AiFi: AI-Enabled WiFi Interference Cancellation with Commodity PHY-Layer Information

Ruirong Chen, Kai Huang and Wei Gao
University of Pittsburgh
USA

ABSTRACT

Interference could result in significant performance degradation in WiFi networks. Most existing solutions to interference cancellation require extra RF hardware, which is usually infeasible in many low-power wireless scenarios. In this paper, we present *AiFi*, a new interference cancellation technique that can be applied to commodity WiFi devices without using any extra RF hardware. The key idea of AiFi is to retrieve knowledge about interference from the locally available physical-layer (PHY) information at the WiFi receiver, including the pilot information (PI) and the channel state information (CSI). AiFi leverages the power of AI to address the possible ambiguity when estimating interference from these PHY information, and incorporates the domain knowledge about WiFi PHY to minimize the neural network complexity. Experiment results show that AiFi can correct 80% of bit errors due to interference and improves the MAC frame reception rate by 18x, with <1ms latency for interference cancellation in each frame.

CCS CONCEPTS

• **Networks** → **Network protocols**;

KEYWORDS

Interference cancellation, WiFi, Artificial Intelligence, pilot information, channel state information

ACM Reference Format:
Ruirong Chen, Kai Huang and Wei Gao. 2022. AiFi: AI-Enabled WiFi Interference Cancellation with Commodity PHY-Layer Information. In *ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3560905.3568537

1 INTRODUCTION

Wireless interference widely exists in today's WiFi networks when multiple devices simultaneously transmit in the same unlicensed WiFi band, and can cause serious network performance degradation with the growth of wireless device population and contention of the limited wireless spectrum.

Commodity WiFi networks combat interference using CSMA/CA [11, 23, 75], which detects channel occupancy via carrier sensing and avoids interference by postponing transmissions (i.e., backoff) if the channel is occupied, but introduce a significant delay due to the long backoff period [50, 61, 77]. Other schemes extend interference avoidance to different wireless technologies [62, 84], and recent cross-technology communication techniques reduce the backoff time by enabling explicit coordination among wireless transmitters [33, 40, 52, 54]. However, these existing schemes based on clear channel assessment cannot fully eliminate the extra delay caused by interference, especially in heavily occupied channels where many embedded devices concurrently transmit.

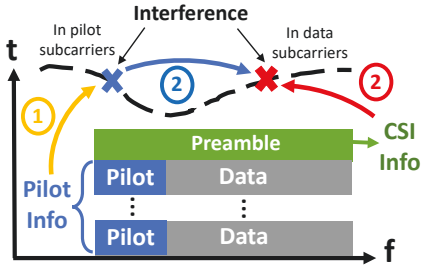


Figure 1: Interference cancellation with commodity PHY information

Instead, interference cancellation [28, 49, 63] uses additional RF hardware to probe the interference signal, which is then removed from the received signal. Similar techniques have also been adopted for full-duplex radios [6, 10, 17–19, 35]. Such extra RF hardware, in most cases, involves multiple RF antennas, RF frontends and PHY-layer controllers. Therefore, adding these extra RF hardware to wireless devices is expensive and infeasible in many wireless scenarios that have strict constraints on the wireless devices' cost, form factor and energy consumption, such as ultra-low-power wireless networks [42, 55, 56, 83], body area networks [7, 58] and industry IoT networks [3, 46]. Commodity MIMO systems, on the other hand, cannot be used to provide such extra hardware, because the different MIMO antennas are controlled by the same MIMO controller and can only be all set as Tx or Rx mode at one time. As a result, most existing schemes use custom RF hardware, which however, cannot be applied to commodity WiFi devices.

To address this limitation, we envision that a fundamental shift on the design methodology of interference cancellation is needed: instead of probing the interference signal on the air using extra RF hardware, knowledge about interference should be retrieved from the local PHY information available at commodity WiFi devices. Such retrieval is possible because the available PHY information, including the pilot information (PI) [1, 4, 70] and channel state information (CSI) [1, 48], exhibit identifiable patterns in both time and frequency domains when interference is present. In particular, in each pilot subcarrier being used in an OFDM-based WiFi system[1],

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568537

[1] OFDM has been used in all mainstream WiFi networks from 802.11a/g to 802.11ac/ax [9]. Old standards (e.g., 802.11b), instead, are obsolete and less used in practice [27].

interference changes the pilot signal's phase over time from linear to non-linear. Interference in each data subcarrier, on the other hand, affects the frequency-domain channel estimation in the subcarrier, which is represented by CSI in the subcarrier's frequency band.

Based on this insight, in this paper we present *AiFi*, a new technique that only uses commodity WiFi devices' locally available PHY information for interference cancellation. As shown in Figure 1, AiFi first calculates the interference in pilot subcarriers by comparing the interfered pilot signal's phase with that of non-interfered pilot signal. Then, it applies such knowledge about pilot subcarriers' interference into regression, to estimate and remove the interference in other data subcarriers. Since the number of WiFi data subcarriers is much larger than that of pilot subcarriers[2], to ensure the estimation accuracy, we further use the CSI in data subcarriers to provide extra frequency-domain information about interference, and use such information to refine the regression.

The major challenge, however, is the possible ambiguity when estimating interference from WiFi PHY information. For example, the phase variation in pilot subcarriers may not uniquely correspond to the interference signal that may have variant amplitudes over time, and channel estimation provided by CSI could be affected by channel distortions caused by random noise or device mobility. Our basic solution to this challenge is to leverage the power of Artificial Intelligence (AI) and use a neural network (NN) to precisely identify and eliminate any ambiguity or inaccuracy in interference estimation and removal. To minimize the NN complexity and meet the timing constraint at WiFi PHY, we explicitly incorporate the domain knowledge about WiFi PHY functionality, such as channel equalization and encoding, as the building blocks in the NN structure. In this way, we can ensure the quality of NN training by avoiding redundant NN structures and training confusions.

More specifically, the WiFi PHY operations ensure continuity across the PI information of different pilot subcarriers, and AiFi utilizes such continuity to perform regression with a deconvolutional NN. After the interference has been estimated from regression, AiFi uses fully-connected NNs to mimic the channel equalization process in the feature space, to remove the estimated interference from the received signal. Furthermore, the accuracy of such removal may be limited when interference is strong and results in very low signal-to-interference-plus-noise ratio (SINR). In this case, we use a long short-term memory (LSTM) network to mimic the commodity WiFi encoder, and further correct data coding errors by restoring the correlation between data payloads in consecutive data symbols.

In practice, although the required PI and CSI information may not be accessible on all commodity WiFi devices, they can be made available on most commodity device models with manageable engineering tweaks or updates on WiFi device drivers or firmware[3]. As a result, AiFi can be applied to many WiFi applications to reduce the lowest SINR requirement for supporting various modulations under interference, especially low-power wireless applications where the power-constrained WiFi devices are incapable of combating interference with increased transmit signal power. Our detailed contributions are as follows:

- We designed unique NN structures that can precisely estimate interference in each data subcarrier, by only using the local PHY information at the WiFi receiver.
- Our NN designs can effectively remove interference from the received WiFi PHY signal, by reflecting WiFi system's domain knowledge in NN models.
- Our design of the LSTM network can correct bit errors across multiple data symbols due to interference, by learning and restoring the long-term correlation among data payloads in these symbols.

We implemented AiFi in an 802.11g network and evaluated AiFi with different interference sources including WiFi, ZigBee, baby monitors and microwave ovens. The performance of AiFi is also evaluated over multiple practical wireless applications, including 1) wireless sensing, 2) webpage loading and 3) online gaming. Our experiment results have the following conclusions:

- AiFi is *accurate*. AiFi is the first system that achieves the performance of the best existing interference cancellation schemes but does not use any extra RF hardware. It can correct 80% of bit errors due to interference, and improves MAC frame reception rate (FRR) by up to 18x under interference. Such improvement reduces the minimally required SINR for different WiFi data rates by >3dB and can potentially improve the wireless network performance by >100%.
- AiFi is *adaptive*. AiFi can well adapt to interference from different signal sources. Even under highly dynamic environmental conditions, it can correct at least 70% of frame reception errors.
- AiFi is *lightweight*. AiFi involves the minimal computation overhead. Its average NN inference time is <1ms per frame, and meets the timing constraints of many network applications. It can largely enhance the user's Quality of Service (QoS) in these applications.

2 BACKGROUND & MOTIVATION

To better understand the design of AiFi, we first introduce the background of WiFi PHY information. We then motivate our design by demonstrating the identifiable patterns of such PHY information with interference, and the ineffectiveness of using a monolithic NN to learn the relationship between these patterns and interference.

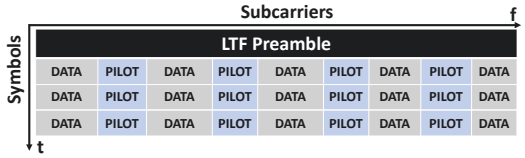


Figure 2: Obtaining PHY information in a WiFi frame

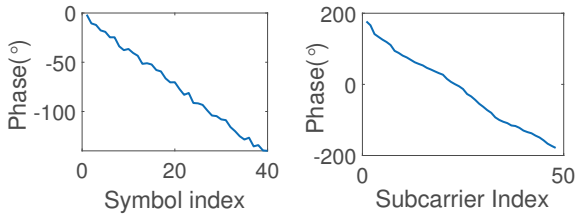
2.1 WiFi PHY Information

As shown in Figure 2, when the channel is invariant within the time duration of a data frame, a WiFi receiver uses the received long-training-field (LTF) frame preamble to compute the CSI for each subcarrier as its channel estimation $H = Y/X$, where X is the predefined LTF signal and Y is the received LTF signal. Further, since the channel may vary over time within the frame duration, the WiFi network embeds a number of pilot subcarriers in each data

[2]In a 20 MHz channel, 802.11g uses 48 data subcarriers and 4 pilot subcarriers, and recent 802.11ax uses 234 data subcarriers and 12 pilot subcarriers [9].

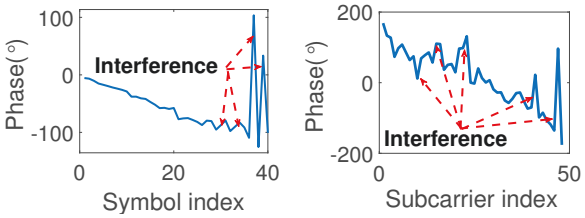
[3]For example, Intel provided a custom driver that allows accessing CSI in the Intel 5300 WiFi chipset [34]. Researchers provided drivers for CSI access on Qualcomm Atheros WiFi chipsets from hardware registers [80].

symbol, and measures such time-domain channel variation from the channel estimations in pilot subcarriers over different symbols.



(a) PI (b) CSI

Figure 3: Linear phase variations in PI and CSI



(a) PI (b) CSI

Figure 4: Non-linear phase variation in PI and CSI when interference is present

Since the transmitted signals in each LTF preamble and pilot subcarrier are pre-defined BPSK bit sequences and are transmitted with a constant sampling rate, there is always a fixed phase difference between every two consecutive CSI samples or PI samples in a clear channel. As a result, the PI's phase variation over time is linear as shown in Figure 3(a), and the CSI's phase variation over different data subcarriers is also linear as shown in Figure 3(b).

However, when the interference signal I is present, the channel estimation in PI or CSI is changed as

$$H_I = Y_I/X = (HX + I)/X = H + I/X, \quad (1)$$

where H is the channel estimation without interference. The interference's impact, characterized by I/X, then distorts the phase variation of PI and CSI to be non-linear, as shown in Figure 4. Other channel variations caused by practical factors, such as device mobility or multi-path effect, can also introduce such non-linearity. However, as we will describe later in Section 3, our NN design in AiFi is able to distinguish between the non-linearity caused by interference and other practical factors, by using the difference between interfered and non-interfered WiFi channel estimations in different environmental conditions as the input to NN models.

Such identifiable patterns in PI and CSI, when interference is present, motivates our design of AiFi that utilizes these PHY information to estimate and remove interference. To precisely estimate and remove the interference in data subcarriers, AiFi uses neural networks to adaptively integrate the time-domain information provided by PI and frequency-domain information provided by CSI.

2.2 Estimating and Removing Interference using Neural Network

To estimate interference using neural networks, the most straightforward approach is to use the available PI and CSI information as the input to train a monolithic neural network, where the transmitted data payload is being used as the output labels in training.

To verify the effectiveness of such training, we conducted preliminary experiments by using a 10-layer convolutional NN with the increasing complexity from 16, 32, 64 to 65,536 layers to learn the correlation between PI/CSI information and the transmitted signal, when WiFi interference is present in the same 2.4 GHz band. Results in Figure 5 show that, when being operated on a RTX A5000 GPU, even when a highly complicated NN is being used and results in a inference latency of >600ms, a very small amount of bit errors can be corrected. The basic reason of such low performance of using a monolithic NN is the high uncertainty of the interference signal and the channel condition that jointly affect the received signal. As a result, it is common that a monolithic NN is confused and even does not converge during training.

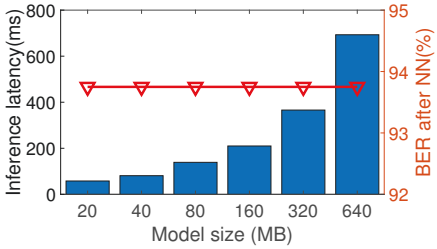


Figure 5: Learning interference using a monolithic NN

Based on these results, we expect that a monolithic NN with much higher representation power is needed to correctly address the possible randomness and abruptness of interference, but using such a complicated NN produces unacceptable computing delay at WiFi devices. Such ineffectiveness of using a monolithic NN, hence, motivates our design of AiFi that uses domain knowledge about WiFi PHY to reduce the NN complexity and avoid possible confusions in NN training.

3 SYSTEM OVERVIEW

As shown in Figure 6, interference cancellation in AiFi builds on the NN design that is guided by the domain knowledge about WiFi PHY functionality. More specifically, AiFi first extracts the interference features from the PI and CSI information at WiFi PHY, and then uses these features to estimate interference in each data subcarrier via regression and refinement based on attention NNs [73, 74]. After that, AiFi removes such interference from the received signal in two steps. First, it removes the interference in each individual data subcarrier by using a fully-connected NN to mimic the WiFi PHY's channel equalization process in the feature space. Second, it further recovers data encoding errors across multiple data subcarriers, by using a LSTM network to mimic WiFi data encoders and restore the correlation between different subcarriers' data payloads in the encoding procedure.

In training, AiFi jointly trains all the involved NN modules in an end-to-end manner with a unified cross-entropy loss function, which aims to minimize the errors in the corrected data payloads after interference cancellation. Its training data is a collection of interfered WiFi signals with known data payloads: these signals and their WiFi PHY information are used as model inputs, and the known data payloads of these signals are be used as output labels. The training includes interfered signals being collected with different interference patterns and channel conditions, to ensure generality and adaptability of the trained NN models.

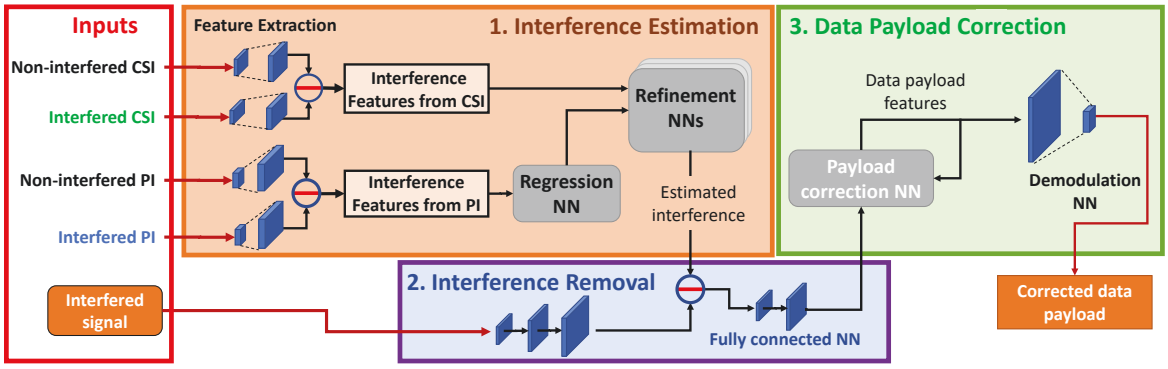


Figure 6: AiFi system overview

3.1 Interference Estimation

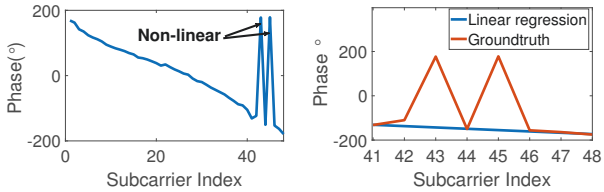
To estimate interference, we first extract features from PI and CSI information provided by WiFi PHY. According to Eq. (1), the interference signal I in a channel is written as

$$I = (H_I - H)X, \tag{2}$$

where X is the transmitted signal, and H_I and H are channel estimations with and without interference. Eq. (2) shows that when X is known, the interference signal I can be uniquely identified by H_I and H. Hence, for both PI and CSI, AiFi separately uses convolutional NNs to extract features from interfered and non-interfered channel estimations, and takes their difference as interference features without requiring any prior knowledge about the patterns of interference signals in different domains. These interference features, then, reflect the information about the interference signal's amplitude and phase in the feature space.

In training, the non-interfered PI and CSI information will be collected from WiFi frames that have a high SINR above 23dB, where the channel is considered as clear without noticeable interference. The interfered and non-interfered channel estimations used in training will be collected in different channel condition settings that are varied by various practical factors, such as device mobility and the surrounding environments. In each setting, the interfered and non-interfered signal samples used in training will be always collected in pairs, so that their only difference is the interference signal. In this way, AiFi ensures that the NN models can remove the channel estimation variations caused by other irrelevant factors from the extraction of interference features.

Based on this design, even though the training data may not cover all the possible domains of interference signals, AiFi can efficiently extract interference features, as long as the NN models are trained to correctly extract features from the non-interfered and interfered channel estimations. Such correctness is ensured due to the following two reasons. First, the variability of channel estimation is constrained by WiFi PHY operations such as channel equalization and is hence smaller than the heterogeneity of interference signals across different domains. Second, using NN models with sufficient representation power makes sure that AiFi can precisely capture the non-linearity in channel estimations, compared to traditional signal processing methods that are limited to extracting linear features from channel estimations [41, 45, 66].



(a) Interference's phase variation (b) Results of linear regression

Figure 7: Interference in data subcarriers

In online inference, we use the aggregate of non-interfered channel estimations collected in different channel conditions, which were used in training, as the non-interfered CSI and PI information for interference estimation. Since non-interfered channel estimations mainly contain linear features as shown in Figure 3 and these linear features have limited variability in different channel conditions, using these features as the reference can still ensure accurate interference estimation in new domains.

With the interference features extracted from PI, AiFi uses regression to interpolate these interference features into each data subcarrier. Standard linear regression, however, fails to correctly capture the non-linear variation of interference over different frequency bands. For example, when the interference's phase exhibits non-linear variation between data subcarriers 41 and 48 as shown in Figure 7(a), using linear regression results in wrong estimation of interference in these subcarriers, as shown in Figure 7(b).

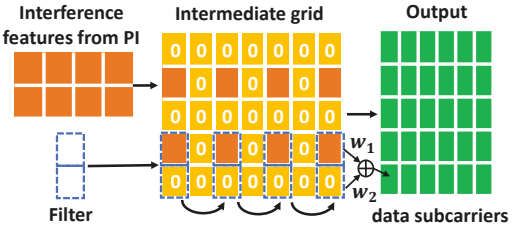


Figure 8: Regression NN

Instead, AiFi uses a deconvolutional NN (`Regression NN` in Figure 6) for such regression, which reverses the procedure of convolutional feature extraction process and learns the generic channel features at data subcarriers to precisely capture their non-linearity. As shown in Figure 8, we first expand the interference features to an intermediate grid via zero padding, and then slide

an 1D filter with trainable parameters over the grid to weigh its components and generate the output. We set the filter size to be 2, to exploit the continuity in the interference's phases between consecutive data subcarriers in commodity WiFi PHY.

However, such interference estimation may not be always accurate, because interference features from PI do not provide frequency-domain information about interference in data subcarriers. AiFi uses interference features from CSI to further refine the interference estimation, and details of such refinement (`Refinement NNs` in Figure 6) are in Section 4.

3.2 Interference Removal

The estimated interference, then, is individually removed from the received WiFi PHY signal in each data subcarrier. In AiFi, after being converted to the feature space, the received signal with interference is applied to a fully-connected NN, which mimics the channel equalization in WiFi PHY for interference removal.

Commodity WiFi adopts Zero-Forcing (ZF) equalization [22, 45] to address the received signal's distortions that are produced during channel propagation, by inversely applying the channel estimation to the signal. Hence, when the interference is estimated from channel estimations in PI and CSI, using ZF equalization to remove the interference is equivalent to subtracting the estimated interference from the received signal in the frequency domain.

To mimic this equalization in the feature space, AiFi uses a fully-connected NN to learn the frequency-domain subtraction of the estimated interference (I) from the received signal (X_I), by adding learnable weights W to such subtraction. The signal after interference removal can hence be written as

$$X = (X_I - I) \cdot W. \tag{3}$$

3.3 Data Payload Correction

Due to the limited signal resolution in PI and CSI information, interference removal described above may not completely remove the interference, when interference is strong and results in very low SINR. In these cases, AiFi further mimics the encoding process in commodity WiFi to correct the decoding errors in data payloads due to interference.

The WiFi encoder correlates each input bit with the previous 6 input bits and interleave the bits that are further modulated into data signals. Similarly, AiFi uses a LSTM network (`Payload Correction NN` in Figure 6) to learn the dependencies between consecutive symbols. In this way, AiFi recovers data payload features from errors, upon detecting contradictions with the learned dependencies.

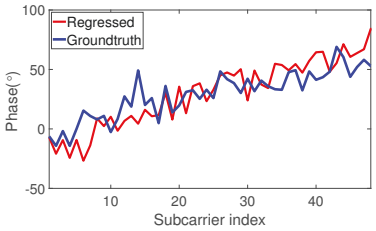


Figure 9: Interference estimation via regression from PI

Eventually, AiFi uses a `Demodulation NN` to replicate the demodulation functionality in WiFi PHY: demodulation at a commodity WiFi receiver transforms the encoded data signal to data payloads,

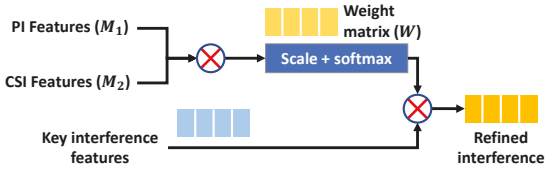
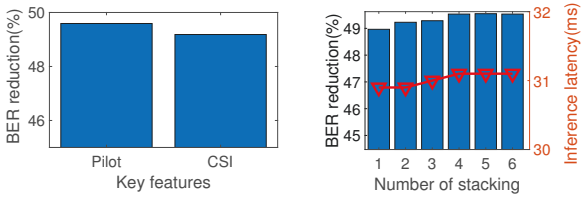


Figure 10: Feature refinement NN

and AiFi's `Demodulation NN` similarly transforms the encoded data signal features from the `Payload Correction NN` to data payloads. Details of such data payload correction are provided in Section 5. In this way, by incorporating these domain knowledge about WiFi PHY operations that are independent from the interference signal into the design of NN models, we ensure that these NN models are trained to learn how interference impacts WiFi data transmission and decoding and further how to correctly remove interference from the received WiFi signal, without assuming any prior knowledge about the interference signal itself. AiFi, hence, can be widely applied to different application scenarios with different interference sources, interference signal patterns and strengths. Such generality will be demonstrated in Section 7.



(a) Choice of key features (b) Stacking refinement NNs

Figure 11: Design choices in feature refinement NN

4 REFINING INTERFERENCE ESTIMATION WITH CSI FEATURES

As shown in Figure 9, interference estimation solely from PI features may be inaccurate, due to PI features' limited resolution in the frequency domain. To further refine such interference estimation with interference features from CSI, our design in AiFi is inspired by attention neural networks [73, 74], and aims to enhance the NN model's cognitive attention to the important interference features that are highlighted in the CSI information.

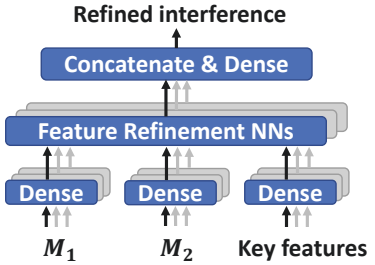


Figure 12: Stacking refinement NNs

To achieve this objective, we train the Refinement NN to learn a weight matrix that captures the correlation between PI and CSI interference features. More specifically, as shown in Figure 10, AiFi takes the PI and CSI interference features as two input masks (M_1

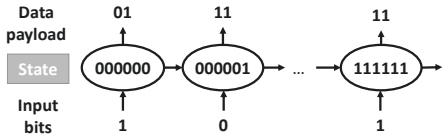


Figure 13: PHY encoder in commodity WiFi

and M_2), and learns the weight matrix (W) by applying softmax operation on the correlation of these two sets of features as

$$W = \text{softmax}(M_1 \cdot M_2 / \sqrt{Scale}), \quad (4)$$

where the scaling constant prevents the correlation result grow large in magnitude and hence pushes the softmax function to a region that has very small gradient.

This weight matrix is applied to the key features that represent the interference in the target data subcarrier to refine the estimation of such interference. In practice, the key features can be either the PI or CSI interference features, and we experimentally verified that using PI interference features as the key features reduces 2% extra bit errors, as shown in Figure 11(a) where Wifi data frames are transmitted with 4dB SINR.

To ensure sufficient learning power in practice, as shown in Figure 12, we further stack multiple refinement NNs to intentionally introduce variation to the input interference features. The outputs of stacked NNs are concatenated and densed to acquire the refined interference features. To balance between the estimation accuracy and NN complexity, we experimentally investigate different numbers of feature refinement NNs being stacked. Results in Figure 11(b) suggest that stacking 4 NNs achieves the highest reduction of bit error rate (BER) from the estimated interference, without unnecessarily incurring extra computing overhead.

5 CORRECTING DATA PAYLOAD ERRORS

In this section, we present how to correct the decoding errors in data payloads due to interference, using a LSTM network to mimic data encoding in commodity WiFi PHY.

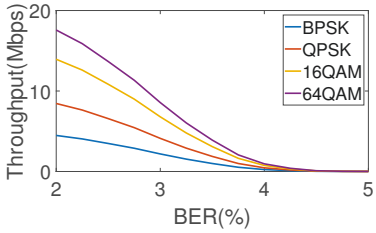


Figure 14: Correcting data payload errors by directly mimicking the commodity WiFi PHY encoder

5.1 Payload Correction NN

The PHY encoder in commodity WiFi correlates every input bit with the previous 6 input bits to output one data payload of two bits, which is then sent to the RF frontend for transmission onto the air. Such encoding process can be modeled a state machine where the payload output of each 6-bit state is jointly determined by the previous state and current input to the state, as shown in Figure 13. Then, an intuitive approach to correct bit errors in data payloads due to interference is to mimic such commodity encoder, by predicting every data bit from the previous 6 bits in data payload.

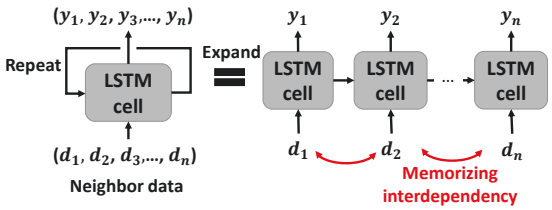
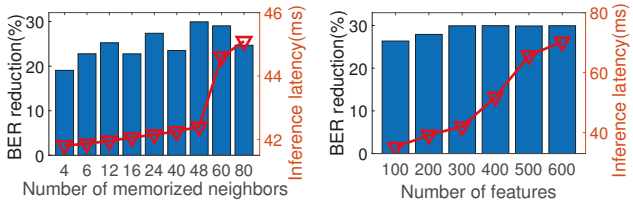


Figure 15: LSTM structure

However, directly mimicking the commodity WiFi encoder has limited capabilities in such error correction. As shown in Figure 14, when the BER caused by interference exceeds 4%, the excessive amount of bit errors caused by interference cannot be corrected and quickly reduce the network throughput to 0. This is because the range of dependency between data payloads in the WiFi encoder is limited to 6 consecutive bits. The long-term dependencies in data payloads are simply ignored, despite their importance in identifying bit errors caused by interference.

Instead, AiFi uses a LSTM network [32, 81] that shares the similar structure as the state machine in encoder to mimic the encoding process, as shown in Figure 15. Since a LSTM network memorizes both the short-term and long-term dependencies between data payloads with a memory cell, it can restore the errors in data payloads by regression based on such dependencies.



(a) Number of memorized neighbors (b) Number of LSTM features

Figure 16: LSTM design choices

To ensure that the LSTM correctly mimics the WiFi encoding process, the number of memory cells in the network should be a multiple of 6. According to our experiment results in Figure 16(a), the LSTM network achieves the highest BER reduction without incurring extra computing latency when memorizing 48 consecutive bits in data payload.

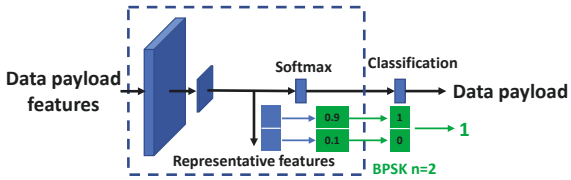
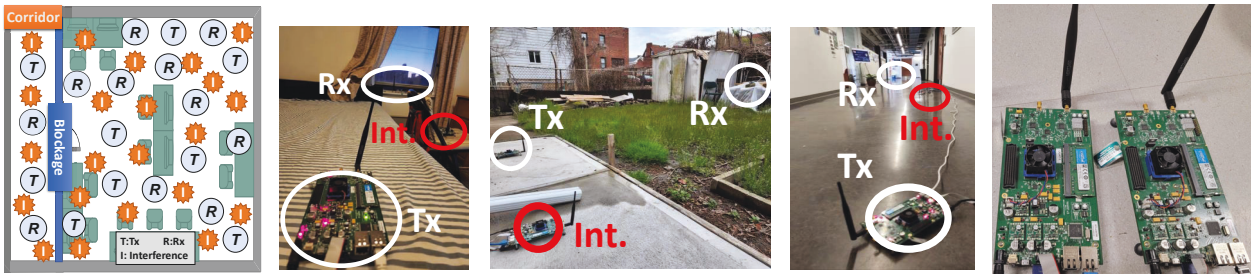


Figure 17: Demodulation NN design

5.2 Demodulation NN

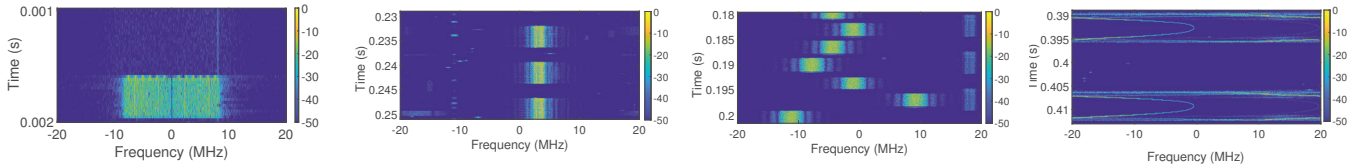
The LSTM network outputs the corrected data payloads in the feature space, which needs to be transformed to data bits for WiFi decoding. We use a Demodulation NN to mimic the WiFi demodulator, which classifies the equalized PHY signal to the data payload based on the signal's phase and amplitude.

As shown in Figure 17 where BPSK demodulation is used as an example, the demodulation NN first compresses the output features



(a) Location settings (b) Test site:room (c) Outdoor environment (d) Test cite: corridor (e) WARP

Figure 18: Evaluation setup



(a) WiFi (b) Zigbee (c) Baby monitor (d) Microwave

Figure 19: Patterns of interferences from different sources

from LSTM with convolution layers to acquire n representative features, where n is the number of possible data payloads that is determined by the current modulation scheme being used[4]. Then, it computes the probability mapping from Softmax, and chooses the output data payload with the highest probability.

Both the accuracy and computing complexity of data payload correction depend on the dimension of data payload features. Our experiment results in Figure 16(b) show that the NN computing cost linearly increases with more features, but its performance of BER reduction saturates when the number of features exceeds 300.

6 IMPLEMENTATION

We build NN models in AiFi with standard TensorFlow 2.8.0 [2] Python APIs. The models are trained with cross-entropy loss function and ADAM optimizer [44] with the learning rate of 10^{-4} in 1000 epochs, and the details of NN models are described as follows.

- The *feature extraction NN* has 3 convolutional layers with 32, 64 and 128 features.
- The *Regression NN* has 3 convolutional layers with 64, 128 and 256 features, and the output are condensed into an array of 96 features. Then, 3 deconvolution layers with 256, 128 and 64 features are used to generate output.
- We implement the *Refinement NNs* using `matmul` and `softmax` functions with 128 features.
- The *Interference Removal NN* uses 128 features and subtracts the interfered signal features with ReLu activation function.
- The *Payload Correction NN* is implemented with a single cell LSTM network with 300 features.
- The *Demodulation NN* is implemented with 3 convolutional layers with 64, 128 and 256 features. The output is passed to a `softmax` function, whose output probability is then used by an `argmax` function for classification.

We deploy the trained NN models on a PC system running Linux Ubuntu 18.04, by loading the NN models into the Linux kernel through the TensorFlow C APIs. The PHY information and the received signal from WiFi PHY, then, are transmitted through an UDP socket in the Linux kernel and can be directly accessed by AiFi's NN models running on GPU, so as to minimize the end-to-end latency of interference cancellation at runtime.

[4]In commodity WiFi that uses QAM modulation, this number is 2 for BPSK, 4 for QPSK, 16 for 16QAM and 64 for 64QAM.

(a) Zigbee (b) Baby Monitor (c) Microwave

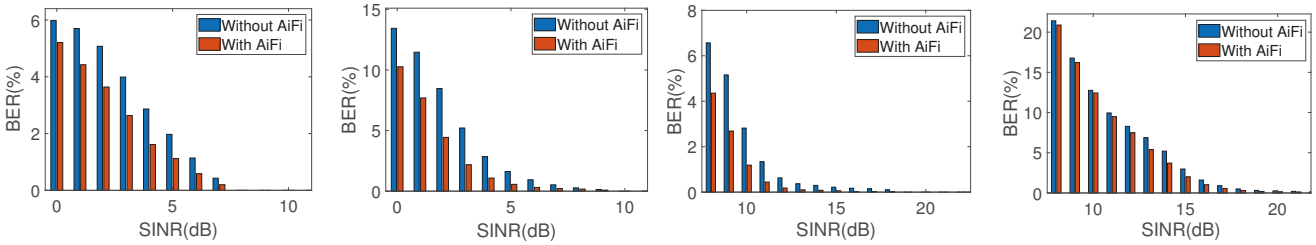
Figure 20: Interference sources

To enable interference cancellation for all modulation schemes being used in WiFi, such as BPSK, QPSK, 16QAM and 64QAM, we individually train four demodulation NNs with different sizes of representative features, as described in Section 5.2, and preload them for online inference.

In practice, the length of a WiFi PHY frame can vary based on the specific size of data payload being transmitted. To accommodate WiFi data frames with various lengths, our implementation builds the NN models in the way that all the data symbols in a frame are being processed in the NNs as a batch. For example, an 802.11g frame that has n symbols over 48 data subcarriers is reorganized into an input matrix with dimensions of $n \times 48$, and a batch size of n is then used.To further speed up the training, in our implementation we simultaneously feed multiple frames as the input to NNs, with a bigger batch size of $m \times n$. Note that in practical TCP transmission, a TCP receiver window is usually used with a buffer to temporarily hold the received data. Such buffer is usually big enough to store multiple received frames, and hence allows AiFi to process multiple frames in a big batch.

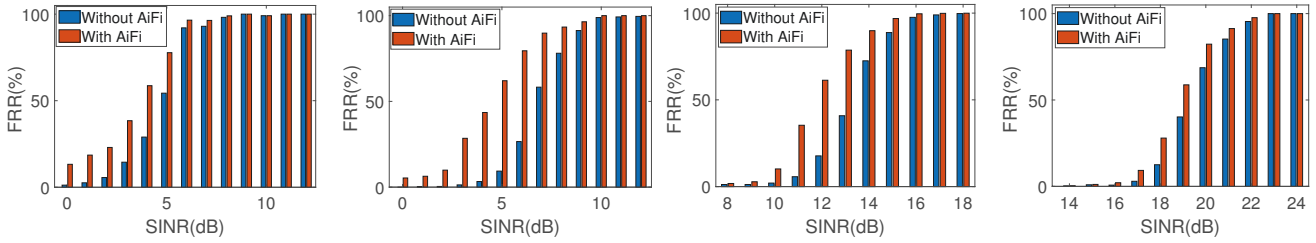
7 PERFORMANCE EVALUATION

As shown in Figure 18, we evaluate AiFi in multiple environments that result in significantly different interference patterns. Our NN



(a) BPSK (b) QPSK (c) 16QAM (d) 64QAM

Figure 21: BER reduction: the minimum SINR required to reach 1% BER is reduced by >4dB.



(a) BPSK (b) QPSK (c) 16QAM (d) 64QAM

Figure 22: FRR improvement: the minimum SINRs required for different modulations are reduced by >3dB.

models are trained based on data collected in a 20m×20m lab site shown in Figure 18(a), where we place the wireless transceivers and interference sources at different locations with variant levels of transmit power to produce different channel conditions and levels of SINR. In our evaluations, data collected in the lab site are randomly split into a training dataset with 80% data and a testing dataset with 20% data, and all evaluation results are averaged over 100 random splits. In all evaluations, the testing dataset is ensured to be different from the training dataset. Further, we also evaluated the performance of AiFi by applying the NN models being trained with the lab site data to other test sites, including a 5m×3m residence room shown in Figure 18(b), an outdoor yard shown in Figure 18(c), and a 15m×1.2m corridor shown in Figure 18(d).

In our experiments, AiFi's NN models are trained using 300k WiFi data frames transmitted in the 2.4GHz band, when different interference sources are present and different WiFi modulation schemes are used. Note that when interference is present, to mitigate its impact and minimize data decoding errors, WiFi rate adaptation always reduces the code rate in all transmissions to the lowest 1/2. Thus, we use the code rate 1/2 in all experiments. NN inference is then executed on a Dell Precision 7820 tower workstation. For experimental evaluation and analysis, we leverage the WiFi reference design on WARP v3 SDR [20] to transmit and receive wireless signals for both training and testing. However, our evaluation results can be fully applied to commodity WiFi devices by modifying their firmware or drivers, without involving extra RF hardware.

We introduce interferences from 1) white Gaussian noise, 2) concurrent WiFi transmissions, 3) commodity Xbee S2C ZigBee transmitters[5], 4) an Anmeate SM24 baby monitor and 5) a Westinghouse WM009 microwave oven, as shown in Figure 20. The interference patterns from these sources, as shown in Figure 19, are significantly different: interferences from WiFi and ZigBee transmitters cover fixed bands, the baby monitor transmits a 4MHz Frequency-hopping spread spectrum (FHSS) signal, and the microwave transmits a continuous wideband signal that covers >40MHz.

We evaluate AiFi's performance of interference cancellation by using bit error rate (BER) and frame reception rate (FRR) as metrics. The performance of AiFi is compared with the following WiFi interference cancellation schemes:

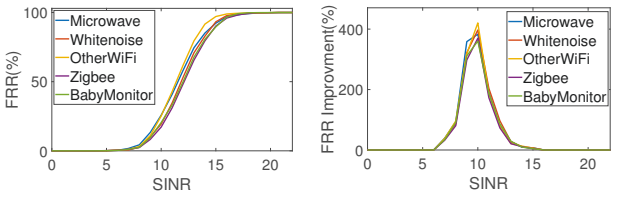
- **OpenRF** [49], which uses MIMO to compute an interference matrix and avoids such interference by using this matrix to instruct WiFi beamforming. OpenRF operates with extra MIMO hardware.
- **802.11n$^+$** [53], which probes the WiFi interference with extra RF antennas and cancels such interference by computing the difference in the channel coefficients between MIMO antennas. 802.11n$^+$ requires additional MIMO antennas to operate.
- **Rodin** [14], which detects the frequency-domain location of narrowband interference and hops the wideband signals to a new spectrum to avoid interference. Rodin adds additional RF frontend circuits in order to mitigate the interference.

7.1 BER Reduction and FRR Improvement

First, we evaluate AiFi's performance of reducing the BER in the received data frames with interference. The results are averaged from using all 5 types of interference sources. As shown in Figure 21, with different WiFi modulation schemes, using AiFi always reduces the minimum SINR required to reach 1% BER by >4dB. In particular, when higher-order modulations are used, the amount of correctable bit errors reduces due to WiFi's higher requirement on channel quality. On the other hand, when the amount of bit errors is very small, the percentage of BER reduction quickly drops to 0%.

Similarly, the amount of bit errors caused by interference grows when interference becomes stronger and SINR correspondingly becomes lower. Results in Figure 21 show that AiFi can significantly

[5]https://www.digi.com/resources/documentation/Digidocs/90002002/. We use 4 Zigbee transmitters with different center frequencies in the same WiFi band to introduce interference to WiFi. These Zigbee transmitters can interfere 50% bandwidth of a WiFi channel and hence greatly reduce WiFi performance.



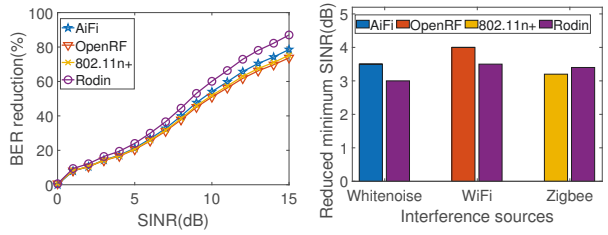
(a) FRR (b) FRR improvement

Figure 23: Correcting frame errors with different interference sources

improve BER in cases of low SINR and strong interference. On the other hand, when SINR is high, the data bit errors caused by interference become fewer, but the percentage of corrected bit errors, in these cases, always remains >50%.

We then evaluate AiFi's performance in improving the FRR by correcting frame errors, which is more difficult because a received data frame is erroneous if any data bit is erroneous. As shown in Figure 22, AiFi can improve the FRR by up to 18x, and such FRR improvement reduces the minimally required SINR to achieve 90% FRR by >3dB with all modulation schemes. In practice, such 3dB difference allows using a higher-order modulation[6], hence potentially leading to 100% improvement of WiFi performance (e.g., by using 16QAM instead of QPSK when SINR is 14-15dB). In addition, with the FRR improvement, AiFi can also reduce the MAC-layer delay by avoiding large TCP backoff window and long DCF backoff timer. Such latency reduction will be further evaluated with practical applications in Section 8.

We also evaluted AiFi's performance of correcting frame errors with different interference sources. As shown in Figure 23(a), AiFi can achieve similar FRR when different interference sources are involved, demonstrating that NN designs in AiFi can well adapt to different interference patterns.



(a) BER reduction (b) SINR improvement

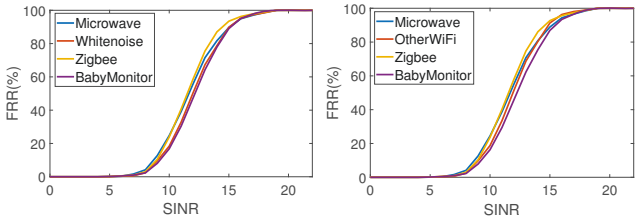
Figure 24: Comparison with existing schemes

We compared AiFi with the existing interference cancellation schemes. As shown in Figure 24(a), AiFi achieves the similar performance with the existing schemes that use extra RF hardware, and Figure 24(b) shows that AiFi reduces the minimally required SINR to reach 1% BER by >10%. Since AiFi does not use any extra RF hardware, it can be easily applied to commodity WiFi devices.

7.2 Generality of AiFi

Interference patterns and wireless channel conditions in different application scenarios and environment settings could be heterogeneous. Once AiFi's NN models have been trained, we expect

[6]In 802.11 standards, the minimum SINRs for using different modulation schemes are mostly defined with 3dB intervals. For example, the minimum SINR to use BPSK and QPSK is 7.7dB and 10.5dB, respectively [72].

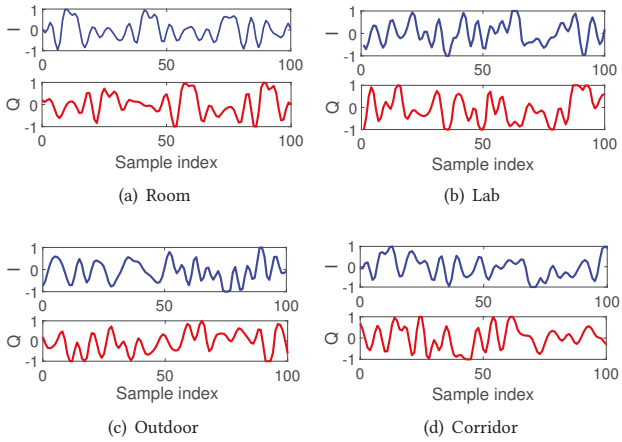


(a) Trained with WiFi (b) Trained with white noise

Figure 25: Generality over different interference sources

that they can be generically applied to different scenarios without having to retrain the models. To verify such generality, we trained AiFi's NN models only using interference signals from concurrent WiFi transmissions and Gaussian white noise in the lab site, and then applied the trained NN models for interference cancellation over other types of interference sources and on other test sites.

First, as shown in Figure 25, the trained model achieves similar FRR when taking signals from different interference sources as inputs. Second, when we transmit the same QPSK WiFi signal as interference with a fixed distance of 2m and SINR at 4dB, Figure 26 shows that the interference signal patterns at different test sites are significantly different. In these cases, as shown in Figure 27, the trained model exhibits <7% difference in BER reduction at these different test sites.



(a) Room (b) Lab

(c) Outdoor (d) Corridor

Figure 26: The interference signal patterns, shown as I/Q signal samples, at different test sites

Based on these results, we conclude that AiFi's NN designs have sufficient representation power to learn the underlying invariant correlation between PI/CSI information and the corresponding interference signal, which is mainly determined by WiFi PHY operations instead of environmental conditions. Hence, AiFi has good generality to be applicable to different wireless scenarios and environment settings.

7.3 Performance with Multiple Types of Co-Existing Interferences

Interference from multiple sources could possibly co-exist. We evaluated AiFi's performance of interference cancellation with multiple types of interference sources co-exist. Results in Figure 28(a) show that AiFi can achieve a similar level of performance in such scenarios with multiple types of interference sources, when the cumulative

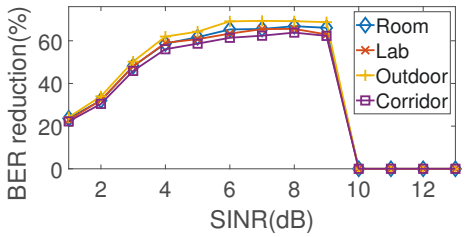
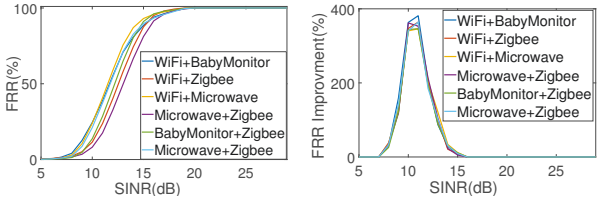


Figure 27: BER reduction at different test sites

SINR of these interference sources varies from 5dB to 20dB. Furthermore, as shown in figure 28(b), even when multiple types of interference sources co-exist, the FRR improvement achieved by AiFi can still be up to 390% when SINR is 10dB. Such improvement is only 2.5% lower than that with a single interference source, as shown in Figure 23(b). These results, hence, demonstrate that AiFi can provide high performance of interference cancellation when multiple types of interference sources co-exist.

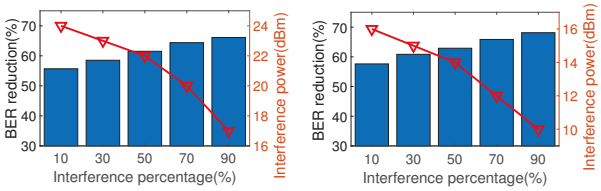


(a) FRR (b) FRR improvement

Figure 28: AiFi performance with multiple types of co-existing interferences

7.4 Impact of Interference in Time and Frequency Domains

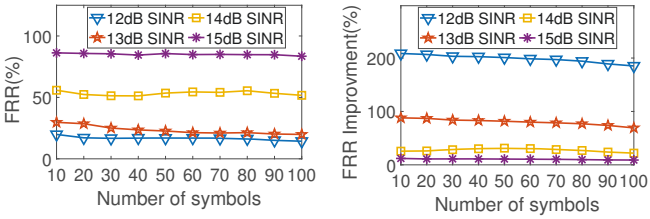
Interferences with the same SINR may impact the WiFi network in different ways, due to their different characteristics in time and frequency domains. For example, the interference signal could be short-time pulses that cover wide bands in the frequency domain, or continuous waves at few carrier frequencies. To investigate the impact of such interference's heterogeneity, besides the practical interference sources being used above, we train AiFi's NN models with completely interfered WiFi QPSK frames in both time and frequency domains, and then test the trained NN models with partially interfered frames in time and frequency domains, by retaining a fixed 4dB SINR but varying the interference's time duration and bandwidth from 10% to 90%, respectively.



(a) Time domain (b) Frequency domain

Figure 29: Performance on interferences with heterogeneous time and frequency domain characteristics

Experiment results in Figure 29(a) show that when applied to interferences with heterogeneous time-domain patterns, AiFi's BER reduction only exhibits <10% variation. Similarly, low variation is shown in Figure 29(b) for interference with different frequency-domain variations. These results further demonstrate AiFi's adaptability and generality over different interference patterns.



(a) FRR after AiFi (b) FRR improvement

Figure 30: Performance with different frame lengths

7.5 Impact of Different WiFi Frame Lengths

In practice, AiFi should be able to correct the errors in frames with different lengths. We conducted experiments to verify this with different levels of SINR. As shown in Figure 30, when the number of data symbols in a frame varies from 10 to 100, the achieved FRR in AiFi has less than 2% variation. Furthermore, Figure 30(b) shows that AiFi's performance in FRR improvement only drops by 5% with the longest frame length, due to the higher chance of burst interference in a longer data frame.

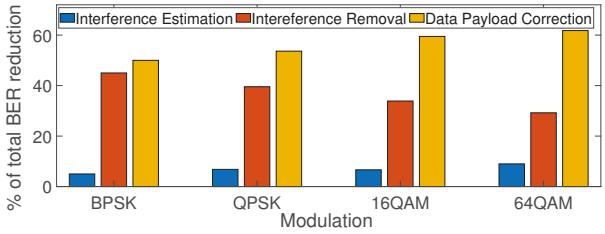


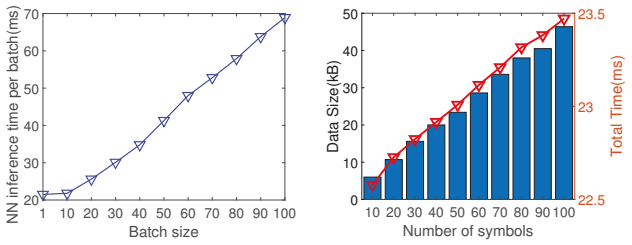
Figure 31: Different NN modules' contributions to bit error correction

7.6 Contribution from Different NN Modules

To investigate the individual NN modules' contributions to interference cancellation, we calculate the percentage of bit errors corrected by each individual NN module, by disabling all the other modules during testing. As shown in Figure 31, the Payload Correction NN makes the biggest contribution to bit error correction, especially when high-order modulation (e.g., 64QAM) is used and more data bits are encoded into one received signal symbol. In these cases, correcting data bit errors needs more fine-grained investigation into the interdependency across multiple data bits and this can only be achieved by the Payload Correction NN. On the other hand, although the Interference Estimation module only makes <20% direct contribution to bit error correction, it is still essential to other NN modules in interference cancellation, because the correctly extracted features help improve the accuracy of interference cancellation in other NN modules.

7.7 Latency of NN Inference

To make AiFi applicable to practical WiFi scenarios, we expect that the NN inference in AiFi is sufficiently lightweight to meet the



(a) Different batch sizes (b) Different frame lengths

Figure 32: Latency of NN inference

timing constraint in WiFi PHY. In our experiments, we evaluate the latency of AiFi's NN inference with different batch sizes and frame lengths. As shown in Figure 32(a), when the batch size is 1, it takes an average of 22ms for AiFi to process one data frame, but when a large batch size of 100 is used, it takes AiFi 72ms to process 100 frames and reduces the per-frame inference latency to 0.72ms.

In practice, such batch frame processing is commonly used in 802.11g/n/ac networks with Automatic Repeat ReQuest (ARQ), which reduces the network latency by only sending out one ACK after a batch of buffered frames [5], and the ARQ window size is usually set as 128 in most Linux systems. The recently proposed Hybrid-ARQ Protocol (HARQ), on the other hand, disables batch processing of frames but has only been adopted in cellular networks [69]. Hence, AiFi's per-frame processing latency in practical WiFi systems can be well controlled within 1ms.

Further, we studied the total time for AiFi to correct a failed frame with different lengths. As shown in Figure 32(b), even when the frame length increased from 10 symbols to 100 symbols, the total latency only increased less than 1ms.

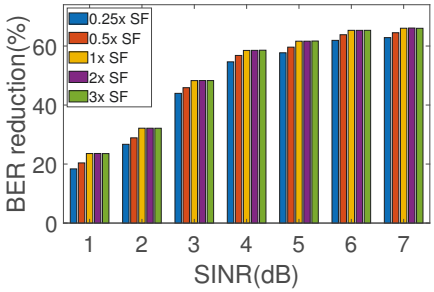


Figure 33: The impact of NN complexity

7.8 Impact of NN Complexity

The complexity of AiFi's NN models may impact their representation power and hence affect the accuracy of data payload correction under dynamic channel conditions. To evaluate such impact of NN complexity, we apply a scaling factor (SF) on the number of features in each layer of the AiFi's NN models[7], to vary the NN complexity.

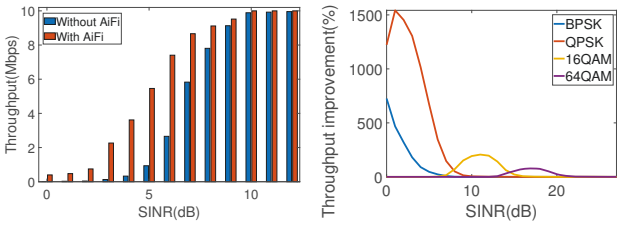
As shown in Figure 33, AiFi's performance of BER reduction exhibits 15%-20% reduction under different levels of SINR, when the complexity of NN models is reduced to 25% (SF=0.25). However, when SF is larger than 1, further increasing the NN complexity only results in another 4% improvement on the performance of bit error correction. In practical applications, the network users can flexibly adjust the NN complexity based on the application's timing constraint and resource conditions of wireless devices. When the local computing resources are abundant, the users can opt to use more complicated NNs for the optimal network performance. On the other hand, in resource-constrained or delay-sensitive applications, more lightweight NNs can be used instead to reduce the latency and resource consumption of AiFi's NN inference.

[7]In our implementation, Feature Extraction NNs and Interference Removal NNs both output 128 signal features, and Payload Correction NN and Demodulation NN process data payloads with 300 features.

8 REAL-WORLD EXPERIMENTATION

In this section, we further examine the applicability of AiFi in improving the network performance in real-world applications. We demonstrate that AiFi's interference cancellation can significantly enhance the Quality of Service (QoS) in the following applications: 1) wireless sensing; 2) webpage loading; 3) online gaming.

In all applications, our evaluations are performed at the application layer by using transport-layer throughput and latency as metrics. In Linux, we set the TCP window size as 416KB via the `net.ipv4.tcp_rmem` command, and retransmission timeout as 200ms via the `TCP_RTO_MIN` variable.



(a) Throughput (b) Throughput improvement

Figure 34: Throughput in wireless sensing

8.1 Wireless Sensing

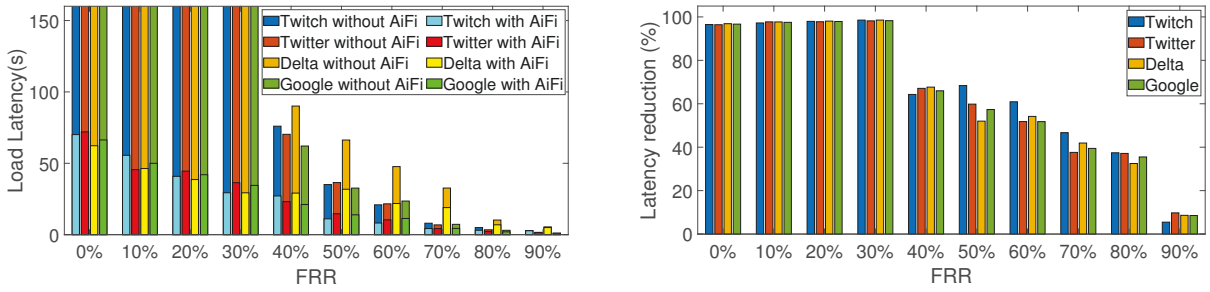
In wireless sensing applications [14, 82], low-cost sensors with small form factors are deployed in a distributed manner and stream sensory data to the backend server. In our evaluations, we examined how AiFi can improve the wireless throughput in sensing applications, by transmitting 10k images from a Raspberry Pi 4 to our WiFi receiver under interference with various levels of SINR.

As shown in Figure 34(a), as long as SINR exceeds 6dB, AiFi can remove the majority of interference and achieves 80% of the maximum throughput (i.e., the throughput without interference). Such throughput improvement, as shown in Figure 34(b), is up to 15x even under strong interference. These results show that AiFi can enable wireless sensing applications in severely interfered wireless channels.

8.2 Webpage Loading

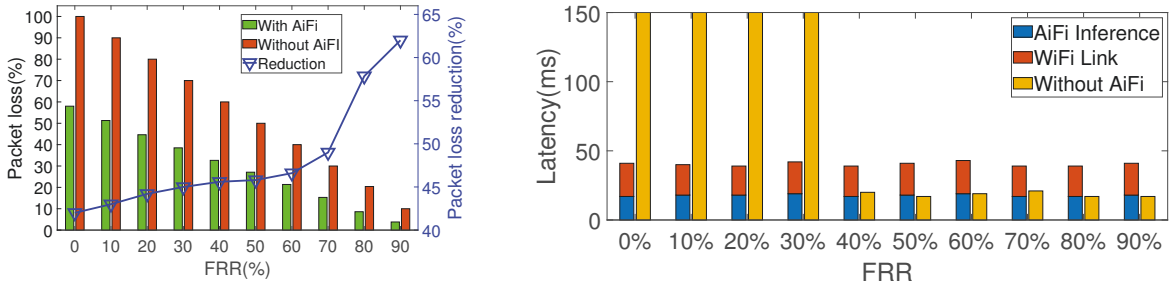
Web browsing is one of the most popular application scenarios using WiFi, and has also become common on mobile and embedded wireless devices. When the WiFi network transmits at 24Mbps, we evaluate how AiFi can reduce the latency of webpage loading by removing the impact of wireless interference. Our evaluation is performed on multiple web pages with different contents and data sizes as listed in Table 1.

As shown in Figure 35(a), when the interference is strong and reduces the FRR to <30%, the HTTP link to the web server cannot be established, resulting in an indefinitely long latency of webpage



(a) Latency (b) Percentage of latency reduction

Figure 35: Latency of webpage loading



(a) Packet Loss (b) Transmission Latency

Figure 36: Performance of online gaming

loading. In these cases, interference cancellation in AiFi is able to transform the webpage loading functionality from impossible to usable, and the average loading latency is within 30s when FRR is at 40%. Furthermore, in scenarios with weaker interference, AiFi's interference cancellation can reduce the latency of webpage loading by at least 42%, as shown in Figure 35(b).

Webpage	Data size (MB)
Google.com	1.4
Delta.com	6.08
Twitter.com	8.21
Twitch.com	14.81

Table 1: Webpage size

In particular, when the FRR reaches 80%, AiFi can constrain the latency of webpage loading within 3s, which ensures satisfiable QoS of web browsing [8, 12, 31].

8.3 Online Gaming

In this section, AiFi takes wireless traffic during online gaming as inputs, and we evaluate how AiFi reduces the packet loss and data transmission latency in real-time online gaming. In our evaluations, we collected the UDP data traffic from League of Legends[8] and Dota2[9] via Wireshark, and transmit such UDP traffic under various interference conditions that result in different levels of FRR.

As shown in Figure 36(a), when interference causes >10% packet loss, AiFi can reduce such amount of packet loss by >63%. In general, as long as the network FRR is higher than 80%, AiFi can effectively restrain the packet loss rate within 10% and hence ensures satisfiable user experience in gaming, according to existing studies about online games' requirements on network link stability [13, 15, 16].

[8]https://www.leagueoflegends.com/
[9]https://www.dota2.com/home

Meanwhile, the improved performance of frame reception also leads to a significant reduction on the data transmission latency, also known as the *ping* value in online gaming. As shown in Figure 36(b), even with very strong interference that reduces the FRR to <30% and incurs unacceptable transmission latency, AiFi can always reduce such latency to <50ms. On the other hand, when the FRR improves, although using AiFi introduces some extra computing latency for NN inference, it still provides significant packet loss reduction, which is critical to good user experience.

9 RELATED WORK

Collision avoidance. Commodity WiFi implements MAC-layer CSMA/CA for collision avoidance [11, 23, 75]. Advanced techniques such as Q-CSMA [26, 60], slotted CSMA [47, 68] and distributed CSMA [39, 67] reduce the network latency by adapting the backoff timer, packet sizes and timing constraints. Other researchers insert custom PHY preambles that can be sensed with a smaller latency, to avoid interference from different wireless technologies such as Zigbee [84] and LTE [62]. These approaches, however, introduce significant delay when the channel is heavily congested. In contrast, AiFi completely avoids such delay by performing interference cancellation in nearly real time.

Cross-technology communication (CTC) [33, 40, 52, 54], on the other hand, advances collision avoidance by enabling explicit coordination between wireless technologies, but still incurs extra delay when waiting for an idle channel to exchange control messages, especially if the channel is intensively occupied.

Interference cancellation. Interference cancellation [28, 49, 63] removes interference from the received signal, by using RF hardware to probe the interference signal on the air. Other backbone-assisted networks [30, 53, 86] cancel interference between clients and WiFi APs by coordinating multiple APs, and full-duplex radios

[6, 10, 17–19, 35] cancel the transmitted signal as self-interference from the simultaneously received signal in another antenna. However, all these techniques build on sensing interference with extra RF hardware, and cannot be applied to commodity wireless devices. In contrast, AiFi does not require any extra hardware in WiFi PHY, allowing easy deployment in commodity WiFi devices.

Zigzag decoding [29], on the other hand, reduces the requirement of extra RF hardware, but still requires access to WiFi PHY to obtain knowledge about the raw received signal. Furthermore, Zigzag decoding is limited to removing interference signal from another WiFi transmitter and utilizes the known patterns of the interfering WiFi signal to achieve lightweight interference cancellation, but it is incapable of canceling interferences from other types of sources. Similarly, early research efforts on packet recovery are mostly limited to specific types of networks [24, 38, 79], and existing successive interference cancellation (SIC) techniques [57, 76] also build on the prior knowledge about both the transmitted data signal and the interference signal. In contrast, the NN models in AIFi are able to remove a large collection of interference signals with heterogeneous signal patterns, without requiring any prior knowledge about such patterns of interference signals.

AI-assisted wireless communication. In recent years, research efforts have been made to utilize modern AI techniques to improve the performance of wireless communication. In particular, deep neural networks (DNNs) have been used to develop better encoders and decoders [36, 37, 43, 78] as the replacement of those being current used in wireless PHY, in order to better addressing the non-linearity in channel conditions and the subsequent bit errors caused by such non-linearity. In contrast, our design of Payload Correction NN is not a new decoder design and does not modify the WiFi PHY-layer in anyway, but instead aims to mimic the WiFi PHY's decoding operations for efficient bit error correction in software. Some other techniques use AI tools for more accurate channel estimation [25, 51, 59, 65, 71], which can contribute to correcting bit errors caused by interference. Most of these techniques, however, design custom neural networks based on specific wireless channel models, and cannot well adapt to the dynamic channel conditions caused by interference. In contrast, since the Payload Correction NN in AiFi is trained with the extracted interference features, it can well adapt to the heterogeneous and dynamic interference patterns in different scenarios and environment settings.

Researchers have also developed and applied DNN-based encoders and decoders to different applications, such as respiration monitoring [85] and jamming removal [21]. While AiFi similarly uses the learning power of NNs to better capture the non-linearity of channel conditions and hence achieve generality across different application domains and environment settings, its objective of interference cancellation is different from the existing work and it hence has a different requirement of the NN's generalizability compared to the existing work. For example, while respiration monitoring requires highly accurate motion tracking to capture each breath, interference estimation in WiFi does not need to be 100% accurate, as the possible estimation errors can be further addressed by Interference Removal and Payload Correction NNs.

10 DISCUSSIONS

Applicability to different WiFi hardware. On most WiFi devices, the PI and CSI information at PHY are stored in hardware registers for PHY channel equalization and demodulation, and can hence be accessed from software via custom WiFi drivers that provide access to these registers. For example, on QualComm Atheros WiFi chipsets, CSI is stored in the `ar9003_hw_set_chain_masks` registers, and can be accessed by reversely engineering the WiFi firmware [80] without any hardware modification. To that end, AiFi can be widely deployed to a large collection of commodity WiFi devices with manageable engineering efforts on WiFi driver or firmware customization.

Impact of device mobility. As we discussed in Section 2.1, device mobility could also cause non-linearity in the WiFi PHY information, due to the Doppler effect being caused. However, even in scenarios with very high mobility (e.g., >100 km per hour), the corresponding frequency shift caused by device mobility, with the WiFi carrier frequency at 2.4 GHz, is capped at a few hundreds of hertz. Such phase shift is much smaller than the smallest data subcarrier spacing in WiFi (312.5 KHz). Therefore, the phase shift and amplitude change caused by device mobility are relatively minor and can be efficiently distinguished from those caused by interference by the NN modules in AiFi.

Acceleration with hardware AI accelerators. In our current implementation, the online inference of AiFi is only executed by CPU. To meet the more strict timing constraint of delay-sensitive applications, such as AR/VR, we can leverage the hardware AI accelerators that have been available on personal wireless devices. For example, neural processing units (NPU) have been made available on smartphones such as Samsung Galaxy S20 and Google Pixel 6, and can be used to further reduce the latency of AiFi's online inference.

Applicability to different wireless technologies. In theory, AiFi is applicable to any OFDM-based wireless systems, as long as the required PHY information is available. However, some wireless systems do not share the same PHY structure or do not provide the PHY information in the same way. For example, LTE networks do not use LTF preambles and have different pilot structures. Adopting AiFi to these systems, hence, requires additional engineering efforts, based on knowledge about PHY information in these systems.

Online model adaptation. In practice, the wireless channel and interference signals may be highly variant over time. To better adapt to such temporal variability, one solution is to adopt active learning approaches [64] for online NN model adaptation, which uses the up-to-date PHY information to re-calculate the NN model weights. Such online model adaptation will be our future work.

11 CONCLUSION

In this paper, we present AiFi, a new wireless system that enables WiFi interference cancellation at commodity WiFi devices without requiring any extra RF hardware. The basic rationale of AiFi design is to extract patterns of interference from the PHY-layer information that is locally available at WiFi receivers. By levering the power of AI and designing NNs from domain knowledge of WiFi PHY, AiFi can reduce 80% of bit errors and improve the FRR by up to 18x.

ACKNOWLEDGMENTS

We thank the anonymous shepherd and reviewers for their comments and feedback. This work was supported in part by National Science Foundation (NSF) under grant number CNS-1812407, CNS-2029520, IIS-1956002, IIS-2205360, CCF-2217003 and CCF-2215042.

REFERENCES

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard 802.11-2012*, March 2012.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[3] S. Aheleroff, X. Xu, Y. Lu, M. Aristizabal, J. P. Velásquez, B. Joa, and Y. Valencia. Iot-enabled smart appliances under industry 4.0: A case study. *Advanced engineering informatics*, 43:101043, 2020.

[4] M. S. Akram. Pilot-based channel estimation in OFDM systems. *Master of Science Thesis*, 2007.

[5] M. Anagnostou and E. Protonotarios. Performance analysis of the selective repeat arq protocol. *IEEE Transactions on Communications*, 34(2):127–135, 1986.

[6] E. Aryafar and A. Keshavarz-Haddad. Fd 2: A directional full duplex communication system for indoor wireless networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1993–2001. IEEE, 2015.

[7] A. Astrin. Ieee standard for local and metropolitan area networks part 15.6: Wireless body area networks. *IE EE Std 802.15. 6*, 2012.

[8] S. Barber. How fast does a website need to be?, 2015.

[9] B. Bellalta. Ieee 802.11 ax: High-efficiency wlans. *IEEE Wireless Communications*, 23(1):38–46, 2016.

[10] D. Bharadia, E. McMilin, and S. Katti. Full duplex radios. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 375–386, 2013.

[11] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans. In *Proceedings of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 392–396. IEEE, 1996.

[12] BlueCorona. How fast should a website load? https://www.bluecorona.com/blog/how-fast-should-website-be/.

[13] M. Bredel and M. Fidler. A measurement study regarding quality of service and its impact on multiplayer online games. In *2010 9th Annual Workshop on Network and Systems Support for Games*, pages 1–6. IEEE, 2010.

[14] E. Chai, J. Lee, S.-J. Lee, R. Etkin, and K. G. Shin. Building efficient spectrum-agile devices for dummies. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 149–160, 2012.

[15] Y.-C. Chang, K.-T. Chen, C.-C. Wu, C.-J. Ho, and C.-L. Lei. Online game qoe evaluation using paired comparisons. In *2010 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2010)*, pages 1–6. IEEE, 2010.

[16] K.-T. Chen, P. Huang, and C.-L. Lei. Effect of network quality on player departure behavior in online games. *IEEE Transactions on Parallel and Distributed Systems*, 20(5):593–606, 2008.

[17] L. Chen, F. Wu, J. Xu, K. Srinivasan, and N. Shroff. Bipass: Enabling end-to-end full duplex. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 114–126, 2017.

[18] T. Chen, M. Baraani Dastjerdi, J. Zhou, H. Krishnaswamy, and G. Zussman. Wideband full-duplex wireless via frequency-domain equalization: Design and experimentation. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[19] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 1–12, 2010.

[20] M. Communications. 802.11 reference design for warp v3. In *warpproject.org/trac/wiki/802.11*, 2013.

[21] K. Davaslioglu, S. Soltani, T. Erpek, and Y. E. Sagduyu. Deepwifi: Cognitive wifi with deep learning. *IEEE Transactions on Mobile Computing*, 20(2):429–444, 2019.

[22] Y. Ding, T. N. Davidson, Z.-Q. Luo, and K. M. Wong. Minimum ber block precoders for zero-forcing equalization. *IEEE Transactions on Signal Processing*, 51(9):2410–2423, 2003.

[23] E. Felemban and E. Ekici. Single hop ieee 802.11 dcf analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases. *IEEE Transactions on Wireless Communications*, 10(10):3256–3266, 2011.

[24] R. K. Ganti, P. Jayachandran, H. Luo, and T. F. Abdelzaher. Datalink streaming in wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 209–222, 2006.

[25] J. Gao, M. Hu, C. Zhong, G. Y. Li, and Z. Zhang. An attention-aided deep learning framework for massive mimo channel estimation. *IEEE Transactions on Wireless Communications*, 21(3):1823–1835, 2021.

[26] J. Ghaderi and R. Srikant. Effect of access probabilities on the delay performance of q-csma algorithms. In *2012 Proceedings IEEE INFOCOM*, pages 2068–2076. IEEE, 2012.

[27] J. Gold. 2.4ghz is a headache for wi-fi users, and it's here to stay. Network World. https://www.networkworld.com/article/3192984/2-4ghz-is-a-headache-for-wi-fi-users-and-it-s-here-to-stay.html.

[28] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM 2011 Conference*, pages 170–181, 2011.

[29] S. Gollakota and D. Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 159–170, 2008.

[30] S. Gollakota, S. D. Perli, and D. Katabi. Interference alignment and cancellation. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 159–170, 2009.

[31] Google. Find out how you stack up to new industry benchmarks for mobile page speed. https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks/.

[32] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[33] X. Guo, Y. He, J. Zhang, and H. Jiang. Wide: physical-level ctc via digital emulation. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 49–60. IEEE, 2019.

[34] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM computer communication review*, 41(1):53–53, 2011.

[35] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 301–312, 2011.

[36] M. V. Jamali, X. Liu, A. V. Makkuva, H. Mahdavifar, S. Oh, and P. Viswanath. Reed-muller subcodes: Machine learning-aided design of efficient soft recursive decoding. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1088–1093. IEEE, 2021.

[37] M. V. Jamali, H. Saber, H. Hatami, and J. H. Bae. Productae: Toward training larger channel codes based on neural product codes. In *ICC 2022-IEEE International Conference on Communications*, pages 3898–3903. IEEE, 2022.

[38] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. *ACM SIGCOMM Computer Communication Review*, 37(4):409–420, 2007.

[39] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, 2009.

[40] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He. Bluebee: a 10,000x faster cross-technology communication via phy emulation. In *Proceedings of ACM SenSys*. ACM, 2017.

[41] Y. Jiang, M. K. Varanasi, and J. Li. Performance analysis of zf and mmse equalizers for mimo systems: An in-depth study of the high snr regime. *IEEE Transactions on Information Theory*, 57(4):2008–2026, 2011.

[42] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of ACM Conference on SIGCOMM*, pages 607–618. ACM, 2014.

[43] M. Kim, W. Lee, J. Yoon, and O. Jo. Building encoder and decoder with deep neural networks: On the way to reality. *arXiv preprint arXiv:1808.02401*, 2018.

[44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] A. Klein, G. K. Kaleh, and P. W. Baier. Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels. *IEEE transactions on Vehicular Technology*, 45(2):276–287, 1996.

[46] T. Kliestik, E. Nica, H. Musa, M. Poliak, and E.-A. Mihai. Networked, smart, and responsive devices in industry 4.0 manufacturing systems. *Economics, Management and Financial Markets*, 15(3):23–29, 2020.

[47] A. Koubaa, M. Alves, and E. Tovar. A comprehensive simulation study of slotted csma/ca for ieee 802.15. 4 wireless sensor networks. In *2006 IEEE international workshop on factory communication systems*, pages 183–192. IEEE, 2006.

[48] R. Kudo, S. M. Armour, J. P. McGeehan, and M. Mizoguchi. A channel state information feedback method for massive mimo-ofdm. *Journal of Communications and Networks*, 15(4):352–361, 2013.

[49] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi. Bringing cross-layer mimo to today's wireless lans. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 387–398, 2013.

[50] B.-J. Kwak, N.-O. Song, and L. E. Miller. Performance analysis of exponential backoff. *IEEE/ACM transactions on networking*, 13(2):343–355, 2005.

[51] L. Li, H. Chen, H.-H. Chang, and L. Liu. Deep residual learning meets ofdm channel estimation. *IEEE Wireless Communications Letters*, 9(5):615–618, 2019.

[52] Z. Li and T. He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of ACM MobiCom*. ACM, 2017.

[53] K. C.-J. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous mimo networks. *ACM SIGCOMM Computer Communication Review*, 41(4):146–157, 2011.

[54] R. Liu, Z. Yin, W. Jiang, and T. He. Lte2b: Time-domain cross-technology emulation under lte constraints. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 179–191, 2019.

[55] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: wireless communication out of thin air. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 39–50, 2013.

[56] V. Liu, V. Talla, and S. Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 67–78. ACM, 2014.

[57] N. I. Miridakis and D. D. Vergados. A survey on the successive interference cancellation performance for single-antenna and multiple-antenna ofdm systems. *IEEE Communications Surveys & Tutorials*, 15(1):312–335, 2012.

[58] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour. Wireless body area networks: A survey. *IEEE Communications surveys & tutorials*, 16(3):1658–1686, 2014.

[59] D. Neumann, T. Wiese, and W. Utschick. Learning the mmse channel estimator. *IEEE Transactions on Signal Processing*, 66(11):2905–2917, 2018.

[60] J. Ni, B. Tan, and R. Srikant. Q-csma: Queue-length-based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks. *IEEE/ACM Transactions on Networking*, 20(3):825–836, 2011.

[61] H. Rahul, H. Hassanieh, and D. Katabi. Sourcesync: a distributed wireless architecture for exploiting sender diversity. *ACM SIGCOMM Computer Communication Review*, 41(4):171–182, 2011.

[62] S. Rathinakumar, B. Radunovic, and M. K. Marina. Cprecycle: Recycling cyclic prefix for versatile interference mitigation in ofdm based wireless systems. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 67–81, 2016.

[63] L. Ruan and V. K. Lau. Dynamic interference mitigation for generalized partially connected quasi-static mimo interference channel. *IEEE Transactions on Signal Processing*, 59(8):3788–3798, 2011.

[64] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan. Active learning in recommender systems. In *Recommender systems handbook*, pages 809–846. Springer, 2015.

[65] J. Schmitz, C. von Lengerke, N. Airee, A. Behboodi, and R. Mathar. A deep learning wireless transceiver with fully learned modulation and synchronization. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2019.

[66] P. Schniter. Low-complexity equalization of ofdm in doubly selective channels. *IEEE Transactions on Signal processing*, 52(4):1002–1011, 2004.

[67] D. Shah and J. Shin. Delay optimal queue-based csma. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):373–374, 2010.

[68] N. Shahin, R. Ali, and Y.-T. Kim. Hybrid slotted-csma/ca-tdma for efficient massive registration of iot devices. *IEEE Access*, 6:18366–18382, 2018.

[69] G. Shishir, R. Sonigra, N. Seshadri, and R. D. Koilpillai. Hybrid-arq protocol for next generation wi-fi systems. In *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 342–350. IEEE, 2021.

[70] S. Sibecas, C. A. Corral, S. Emami, G. Stratis, and G. Rasor. Pseudo-pilot ofdm scheme for 802.11 a and r/a in dsrc applications. In *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, volume 2, pages 1234–1237. IEEE, 2003.

[71] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019.

[72] J. Thomson, B. Baas, E. M. Cooper, J. M. Gilbert, G. Hsieh, P. Husted, A. Lokanathan, J. S. Kuskin, D. McCracken, B. McFarland, et al. An integrated 802.11a baseband and MAC processor. In *IEEE International Solid-State Circuits Conference*, pages 126–451, 2002.

[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[74] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *stat*, 1050:20, 2017.

[75] X. Wang and K. Kar. Throughput modelling and fairness issues in csma/ca based ad-hoc networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 1, pages 23–34. Ieee, 2005.

[76] M. Wildemeersch, T. Q. Quek, M. Kountouris, A. Rabbachin, and C. H. Slump. Successive interference cancellation in heterogeneous networks. *IEEE transactions on communications*, 62(12):4440–4453, 2014.

[77] P. K. Wong, D. Yin, and T. T. Lee. Analysis of non-persistent csma protocols with exponential backoff scheduling. *IEEE Transactions on Communications*, 59(8):2206–2214, 2011.

[78] D. Wu, M. Nekovee, and Y. Wang. Deep learning-based autoencoder for m-user wireless interference channel physical layer design. *IEEE Access*, 8:174679–174691, 2020.

[79] J. Xie, W. Hu, and Z. Zhang. Revisiting partial packet recovery in 802.11 wireless lans. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 281–292, 2011.

[80] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wi-fi. *IEEE Transactions on Mobile Computing*, 18(6):1342–1355, 2018.

[81] Y. Yu, X. Si, C. Hu, and J. Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[82] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang. Mobile edge computing and networking for green and low-latency internet of things. *IEEE Communications Magazine*, 56(5):39–45, 2018.

[83] P. Zhang, J. Gummeson, and D. Ganesan. Blink: A high throughput link layer for backscatter communication. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 99–112. ACM, 2012.

[84] X. Zhang and K. G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *IEEE INFOCOM*, 2013.

[85] T. Zheng, Z. Chen, S. Zhang, C. Cai, and J. Luo. More-fi: Motion-robust and fine-grained respiration monitoring via deep-learning uwb radar. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 111–124, 2021.

[86] W. Zhou, T. Das, L. Chen, K. Srinivasan, and P. Sinha. Basic: Backbone-assisted successive interference cancellation. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 149–161, 2016.