, cancelling jammer is essential to maintain the quality of the communications. With that goal, we developed JaX, to the best of our knowledge, the first work in the literature that addresses DL-based *unified jamming detection-cancellation*. Our inspiration for CNN comes from the successful applications in various tasks of wireless communications [24, 27, 29] and security applications [25, 51]. JaX can detect the presence of a jammer emission with over 99% accuracy and enhance the RF receiver to reduce the Bit Error Rate to as low as 10^{-6} against a jammer whose transmitting power is 19 dB stronger than the legitimate sender.

ACKNOWLEDGMENTS

This work was partially supported by grants NAVY/N00014-20-1-2124, NCAE-Cyber Research Program, and NSF/DGE-1661532.

REFERENCES

[1] 2021. GNURadio. https://www.gnuradio.org

[2] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 10 (2014), 1533–1545. https://doi.org/10.1109/TASLP.2014.2339736

[3] Baruch Awerbuch, Andrea Richa, and Christian Scheideler. 2008. A jamming-resistant MAC protocol for single-hop wireless networks. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*. 45–54.

[4] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[5] Jerry T Chiang and Yih-Chun Hu. 2010. Cross-layer jamming detection and mitigation in wireless broadcast networks. *IEEE/ACM Transactions on networking* 19, 1 (2010), 286–298.

[6] CISCO Meraki. 2018. SNR and Wireless Signal Strength. https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_(SNR)_and_Wireless_Signal_Strength.

[7] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. Wiley.

[8] Jing Dong, Reza Curtmola, and Cristina Nita-Rotaru. 2009. Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks. In *Proceedings of the second ACM conference on Wireless network security*. 111–122.

[9] Ettus Research. 2019. USRP: Universal software radio peripheral. https://www.ettus.com/.

[10] FCC. 2020. Jammer Enforcement. https://www.fcc.gov/general/jammer-enforcement.

[11] FCC. 2022. FCC Upholds Fine for Jammer Used to Block Workers' Phone Use. https://www.fcc.gov/document/fcc-upholds-fine-jammer-used-block-workers-phone-use.

[12] Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. 2019. The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares. In *Advances in Neural Information Processing Systems*.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*. 315–323.

[14] Guoan Han, Liang Xiao, and H. Vincent Poor. 2017. Two-dimensional anti-jamming communication based on deep reinforcement learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2087–2091. https://doi.org/10.1109/ICASSP.2017.7952524

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning Image Recognition. In *CVPR'16*.

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*.

[17] Tao Jin, Guevara Noubir, and Bishal Thapa. 2009. Zero Pre-Shared Secret Key Establishment in the Presence of Jammers. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New Orleans, LA, USA) *(MobiHoc '09)*. Association for Computing Machinery, New York, NY, USA, 219–228. https://doi.org/10.1145/1530748.1530779

[18] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. arXiv:1404.2188 [cs.CL]

[19] Andrew J Kerns, Daniel P Shepard, Jahshan A Bhatti, and Todd E Humphreys. 2014. Unmanned aircraft capture and control via GPS spoofing. *Journal of Field Robotics* 31, 4 (2014), 617–636.

[20] Loukas Lazos, Sisi Liu, and Marwan Krunz. 2009. Mitigating Control-Channel Jamming Attacks in Multi-Channel Ad Hoc Networks. In *Proceedings of the Second ACM Conference on Wireless Network Security* (Zurich, Switzerland) *(WiSec '09)*. Association for Computing Machinery, New York, NY, USA, 169–180. https://doi.org/10.1145/1514274.1514299

[21] Yuchen Li, Jered Pawlak, Joshua Price, Khair Al Shamaileh, Quamar Niyaz, Sidike Paheding, and Vijay Devabhaktuni. 2022. Jamming Detection and Classification in OFDM-Based UAVs via Feature-and Spectrogram-Tailored Machine Learning. *IEEE Access* 10 (2022), 16859–16870.

[22] Songyi Liu, Yifan Xu, Xueqiang Chen, Ximing Wang, Meng Wang, Wen Li, Yangyang Li, and Yuhua Xu. 2019. Pattern-aware intelligent anti-jamming communication: A sequential deep reinforcement learning approach. *IEEE Access* 7 (2019), 169204–169216.

[23] Yao Liu, Peng Ning, Huaiyu Dai, and An Liu. 2010. Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication. In *2010 Proceedings IEEE INFOCOM*. 1–9. https://doi.org/10.1109/INFCOM.2010.5462156

[24] Hai N Nguyen and Guevara Noubir. 2022. Universal Beamforming: A Deep RFML Approach. In *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. 165–172.

[25] Hai N. Nguyen, Tien Dang Vo-Huu, Triet Vo-Huu, and Guevara Noubir. 2019. Towards Adversarial and Unintentional Collisions Detection Using Deep Learning. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning, WiseML@WiSec 2019, Miami, Florida, USA, May 14, 2019*. 22–24.

[26] Hai N Nguyen, Marinos Vomvas, Triet Vo-Huu, and Guevara Noubir. 2021. Wideband, Real-time Spectro-Temporal RF Identification. In *Proceedings of the 19th ACM International Symposium on Mobility Management and Wireless Access*. 77–86.

[27] Hai N Nguyen, Marinos Vomvas, Triet D Vo-Huu, and Guevara Noubir. 2023. WRIST: Wideband, Real-time, Spectro-Temporal RF Identification System using Deep Learning. *IEEE Transactions on Mobile Computing* (2023).

[28] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. 2020. CUDA, release: 10.2.89. https://developer.nvidia.com/cuda-toolkit

[29] Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. 2016. Convolutional Radio Modulation Recognition Networks. In *Engineering Applications of Neural Networks*.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Marin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 8024–8035.

[31] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V Krishnamurthy. 2010. Denial of service attacks in wireless networks: The case of jammers. *IEEE Communications surveys & tutorials* 13, 2 (2010), 245–257.

[32] Christina Pöpper, Mario Strasser, and Srdjan Capkun. 2009. Jamming-resistant broadcast communication without shared keys.. In *USENIX security Symposium*. 231–248.

[33] John G Proakis and Masoud Salehi. 2008. Digital communications.

[34] Project Zero. 2017. Over The Air: Exploiting Broadcom's Wi-Fi Stack. https://www.crowdsupply.com/fairwaves/xtrx

[35] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. 2010. A jamming-resistant mac protocol for multi-hop wireless networks. In *International Symposium on Distributed Computing*. Springer, 179–193.

[36] M. Schulz, D. Wegemer, and M. Hollick. 2016. DEMO: Using NexMon, the C-based WiFi firmware modification framework. In *Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '16)*. ACM.

[37] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. In *arXiv:1409.1556*.

[38] Michael Spuhler, Domenico Giustiniano, Vincent Lenders, Matthias Wilhelm, and Jens B Schmitt. 2014. Detection of reactive jamming in DSSS-based wireless communications. *IEEE Transactions on Wireless Communications* 13, 3 (2014), 1593–1603.

[39] Mario Strasser, Christina Popper, Srdjan Capkun, and Mario Cagalj. 2008. Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. 64–78. https://doi.org/10.1109/SP.2008.9

[40] Mario Strasser, Christina Pöpper, and Srdjan Čapkun. 2009. Efficient Uncoordinated FHSS Anti-Jamming Communication. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New Orleans, LA, USA) *(MobiHoc '09)*. Association for Computing Machinery, New York, NY, USA, 207–218. https://doi.org/10.1145/1530748.1530778

[41] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*. PMLR, 1139–1147.

[42] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the Requirements for Successful GPS Spoofing Attacks. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*.

[43] Triet D. Vo-Huu, Erik-Oliver Blass, and Guevara Noubir. 2013. Counter-Jamming Using Mixed Mechanical and Software Interference Cancellation. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks* (Budapest, Hungary) *(WiSec '13)*. Association for Computing Machinery, New York, NY, USA, 31–42. https://doi.org/10.1145/2462096.2462103

[44] Matthias Wilhelm, Ivan Martinovic, Jens B. Schmitt, and Vincent Lenders. 2011. Short paper: reactive jamming in wireless networks: how realistic is the threat?. In *Proceedings of the fourth ACM conference on Wireless network security* (Hamburg, Germany) *(WiSec '11)*. ACM, New York, NY, USA, 47–52. https://doi.org/10.1145/1998412.1998422

[45] Liang Xiao, Huaiyu Dai, and Peng Ning. 2012. Jamming-Resistant Collaborative Broadcast Using Uncoordinated Frequency Hopping. *IEEE Transactions on Information Forensics and Security* 7, 1 (2012), 297–309. https://doi.org/10.1109/TIFS.2011.2165948

[46] Wenyuan Xu, Wade Trappe, and Yanyong Zhang. 2008. Anti-jamming timing channels for wireless networks. In *Proceedings of the first ACM conference on Wireless network security*. 203–213.

[47] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. 2005. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. 46–57.

[48] Qiben Yan, Huacheng Zeng, Tingting Jiang, Ming Li, Wenjing Lou, and Y. Thomas Hou. 2014. MIMO-based jamming resilient communication in wireless networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 2697–2706. https://doi.org/10.1109/INFOCOM.2014.6848218

[49] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. 2018. Adaptive Methods for Nonconvex Optimization. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[50] Huacheng Zeng, Chen Cao, Hongxiang Li, and Qiben Yan. 2017. Enabling jamming-resistant communications in wireless MIMO networks. In *2017 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.

[51] Yu Zhang, Bo Jiu, Penghui Wang, Hongwei Liu, and Siyuan Liang. 2021. An end-to-end anti-jamming target detection method based on CNN. *IEEE Sensors Journal* 21, 19 (2021), 21817–21828.