Detection and Classification of Smart Jamming in Wi-Fi Networks Using Machine Learning

Zhengguang Zhang and Marwan Krunz
Department of Electrical and Computer Engineering, University of Arizona, USA
{zhengguangzhang,krunz}@arizona.edu

***Abstract*—Smart adversaries can exploit the publicly known frame structure of OFDM-based Wi-Fi protocols to disrupt communications by strategically jamming specific time samples or specific subcarriers. Such attacks are very difficult to detect by traditional techniques like spectral analysis and signal strength indicators. Machine learning (ML) based methods have been proposed to tackle this problem. However, existing ML methods are computationally intensive and perform well only at low signal-to-jamming power ratios (SJRs). In this paper, we propose a computationally efficient deep convolutional neural network (DCNN) consisting of only four convolution layers to detect and classify several smart jamming attacks in Wi-Fi networks. To deal with the time-frequency selectivity of smart jamming, we apply the continuous wavelet transform (CWT) to partially overlapped segments of the received I/Q samples to extract features. The scalogram of the CWT is used as input to the DCNN. We focus on three smart jamming attacks: preamble jamming, pilot jamming, and interleaving jamming. These attacks share similar characteristics, making their differentiation particularly challenging. Our proposed classifier achieves high accuracy in detecting and classifying these jamming attacks across a range of SJRs, from -6 dB to 15 dB, with an overall classification accuracy of 98%. Even at high SJR levels, the accuracy remains high at around 90%. We also train the classifier to be robust against partial preamble jamming and pilot jamming, The resulting classification accuracy is over 90% at SJRs up to 12 dB. Additionally, we compare our classifier with one that uses the spectrogram (short-time Fourier transform) as input to the DCNN, and demonstrate the superior performance of the proposed scalogram-based classifier, particularly in the high SJR regime.**

***Index Terms*—Smart Jamming classification, Deep Neural Networks, Wavelet analysis, Wireless security, Wi-Fi networks.**

I. Introduction

Widespread adoption of Wi-Fi networks has made them a prime target for adversarial attacks. These attacks exploit the deterministic and publicly known structure of the Wi-Fi frame to launch various attacks. One notable category of these attacks is smart jamming. Over the past decades, numerous smart jamming attacks on Wi-Fi networks have been identified. *Preamble jamming* is one such attack. This attack is launched against legacy Wi-Fi preambles [1]–[3], aiming at disrupting the detection and synchronization processes of Wi-Fi frames. Modern Wi-Fi networks that implement orthogonal frequency-division multiplexing (OFDM) are particularly susceptible to smart jamming. For instance, the so-called *pilot jamming* specifically targets pilot subcarriers, which are crucial for channel estimation and phase tracking [4]. *Interleaving Jamming* [5] involves jamming every three data subcarriers that carry adjacent non-interleaved bits to cause burst errors. Zhao et al. [6] transmitted jamming signals of sufficient frequency offsets from a few target subcarriers to sabotage the orthogonality of OFDM systems. Recent works [7], [8] introduced attacks that modify the Signaling Fields of the frame preamble to jam various versions of Wi-Fi. All these jamming attacks operate at the Physical (PHY) layer and result in denial-of-service (DoS), which can potentially be used as a basis for launching higher-layer attacks. Accordingly, there is a critical need for *automated* detection and classification of smart jamming attacks, so that appropriate countermeasures, such as those proposed in [3], [5], [7]–[10], can be implemented to mitigate the impact of these attacks in real-time.

The low duty cycle (time domain) and/or narrow and constantly changing frequencies associated with smart jamming pose significant challenges for detection methods. In particular, traditional jamming detection techniques that rely on spectral analysis or signal strength are ineffective against these attacks. The authors in [6], [8] proposed detection approaches tailored to their respective smart attacks. However, these approaches suffer from performance degradation under certain signal-to-jamming power ratios (SJRs) or at specific attack locations. Moreover, these methods cannot differentiate between different smart jamming attacks. For example, by using frequency-domain analysis proposed in [6], the attacks in [6] would be misclassified as pilot jamming [4] or interleaving jamming [5] as all these attacks exhibit narrow-band characteristics. Additionally, jamming attacks discussed in [3], [4], [6] result in a rotated constellation of received signals, making them indistinguishable through constellation analysis. Classifying these smart jamming based on long-term error rates is infeasible because such errors can be attributed to a variety of attacks or poor channel quality, not to mention the long inference time.

Recently, researchers have turned to machine learning (ML) techniques to detect and classify smart jamming attacks. They have applied several time-frequency transforms (TFT), including short-time Fourier transform (STFT), continuous wavelet transform (CWT), and Choi–Williams transform, to the re-

This research was supported in part by NSF (grants CNS-1563655, CNS-1731164, and IIP-1822071) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

ceived signal to generate images that are fed into ML models. Various ML models were explored in [11] to detect and classify four jamming attacks on Wi-Fi-connected unmanned aerial vehicles (UAVs). These models incorporated features like signal-to-noise ratio (SNR), energy thresholds, and key OFDM parameters, and they were built on Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), among others. While the RF model achieved an accuracy of approximately 92%, the accuracy of other models ranged from 75% to 85%. Additionally, four popular DCNN models (AlexNet, VGG-16, ResNet-50, and EfficientNet-B0) were trained using the spectrogram image of the received signal. Although these models reported high accuracy exceeding 90%, critical information such as the SJRs, spectrogram parameters, and the number of signal samples used for the spectrogram were not provided. Moreover, it is worth noting that the four attacks studied in [11] exhibit distinct characteristics, including single-subcarrier jamming versus successive-subcarrier jamming and shot-noise jamming versus continuous barrage jamming across the entire spectrum. Gecgel et al. focused on detecting and classifying barrage jamming and jamming of the reference and synchronization signals in LTE networks [12]. They applied DNNs and support vector machines (SVMs) to spectral images obtained from three different TFTs. However, the accuracy they got at SJRs of $5\,$dB and $10\,$dB ranged from 25% to 70%. Such performance is unacceptable as these two SJR levels are considered moderate for a successful attack, given that reference and synchronization signals occupy only a small portion of time-frequency resources in an LTE frame. Most importantly, due to the fundamental differences between LTE and Wi-Fi protocols, the classifiers in [12] cannot be directly applied to Wi-Fi networks.

While it appears intuitive to use TFTs to preprocess the signal before applying ML classification, the utilization of spectrogram and scalogram images raises two concerns. First, cropping these images to fit specific sizes for ML models may lead to the loss of essential features. Second, training with images represented in three dimensions can be computationally intensive, particularly for the DNNs employed in [11].

In this paper, we develop a DCNN to automatically classify smart jamming attacks on Wi-Fi systems. Our classifiers use a time-frequency representation of the received signal as input. Specifically, we propose to feed the DCNN with the scalogram arrays of the CWT, which is applied to partially overlapped segments of the received I/Q samples. Our approach reduces the dimension and complexity of the classifier while preserving essential information. This is in contrast to cropped or resized spectrogram or scalogram images commonly used in the literature. Particularly, our proposed DCNN has less than 1% trainable parameters of those used in [11], [12]. We analyzed the proposed scalogram-based DCNN classifier by considering three smart jamming attacks in Wi-Fi systems: preamble jamming, pilot jamming, and interleaving jamming. These attacks were selected because they exhibit similar characteristics, making their classification particularly challenging.

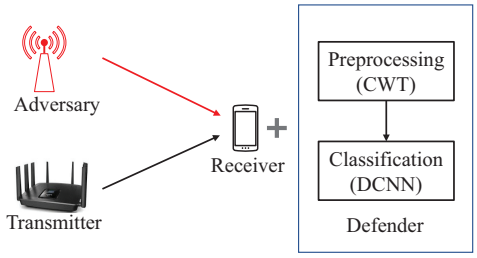


Fig. 1. System model.

We also adapt these attacks, initially designed for legacy Wi-Fi networks, to more recent IEEE 802.11ac Wi-Fi networks. As there is no publicly available dataset for such attacks, we generate synthetic data by simulating legitimate Wi-Fi signals and applying the respective attacks using MATLAB. Our classifier also naturally serves as a detector since we add a class that represents normal unjammed Wi-Fi signals. We employ Morlet and Gaussian wavelets for the CWT and observe only minor differences between their performance. Both wavelets achieve 98% accuracy across a wide range of SJRs between $-6\,$dB and $15\,$dB. Even at a high SJR of $15\,$dB, the accuracy is still around 90%. We further experiment with a shallower DCNN, resulting in around 10% reduction in accuracy only at the high SJR of $15\,$dB. Furthermore, we study the impact of replacing the scalogram arrays with the spectrogram arrays as input. Meanwhile, we make slight adjustments to the DCNN architecture to accommodate the input size. However, we observe that with this modification, the accuracy is significantly degraded at SJRs of $12 \sim 15\,$dB, demonstrating the superior performance of the proposed scalogram-based approach.

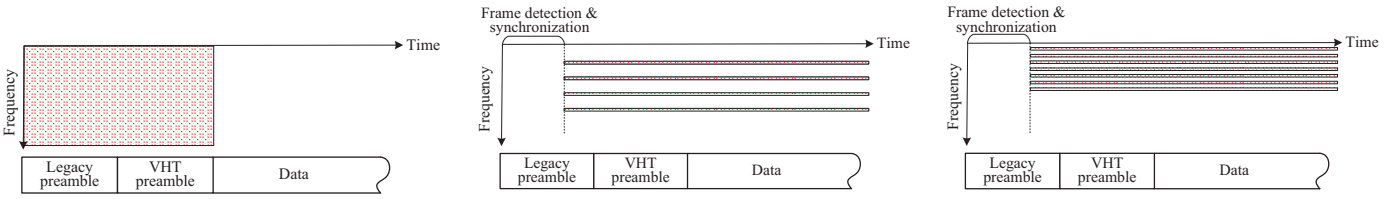
II. System and Attack Models

A. System Model

As shown in Fig. 1, we consider a system that consists of a legitimate Wi-Fi transmitter-receiver pair and an adversary, whose goal is to launch smart jamming attacks against legitimate Wi-Fi signals. We define SJR as the power ratio between the legitimate signal and the jamming signal at the receiver while the jamming is active. The defender resides at the receiver and has two modules: signal preprocessing and DCNN-based classification. The defender does not know a priori the presence of jamming nor the specific type of jamming attacks it encounters. It samples the received baseband I/Q signals and tries to detect the occurrence of a jamming attack during the sampled segment, and further identifies the type of jamming attack in progress.

B. Attack Models

In this paper, we consider three typical smart jamming attack models. Without loss of generality, we illustrate them in an IEEE 802.11ac network, which is a representative example of OFDM-based Wi-Fi networks.



(a) Preamble jamming (b) Pilot jamming (c) Interleaving jamming

Fig. 2. Three smart jamming attacks considered in this paper. (The Data field should be longer in reality but cropped here to save space.)

1) Preamble Jamming: As shown in Fig. 2, the data of an IEEE 802.11ac frame is prepended by a preamble, which has two parts: the legacy preamble and the very high throughput (VHT) preamble. Wi-Fi networks rely on the preamble to implement frame detection, timing, frequency synchronization, channel estimation, and PHY-layer signaling. As depicted in Fig. 2(a), the adversary transmits Gaussian noise across the receiver's bandwidth to jam the preamble. The adversary can target either the entire preamble (full preamble jamming) or specific portions of it, such as the VHT preamble (VHT preamble jamming), to disable its functionality, hence, disrupting the reception of the frame.

2) Pilot Jamming: In the IEEE 802.11ac system, a specific set of subcarriers at indices k_P are designated as pilot subcarriers to correct frequency offsets and phase noise over time. For example, in a 20 MHz channel, pilot subcarriers are at indices $k_P = \{\pm 21, \pm 7\}$. Fig. 2(b) shows pilot jamming [4] that targets all pilot subcarriers to distort them, and further disrupt the data recovery. We call the attacks that target a subset of the pilot subcarriers partial pilot jamming. To successfully launch a pilot jamming, the adversary needs first to detect the frame and synchronize with the carrier frequency of the receiver. Subsequently, the adversary transmits Gaussian noise on subcarriers at indices $k \in k_P$ across the remaining frame.

3) Interleaving Jamming: Interleaving jamming [5] exploits the fact that two adjacent data bits are separated by a few (four in IEEE 802.11ac) data subcarrier (DSC) spacings after the first-round permutation of the interleaver. By jamming every four DSCs, burst errors are induced that are hard to be corrected by the channel decoder. Fig. 2(c) shows one scenario of such attack, where 7 DSCs at indices $k_I = \{4, 8, 12, 16, 20, 24, 28\}$ are jammed by Gaussian noise. Similar to pilot jamming, frame detection and synchronization are conducted by the adversary before transmitting the noise throughout the remaining frame.

These smart jamming attacks focus energy on selected time or frequency domains. Therefore, they can succeed with a power lower than the power of the legitimate signal, i.e., a high positive SJR. To validate this, we conduct simulations of these attacks over a 20 MHz channel, targeting IEEE 802.11ac frames whose data is encoded with a rate of $3/4$ and modulated by 16-QAM. The results show that pilot jamming is the most energy-efficient, which leads to frame errors at an SJR up to 23 dB. Whereas the highest achievable SJRs for successful interleaving jamming and preamble jamming are 20 dB and 10 dB, respectively. Besides, at such high SJRs, these jamming attacks cannot be identified by an Oscilloscope or a spectrum analyzer.

III. Proposed Defense

A. Pre-processing with CWT

As demonstrated in Section II-B, smart jamming attacks conceal the jamming signal in either time and/or frequency domains by jamming for a short time or narrow time-variant frequencies. To analyze and identify the specific type of jamming, it is intuitive to examine the time-frequency representation of the signal. Fourier transform (FT) is useful for analyzing stationary signals. However, for non-stationary signals that exhibit changes in frequency and amplitude over time, different signals with the same frequency components but occurring at different times produce identical FT results. STFT and CWT overcome this limitation, as they can capture time-frequency characteristics simultaneously. STFT, with a fixed window size, suffers from poor frequency or time resolution if the window length is excessively narrow or wide, respectively. On the other hand, CWT, with varying window sizes, provides a multi-resolution representation of the signal at different positions [13]. This allows for the extraction of local spectral and temporal information simultaneously, making it an ideal approach for analyzing smart jamming attacks.

CWT is defined as a convolution between the signal $y(t)$ and a basis function $\psi(t)$ known as the wavelet [14], i.e.,

$$W(a,\tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} y(t) * \psi(\frac{t-\tau}{a}) dt \qquad (1)$$

where a is the scale parameter that determines the frequency resolution and τ is the position parameter of the wavelet that determines the time resolution. More specifically, a small a compresses the wavelet to reflect high-frequency details, and vice versa. In addition to adjusting the scale and position parameters when applying CWT, it is also possible to select different wavelet functions depending on the characteristics of the signal under consideration. In this work, we primarily use two complex wavelets [15] for the preprocessing of the received complex signals. The complex Morlet Wavelet is given by:

$$\psi(t) = \frac{1}{\sqrt{\pi f_b}} \exp(2j\pi f_c t) \exp(\frac{-t^2}{f_b}) \qquad (2)$$

where f_b is the bandwidth parameter and f_c is the center frequency of the wavelet. The Gaussian Derivative Wavelet

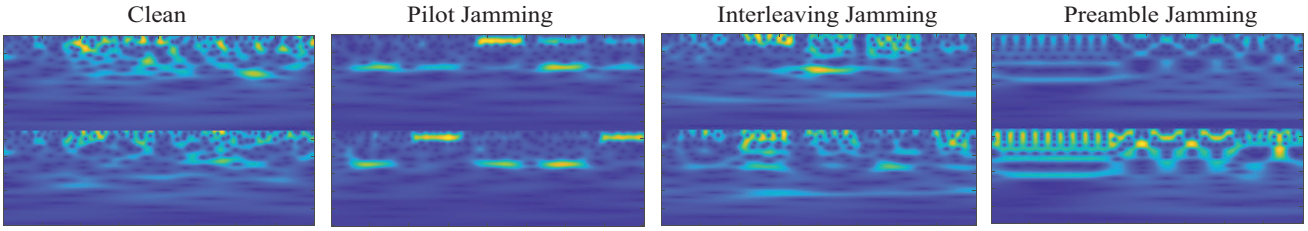


Fig. 3. Example scalogram images by Morlet CWT of the clean signal and the signals subjected to three smart jamming attacks.

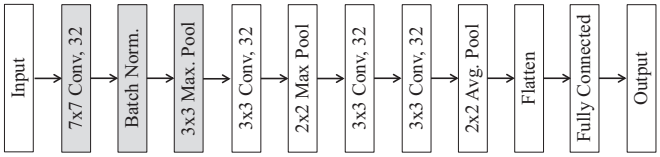


Fig. 4. DCNN_1 architecture used for smart jamming classification.

TABLE I
NUMBER OF TRAINABLE PARAMETERS OF THE PROPOSED CLASSIFIER WITH INPUT SIZE 400×100 AND STATE-OF-ART CLASSIFIERS IN [11].

Proposed	AlexNet	VGG-16	ResNet-50
32292	62000000	138000000	23000000

("gaus1") is the first-order derivative of the function:

$$\psi(t) = C \exp(-jt) \exp(-t^2) \tag{3}$$

where C satisfies the norm of the first-order derivative of $|\psi|$ equal to 1.

To detect stealthy jamming signals effectively, we employ a sliding window approach to sample partially overlapping segments with a sample rate of F_s. Denote the window size as w and the stride as s, in the units of OFDM symbols. After applying CWT to a received signal segment, we can get a scalogram array of size $T \times F$, where $T = wF_s$ and F is the number of frequencies for CWT. Fig. 3 depicts example scalogram images of the signal in the absence of jamming and signals subjected to three smart jamming within a 20 MHz channel. The images are obtained by CWT with a Morlet wavelet whose parameters are $f_b = 2$ Hz and $f_c = 1$ Hz. Here, the signal segment includes 400 baseband I/Q samples over $20\,\mu s$. The x-axis is the time, while the y-axis is the frequency. In the absence of jamming, the energy distribution is low and dispersed across the entire time-frequency plane. However, due to pilot jamming, the scalogram exhibits concentrated high energy around four frequency components, extending over time. In contrast, interleaving jamming displays high-energy concentrations across several frequencies in two separate domains. Lastly, the energy distribution resulting from preamble jamming aligns with the periodicity of the preamble in the time domain and extends over a broad frequency range. These distinct energy patterns in the scalogram images validate our previous analysis and provide useful insights for identifying different types of jamming attacks.

B. DCNN-based Classifier

There are many popular DCNNs (e.g.,ResNet) that are good at image classification and widely used for signal classification recently. However, we decide to design our own DCNN for two main reasons. First of all, those popular DCNNs require specific image sizes, which can limit the time-frequency resolution of the scalogram. Cropping or resizing the scalogram images to fit these specific input dimensions may result in the loss of essential features. Secondly, training images can be computationally intensive, as reported in [11]. Therefore, we propose a computationally efficient DCNN architecture called DCNN_1, shown in Fig. 4, as the defender's classifier. Instead of using the scalogram image, we propose to directly use the $T \times F$ scalogram array of the sampled signal segment as the input to the DCNN_1. The classifier consists of four convolutional layers, three pooling layers, and one fully-connected layer. The kernel size and the number of filters for each layer are specified in the respective blocks, and a stride size of 2 is used for most convolutional and pooling layers, except for the second and third convolutional layers. The dense layer is connected to the output layer, allowing the classifier to predict one of four classes. In Table I, We compare the number of trainable parameters of our proposed DCNN_1 and models used in [11]. Our proposed model requires less than 1% of the parameters used in other models.

Throughout the network, rich features are extracted by the convolutional layers to capture the time-frequency characteristics of the signal segment. The two pooling layers reduce the dimensionality of the features, thereby improving generalization and computational efficiency. ReLU activation functions are used in all convolutional layers, while the output layer utilizes Softmax activation. The Adam optimizer with a learning rate denoted as r. To mitigate overfitting, early stopping is applied by monitoring the categorical cross-entropy loss for the validation dataset with a patience of three.

IV. PERFORMANCE EVALUATION

A. Data Collection

As no open-source dataset for Wi-Fi traces subjected to these three smart jamming attacks is available, we generate a synthetic dataset using the Matlab WLAN Toolbox [16]. The parameters of the 802.11ac system used for the Wi-Fi trace are listed in Table II. We create legitimate Wi-Fi traces in which each frame carries 1500 bytes of data. With this configuration, each frame includes 88 OFDM symbols within which the first 5 symbols are the legacy preamble and the following 5 symbols are the VHT preamble. We consider a typical indoor environment with no large-scale fading. To

TABLE II
PARAMETERS FOR DATA GENERATION.

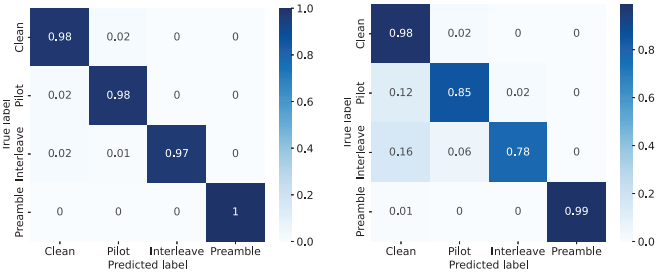
Parameter	Value	Parameter	Value	Parameter	Value
Channel Model	TGac Model-B	Frequency	5.25 GHz	Bandwidth	20 MHz
Frequency offset	0	SNR	20 dB	SJR	$-6:3:15$ dB
data size/frame	1500 bytes	Modulation	16-QAM	Coding rate	$3/4$
OFDM symbols/frame	88	Window size	5 OFDM symbols	Sample stride	1 OFDM symbol

TABLE III
SJR AVERAGED OVER THE WHOLE FRAME CORRESPONDS TO SJR_1 WHEN THE JAMMING IS ACTIVE FOR PREAMBLE JAMMING (UNIT: DB).

SJR	-6	-3	0	3	6	9	12	15
SJR_1	3.5	6.5	9.5	12.5	15.5	18.5	21.5	24.5

obtain a realistic dataset, we apply the TGac Channel Model-B [17] and add AWGN on top of it. The SNR is set to 20 dB, which is 5 dB higher than the minimum requirement for reliable communication of the data modulated by 16-QAM and coded by a rate of $3/4$. For simplicity, we assume no frequency offset among the three parties involved. In the case of pilot jamming and interleaving jamming, the attacks are assumed to be active after the adversary detects and synchronizes with the legitimate frame at the end of the fourth OFDM symbols and last throughout the remaining duration of the frame, despite targeting only a few subcarriers. For each jamming scenario, the adversarial signal is generated in accordance with Section II-B.

Generating thousands of packets for the dataset is infeasible, so we employ data augmentation techniques to expand our dataset. However, traditional data augmentation methods like flipping, rotation, and scaling could distort the temporal and frequency features of RF signals, making them unsuitable for our purposes. Consequently, we opt for a sliding window approach to augment our data. With the window size of 5 OFDM symbols and a stride of 1 OFDM symbol, two consecutively sampled segments have an 80% overlap. This window size allows the defender to capture per-symbol frequency-domain features for pilot jamming and interleaving jamming, as well as time-domain features for full or partial preamble jamming. Moreover, a stride of one OFDM symbol helps prevent misdetection. The presence of multipath fading and AWGN in the channel ensures sufficient difference between the non-overlapping portion of the two segments, yet belong to the same class. However, the overlapping may cause overfitting. To avoid any correlation between the training, validation, and test datasets, we separately generate Wi-Fi frames for each of them using different random seeds for data and channels. It is notable that even if the jammed signal only appears in part of the window, we still label the segment as the corresponding jamming. For each class, we generate 3000 segments for training, and 500 segments each for validation and testing at a specific SJR. In total, we collected 128000 segments of received signals across eight SJRs (see Table III), which are further preprocessed by CWT as explained in Section III-A.



(a) Overeall (b) SJR = 15 dB

Fig. 5. Confusion matrices of the optimal classifier under jamming attacks.

B. Optimal Classifier Performance

We first employ CWT with Morlet for preprocessing and the DCNN_1 in Fig. 4 for the classifier. The optimizer is Adam with a learning rate $r = 0.001$, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$. The batch size is set to 128 which runs for $10 \sim 20$ epochs. Here, we only consider the full preamble jamming and the other two smart jamming attacks. In practice, the defender does not know the SJR in advance. So we train the classifier with a mixture of data from all SJR values so that the defender can identify which class the test sample is regardless of the SJR. However, for evaluation purposes, we still test the performance at each SJR value in addition to the overall SJRs. As seen from Fig. 5(a), when tested over all SJRs, the proposed approach can predict each class with an accuracy greater than 97%. The specific classification accuracy at each SJR is plotted in Fig. 6, where the accuracy is above 98% for SJR less than 15 dB and suddenly drops to 90% at the challenging SJR of 15 dB. The detailed performance at 15 dB SJR is shown in Fig. 5(b). Notably, the defender occasionally misclassifies the stealthy pilot jamming and interleaving jamming to clean or each other. Overall, the classifier achieves high precision and recall of 0.9826 and 0.9825, respectively.

C. Impact of Different TFTs and CNN structure

We further form a simpler DCNN_2 by removing the first three shaded layers of DCNN_1 in Fig. 4. This change leads to a decrease in accuracy of approximately 10% only at a 15 dB SJR by around . To study the impact of different wavelets, we compare Morlet wavelets with Gaussian derivative wavelets, while using the same DCNN_1 model. We can see from Fig. 6 that there is no performance gap between the two wavelets. Finally, we exploit STFT instead of the CWT for preprocessing and the window size for the STFT is 64. Due to the change in the input size, we reduce the kernel size of the first convolutional layer from 7×7 to 2×2. The accuracy is comparable to the ones by CWT up to 9 dB SJR. But when it

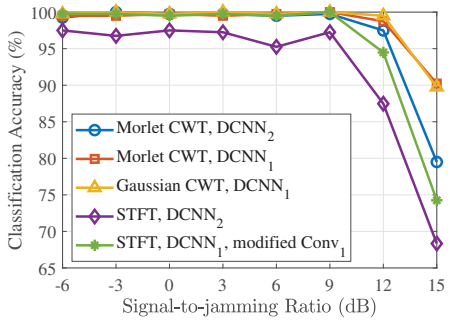


Fig. 6. Classification accuracy obtained by various approaches.

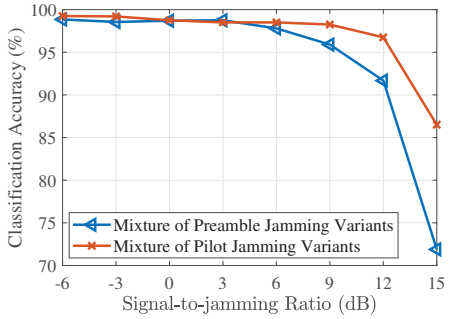


Fig. 7. Classification accuracy vs. SJRs for Morlet CWT DCNN_1 on datasets include VHT preamble jamming and partial pilot jamming.

comes to high SJRs, the accuracy is degraded by 5%. When we use STFT for the DCNN_2 classifier, the accuracy even deteriorates.

D. Robustness against Partial Jamming

To enhance the robustness of our classifier, we trained the Morlet-DCNN_1 model using a combination of full preamble jamming and VHT preamble jamming samples, while keeping samples for the other three classes fixed. The classification accuracy across various SJRs is shown in Fig. 7, where it significantly decreases from 95% to around 73% as the SJR increases from $9\,\text{dB}$ to $12\,\text{dB}$. This reduction in accuracy is primarily due to VHT preamble jamming, where the sampled window sometimes covers only a small portion of the jammed signal. However, the classifier makes multiple decisions as it slides the window along the received signal, ultimately enabling accurate detection and classification of the attack. We also consider partial pilot jammings to train a robust classifier. In this case, we train the model with a mixture of full pilot jamming and various partial pilot jammings that target $1 \sim 4$ pilot subcarriers at random indices. The classifier is quite robust against partial pilot jamming and achieves an accuracy above 95% for SJRs up to $12\,\text{dB}$. However, when the SJR increases to $15\,\text{dB}$, the accuracy drops to around 86%. Indeed, the scalograms of various partial pilot jamming exhibit obvious frequency selectivity, showing horizontal bars resembling those in Fig. 3, although the number of bars varies.

V. Conclusion

We proposed to detect and classify smart jamming attacks in Wi-Fi networks using a DCNN-based classifier with the scalogram array of the received signal as the input. We demonstrated that CWT performs better than STFT in extracting time-frequency features of the jammed signal, which improves the classification accuracy, especially in the challenging high SJR regime. Our proposed approach also has a much lower complexity when compared with state-of-the-art approaches while beating the accuracy of those approaches. Though the classifier does not know the SJR in advance, it achieves a classification accuracy above 98% across a range of SJRs from $-6\,\text{dB}$ to $15\,\text{dB}$, and the accuracy at $15\,\text{dB}$ SJR is 90%. We also improved the robustness of our classifier against partial preamble jamming and partial pilot jamming. In these cases, we achieved accuracy over 90% for SJR up to $12\,\text{dB}$. It demonstrates the practicality and effectiveness of the proposed approach. Our approach can be generalized to detect and classify any other smart jamming with necessary adaptation on the CWT, DCNN structure, and windowing.

References

[1] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan, "Understanding and mitigating the impact of RF interference on 802.11 networks," in *Proc. of the ACM SIGCOMM 2007 Conf.*, Aug. 2007, pp. 385–396.

[2] M. J. L. Pan, T. C. Clancy, and R. W. McGwier, "Jamming attacks against OFDM timing synchronization and signal acquisition," in *Proc. of the IEEE MILCOM 2012 Conf.*, Oct. 2012, pp. 1–7.

[3] H. Rahbari, M. Krunz, and L. Lazos, "Swift jamming attack on frequency offset estimation: The achilles' heel of OFDM systems," *IEEE Trans. on Mobile Comput.*, vol. 15, no. 5, pp. 1264–1278, May 2016.

[4] T. C. Clancy, "Efficient OFDM denial: Pilot jamming and pilot nulling," in *Proc. of the IEEE ICC 2011 Conf.*, June 2011, pp. 1–5.

[5] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Interleaving jamming in Wi-Fi networks," in *Proc. of the ACM WiSec 2016 Conf.*, July 2016, pp. 31–42.

[6] S. Zhao, Z. Lu, Z. Luo, and Y. Liu, "Orthogonality-sabotaging attacks against OFDMA-based wireless networks," in *Proc. of the IEEE INFOCOM 2019 Conf.*, May 2019, pp. 1603–1611.

[7] Z. Zhang and M. Krunz, "Preamble injection and spoofing attacks in Wi-Fi networks," in *Proc. of the IEEE GLOBECOM 2021 Conf.*, Dec 2021, pp. 1–6.

[8] Z. Zhang and M. Krunz, "SIGTAM: A tampering attack on Wi-Fi preamble signaling and countermeasures," in *Proc. of the IEEE CNS 2022 Conf.*, Oct 2022, pp. 1–9.

[9] C. Shahriar, R. McGwier, and T. C. Clancy, "Performance impact of pilot tone randomization to mitigate OFDM jamming attacks," in *Proc. of the IEEE CCNC 2013 Conf.*, Jan. 2013, pp. 813–816.

[10] B. Davis and B. DeBruhl, "Mitigating interleaving jamming of IEEE 802.11," in *Proc. of the IEEE CSCI 2020 Conf.*, 2020, pp. 16–21.

[11] Y. Li *et al.*, "Jamming detection and classification in OFDM-Based UAVs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16 859–16 870, 2022.

[12] O. A. Topal, S. Gecgel, E. M. Eksioglu, and G. Karabulut Kurt, "Identification of smart jammers: Learning-based approaches using wavelet preprocessing," *Phys. Commun.*, vol. 39, no. 101029, pp. 1–9, April 2020.

[13] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[14] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[15] Mathworks. (2023) Wavelet families. [Online]. Available: https://www.mathworks.com/help/wavelet/ug/wavelet-families-additional-discussion.html

[16] Mathworks. (2021) Matlab WLAN toolbox. [Online]. Available: https://www.mathworks.com/help/wlan/

[17] G. Breit, H. Sampath, S. Vermani *et al.*, "TGac Channel Model Addendum," IEEE, Report Doc. IEEE 802.11-09/0308r12, mar 2010.