`mmFingerprint`: A New Application Fingerprinting Technique via mmWave Sensing and Its Use in Rowhammer Detection

Sisheng Liang[1], Zhengxiong Li[2], Chenxu Jiang[3], Linke Guo[3], and Zhenkai Zhang[1]

[1]School of Computing, Clemson University
[2]Department of Computer Science and Engineering, University of Colorado Denver
[3]Holcombe Department of Electrical and Computer Engineering, Clemson University
sishenl@clemson.edu, zhengxiong.li@ucdenver.edu, {chenxuj, linkeg, zhenkai}@clemson.edu

Abstract. Application fingerprinting is a technique broadly utilized in diverse fields such as cybersecurity, network management, and software development. We discover that the mechanical vibrations of cooling fans for both the CPU and power supply unit (PSU) in a system strongly correlate with the computational activities of running applications. In this study, we measure such vibrations with the help of mmWave sensing and design a new application fingerprinting approach named `mmFingerprint`. We create a prototype of `mmFingerprint` and demonstrate its effectiveness in distinguishing between various applications. To showcase the use of `mmFingerprint` in cybersecurity for defensive purposes, we deploy it in a real computer system to detect the execution of reputable Rowhammer attack tools like TRRespass and Blacksmith. We find that the detection can reach a very high accuracy in practical scenarios. Specifically, the accuracy is 89% when exploiting CPU fan vibrations and nearly 100% when leveraging PSU fan vibrations.

Keywords: Application fingerprinting · mmWave sensing · physical side-channel · Rowhammer detection.

1 Introduction

Fingerprints are unique attributes that objects possess, and can be used to differentiate one from another despite their similarities [1]. This concept naturally extends into the digital world, where we see its application in the form of application fingerprinting. Generally speaking, application fingerprinting is a process that identifies, detects, and catalogs running applications based on distinctive elements, such as patterns in data usage, computation/network behavior, or specific configurations within the application's code.

In recent years, application fingerprinting techniques have been widely employed in various areas, including cybersecurity, network management, and software development. As representative examples in cybersecurity, not only can

these techniques be exploited for compromising user privacy [2, 3], but they can also be employed for defensive purposes, such as detecting the use of illicit programs (e.g., those for crypto mining and password cracking) on high-performance computing systems [4, 5] and identifying the execution of denial-of-service (DoS) or other malicious software [6, 7].

The practice of application fingerprinting leveraging side-channel information has gained considerable popularity. This is because side-channel information, such as power consumption [8], and electromagnetic radiation [3, 9], are inevitable byproducts of any computation and can be hardly suppressed by external adversaries [10]. More importantly, the information correlates with the ongoing computation activities, making side-channel-based application fingerprinting possible.

In this paper, we propose a novel approach leveraging certain physical side-channel information obtained through mmWave sensing to achieve application fingerprinting that can be used to replace or complement traditional application fingerprinting methods as present in Figure 1. The foundation of our approach is built on the observation that different applications generate varying computation activities, which modulate the speed of the cooling fan. These modulated cooling fan speeds can reveal the computation activities. Therefore, accurately measuring these speed variations becomes the key. Equipped with advanced range and vibration sensing techniques, mmWave sensing, our method can measure fine speed variations with high precision. By monitoring the vibration patterns incurred by the speed of the cooling fan, our technique employs features engineering and deep neural networks to extract features and then uses a deep learning classifier to distinguish the applications.

Compared to the conventional application fingerprinting methods using network traffic statistics[11, 12], our approach has the following advantages: (1) We can indirectly monitor the computational actions of an application through the fan's status. This is particularly beneficial when the application does not generate any network traffic or when some applications alter the characteristics of the network traffic to make it seem legitimate [13]. (2) Our system provides non-intrusive and remote monitoring. It cannot be easily suppressed by external adversaries due to the contact-less fashion. (3) It does not add performance overhead to the target computing system.

Alongside the introduction of our new application fingerprinting technique, we also demonstrate its practical use in the field of detecting the execution of malicious programs. Specifically, we show that our fingerprinting technique can accurately identify potential Rowhammer attempts carried out by certain existing tools. We concentrate on this type of threat for two main reasons: the severity of Rowhammer attacks and the prevalent use of established tools in the initial reconnaissance phase.

Firstly, Rowhammer attacks pose substantial and ongoing threats to computer systems, leading to numerous exploitations such as sandbox escaping, privilege escalation [14, 15], cryptography subversion [16], denial of service [17, 18, 19], and even confidentiality breaches [20]. Although there are many mitiga-

tion strategies proposed, including counter-based methods such as [21, 22, 23], and Target Row Refresh (TRR) that is implemented in the current off-the-shelf DDR4 DRAMs by major vendors. However, advanced Rowhammer attack techniques such as TRRespass [24] and Blacksmith [25] have circumvented TRR. The effectiveness of counter-based defenses becomes questionable for this new type of many-sided Rowhammer attack.

Secondly, before launching a real Rowhammer attack, an attacker must inspect and scan the system to determine if its memory is susceptible to the Rowhammer effect. It is highly likely that during this reconnaissance phase, the attacker will utilize one or more reputable and effective tools, such as TRRespass [24] and Blacksmith [25], for such a purpose. These tools are known for their efficiency in hammering standard DDR4 DRAM modules, even those under the protection of TRR, aiding the attacker in swiftly identifying exploitable bits.

We evaluate `mmFingerprint` using data gathered from a CPU cooling fan and a PSU cooling fan, each subjected to ten different applications. These include two of the latest and most potent Rowhammer attack tools as well as harmless applications like the SPEC 2006 benchmark, YouTube, and system idle states. `mmFingerprint` demonstrates robust performance across these applications, achieving accuracy ranging from 0.69 to 1.00 in various scenarios. Notably, it can detect known Rowhammer attacks with near-perfect accuracy. Our findings indicate that the approach we've introduced is a feasible method for detecting Rowhammer attacks when established tools are used during the preliminary reconnaissance phase.

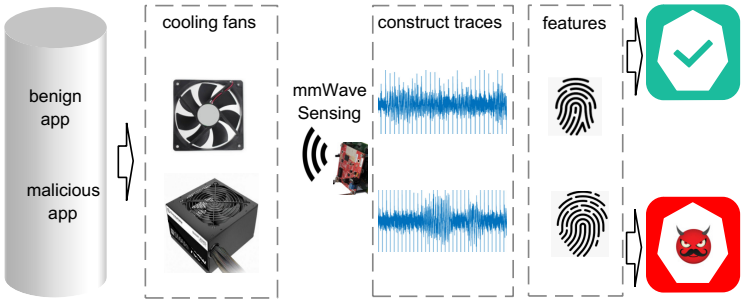


Fig. 1: `mmFingerprint` is based on monitoring the fan status through mmWave sensing and it can be used to detect if malicious applications are running.

The main contributions of this paper include:

– We introduce an innovative approach to application fingerprinting that capitalizes on side-channel information from cooling fans and mmWave sensing technology. This method identifies applications by picking up the subtle vibration differences on the cooling fan induced by the computation activities.

To the best of our knowledge, this is the first time that mmWave sensing has been applied to the context of application fingerprinting.
- We exemplify its defensive application by illustrating how it can detect Rowhammer attacks executed with recognized hammering tools during the reconnaissance process. We are the first to introduce mmWave sensing in the detection of Rowhammer attacks. It provides a new research vision in this area.
- The proposed `mmFingerprint` can efficiently recognize the most sophisticated Rowhammer attempts with reputable tools during the reconnaissance phase. The accuracy of this method can reach up to 100% percent.

2 Background

2.1 Advanced Configuration and Power Interface

The Advanced Configuration and Power Interface (ACPI) specification is an industry-wide standard that enables sophisticated operating system-directed configuration and power management for both individual devices and whole systems via the motherboard. [26]. It is comprised of both software and hardware elements. Devices and processors can run on different states based on the necessity to maintain a balance among power saving, heat dissipation, and performance. For example, it defines four useful states for a processor: the C0 state, where the CPU is doing useful work; the C1 (Halt) state, a light sleep state where the processor isn't executing instructions; the C2 (Stop-Clock) state, a deeper sleep state where power to the core is shut off; and the C3 (Sleep) state, an even deeper sleep state where the cache's context is lost and power to the cache is shut off. ACPI allows the OS to play a role in the thermal management of the system while maintaining the platform's ability to mandate cooling actions as necessary. It defines two cooling modes, Active and Passive. In the passive cooling mode, OS reduces the power consumption of devices at the cost of system performance to reduce the temperature of the system. While in active cooling, OS increases the power consumption of the system (for example, turning on a fan) to reduce the temperature of the system [26].

The OS active cooling mode needs support from the hardware such as the thermal sensor, cooling fan, and fan speed controller. The cooling fans are important computer components that help dissipate the heat generated by electronic components such as CPU, GPU, and the power electronics in the power supply. Most modern computer systems use temperature-controlled fan speed control mechanisms to regulate CPU and GPU cooling fan speeds. These mechanisms use hardware sensors to monitor CPU temperature and adjust the fan speed accordingly. Usually, the speed is a function of the temperature. This function can be selected from different working modes in the BIOS of some modern motherboards. The speed control approaches described include on-off, linear, and pulse width modulation (PWM) [27].

2.2 mmWave Sensing

The high-resolution frequency-modulated continuous-wave (FMCW) mmWave radar has been widely used in automotive and industrial applications recently due to the low cost [28]. It can be used to detect objects by estimating the range, velocity, and angle [29]. The mmWave radar transmits serial FMCW signals and receives the corresponding reflection signals from surrounding objects. Mixing the transmitted signal and received signal produces an intermediate-frequency (IF) signal, which can be used to estimate range, velocity, and angle. By tracking changes in the estimated range over a specific time step, the variation can be considered the object's vibration. The derivation of vibration is widely used in speech eavesdropping and reconstruction [30, 31, 32], vibration monitoring [33].

The estimation of range with coarse resolution can be achieved by applying a range FFT to the IF signal. With a 4GHz bandwidth FMCW mmWave device, the resolution stands at 3.75 cm [31]. This level of resolution suffices for many applications, like detecting objects in automotive settings. However, it falls short for applications that need a higher degree of detail, such as sound reconstruction and subtle vibration tracking, which typically require finer resolution. For these applications, a high-resolution range (e.g. 1 mm or even smaller) can be extracted from the phase value corresponding to the target range.

2.3 Rowhammer Attacks

Rowhammer attacks are a class of security exploits that target a hardware vulnerability in dynamic random-access memory (DRAM). By repeatedly accessing some DRAM rows, an attacker can cause unintended bit flips in neighboring rows by accelerating capacitor charge leakage, potentially leading to unauthorized access or privilege escalation, etc. The execution of a Rowhammer attack involves three phases by the attackers [10].

- Phase 1, the attacker scans the DRAM addresses by repeatedly accessing certain DRAM rows to search for exploitable bit flips. For example, with the addresses mapping information obtained by reverse engineering before the attack, the attacker can explore Rowhammer scanning by accessing two addresses from the same bank but not in the same row. When bit flips are found, the attacker can record the corresponding physical address for later use.
- Phase 2, The attacker redirects the target's sensitive security data to the vulnerable location identified in the first step.
- Phase 3, the attacker flips the bits when the security-critical data is placed at the location where it is flippable according to the second step. Then, the attacker can achieve his design goals such as privilege escalation, cryptography subversion, denial of service, and confidentiality breaching from this step.

Major DRAM vendors have widely adopted the Targeted Row Refresh (TRR) strategy to counteract Rowhammer attacks on the DDR4 memory. When the

number of accesses to a particular row surpasses a set threshold, a refresh (or activation) is issued to the neighboring rows. This action recharges these adjacent rows, thereby safeguarding them from being flipped. However, several advanced Rowhammer tools have recently been developed to bypass this TRR mitigation strategy, implemented by leading manufacturers on certain DDR4 DRAMs. Examples of such tools include TRRespass [24] and Blacksmith [25]. These tools are typically employed by attackers during the reconnaissance phase of a Rowhammer attack due to their efficacy. TRRespass utilizes a many-sided hammering technique to trigger bit flips and circumvent the TRR by generating a high volume of accesses to different DRAM rows in the same bank during the refresh window. Meanwhile, Blacksmith optimizes the row access pattern to achieve higher efficiency than TRRespass in triggering bit flips by adjusting the offset and intensity of hammering.

3 `mmFingerprint`

In this section, we present a robust technique called `mmFingerprint`, designed for application fingerprinting in systems that incorporate a CPU cooling fan or a power supply fan. These applications impact the CPU temperature or power electronics in the PSU, which subsequently alters the speed of the CPU fan or PSU fan. The `mmFingerprint` tool is adept at identifying such minor shifts in fan speed. The system can differentiate among various applications by analyzing the vibrations in the CPU cooling fan or power supply unit (PSU) fan, without requiring direct physical interaction. `mmFingerprint` employs advanced signal processing methods to detect these subtle vibrations.

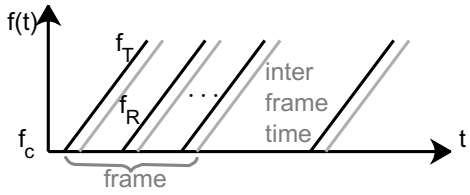


Fig. 2: mmWave FMCW chirps

3.1 Estimating Displacement Using mmWave Technology

mmWave radar adopts the Frequency Modulated Continuous Wave (FMCW) chirps for distance measuring [29]. Estimating the distance between the transmitter and receiver can be achieved by measuring the time delay and phase shift of mmWave signals. Furthermore, mmWave sensing enables the determination of object displacement by analyzing the range difference of the same object over

a given time interval. For example, given a sinusoidal FMCW transmit signal represented by

$$x_T(t) = \cos\left(2\pi f_c t + \pi S t^2\right) \ , \tag{1}$$

where f_c is the start frequency of the chirp and S is the frequency slope of the chirp. The time delay between the transmitter signal and the receiver signal can be represented as

$$\tau = 2d/c \ , \tag{2}$$

where τ is the time delay; d is the distance from the antenna to the target; c denotes the speed of light. The mixer combines the incoming and outgoing signals to generate the intermediate frequency (IF) signal. After the high-frequency components are eliminated by a low-pass filter, the low-frequency elements remain in the IF signal, which can be represented by

$$x_{\mathrm{IF}}(t) = \mathrm{LPF}\left\{x_T(t)x_R(t)\right\} = A\cos\left(2\pi f_{\mathrm{IF}} t + \phi_{\mathrm{IF}}\right) \ . \tag{3}$$

The intermediate frequency f_{IF} can be represented by the difference between the transmit signal frequency $f_T(t)$ and receiver signal frequency $f_R(t)$, as shown in

$$f_{\mathrm{IF}} = f_T(t) - f_R(t) = S\tau \ , \tag{4}$$

which can be further represented by the chirp frequency slope S and time delay τ according to the geometric relationship between the intermediate frequency and the frequency slope of the chirp as presented in figure 2. The intermediate signal initial phase can be determined from (1) at the time instant τ when the reflected signal just arrives at the antenna, which can be represented as

$$\phi_{\mathrm{IF}} = 2\pi f_c \tau + \pi S \tau^2 \approx 2\pi f_c \tau. \tag{5}$$

It can be approximated because f_c is much larger than $S\tau$ [29].

Finally, from (2) and (4) the distance and frequency relation can be represented as

$$d = S\tau = c f_{\mathrm{IF}}/(2S). \tag{6}$$

By performing the FFT operation to the intermediate signal (range FFT), the ranges can be obtained according to this equation. However, the range resolution is only 3.75 cm for a 4 GHz continuous bandwidth mmWave radar such as the TI IWR1642BOOST since the range resolution is determined by $c/(2B)$, where B is the chirp bandwidth [34]. This resolution is enough for applications such as distance detection in vehicles. However, it is not effective for applications requiring 1-mm or even better resolution such as voice recovery. Fortunately, we can derive a high-resolution range from phase based on (2) and (5), which can be represented as

$$\phi_{\mathrm{IF}} = 2\pi f_c \tau = 4\pi d/\lambda \ , \tag{7}$$

where λ is the wavelength of mmWave signal at frequency f_c. Differentiating both sides of the equation (7) results in $\Delta d = \lambda \Delta \phi_{\text{IF}}/4\pi$, where Δd is the small range displacement for a target during a short time; $\lambda \approx 4mm$ is mmWave signal wavelength for a 77-81Ghz mmWave radar. $\Delta \phi_{\text{IF}}$ is the corresponding phase displacement for the same target. The displacement calculated through phase yields a better range resolution than that derived from range FFT which is 3.75 cm for a 4Ghz bandwidth mmWave radar.

3.2 Locate the Cooling Fan with mmWave Radar

First, `mmFingerprint` locates the target cooling fan with mmWave sensing. `mmFingerprint` conducts a range-FFT over each chirp on the gathered Intermediate Frequency (IF) data. Different frequency components represent distinct reflective signals from various objects in the surrounding environment. Identifying the desired frequency bin (range bin) among numerous bins can be challenging. We monitor various range bins across several consecutive frames, as shown in figure 3. Each peak represents an object. We identify the correct range bin by locating the right peak and verifying it with a measured distance from a ruler. Second, once the target range bin has been located, `mmFingerprint` extracts the phase value at the target bin by calculating the phase angle from the complex values at the peak. According to equation (7), the phase value is proportional to the target distance.

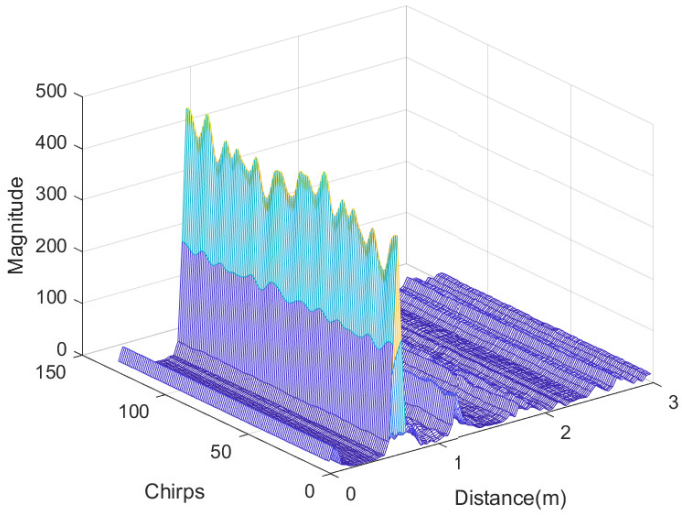
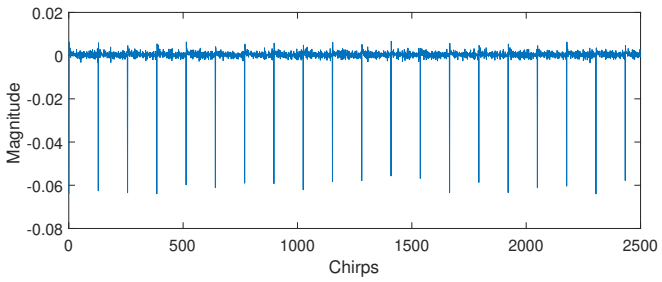
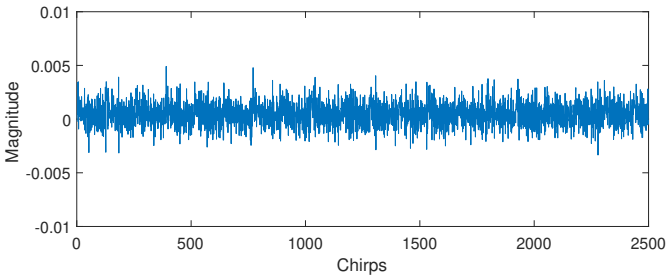


Fig. 3: Range-FFT across many chirps.



(a) Time serials with spikes.



(b) Time serials after removing spikes.

Fig. 4: Reconstructed displacement time serials traces.

3.3 Time Serials Trace Construction

Using the phase data from the designated bin, `mmFingerprint` initially creates the distance-time series traces for a targeted object. It does this by joining together the phase values obtained at the targeted bin from every chirp, over a multitude of continuous frames. Then, range displacement is derived from the distance-time series trace according to $\Delta d_n = d_{n+1} - d_n$, where d_{n+1} is the distance at discrete time $n+1$ and d_n is the distance at discrete time n. Therefore, the range displacement is sampled at the sample rate of the chirp rate.

Removing the spikes: The mmWave radar produces chirps in frames, in a non-continuous fashion. There is a noticeable surge at the start of each frame due to the first two data points, and these surges significantly exceed other phase values as shown in figure 4(a). In order to mitigate the influence of these abnormal data points on the classification process, we replace them with the final data point from the preceding frame. This strategy facilitates a seamless transition from one frame to the next. As shown in figure 4(b), the range displacement trace oscillates around zero in a more symmetrical way. The useful side-channel information encoded into the recovered time series trace can be exploited to infer the computing activities.

3.4 Fan Responses and Correlations

A time series trace can be constructed from the displacement derived from the phase with the aforementioned method. Applying the Fast Fourier Transform (FFT) to the time domain signal is a common technique used to analyze frequency components and extract features from signals. By converting the signal from the time domain to the frequency domain, we can examine the distribution of frequencies present in the signal and identify specific patterns or characteristics. Figure 6 presents the frequency components of the two time-series traces for different loads. The frequency distributions for these two traces display uniqueness. When the CPU executes different applications, the computational tasks vary, resulting in unique fan vibration patterns. We leverage these specific traits to distinguish and classify various applications.

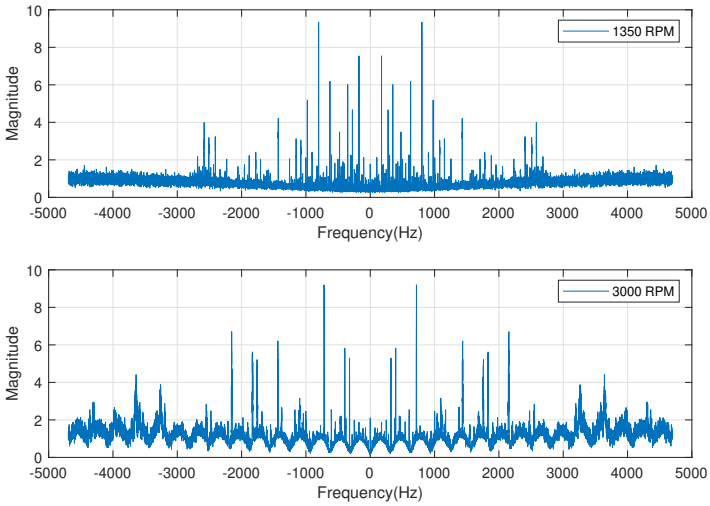


Fig. 5: Frequency components of `mmFingerprint` responses to different fan speeds.

3.5 Features Extraction and Applications Classification

Numerous methods exist for extracting the features from time-series data. One approach involves the manual extraction of these features by performing signal analysis, such as Fast Fourier Transform (FFT). Another method is to utilize deep neural networks (DNN) for feature extraction. By employing trained DNN layers, we can extract complex features. Different applications are subsequently categorized based on the features extracted from the mmWave vibration traces.

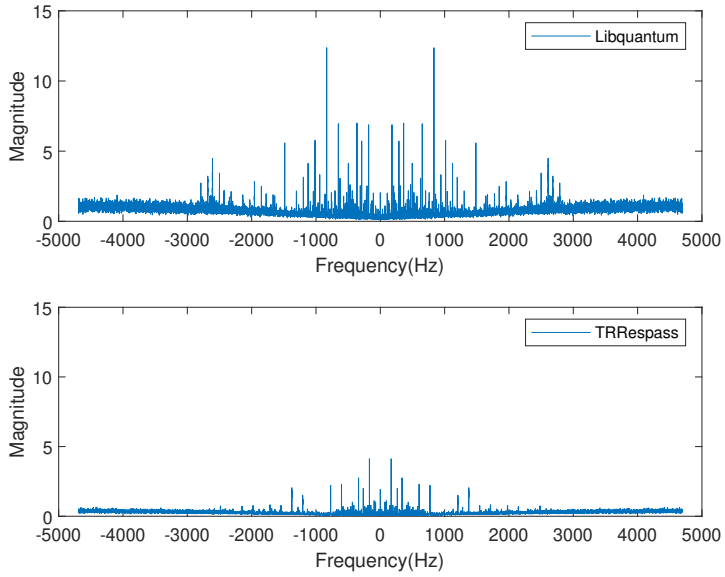


Fig. 6: Correlation between applications from fan response in the frequency domain.

To eliminate the necessity for manual feature crafting, we opt for a machine-learning approach to extract features and classify the workload traces. This job can be accomplished using either a KNN classifier or deep neural networks. To attain high precision, we choose state-of-the-art deep neural networks (DNN).

In terms of the DNN model, we choose to use Convolutional Neural Network (CNN) over Recurrent Neural Network (RNN), even though the workload power traces belong to time series data. One of the primary concerns is that RNN usually suffers from the over-fitting problem more severely when training on long time series [35]. To be specific, we use the ResNet10 architecture that is described in [36] as the classifier in this work.

4 Evaluations

4.1 Experiment Setup

We use a Texas Instruments IWR1642BOOST evaluation board to transmit and receive chirps. The IWR1642 chip can generate chirps with continuous frequency bands of 76 $\sim$ 77 GHz and 77 $\sim$ 81 GHz. The evaluation board integrates two Tx antennas and four Rx antennas. We use the two Tx channels sending out the same FMCW chirps with a continuous band of 3.98 GHz. We use DCA1000EVM evaluation board to extract data samples at a rate of 2.1 Msps. The frame

duration is 15 ms with 128 chirps in each frame. The antenna is placed 0.6 meters away from the CPU fan with no obstacles in between. The chirps are reflected off the cooling fan and captured by the four Rx antennas. In each case, the positions of the antenna and the target machine are kept constant to eliminate the effects of position movement on the reflected signals.

The targeted machine is equipped with a Gigabyte GA-H170-D3HP motherboard, an i7-6700 CPU, an Intel E97379-003 CPU cooling fan, and an Apevia ATX-SN1050W power supply. The deep learning classifier is built with Keras, using Tensorflow as its backend. This classifier is implemented on a desktop computer powered by an Intel i7-9700K CPU, with 64 GB of DRAM, and an Nvidia RTX3090 GPU.

4.2 Threat Model for Detecting Rowhammer Attempts using Reputable Tools

Assume an attacker plans to initiate a Rowhammer attempt on a targeted computer system equipped with DDR3 or DDR4, a CPU cooling fan, or a power supply cooling fan. Before the Rowhammer attack, the attacker must scan the memory addresses to determine if the computing systems are vulnerable to Rowhammer attacks. Due to their effectiveness, it is highly likely that the attacker used the most advanced Rowhammer attack tools such as TRRespass and Blacksmith for this reconnaissance process to circumvent the TRR implemented by major vendors in DDR4 DRAMs. Considering the extremely low likelihood of discovering exploitable bit flips within a short time, the attacker would need to scan the DRAM intensively to identify vulnerable bits, recording this information for future exploitation. This step typically requires a significant amount of time. We can set up a millimeter-wave (mmWave) radar at a predetermined distance from the cooling fan of either the CPU or power supply, ensuring that there are no obstructions in the path. This arrangement is feasible for most desktops and servers since their cooling fans are typically visible through ventilation openings. With its high-precision detection capabilities, our system can discern even the smallest variations in the vibrations of the cooling fan during computational processes.

4.3 CPU Cooling Fan Side-Channel

We assess the CPU cooling fan side-channel across various applications, as outlined in Table 1. We select several benign applications and two of the most effective Rowhammer tools against TRR named TRRespass [24] and blacksmith [25]. These benign applications include system idle, playing a video with vlc player, and opening the YouTube webpage. We also evaluate some SPEC 2006 benchmarks including data compression application `bzip2` (integer), quantum computation simulator `libquantum` (integer), playing the game of Go `gobmk` (integer), fluid dynamics simulation `lbm` (floating-point), quantum chromodynamics simulation`milc` (floating-point). For each workload, we construct 500 individual traces, each lasting 0.96 seconds with 8192 equivalent samples.

Table 1: Evaluated applications

Label	applications	Notes
0	blacksmith	Rowhammer
1	bzip2	CINT
2	gobmk	CINT
3	idle	
4	lbm	CFP
5	libquantum	CINT
6	milc	CFP
7	TRRespass	Rowhammer
8	vlc	video
9	Youtube	

Dataset The dataset is composed of ten distinct classes, which are divided into training and test sets at a proportion of 80% and 20%, respectively. The deep learning classifier undergoes training for 500 epochs using the training dataset and its performance is subsequently evaluated on the test dataset.

Evaluation metrics and results The effectiveness of `mmFingerprint` is assessed using precision, recall, and F1-score as performance measures. The evaluation confusion matrix is presented in figure 7(a) and the precision, recall, and F1-score are shown in table 2. The `mmFingerprint` has demonstrated an impressive ability to categorize ten distinct classes with an overall accuracy rate of 0.89. Additionally, it exhibits an almost flawless accuracy rate nearing 1.00 when distinguishing two specific Rowhammer tools, data compression `bzip2`, and playing youtube from other applications. The classifier can recognize `gobmk` with perfect precision, but a slightly lower recall of 0.93, which has lowered the F1-score to 0.97. This suggests that the model occasionally misses true positives for this class. `mmFingerprint` has relatively lower precision recognizing `idle`, `lbm`, and `libquantum`, but the model has good recall for these classes. This indicates the model occasionally misclassifies other instances as these classes, but does well in identifying true instances of these classes. The lowest F1-scores on distinguishing `milc` and `vlc`, suggesting that the model struggles the most with these classes. When dealing with `milc`, the model struggles to correctly identify all true instances (recall of 0.59), and for `vlc`, it frequently misclassifies other instances as this class (precision of 0.92), leading to lower F1-scores. Overall, `mmFingerprint` performs well on most classes, especially for Rowhammer tools.

4.4 Power Supply Cooling Fan Side-Channel

To assess the efficiency of `mmFingerprint` when dealing with power supply coiling fan side-channel, we conduct evaluations using the same applications shown in table 1. We collect 500 traces for each workload and they are split into training and test sets at a proportion of 80% and 20%,

Table 2: Evaluation of CPU Fan

Label	precision	recall	f1-score
0	0.99	1	1
1	1	1	1
2	1	0.93	0.97
3	0.69	0.95	0.8
4	0.75	0.85	0.8
5	0.78	0.92	0.84
6	0.79	0.59	0.68
7	1	1	1
8	0.92	0.56	0.69
9	1	1	1

Table 3: Evaluation of Power Fan

Label	precision	recall	f1-score
0	1	1	1
1	1	1	1
2	0.99	1	1
3	1	1	1
4	1	0.99	0.99
5	0.99	1	0.99
6	1	1	1
7	1	1	1
8	1	1	1
9	1	0.99	1

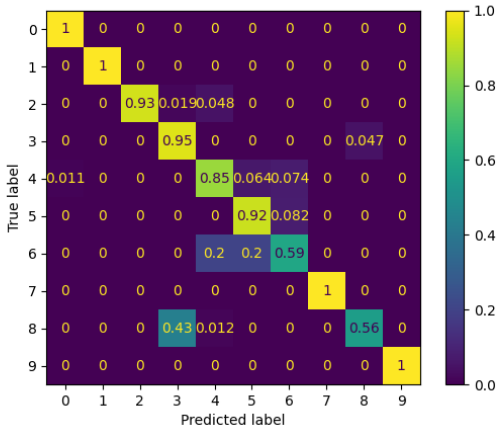
`mmFingerprint` performs well on the power supply cooling fan. The precision, recall, and F1-score are presented in table 3 and the confusion matrix is shown in figure 7(b). A precision of 1.00 means there were no false positive instances. It presents an almost absolute accuracy rate nearing 1.00 when classifying blacksmith, data compression `bzip2`, system idle, `milc`, TRRespass, and `vlc` from other applications. It exhibits a slightly lower precision of 0.99 when classifying `gobmk` and `libquantum`, which still indicates a high accuracy. Recall measures the ratio of correctly predicted positive instances to all instances that are actually positive. Like precision, a recall of 1.00 indicates a perfect score. All classes have a recall of 1.00, except for `lbm` and Youtube which have a slightly lower recall of 0.99. Overall, `mmFingerprint` can recognize different applications with high performance.

5 Related Work

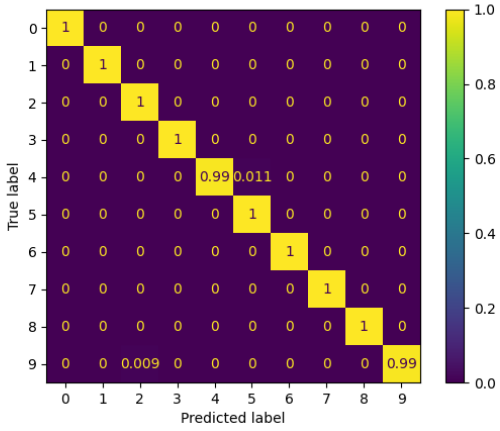
mmWave sensing The ability of mmWave sensing to accurately detect micro-vibrations underscores its effectiveness. It employs high-frequency radar waves, which are adept at identifying minute alterations in the phase or amplitude of reflected signals, enabling the detection of minute displacements, typically associated with vibrations. We summarize the most recent and important findings related to security and privacy, emphasizing the capabilities of mmWave sensing technology.

These applications include speech recovery such as WaveEar [37], through wall sound reconstruction such as Wavesdropper [38], eavesdropping speech of phone call such as mmEve [31], mmSpy [32] mmEcho [30], construction of a Covert Channel using the mmWave sensing of the status of cooling fan [39], lunching a spoofing attack to vehicles [40], user verification for IoT devices [41]. However, to the best of our knowledge, no studies have yet utilized mmWave sensing for the detection of malicious workloads.

Rowhammer Ever since the inaugural Rowhammer attack [42], the spectrum of these attacks has broadened with numerous variants coming to light. In



(a) Confusion matrix of CPU fan side-channel.



(b) Confusion matrix of power supply fan side-channel.

Fig. 7: Confusion matrix.

response, the research community and major DRAM vendors have put forward a wide array of proposed defenses against these diverse Rowhammer onslaughts. The first category is performance counters based Rowhammer detection such as [21],[43],[44]. A second category physically isolates all rows by making only every second row accessible to programs [45]. This method can be circumvented by half-double hammering [46]. Another important way is the Target Row Refresh (TRR) adopted by major DRAM vendors for off-the-share DDR4 DRAMs. This technique is proved to be ineffective for many-sided Rowhammer attacks [24]

and half-double hammering [46]. Researchers leveraged EM side-channel to detect the Rowhammer attacks in [10]. But it is unclear whether this can detect the new, sophisticated many-sided hammering and half-double hammering or not.

6 Discussion

In this section, we discuss some situations `mmFingerprint` can be applied and some limitations.

Although it can not always achieve a 100% detection accuracy, it can significantly improve the detection performance through this new detection method. Moreover, it can complement other existing defense solutions. This system also has the capability to monitor several cooling fans simultaneously. To illustrate, after the application of range FFT, multiple range bins are generated, each corresponding to a specific distance. We can derive varied phase data from these different range bins, which allows us to monitor objects at different distances, thereby observing various cooling fans concurrently. Furthermore, the method presented can potentially be expanded to encompass High-Performance Computing Centers, allowing for the monitoring of illicit applications. An illustration of this would be its application in the detection of unauthorized Cryptocurrency mining activities.

However, certain limitations exist. Detecting minute changes can be challenging, particularly when the execution time is short because the equivalent sampling rate is about 10 kHz with the device we use. Based on the Nyquist sampling theorem, the highest frequency it can sample is less than 5 kHz. The sampling rate is insufficient for capturing applications that have a short execution time, such as those lasting only a few hundred microseconds or less.

7 Conclusion

In our study, we propose a novel application fingerprinting system capable of detecting harmful applications based on the physical side-channel of a cooling fan, specifically focusing on detecting Rowhammer attacks using reputable tools. This system differentiates between the specific characteristics of various applications by utilizing millimeter-wave sensing technology and a machine learning model. Our approach has undergone rigorous assessments, which include evaluations of applications encompassing advanced Rowhammer attack tools like TRRespass and Blacksmith, as well as SPEC2006 benchmarks. These evaluations confirm the high precision of our technique across various scenarios.

8 Acknowledgment

This work is supported in part by the National Science Foundation (CNS-2147217). The authors would like to thank the anonymous reviewers for their comments and suggestions that help us improve the quality of the paper.

Bibliography

[1] Neal R Wagner. Fingerprinting. In *1983 IEEE Symposium on Security and Privacy*, pages 18–18. IEEE, 1983.

[2] Li Yang, Yifang Zhi, Teng Wei, Shui Yu, and Jianfeng Ma. Inference attack in android activity based on program fingerprint. *Journal of Network and Computer Applications*, 127:92–106, 2019.

[3] Nikolay Matyunin, Yujue Wang, Tolga Arul, Kristian Kullmann, Jakub Szefer, and Stefan Katzenbeisser. Magneticspy: Exploiting magnetometer in mobile devices for website and application fingerprinting. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 135–149, 2019.

[4] Dragos Draghicescu, Alexandru Caranica, Alexandru Vulpe, and Octavian Fratu. Crypto-mining application fingerprinting method. In *2018 International Conference on Communications (COMM)*, pages 543–546. IEEE, 2018.

[5] Pengfei Zou, Ang Li, Kevin Barker, and Rong Ge. Detecting anomalous computation with rnns on gpu-accelerated hpc machines. In *Proceedings of the 49th International Conference on Parallel Processing*, pages 1–11, 2020.

[6] Muhammad Ejaz Ahmed, Saeed Ullah, and Hyoungshick Kim. Statistical application fingerprinting for ddos attack mitigation. *IEEE Transactions on Information Forensics and Security*, 14(6):1471–1484, 2018.

[7] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. Automated worm fingerprinting. In *OSDI*, volume 4, pages 4–4, 2004.

[8] Yimin Chen, Xiaocong Jin, Jingchao Sun, Rui Zhang, and Yanchao Zhang. Powerful: Mobile app fingerprinting via power analysis. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[9] Haider Adnan Khan, Nader Sehatbakhsh, Luong N Nguyen, Milos Prvulovic, and Alenka Zajić. Malware detection in embedded systems using neural network model for electromagnetic side-channel signals. *Journal of Hardware and Systems Security*, 3:305–318, 2019.

[10] Zhenkai Zhang, Zihao Zhan, Daniel Balasubramanian, Bo Li, Peter Volgyesi, and Xenofon Koutsoukos. Leveraging em side-channel information to detect rowhammer attacks. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 729–746. IEEE, 2020.

[11] Jianfeng Li, Hao Zhou, Shuohan Wu, Xiapu Luo, Ting Wang, Xian Zhan, and Xiaobo Ma. {FOAP}:{Fine-Grained}{Open-World} android app fingerprinting. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1579–1596, 2022.

[12] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescape. Traffic classification of mobile apps through multi-classification. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[13] Clemens Kolbitsch, Paolo Milani Comparetti, Christopher Kruegel, Engin Kirda, Xiao-yong Zhou, and XiaoFeng Wang. Effective and efficient malware detection at the end host. In *USENIX security symposium*, volume 4, pages 351–366, 2009.

[14] Mark Seaborn and Thomas Dullien. Exploiting the dram rowhammer bug to gain kernel privileges. *Black Hat*, 15:71, 2015.

[15] Yuan Xiao, Xiaokuan Zhang, Yinqian Zhang, and Radu Teodorescu. One bit flips, one cloud flops: Cross-vm row hammer attacks and privilege escalation. In *25th {USENIX} security symposium ({USENIX} security 16)*, pages 19–35, 2016.

[16] Sarani Bhattacharya and Debdeep Mukhopadhyay. Curious case of rowhammer: flipping secret exponent bits using timing analysis. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 602–624. Springer, 2016.

[17] Yeongjin Jang, Jaehyuk Lee, Sangho Lee, and Taesoo Kim. Sgx-bomb: Locking down the processor via rowhammer attack. In *Proceedings of the 2nd Workshop on System Software for Trusted Execution*, pages 1–6, 2017.

[18] Onur Mutlu. The rowhammer problem and other issues we may face as memory becomes denser. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 1116–1121. IEEE, 2017.

[19] Daniel Gruss, Moritz Lipp, Michael Schwarz, Daniel Genkin, Jonas Juffinger, Sioli O'Connell, Wolfgang Schoechl, and Yuval Yarom. Another flip in the wall of rowhammer defenses. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 245–261. IEEE, 2018.

[20] Andrew Kwong, Daniel Genkin, Daniel Gruss, and Yuval Yarom. Rambleed: Reading bits in memory without accessing them. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 695–711. IEEE, 2020.

[21] Zelalem Birhanu Aweke, Salessawi Ferede Yitbarek, Rui Qiao, Reetuparna Das, Matthew Hicks, Yossi Oren, and Todd Austin. Anvil: Software-based protection against next-generation rowhammer attacks. *ACM SIGPLAN Notices*, 51(4):743–755, 2016.

[22] Eojin Lee, Ingab Kang, Sukhan Lee, G Edward Suh, and Jung Ho Ahn. Twice: Preventing row-hammering by exploiting time window counters. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 385–396, 2019.

[23] Yeonhong Park, Woosuk Kwon, Eojin Lee, Tae Jun Ham, Jung Ho Ahn, and Jae W Lee. Graphene: Strong yet lightweight row hammer protection. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1–13. IEEE, 2020.

[24] Pietro Frigo, Emanuele Vannacc, Hasan Hassan, Victor Van Der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Trrespass: Exploiting the many sides of target row refresh. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 747–762. IEEE, 2020.

[25] Patrick Jattke, Victor Van Der Veen, Pietro Frigo, Stijn Gunter, and Kaveh Razavi. Blacksmith: Scalable rowhammering in the frequency domain. In

2022 IEEE Symposium on Security and Privacy (SP), pages 716–734. IEEE, 2022.

[26] UEFI Forum, Inc. Advanced configuration and power interface (acpi) specification. `https://uefi.org/sites/default/files/resources/ACPI_Spec_6_5_Aug29.pdf`, 2022. accessed: 06.12.2023.

[27] David Hanrahan. Fan-speed control techniques in pcs. *Analog Dialogue*, 34(4):34–04, 2000.

[28] Igal Bilik, Oren Longman, Shahar Villeval, and Joseph Tabrikian. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE signal processing Magazine*, 36(5):20–31, 2019.

[29] Xinrong Li, Xiaodong Wang, Qing Yang, and Song Fu. Signal processing for tdm mimo fmcw millimeter-wave radar sensors. *IEEE Access*, 9:167959–167971, 2021.

[30] Pengfei Hu, Wenhao Li, Riccardo Spolaor, and Xiuzhen Cheng. mmecho: A mmwave-based acoustic eavesdropping method. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 836–852. IEEE Computer Society, 2022.

[31] Chao Wang, Feng Lin, Tiantian Liu, Kaidi Zheng, Zhibo Wang, Zhengxiong Li, Ming-Chun Huang, Wenyao Xu, and Kui Ren. mmeve: eavesdropping on smartphone's earpiece via cots mmwave device. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 338–351, 2022.

[32] Suryoday Basak and Mahanth Gowda. mmspy: Spying phone calls using mmwave radars. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1211–1228. IEEE, 2022.

[33] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

[34] Sandeep Rao. Introduction to mmwave sensing: Fmcw radars. *Texas Instruments (TI) mmWave Training Series*, pages 1–11, 2017.

[35] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[36] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[37] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 14–26, 2019.

[38] Chao Wang, Feng Lin, Zhongjie Ba, Fan Zhang, Wenyao Xu, and Kui Ren. Wavesdropper: Through-wall word detection of human speech via commer-

cial mmwave devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–26, 2022.

[39] Zhengxiong Li, Baicheng Chen, Xingyu Chen, Huining Li, Chenhan Xu, Feng Lin, Chris Xiaoxuan Lu, Kui Ren, and Wenyao Xu. Spiralspy: Exploring a stealthy and practical covert channel to attack air-gapped computing devices via mmwave sensing. In *The 29th Network and Distributed System Security (NDSS) Symposium 2022*. The Internet Society, 2022.

[40] Rohith Reddy Vennam, Ish Kumar Jain, Kshitiz Bansal, Joshua Orozco, Puja Shukla, Aanjhan Ranganathan, and Dinesh Bharadia. mmspoof: Resilient spoofing of automotive millimeter-wave radars using reflect array. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1971–1985. IEEE Computer Society, 2022.

[41] Yudi Dong and Yu-Dong Yao. Secure mmwave-radar-based speaker verification for iot smart home. *IEEE Internet of Things Journal*, 8(5):3500–3511, 2020.

[42] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014.

[43] Daniel Gruss, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. Flush+ flush: a fast and stealthy cache attack. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 13th International Conference, DIMVA 2016, San Sebastián, Spain, July 7-8, 2016, Proceedings 13*, pages 279–299. Springer, 2016.

[44] Tianwei Zhang, Yinqian Zhang, and Ruby B Lee. Cloudradar: A real-time side-channel attack detection system in clouds. In *Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016, Paris, France, September 19-21, 2016, Proceedings 19*, pages 118–140. Springer, 2016.

[45] Radhesh Krishnan Konoth, Marco Oliverio, Andrei Tatar, Dennis Andriesse, Herbert Bos, Cristiano Giuffrida, and Kaveh Razavi. Zebram: comprehensive and compatible software protection against rowhammer attacks. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 697–710, 2018.

[46] Andreas Kogler, Jonas Juffinger, Salman Qazi, Yoongu Kim, Moritz Lipp, Nicolas Boichat, Eric Shiu, Mattias Nissler, and Daniel Gruss. {Half-Double}: Hammering from the next row over. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3807–3824, 2022.