JamGuardian - Milcom '24

***Abstract*—Despite the unprecedented growth of wireless communication with the emerging applications in Wi-fi, bluetooth and IoT, it is still vulnerable to inferences and jamming from both independent (e.g., microwave oven) and adversarial sources (e.g., enemy jammers). Although techniques like channel retreat, frequency hopping, jammer localization and avoidance exist to encounter jamming and interference, the very basic form where the a receiver can distinguish a sender's data from the superimposed signal from multiple transmitting entities in the same frequency channel, is still an open problem. We presents a novel approach of inferring sender's bit by looking at received amplitude levels in each bit period of sender and intelligently varying sender's rate as required, thereby making it challenging for jammers to disrupt communication consistently. The techniques we use for this purpose include renormalization to map received symbols to appropriate levels, heuristic-based historical prediction for inferring regions of uncertainty in the signals, and varying transmission time of each bit to evade the jammer. Extensive simulations demonstrate that our approach achieves a ∼2x higher accuracy supporting its effectiveness in maintaining reliable communication.**

***Index Terms*—Wireless network, interference, jamming**

I. Introduction

Shadman: We'll have to remove the points, write them as paragraphs and connect them coherently. Still just a draft, can be modified. Shadman: The following is a skeleton:

- Motivation: Wireless is very important in modern military communication - large scale deployment, rescue operation, coordination activities. Convergence in 5G/6G moving from older techniques e.g., adhoc.
- Putting the work into context: Wireless communication prone to threats from adversaries. Countermeasures on integrity violation, confidentiality solved with encryption, availability not solved because of jamming/interference. Existing techniques does ... to mitigate jamming but they cannot handle ...
- Technique: We design a technique that applies domain knowledge to narrow down the subspace of sender data inference. Our technique first maps the received symbols to appropriate levels and subtracts the known amplitude range from the recived ones. We also augment the capability of signal learning with embedded preambles or during the periods of silence to have more educated guesses on jammer's signal when required. Furthermore, we adopt historical filtering to subtract long-term trends in jamming patterns so that it averages out the highly fluctuating random jamming attempts. Moreover, we vary the sender's rates pseudo-randomly to evade the jammer as that helps with their rates being different and thereby inferring sender's data with higher certainty. Finally we combine these strategies and pick the most appropriate ones by checksum matching.
- Evaluation: We do extensive evaluation study of our approach through simulations by varying different parameters including but not limited ot the sender dataset size, sender to jammer rates etc. and compare it against the baseline ??. The results demonstrate the merit of our approaches achieving ??x higher accuracy (in terms of sender data reconstruction) and ??x higher throughput.
- Contributions - 1 paragraph
- Paper structure - 1 paragraph

Due to the prevalence of wireless communications and the availability of hardware, jamming has become a massive concern, [1]. In the past decades, wireless communication has become ubiquitous. Anyone in any part of the globe now has access to immediate communication with practically anybody else. While public use of wireless communication has increased, military use of the same technologies has all but exploded. Wireless communications provide the ability to communicate orders from command and control to the front lines, control autonomous drones, create large-scale surveillance networks, coordinate rescue efforts, and much more. All these use cases cause wireless communication to permeate the landscape of modern military applications. Furthermore, new 5G/6G technologies provide an even larger scope to utilize wireless communications in military applications (stronger and more widespread IoT, Wi-Fi, Bluetooth, etc.).

Wireless communication allows for long-range communication with little infrastructure required. However, the act of transmitting signals through a shared medium (generally air) is inherently unsafe, especially with an adversarial actor who tries to disrupt the signal. While electromagnetic signals can be altered or intercepted in transit to their destination, checksums and redundant transmissions help with integrity validation. Furthermore, physical checks like checking codes based on a physical reference, provide a nearly impenetrable method of ensuring received transmissions contain the correct information within. Besides, problems regarding a transmission's confidentiality can be solved through encryption of a sender's transmission. However, neither of these techniques can rectify a signal from the effects of jamming and interference resulting in poorer availability. Existing techniques try to avoid jamming actions from occurring by targeting the frequency or energy of a transmission [2], [3], but these techniques require devices that can transmit and receive either on multiple frequencies in quick succession or at vastly different energy levels.

In this work, we design a technique that attempts to mitigate jamming in the presence of adversaries transmitting on the

Identify applicable funding agency here. If none, delete this.

same frequency channel. Our technique applies domain knowledge to narrow down the subspace of sender data inference. Specifically, it maps the received symbols to appropriate levels and subtracts the known amplitude range from the received ones. We utilize heuristics to infer jammer's signal in uncertain regions to approximate this known amplitude. We also vary the sender's schedule pseudo-randomly to evade the jammer as that helps with their schedules being different, thereby inferring the sender's data with higher certainty.

We run simulations extensively by varying parameters like sender dataset size, sender to jammer schedule ratio etc. to measure the performance of our algorithm in terms of accuracy. The results compared to baseline approaches, see Section VI, demonstrate the merit of the new approaches, achieving 2x higher accuracy of sender data reconstruction.

Contribution. In this paper we provide the following contributions

1) We propose a new algorithmic method of decoding signals possibly jammed by a Jammer
2) We suggest an argument for adopting rate-hopping as a method to increase the efficacy our decoding algorithms
3) We demonstrate the merits of the aforementioned methods in unison between a three party system of a sender, jammer, and receiver through extensive simulations.

The rest of this paper is organized as follows. Section II discusses related body of works. Section III defines the problem more formally, and Section IV outlines our approach to solving it. The algorithms we utilize to decode transmissions are provided in Section V. Sections VI summarizes our findings, while Section VII concludes the paper with a brief summary and possible future directions.

II. Related Work

Shadman: A high level outline, may need to change as appropriate. The following is a skeleton

- Existing jamming mitigation techniques (e.g., jammer localization, frequency hopping etc.) - 2 paragraphs
- Related work in information theory (communication in presence of adversary) - 1 paragraph

To combat the threat of jamming, significant research has been put towards mitigating its effects. Physically, new sixth-generation communication infrastructure has been created to allow for varied jamming-resistant communication. It allows for the decoupling of transmitters and receivers for safer transmissions [4], and the implementation of Visible Light Communication (VLC) to facilitate jamming resilient communication in smaller settings, like indoors [5]. Despite providing some jamming resistance, VLC is still vulnerable to many attacks, including but not limited to DDoS and eavesdropping [6]. Further, these techniques only apply to new 6G communication networks; an infrastructure may not even be adopted until 2030 [7].

Without the physical upgrades 6G will provide, clever techniques must be employed. There has been much research into techniques that avoid a jammer entirely. Mpitziopoulos et. al [8] discusses the methodology behind decreasing transmission power to remain undetected by an adversarial jammer, although, decreasing transmission power causes increased data loss [9], making this method unreliable for long-range communication. Transmission data can also be encoded through Ambient Backscatter Communication, allowing signals to be transmitted while appearing as white noise [10]. Masking data as background noise makes it less likely for a jammer to pick up on the transmission. However, as described in [11], this approach does not work well in the scenario with a high-power jammer, unless the transmitter itself has unlimited-power.

In the presence of a jammer, there are multiple methods that a transmitter and receiver duo can employ to avoid or fight against the jammer. Foremost, the transmitter/ receiver duo can alter their frequency to transmit on some new, unjammed, channel [2]. While this method of anti-jamming works best in the case of a spot jammer [8], applying this technique to other types of jammers, like a sweep jammer [8] (one that sweeps available channels and jams on each), will yield worse results. In the case of a mobile transmitter and receiver, a device can elect to physically move out of the area affected by the jammer, a technique called spatial retreat, effectively removing the threat [12]. A transmitter/ receiver duo can also act in response to the presence of a jammer by increasing the power of transmissions [3], and thereby forcing the jammer and transmitter to compete in the transmission space. If the new power of a transmitter be high enough, it completely overpowers the jammer's attempts. These approaches overcome the effects of a jammer, but they require extemely specialized equipment to perform.

The transmitter/ receiver duo can also use jamming to avoid an adversarial party through a process called friendly jamming in many different ways. One such use case is maintaining wireless communication confidentiality. In [13], a process to fool an eavesdropper through selectively jamming repeated transmissions is developed. By transmitting a message twice and having the receiver, in this case, a transceiver, jam parts of each message, the entire transmission can be rectified without the ability of a potential adversary to understand the transmission. Likewise, in [14], dialog codes are developed. These work similarly to [13]: selectively jamming part of a message in a manner that is reversible to obfuscate its meaning to any eavesdropper. In the case of [14], this means jamming one of every two bytes, and then using previously defined dialog codes to rectify the jammed data. This effectively makes the transmission intelligible, while allowing for correct decoding.

Pelechrinis et al. [15] have designed a system they term ARES to determine the effects of applying a rate hopping evasion scheme to avoid the effects of jamming across various types of jamming decives. However, they were most concerned about the throughput of such a design, rather than the possible schemes in which to decode the jammed signals.

Further, in [16], rate adaptation is tested through a game theory lens, in conjunction with frequency hopping, in order to determine the best approach to avoid jamming. However, they

were most concerned with avoiding jamming, not decoding a jammed signal, and compared the combined effects of frequency hopping and rate adaptation, instead of rate adaptation alone.

III. Problem Statement

Shadman: Somewhat complete.

Assume, a wireless communication takes place for n time slots on the same frequency channel where three entities - a sender, a jammer and a receiver interact with each other. The sender's data, $S = x_1, x_2, \dots x_s$, and the jammer's data, $J = y_1, y_2, \dots y_j$. Each symbol in the transmitted signals can be either 0 or 1, i.e., $\forall x \in S, x \in \{0, 1\}$ and $\forall y \in J, y \in \{0, 1\}$.

Suppose, the receiver's received signal, $R = z_1, z_2, \dots z_n = \phi(S, J)$ where $\phi(.)$ captures the characteristics of the combination of S and J. Since the signals in wireless communication gets added up in case of overlapping, let us consider that ϕ adds up the values, x and y, in each time slot, for simplicity. In other words, as R consists of n symbols, each corresponding to one time slot, and each of x and y can span over multiple time slots, the received symbol at t-th time instant, $z_t = x_t + y_t$ where $(x_t \in \{x_1, x_2 \dots x_s\})$ and $y_t \in \{y_1, y_2 \dots y_j\})$ denote the symbols, that span the t-th time slot, transmitted by the sender and the jammer respectively. Therefore, the symbols in the received signal can evaluate to 0, 1, or 2, i.e., $\forall z \in R, z \in \{0, 1, 2\}$.

Let $\psi(.)$ be a mechanism to estimate S from R, i.e., $\psi(R) = \overline{S}$. Assume, $\Delta(S, \overline{S})$ denotes the distance between S and $\overline{S}$, which captures the error in reconstructing S, e.g., bit error rate in wireless communication. So, given R, our goal is to find $\psi(.)$ that minimizes $\Delta(S, \overline{S})$.

IV. Solution Approach

Shadman: A high level outline:

- Observations - 1 paragraph
- Challenges - 1 paragraph
- High level overview - 2 paragraph

A. Observations

Multiple characteristics of the received signal provide deterministic insight into the true value of a sender's data. Given any set of analog signals received by the receiver, we can translate them into digital counterparts. Consider the function $\phi(S, J)$, described in Section III. For this function, 0 represents the receiver receiving a 0 from both transmitters, 1 represents a 1 bit from either the sender of the jammer, and 2 represents receiving a 1 bit from both transmitting devices.

Let us term 0 and 2 as extreme states of $\phi(.)$, and the times these states occur as extreme times. The extreme states of $\phi(.)$, provide deterministic information regarding the true value of the sender's data at that point in time. The only way for $\phi(.)$ to evaluate to 0 is when both the sender and jammer transmit a 0 bit. Similarly, the only scenario where $\phi(.) = 2$ occurs when the sender and the jammer both transmit a data bit of 1. Thus, whenever a receiver receives a symbol equal to either of the extreme states, the sender's transmission can be deterministically calculated. Conversely, receiving a 1 does not provide any deterministic insight into the true transmission value of the sender because $\phi(.) = 1$ can occur in two distinct scenarios: the sender sending a 1 and jammer sending a 0, or the sender sending a 0 and jammer sending a 1.

Thus, it follows that to ensure the highest accuracy between transmitted sender signals and decoded receiver signals we must create the most extreme times. Doing so will allow the receiver to deterministically recreate the sender's transmission, as described above. Ideally, if a system is able to maintain an extreme state over it's entire time, the sender's transmission will be perfectly decoded.

B. Overview

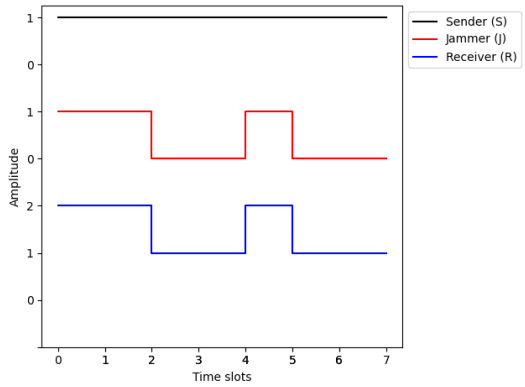
Let us term the time the sender transmits each bit over as its schedule. So, a sender bit remains unchanged over the course of one schedule. Any variation, therefore, in $\phi(.)$ over the course of a single schedule must result from variation in the jammer's transmission. Since the sender must be constant, we can limit the co-domain of $\phi(.)$ for each schedule. Consider the case where the sender bit is 0. The jammer bit can only vary between $\{0, 1\}$. Thus, $\phi(.)$ must vary between 0 and 1. Consequently, if the sender's current transmission is 1, $\phi(.)$ can only vary between 1 and 2. We can use the observation of extreme states and the restricted co-domain to deterministically decode many more symbols. Consider one sender bit transmitted for one schedule of time. Throughout that schedule, $\phi(.) = \{0, 1\}$, or $\phi(.) = \{1, 2\}$. Thus, if the receiver receives any symbol that is a 0 or 2 over the course of the schedule, it can determine the entire sender bit over the schedule to be 0 or 1, respectively.

Consider the case in Figure 1a. Here, the sender has a schedule of 7, and is transmitting a 1. At the same time, the jammer, with a schedule of 1, transmits 1100100. As a result, $\phi(.)$ will decode to 2211211. Since this schedule contains at least one 2, we can definitively say that the sender was transmitting a 1. Thus, we can correctly decode the entire sender's data over a schedule as long as an extreme state is present in that schedule.

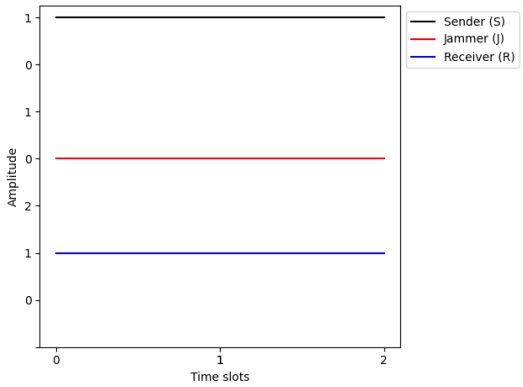
However, now consider the case in 1b. Here, the sender has a schedule of 2, while transmitting a 1. The jammer has a longer schedule, and is transmitting a 0. Thus, $\phi(.)$ will decode to 11. Since this schedule does not contain an extreme state, it cannot be deterministically decoded, so we would have to rely on other techniques.

From these two example, we can notice a pattern: when the jammer's schedule is much shorter than the sender's schedule, extreme values are more likely to be present, due to the jammer transmitting both 0 and 1 during the schedule. When the jammer's schedule is much longer than the sender's, we need other techniques for better accuracy, for which we currently rely on heuristics described in Section V, specifically Algorithm 3. However, a troublesome scenario occurs when the sender and jammer have nearly the same schedule.

To solve the problem of similar schedules, we propose a technique called *rate hopping*. Rate hopping refers to the



(a) Extreme States Present



(b) Extreme States Not Present

Fig. 1: Decoding Cases

action of varying the sender's schedule in a manner such that the receiver and sender are both aware of the schedule change. This can happen through either some predefined progression through a pseudo-randomly generated set of schedules, or by communicating it directly between the two devices. In doing so, we hope to transition the sender's schedule out of the vicinity of the jammer's schedule, resulting in a more desirable scenario.

C. Challenges

The challenge of creating extreme states of $\phi(.)$ lies in the fact that it only occur when both the controllable sender, and an uncontrollable jammer, transmit the same bit, which may not occur frequently, if at all.

Furthermore, in order to ensure throughput integrity, the schedules as described in Section IV-B must be respected unless a new agreement between the sender and receiver is made. Inevitably, this will result in times where a non-extreme state is created. For example, assume the system was in an extreme state to begin. Any sender bit transition, $0 \rightarrow 1$ or $1 \rightarrow 0$, will result in the undesired state where $\phi(.) = 1$ unless the jammer also switches bit simultaneously, which is extremely unlikely.

To compensate the lack of extreme states in a schedule, we make educated guesses, resulting in decreased accuracy. We discuss these methods that either aim to increase the schedules where an extreme state is present, or enhance the accuracy of the educated guesses in the following section.

V. Algorithm

Shadman: High level outline.

- Discussion of algorithms for the cases of comparable and different rates of jammer and sender
- Algorithm for cases with different rates of jammer and sender (incl. pseudocode or steps)
- Rate hopping algorithm

In this section, we discuss a method in which a string of digital bits, the output of $\phi(.)$, can be leveraged to recreate the sender's transmission. Within the main framework of this algorithm, which we call $MinMax$, multiple variations exist, which excel in recreating sender transmissions under different conditions.

When decoding a received transmission, we first need to group based on sender bits by utilizing the history of sender schedules. Doing so will allow us to perform proper schedule-wide decoding. This functionality is abstracted away in the $Group(Signals, Schedules)$ function. While schedule-wide decoding requires more complexity and overhead, its ability to allow us to decode entire schedules more than make up for the performance penalty.

The algorithm we will modify will be our previously defined $MinMax$ algorithm. Our basic implementation of $MinMax$ is described in Algorithm 1.

Algorithm 1: MinMax($RecieivedSigs$, $Schedule$)

Input: Cumulative received data bits at points in time $RecieivedSigs$, Sender Schedule at points in time $Schedule$
Output: Sender_data

1.1 $SBR \leftarrow Group(RecieivedSigs, Schedule)$
1.2 $n \leftarrow Length(SBR)$
1.3 $S \leftarrow \{\}$
1.4 **for** $i = 1 \ldots n$ **do**
1.5 **if** $0 \in SBR_i$ ***and*** $2 \notin SBR_i$ **then**
1.6 $S.insert(0)$
1.7 **else if** $2 \in SBR_i$ ***and*** $0 \notin SBR_i$ **then**
1.8 $S.insert(1)$
1.9 **else**
1.10 $S.insert(RandomChoice(\{0, 1\}))$
1.11 **return** S

Line 1.1 groups the received signals per schedule they occurred in in order to allow for decoding over an entire schedule. Lines 1.4-1.10 are where the data is processed and decoded. For each schedule, the following happens. Line 1.5 checks if a 0 is present in the schedule. If so, it decodes the schedule to 0. Line 1.7 does the opposite, checking if 2 is present and decoding to 1. If neither is the case, ie. extreme states are not present, we decode based on randomly guessing, on line 1.10.

We can further improve Algorithm 1 with heuristics, specifically the case where the true value of the sender's transmission is unknown. Thus, we must make educated guesses to rectify the potential true value of the sender's transmission. This is where previous observations regarding the behavior of the

sender and jammer devices and their relative rates come into play.

One heuristic that we can leverage revolves around the history of the decoded schedules. One assumption we could make about the sender is that it tends to transmit the same bit multiple times in a row. Using this notion about the sender, we can take advantage of previously decoded schedules to inform a current schedule without extreme states present. The implementation of this decoding method is defined in Algorithm 2.

Algorithm 2: MinMaxHistory($RecievedSigs$, $Schedule$, t)

Input: Cumulative received data bits at points in time $RecievedSigs$, Sender Schedule at points in time $Schedule$, Number of schedules to look back t
Output: Sender_data

$SBR \leftarrow Group(RecievedSigs, Schedule)$
$n \leftarrow Length(SBR)$
$S \leftarrow \{\}$
for $i = 1 \dots n$ **do**
  if $0 \in SBR_i$ ***and*** $2 \notin SBR_i$ **then**
    $S_i \leftarrow 0$ $S.insert(0)$
  else if $2 \in SBR_i$ ***and*** $0 \notin SBR_i$ **then**
    $S.insert(1)$
  else
    $num\_zero \leftarrow Count(\{S_{i-t-1}, \dots, S_{i-1} = 0)$
    $num\_one \leftarrow Count(\{S_{i-t-1}, \dots, S_{i-1} = 1)$
    if $num\_zero > num\_one$ **then**
      $S.insert(0)$
    else
      $S.insert(1)$
return S

Tne final piece of information we can utilize is the jammer's current transmission. If we can determine what this transmission is, we can assume this is the case for future schedules until we can see it is not. In the case where the jammer's schedule is long compared to the sender's, the scenario where extreme states are less likely, applying local knowledge of the jammer's transmission will allow for proper decoding. Fortunately, we can obtain the current jammer transmission quite trivially, given extreme states: if $\phi(.) = 0$, the jammer must be transmitting a 0, and the opposite if $\phi(.) = 2$. This heuristic is formally described in Algorithm 3.

As discussed previously, the $MinMax$ algorithm and all its corresponding heuristics suffer in cases where the jammer and sender schedules are quite similar. In this scenario, extreme states are less likely to occur, and local knowledge of the jammer's transmission is less valuable for the decreased likelihood of a jammer's transmission persisting over multiple schedules. We propose the idea of changing sender schedule randomly to evade the similarity with jammer schedule. Moreover, we do this repeatedly to random lengths for the schedules so that even if the jammer is intelligent and tries to adjust its schedule, we can still expect to achieve better performance. We name the idea of changing schedules *rate hopping*.

The heuristic we employ to determine when to rate-hop relies on comparing the most recent time between a change in the receiver's signal, a change in the value of $\phi(.)$, to the current schedule. Should this time be less than the current schedule, we should choose to perform a rate-hop. This heuristic is defined in Algorithm 4.

Algorithm 3: MinMaxMyopic($RecievedSigs$, $Schedule$)

Input: Cumulative received data bits at points in time $RecievedSigs$, Sender Schedule at points in time $Schedule$
Output: Sender_data

$SBR \leftarrow Group(RecievedSigs, Schedule)$
$n \leftarrow Length(SBR)$
$S \leftarrow \{\}$
$jammer\_sig \leftarrow 0$
for $i = 1 \dots n$ **do**
  if $0 \in SBR_i$ ***and*** $2 \notin SBR_i$ **then**
    $S.insert(0)$
    $jammer\_sig \leftarrow 0$
  else if $2 \in SBR_i$ ***and*** $0 \notin SBR_i$ **then**
    $S.insert(1)$
    $jammer\_sig \leftarrow 1$
  else
    $S.insert(1 - jammer\_sig)$
return S

Algorithm 4: RateHopping($Times, CurrentSchedule$)

Input: Times where $\phi(.)$ changes $Times$, Current Sender Schedule $CurrentSchedule$
Output: Should_Rate_Hop

$n \leftarrow Length(Times)$
if $Times_n - Times_{n-1} < CurrentSchedule$ **then**
  return $true$
return $false$

Should Algorithm 4 return a $true$ value, we would choose a new randomly generated schedule of an acceptable value to rate hop to. This new value would then be communicated to the sender.

VI. Evaluation

Shadman: High level outline. Shadman:

- Environment - 1 paragraph (on what configuration of machine and software the experiments were run), hardware description of hackrf etc. (which we can do later)
- Experimental settings - One subsection for parameters and performance metrics, another for methods
- Results & explanation

A. Environment

We implement the algorithms with Python 3.12, and run the experiments on a machine equipped with Apple M2 Pro processor with 16GB of memory.

B. Experimental Settings

1) Methods: As described in V, this paper compares two main algorithms: $MinMax$, described in Section V, and $Baseline$, defined below, and their variants. We compared these two algorithms, along with their respective heuristics, through three separate parameters.

Baseline algorithms work on a similar, yet simpler premise to $MinMax$, aligned with the traditional approach of reading bits of the wire as in wired communication. Over the course of each sender schedule, it averages the symbols received. Should that average be exactly 0 or 1, it decodes to the corresponding value: 0 to 0 and 1 to 1. If the average is not exactly either extreme state, it will decode to -1, marking the schedule as illegible, representing an error in the transmission. This implementation is called *Baseline Sender Bitwise*. In the illegible case, we have two heuristics. The first is to take the majority symbol, if 1 appears more than 0, decode to 1, and vice versa. This is called *Baseline Majority*. The other option is to randomly guess the value of the sender's transmission in that schedule. This is termed *Baseine Random*.

2) Parameterization and Performance Metrics: We mainly vary three parameters to evaluate the performance of our proposed algorithms, namely, sender to jammer schedule ratio, dataset size in terms of # of sender bits and the history window size in terms of bits, as delineated in Table I. As discussed in Sections IV and V, this schedule ratio is an important factor in determining the efficacy of our algorithms. We keep jammer's schedule fixed at 128 and vary sender's schedule to achieve the different schedule ratios. Moreover, the length of the sender's transmission. By varying this length, we would be able to determine how each algorithm variant performs, regardless of schedule ratio. In this regard, the dataset we have actually used are purely synthetic, but the bits of a sender were fixed while that of a jammer were randomly generated for each simulation. Finally, history window length helps assess relative utility of different heuristics.

Table I lists the all values of the parameters and their default values in boldface. When performing experiments, we maintained every parameter as default, except for the one being measured.

Parameter	Value
Sender to Jammer Schedule Ratio	$\mathbf{2^3}, 2^4, ..., 2^{11}$
Dataset Size (# of Sender bits)	$2^7, 2^8, ..., \mathbf{2^{16}}, ..., 2^{20}$
History window size	$2^0, 2^1, ..., \mathbf{2^7}, ..., 2^{10}$

TABLE I: Parameters for Experiments

In order to determine the accuracy of each of our algorithms, ie. compute $\Delta(S, \overline{S})$ where $\overline{S}$ is the decoded sender transmission, we simulated a potential transmission history with the given parameters. For each set of parameters the simulation was run 10 times to average out outliers in the results due to randomness, with their accuracies calculated as $\frac{Number\ of\ Bits\ Decoded\ Correctly}{Number\ of\ Bits\ Transmitted\ By\ Sender}$. After simulating ten times, the average of all the accuracies was computed. This is the accuracy reported in the results.

C. Results

1) Varying Schedule Ratio: We varied the schedule ratio between the sender and jammer, without engaging in rate hopping in the first experiment. Figure 2 shows $MinMax$ algorithm variations performed better when the schedule ratio was far from 1 (sender rate $\leq 2^5$, $\geq 2^9$) , as expected. With

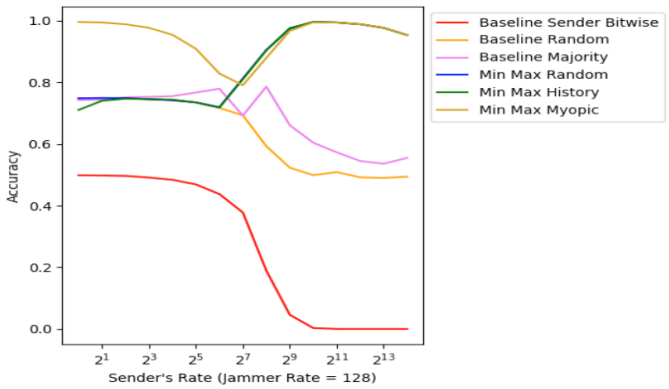


Fig. 2: Evaluating algorithms varying schedule ratio

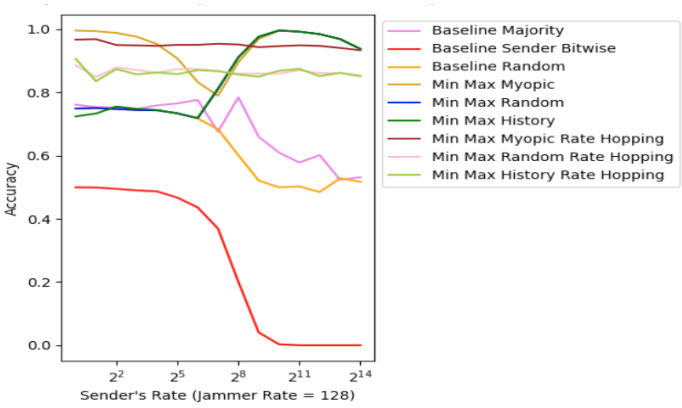


Fig. 3: Evaluating rate hopping varying initial schedule ratio

the sender's schedule being between 2^0 and 2^5 (schedules that make the sender transmit faster than the jammer), although all variants of $MinMax$ outperformed baseline, Algorithm 1 and Algorithm 2 suffered relative to Algorithm 3, achieving around 80% accuracy, due to their heuristics not being able to account for a lack of extreme states. However, Algorithm 3 performed astoundingly, with a near perfect decoding rate, since jammer schedules would persist in multiple sender schedules. On the other hand, when the sender's rate was between 2^9 and 2^{14}, significantly slower than the jammer, all three algorithms performed well, approaching 100% accuracy, given the expected abundance of extreme states. In the middle, with sender rate between 2^6 and 2^8, each of the $MinMax$ algorithms suffered dropping to around 75% accuracy, due to similar schedule ratios.

Next we included rate hopping, demonstrating that this approach can get around the lower accuracy for similar initial rates (as sender's rate changes over time pseudo-randomly) without much compromise in the better perfoming regions. From Figure 3, we notice that in the critical zone of the center of the graph, from around a sender initial schedule of 2^6 to a ratio of about 2^8, all the $MinMax$ algorithms with rate hopping outperformed their non-rate-hopping counterparts, by up to 20%. Evidently, in these cases, rate-hopping was able to pull the transmitter/receiver duo into a schedule ratio more favorable in creating extreme states. However, in the cases with vastly different schedules, some of the rate-hopping algorithms

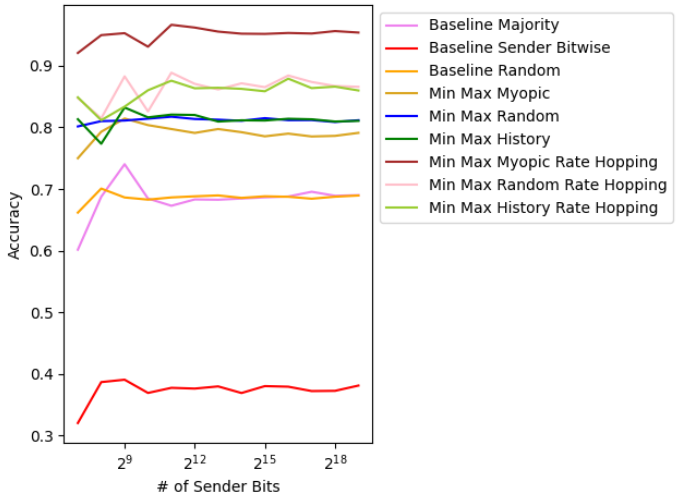


Fig. 4: Evaluating rate hopping varying dataset size, ratio = 1

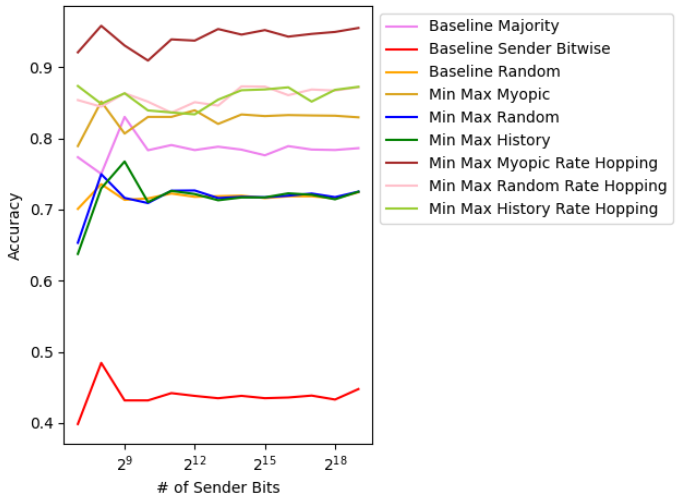


Fig. 5: Evaluating algorithms varying dataset size, ratio = $\frac{1}{4}$

under performed. This is most likely due to rate hopping changing the sender's schedule to one that is similar to the jammer's, and thus making it difficult to decode transmissions.

2) Varying Dataset Size: We have also verified the decoding capability of our algorithms against a varied transmission length (Figure 4). This experiment was performed at a sender schedule of 128, equivalent to a sender to jammer schedule ratio of 1. The flattening of the accuracy curves is a direct result of a constant rate ratio. However, the variation present in smaller dataset sizes is likely due to outliers in a few of the transmission decoding, that get smoothed out over time. The algorithms with rate-hopping perform up to 20% better when the schedules of the sender and jammer are the same.

We also conducted this experiment with two more schedule ratios: $\frac{1}{2}$, and another with a ratio of 2. These two experiments allowed us to verify the algorithms in both scenarios: a faster sender and a slower sender (results in Figure 5 and 6, respectively). Note that, the schedule ratio being different, the relative benefits of rate hopping diminishes slightly.

The flat curves, characteristic of a constant schedule ratio,

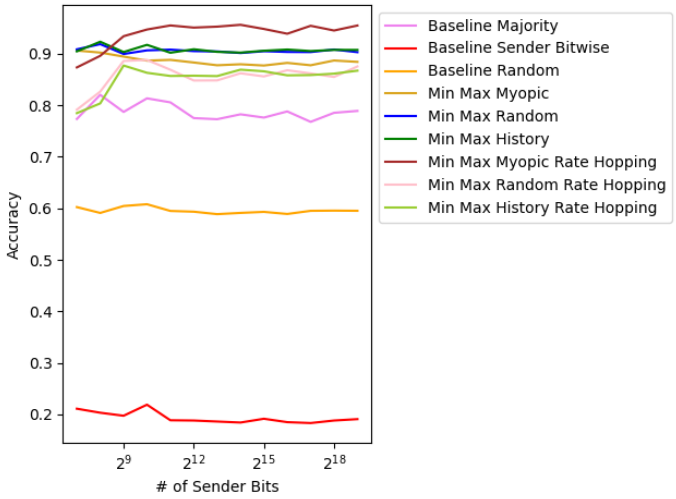


Fig. 6: Evaluating algorithms varying dataset size, ratio = 4

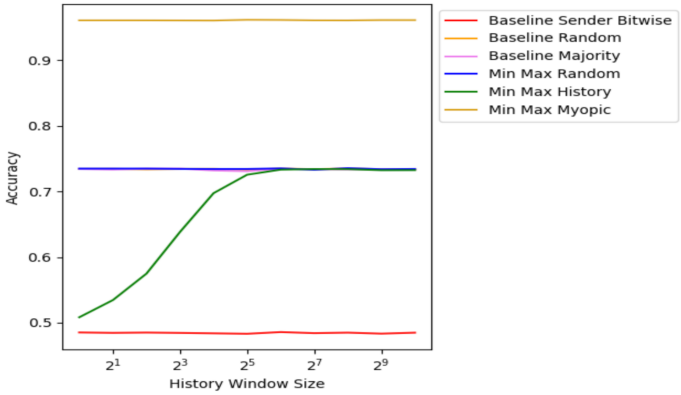


Fig. 7: Evaluating algorithms varying window history size

are present for each of these graphs. In the case where the schedule ratio is $\frac{1}{4}$, Algorithm 3 is the one that performs well. The rest of the algorithms cluster around 60% to 70% accuracy. This disparity is because of considering most recent jammer bit while that is expected to sustain across multiple sender schedules. However, in the case where the schedule ratio is 4, we see all Algorithms 1 - 3 shoot up to near-perfect accuracy due to the abundant presence of extreme values.

3) Varying Window Size: Figure 7 shows the results when we varied the size window of history used in Algorithm 2. So, only the performance of this variant of $MinMax$ varies. Interestingly, after a certain point, here 2^5, the size of the window doesn't matter, while smaller window sizes degrade performance. This is likely due to the history window containing a higher percentage of random bits, augmented during the initial periods on uncertainty.

VII. Conclusion

Shadman: Need to expand into paragraph(s).

- Applications
- Solution summary
- Future directions

This paper provides a comprehensive dive into a new method of decoding jammed signals in a sender, jammer,

and receiver system. It also provides new algorithms for this decoding process and compares their accuracies in a rate-hopping vs. non-rate-hopping scenario. Through our new algorithm, $MinMax$, we achieved near-perfect reconstruction of a jammed transmission in ideal cases. By utilizing rate-hopping on top of that, non-ideal cases became more accurate. By applying these two devices in wireless communication systems, the effects of adversarial jammers diminish. Thus, in military applications, $MinMax$ and rate hopping can be utilized, in conjunction with other methods, to defeat adversaries.

Our work paves the way to future researches in mitigating jamming in a specific frequency channel. First, we can take into account the intelligence and adaptivity of a jammer, who can listen to sender transmission and adjust jamming strategies accordingly since the adversaries in the real world may jam sporadically or act deliberately. Second, how our algorithm works on real hardwares, emulated by software defined radio, in the presence of noise, attenuation, fading etc. calls for further implementation and experiments. Moreover, how to improve rate hopping with better heuristics, potentially with a deep learning approach may be an interesting direction. And finally, applying knowledge of coding theory to further mitigate impacts of jamming in case of adaptive smart adversaries calls for follow-up works.

REFERENCES

[1] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[2] Y. Arjoune and S. Faruque, "Smart jamming attacks in 5g new radio: A review," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1010–1015.

[3] W. Xu, "On adjusting power to defend wireless networks from jamming," in *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, 2007, pp. 1–6.

[4] S. A. H. Mohsan, A. Mazinani, W. Malik, I. Younas, N. Q. H. Othman, A. Hussain, and A. Mahmood, "6g: Envisioning the key technologies, applications and challenges," *International Journal of Advanced Computer Science and Applications*, vol. 11, 01 2020.

[5] S. Soderi, A. Brighente, F. Turrin, and M. Conti, "Vlc physical layer security through ris-aided jamming receiver for 6g wireless networks," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2022, pp. 370–378.

[6] X. Lu, L. Xiao, P. Li, X. Ji, C. Xu, S. Yu, and W. Zhuang, "Reinforcement learning-based physical cross-layer security and privacy in 6g," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 425–466, 2023.

[7] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6g: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[8] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[9] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, 2004, pp. 289–298.

[10] N. Van Huynh, D. T. Hoang, X. Lu, D. Niyato, P. Wang, and D. I. Kim, "Ambient backscatter communications: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2889–2922, 2018.

[11] N. V. Huynh, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, and M. Mueck, "Defeating smart and reactive jammers with unlimited power," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[12] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 245–257, 2011.

[13] S. Gollakota and D. Katabi, "Physical layer wireless security made fast and channel independent," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1125–1133.

[14] A. Arora and L. Sang, "Dialog codes for secure wireless communications," in *2009 International Conference on Information Processing in Sensor Networks*. IEEE, 2009, pp. 13–24.

[15] K. Pelechrinis, I. Broustis, S. Krishnamurthy, and C. Gkantsidis, "Ares: an anti-jamming reinforcement system for 802.11 networks," 12 2009, pp. 181–192.

[16] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Game theoretic anti-jamming dynamic frequency hopping and rate adaptation in wireless systems," in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2014, pp. 247–254.