Information Security Advances with a Focus on AI & Privacy Preserving

frostwing98

COSEC of Nanjing University

February 14, 2019

Table Of Contents I

- ~~Traditional~~

- ~~Traditional~~
  - ~~Vulnurables~~

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~
  - ~~Cryptography~~

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~
  - ~~Cryptography~~
- Cross-Disciplinary

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~
  - ~~Cryptography~~
- Cross-Disciplinary
  - Fuzzing

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~
  - ~~Cryptography~~
- Cross-Disciplinary
  - Fuzzing
  - Privacy

- ~~Traditional~~
  - ~~Vulnurables~~
  - ~~Cybersecurity~~
  - ~~Cryptography~~
- Cross-Disciplinary
  - Fuzzing
  - Privacy
  - ~~Security-schema~~

AI fuzzing

Fuzzing:

- an automated software **testing** technique
- provides invalid, unexpected, or random data as inputs to a computer program
- monitors exceptions such as crashes, failing built-in code assertions, or potential memory leaks

Two main categories:

- Fuzzing for AI
- AI for fuzzing

Outline I

Fuzzing for AI

Target: AI Components

- neuron coverage[1]
- layer coverage[2]
- formal security method[3]

[1] [Kexin Pei et al., 2017]DeepXplore: Automated Whitebox Testing of Deep Learning Systems

[2] [Lei Ma et al., 2018]DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems

[3] [Shiqi Wang et al., 2018]Formal Security Analysis of Neural Networks using Symbolic Intervals

AI for fuzzing

Method: AI

- RNN-based(LSTM+AFL)[4]
- CNN-based(CNN+gradient descent)[5]
- RL(Q-Learning)[6]

[4] [Mohit Rajpal et al., 2017]Not all bytes are equal: Neural byte sieve for fuzzing

[5] [Dongdong She et al., 2017]NEUZZ: Efficient Fuzzing with NeuralProgram Smoothing

[6] [Konstantin Bottinger et al., 2018]Deep Reinforcement Fuzzing

Testing

- Blackbox, testing functions without peering into internal structures or workings
- Whitebox, testing internal structures or workings of an application
- Greybox, tests improper structure-caused defects, if any

Coverage

Software testing measurement for describing the degree to which the source code of a program is executed

- Edge Coverage
- Function Coverage
- Statement Coverage

CNN,RNN

Short view:

- MLP: Simplest DNN with fully-connected layers
- CNN: +Hypo:Space-correlation, everywhere in CV
- RNN: +Hypo:Time-correlation, usually used in Speech Analytics

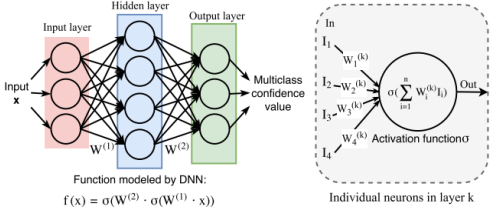
Outline I

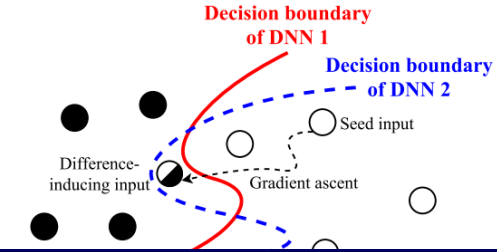
DeepXPlore[1]

- Neuron coverage: coverage of neurons with outputs exceeding preset thresholds
- Goal: Optimize neuron coverage
- How: Gradient Descending aiming to find maximal value

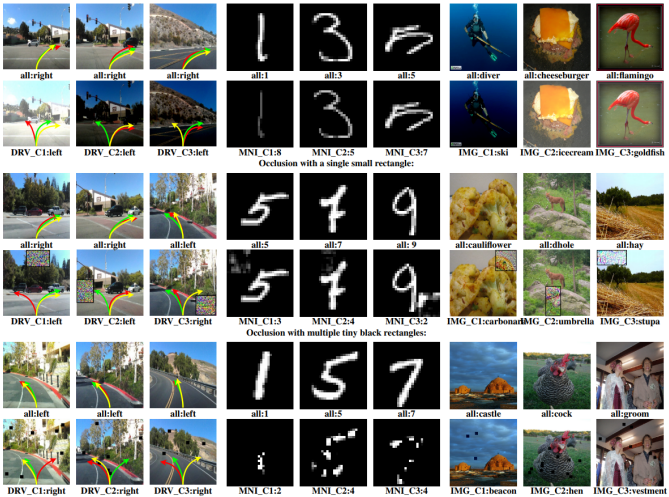
[1] [Kexin Pei et al., 2017]DeepXplore: Automated Whitebox Testing of Deep Learning Systems

Related Works-Fuzzing for AI





Related Works-Fuzzing for AI



DeepGauge[2]

- Neuron coverage is not enough:
  - k-multisection Neuron Coverage
  - Neuron Boundary Coverage (Corner Region Coverage)
  - Strong Neuron Activation Coverage (Corner Case Coverage)
- Layer coverage:
  - Top-k Neuron Coverage
  - Top-k Neuron Patterns

[2] [Lei Ma et al., 2018]DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems

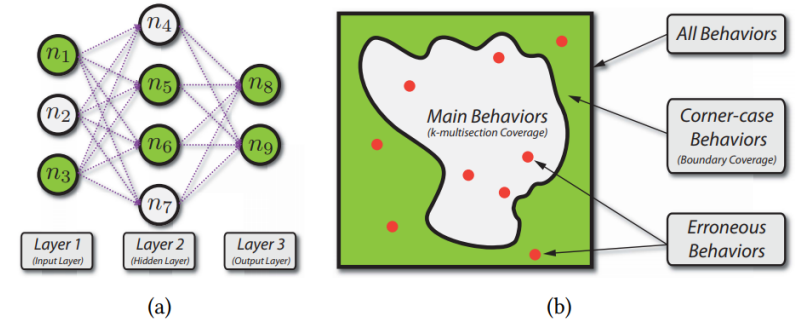


Figure 1: (a) An example of a fully connected DNN. (b) Behaviors of DNNs and relations between defined coverage criteria (the red points denote erroneous behaviors therein).

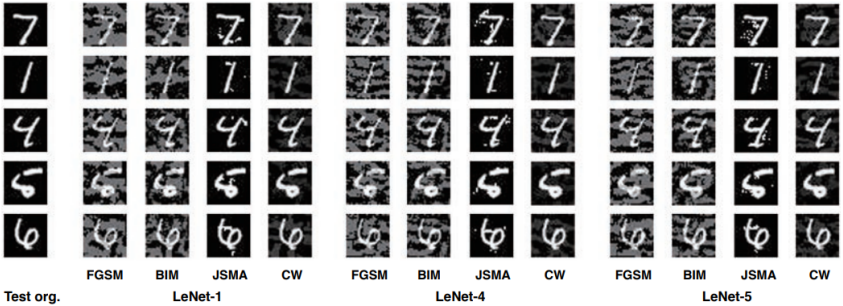


Figure 2: Examples of original sampled test data from MNIST in comparison to the ones generated by each adversarial technique on the corresponding studied DNN models.

ReluVal[3]

- Formal Security: Mathematically declared secure properties
- Goal: Achieve an exhaustive, high-performance analysis method
- How: Symbolic intervals and Interval analysis

[3] [Shiqi Wang et al., 2018]Formal Security Analysis of Neural Networks using Symbolic Intervals

- Existing adversarial testing models:
  - No guarantee of non-existence of adversarial examples
  - My conjecture:
    - Tend to overestimate
    - The example might not be applied to real life
- High overhead of SMT
  - especially for non-linear, non-convex function

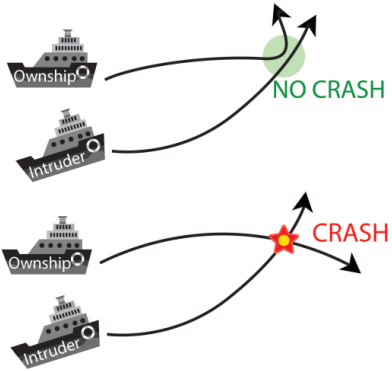
Goal:

A system for **formally** checking **security properties** of **Relu-based DNNs**

- High efficiency: ”200 times on average”
- High accuracy: ”a variety of optimizations to improve accuracy”

- Target system: ACAS Xu
- Security property: input-output-based
  - To security property
- Attacker model: similar to adversarial examples:

given a computer vision DNN f, the attacker solves following optimization problem: $min(L_p(x' - x))$ such that $f(x) \neq f(x')$, where $L_p(\cdot)$ denotes the p-norm and $x' - x$ is the perturbation applied to original input x. In other words, the security property of a vision DNN being robust against adversarial perturbations can be defined as: for any x' within a L-distance ball of x in the input space, $f(x) = f(x')$.



- Main method: interval analysis
- *Optimization*$_1$: symbolic Interval
- *Optimization*$_2$: iterative refinement:
  (existence of Lipschitz Consistency)

A Working Example: aiming to verify whether safe or not

$$\textit{Distance}: x,$$

$$\textit{Approaching angle}: y$$

$$\textit{Safe property}: x \in [4,6]\, y \in [1,5]$$

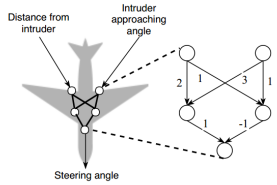
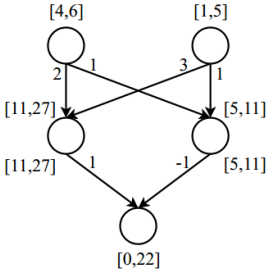


Figure 2: Running example to demonstrate our techniques.



(a) Naive interval propagation

Dependency error

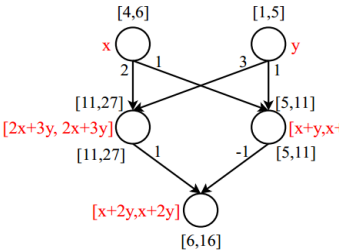
- Naively computing output intervals in this way suffers from high errors as it computes extremely loose bounds.
- Only a highly conservative estimation of the output range, too wide to be useful for checking any safety property.

Symbolic Interval and Iterative Refinement

- Symbolic interval propagation
  - explicitly represent the intermediate computations of each neuron in terms of the symbolic intervals that encode the interdependency of the inputs to minimize overestimation
- Iterative refinement
  - The dependency error for Lipschitz continuous functions decreases as the width of intervals decreases
  - Therefore, we can bisect the input interval by evenly dividing the interval into the union of two consecutive sub-intervals and reduce the overestimation

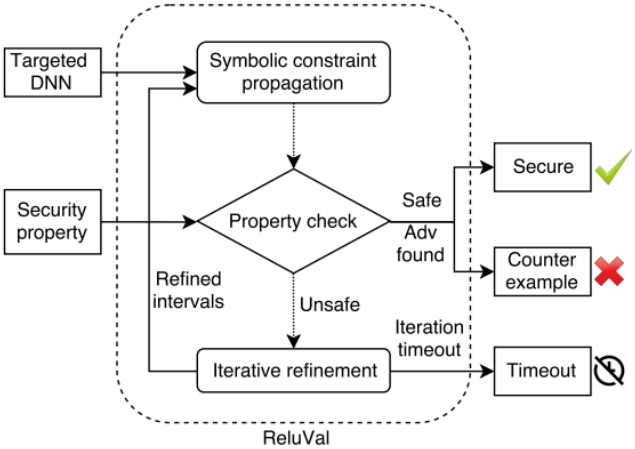
Symbolic Interval and Iterative Refinement

- Symbolic interval propagation
  (algebraic operand preservation)
- Iterative refinement
  (even interval division)



(b) Symbolic interval propagation

(c) Iterative bisection and refinement


Targeted DNN
Symbolic constraint propagation
Security property
Property check
Safe
Adv found
Secure
Counter example
Refined intervals
Unsafe
Iteration timeout
Iterative refinement
Timeout
ReluVal

Source	Properties	Networks	Reluplex Time (sec)	ReluVal Time (sec)	Speedup
Security Properties from [25]	ϕ_1	45	>443,560.73*	14,603.27	>30×
	ϕ_2	34*[2]	123,420.40	117,243.26	1×
	ϕ_3	42	35,040.28	19,018.90	2×
	ϕ_4	42	13,919.51	441.97	32×
	ϕ_5	1	23,212.52	216.88	107×
	ϕ_6	1	220,330.82	46.59	4729×
	ϕ_7	1	>86400.0*	9,240.29	>9×
	ϕ_8	1	43,200.01	40.41	1069×
	ϕ_9	1	116,441.97	15,639.52	7×
	ϕ_{10}	1	23,683.07	10.94	2165×
Additional Security Properties	ϕ_{11}	1	4,394.91	27.89	158×
	ϕ_{12}	1	2,556.28	0.104	24580×
	ϕ_{13}	1	>172,800.0*	148.21	>1166×
	ϕ_{14}	2	>172,810.86*	288.98	>598×
	ϕ_{15}	2	31,328.26	876.80	36×

* Reluplex uses different timeout thresholds for different properties.

Table 1: ReluVal's performance at verifying properties of ACAS Xu compared with Reluplex. ϕ_1 to ϕ_{10} are the properties proposed in Reluplex [25]. ϕ_{11} to ϕ_{15} are our additional properties.

# Seeds	CW	CW Miss	ReluVal	ReluVal Miss
50	24/40	40.0%	40/40	0%
40	21/40	47.5%	40/40	0%
30	17/40	58.5%	40/40	0%
20	10/40	75.0%	40/40	0%
10	6/40	85.0%	40/40	0%

Table 2: The number of adversarial inputs CW can find compared to ReluVal on 40 adversarial ACAS Xu properties. The third column shows the percentage of adversarial properties CW failed to find.

P	Adv Range	Adv	Timeout	Non-adv
S_1	$[6402.36, 10000]$	98229	1	163915
S_2	$[-0.2, -0.186]$ and $[-0.103, 0]$	18121	2	14645
S_3	$[-0.1, 0.0085]$	17738	1	15029

Table 3: The second column shows the input ranges containing at least one adversarial input, while the rest of ranges are found by ReluVal to be non-adversarial. The last three columns show the number of total sub-intervals checked by ReluVal with a precision of $e-6$.

Outline I

NEUZZ[4]

- Goal: Find lightweight solutions with efficiency and reasonable accuracy for fuzzing
- How: use CNN to train with inputs: compute edge coverage
- How: Gradient decent twith sign(increase/decrease by 1)

[4] [Dongdong She et al., 2017]NEUZZ: Efficient Fuzzing with NeuralProgram Smoothing

Neural byte sieve[5]

- Goal: Use Machine Learning to learn guiding strategy based on input history and code coverage

[5] [Mohit Rajpal et al., 2017]Not all bytes are equal: Neural byte sieve for fuzzing

Deep Reinforce Learning[6]

- Formalize fuzzing procedure into reinforcement learning

[6] [Konstantin Bottinger et al., 2018]Deep Reinforcement Fuzzing

Outline I

Data Poisoning

- Given a picture
- Calculate how much perturbation to add
- Goal: Make Classifier misclassify or do other jobs

[7] [Ian J. Goodfellow et al., 2014]Explaining and Harnessing Adversarial Examples

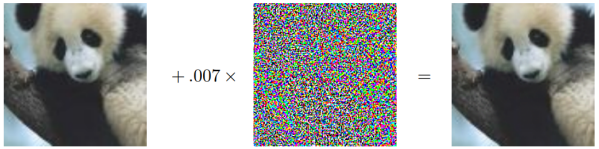


Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

$$\eta = \epsilon sign(\bigtriangledown_x J(\theta, x, y))$$

Acoustic Eavesdropping

- Previous method: weak assumption, strong adversarial examples
- Scene: Known keyboard type, known user habit of typing(sounds)
- Goal: Learn the user typing and tell keystroke

[8] [A. Compagno et al., 2017]Don't Skype and Type! Acoustic Eavesdropping in Voice-Over-IP

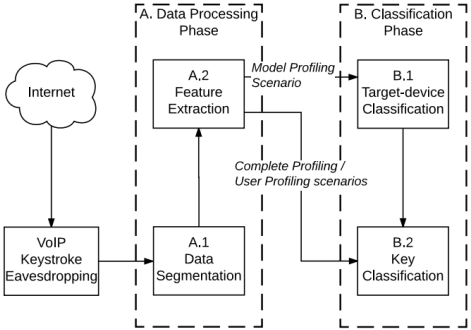


Figure 2: *S&T attack* steps.

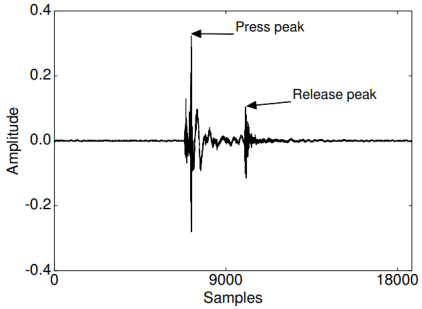
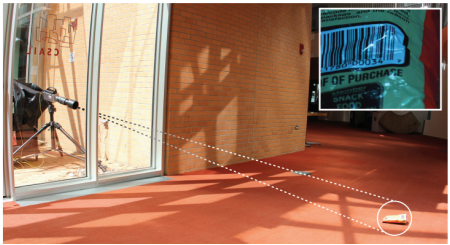


Figure 3: Waveform of the "*A*" key, recorded on an Apple Macbook Pro 13" laptop.

The Visual Microphone

- Target: Human speaking
- Gadget: crabchip/tissue/flower...+high frame rate camera
- Goal: Rebuild sound signal from vibration

[9] [Abe Davis et al., 2014]The Visual Microphone: Passive Recovery of Sound from Video



(a) Setup and representative frame

(b) Input sound (c) Recovered sound

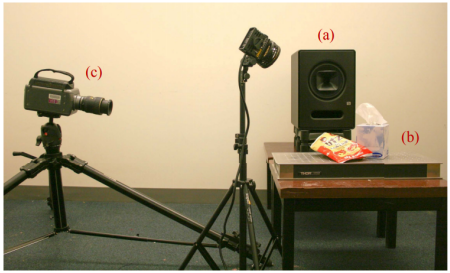
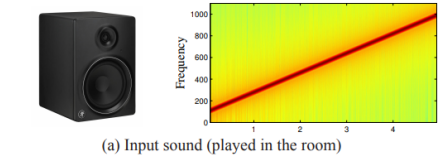


Figure 4: *An example of our controlled experimental setup. Sound from an audio source, such as a loudspeaker (a) excites an ordinary object (b). A high-speed camera (c) records the object. We then recover sound from the recorded video. In order to minimize undesired vibrations, the objects were placed on a heavy optical plate, and for experiments involving a loudspeaker we placed the loudspeaker on a separate surface from the one containing the objects, on top of an acoustic isolator.*

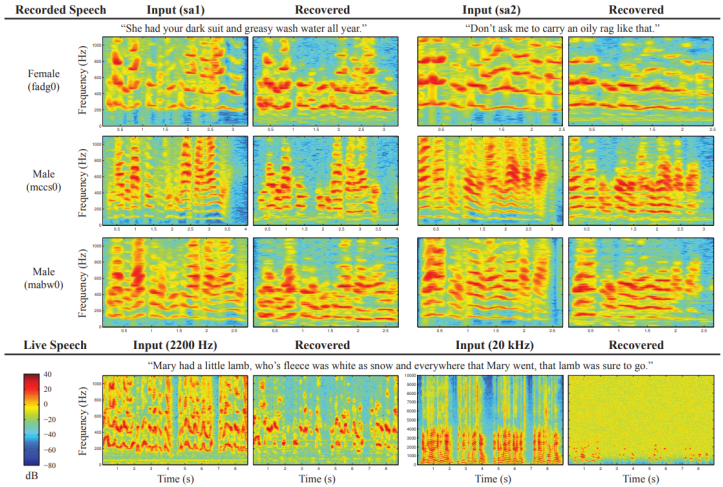
Related Work



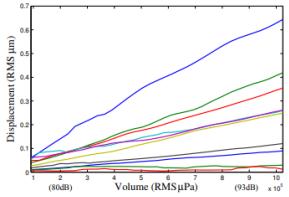
(a) Input sound (played in the room)

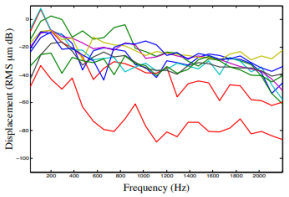
(b) Reconstructed sound

Related Work



	Measured displacement increase rate (μm/Pa)
crabchips	62.4
greenteabox	41.1
tissue	32.9
foiltogo	20.6
kitkat	21.6
afoil	24.6
rose	10.6
foamcup	8.6
chobani	1.2
teapot	1.1





(a) Displacement coefficients at 300Hz (b) Motion vs. sound volume (c) Frequency responses

Figure 7: *Object motion as function of sound volume and frequency, as measured with a laser Doppler vibrometer. Top: the objects we measured, ordered according to their peak displacement at 95 dB, from left (larger motion) to right (smaller motion). (b) The RMS displacement (micrometers) vs RMS sound pressure (Pascals) for the objects being hit by a calibrated 300Hz sine wave linearly increasing in volume from 57 decibels to 95 decibels. Displacements are approximately linear in Pascals, and are all in the order of a micrometer (one thousandths of a millimeter). (c) The frequency responses of these objects (Power dB vs frequency), based on their response to a ramp of frequencies ranging from 20Hz to 2200Hz. Higher frequencies tend to have weaker responses than lower frequencies. Frequency responses are plotted on a dB scale, so the relative attenuation of higher frequencies is quite significant.*

Dolphin Attack

- Target:Voice Assistant
- Method: Convert human voice command into higher frequencies
- Result: iPad、iPhone、MacBooks、Apple Watch、Amazon Echo, ThinkPad T440p

[10] [Guoming Zhang et al., 2017]Dolphin Attack: Inaudible Voice Commands

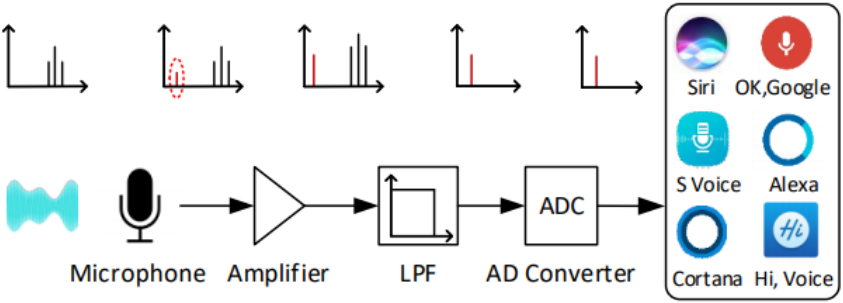


Figure 3: An illustration on the modulated tone traversing the signal pathway of a voice capture device in terms of FFT.

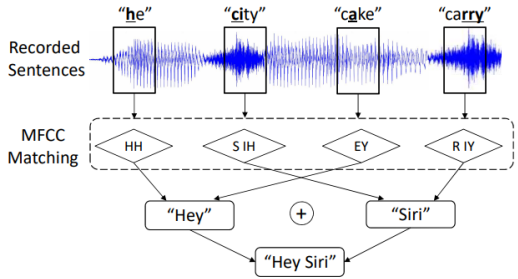


Figure 8: Concatenative synthesis of an activation command. The MFCC feature for each segment in a recorded sentence is calculated and compared with the phonemes in the activation command. After that, the matched voice segments are shuffled and concatenated in a right order.

Manuf.	Model	OS/Ver.	SR System	Attacks		Modulation Parameters		Max Dist. (cm)	
				Recog.	Activ.	f_c (kHz) & [Prime f_c] ‡	Depth	Recog.	Activ.
Apple	iPhone 4s	iOS 9.3.5	Siri	√	√	20–42 [27.9]	≥ 9%	175	110
Apple	iPhone 5s	iOS 10.0.2	Siri	√	√	24.1 26.2 27 29.3 [24.1]	100%	7.5	10
Apple	iPhone SE	iOS 10.3.1	Siri	√	√	22–28 33 [22.6]	≥ 47%	30	25
			Chrome	√	N/A	22–26 28 [22.6]	≥ 37%	16	N/A
Apple	iPhone SE †	iOS 10.3.2	Siri	√	√	21–29 31 33 [22.4]	≥ 43%	21	24
Apple	iPhone 6s *	iOS 10.2.1	Siri	√	√	26 [26]	100%	4	12
Apple	iPhone 6 Plus *	iOS 10.3.1	Siri	×	√	— [24]	—	—	2
Apple	iPhone 7 Plus *	iOS 10.3.1	Siri	√	√	21 24-29 [25.3]	≥ 50%	18	12
Apple	watch	watchOS 3.1	Siri	√	√	20–37 [22.3]	≥ 5%	111	164
Apple	iPad mini 4	iOS 10.2.1	Siri	√	√	22–40 [28.8]	≥ 25%	91.6	50.5
Apple	MacBook	macOS Sierra	Siri	√	N/A	20-22 24-25 27-37 39 [22.8]	≥ 76%	31	N/A
LG	Nexus 5X	Android 7.1.1	Google Now	√	√	30.7 [30.7]	100%	6	11
Asus	Nexus 7	Android 6.0.1	Google Now	√	√	24–39 [24.1]	≥ 5%	88	87
Samsung	Galaxy S6 edge	Android 6.0.1	S Voice	√	√	20–38 [28.4]	≥ 17%	36.1	56.2
Huawei	Honor 7	Android 6.0	HiVoice	√	√	29–37 [29.5]	≥ 17%	13	14
Lenovo	ThinkPad T440p	Windows 10	Cortana	√	√	23.4–29 [23.6]	≥ 35%	58	8
Amazon	Echo *	5589	Alexa	√	√	20-21 23-31 33-34 [24]	≥ 20%	165	165
Audi	Q3	N/A	N/A	√	N/A	21–23 [22]	100%	10	N/A

‡ Prime f_c is the carrier wave frequency that exhibits highest baseband amplitude after demodulation. — No result

† Another iPhone SE with identical technical spec.

* Experimented with the front/top microphones on devices.