I Spy You: Eavesdropping Continuous Speech on Smartphones via Motion Sensors

SHIJIA ZHANG, The Pennsylvania State University, USA
YILIN LIU, The Pennsylvania State University, USA
MAHANTH GOWDA, The Pennsylvania State University, USA

This paper presents *iSpyU*, a system that shows the feasibility of recognition of natural speech content played on a phone during conference calls (Skype, Zoom, etc) using a fusion of motion sensors such as accelerometer and gyroscope. While microphones require permissions from the user to be accessible by an app developer, the motion sensors are zero-permission sensors, thus accessible by a developer without alerting the user. This allows a malicious app to potentially eavesdrop on sensitive speech content played by the user's phone. In designing the attack, *iSpyU* tackles a number of technical challenges including: (i) Low sampling rate of motion sensors (500 Hz in comparison to 44 kHz for a microphone). (ii) Lack of availability of large-scale training datasets to train models for Automatic Speech Recognition (ASR) with motion sensors. *iSpyU* systematically addresses these challenges by a combination of techniques in synthetic training data generation, ASR modeling, and domain adaptation. Extensive measurement studies on modern smartphones show a word level accuracy of 53.3 – 59.9% over a dictionary of 2000-10000 words, and a character level accuracy of 70.0 – 74.8%. We believe such levels of accuracy poses a significant threat when viewed from a privacy perspective.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile devices**; • **Security and privacy → Embedded systems security**.

Additional Key Words and Phrases: IoT Security, Speech Privacy

ACM Reference Format:
Shijia Zhang, Yilin Liu, and Mahanth Gowda. 2022. *I Spy You*: Eavesdropping Continuous Speech on Smartphones via Motion Sensors . *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 197 (December 2022), 31 pages. https://doi.org/10.1145/3569486

1 INTRODUCTION

There is a surge in Internet-of-Things (IoT) devices in applications including smart-homes, industrial automation, self-driving cars [22, 71] with rich sensing capabilities. However, such a rich ecosystem of IoT sensors are often considered "double-edged swords" since they leak private information [9, 18, 47, 89]. To understand what level of leakage is appropriate, the key question boils down to: *How much information can be inferred from a given sensor data?* While this question is a subject of an active area of research, emerging advances in hardware, software, and machine learning warrant constant attention to information leakage. In this context, this paper asks the following question: *Can motion sensor data (also called as IMU - Inertial Measurement Unit) from accelerometer and gyroscope be used to eavesdrop on continuous speech content played on a smartphone (during skype, zoom calls, interaction with voice assistants, etc.)?* If so, it could impose a serious privacy threat. A malicious app disguised

Authors' addresses: Shijia Zhang, scarlettzhang27@psu.edu, The Pennsylvania State University, USA; Yilin Liu, yzl470@psu.edu, The Pennsylvania State University, USA; Mahanth Gowda, mahanth.gowda@psu.edu, The Pennsylvania State University, USA.

2474-9567/2022/12-ART197 $15.00
https://doi.org/10.1145/3569486

as an activity tracker can eavesdrop on sensitive speech content. Unlike microphones which require explicit permissions from the user for access by app developers, motion sensors (accelerometer and gyroscope) have unrestricted access, thus providing a side channel for eavesdropping speech.

Detailed in the threat model in Section 4, accelerometers and gyroscopes are zero-permission sensors on popular mobile operating systems like Android, thus allowing their free use by developers without alerting the user. When speech content is played on a smartphone, these sensors can record the vibrations, thus causing information leakage from the loudspeaker to the motion sensors. As shown in prior works, an adversary can disguise an app (for example, as a fitness tracking app) for eavesdropping on speech content [3, 6, 47, 89], which will be used for performing the attack by exploiting the above leakage. In particular, recent android smartphones such as OnePlus 9 Pro, Samsung S20 and Huawei P20 provide unrestricted access to sensor data with sampling rates upto 500 Hz when sampled in SENSOR_DELAY_FASTEST mode [4, 6]. This covers a key range of human speech frequencies, thus allowing a malicious app developer to eavesdrop speech content and compromise the privacy of users. Given that Android OS has a market share of about 72% worldwide , we believe this is a critical concern related to speech privacy [48].

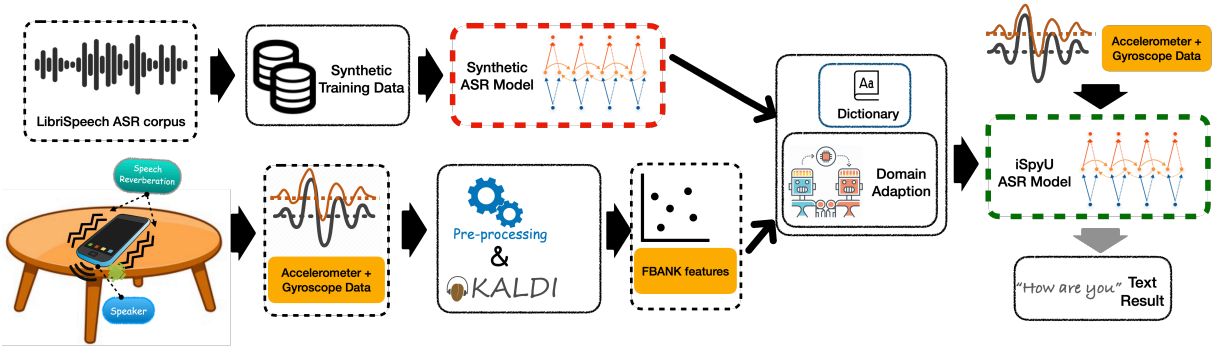


Fig. 1. Overall Architecture of *iSpyU* for spying smartphone speakers via motion sensors (accelerometer and gyroscope): Synthetic training data generated from large scale speech datasets is combined with small scale training data from real-world motion sensors – this generates *iSpyU*'s speech recognition models.

Exploiting the above vulnerability, we propose *iSpyU* (IMU based Spying), a system that demonstrates the feasibility of an automatic speech recognition (ASR) attack for converting such motion sensor vibrations into text. The COVID-19 pandemic has forced people to embrace remote working. Many companies, including Facebook and Twitter, plan for permanent remote working jobs even after the pandemic ends [66]. Given that smartphones are actively involved in many virtual video/audio conferencing calls (Zoom, Skype, etc) to deliver the speech content from the call, we believe the ability to eavesdrop on spoken natural languages and convert them into text is a critical concern.

Prior works in this area include Gyrophone [47], Spearphone [3], and AccelEve [6]. Gyrophone uses gyroscope to detect 11 digits with an accuracy of 26%. Spearphone classifies 58 words spoken in isolation with 67% accuracy using accelerometers. AccelEve can classify 10 digits and 26 alphabets with 55% accuracy. However, the attack capabilities shown in prior works made unrealistic assumptions with the attack only being limited to digits,

Table 1. Scope of *iSpyU* in the context of key prior works on motion sensors. *iSpyU+* is a version of *iSpyU* where we assume a smaller dictionary size of 2000 words at test time for decoding each sentence. To our best knowledge, *iSpyU* is the first work on continuous speech recognition with ASR over thousands of words in the dictionary.

System	Sensor	Source of Sound	Recognition Task	Dictionary Size	Accuracy
Gyrophone [47]	Gyroscope	External Loudspeaker (Subwoofer)	Isolated Words	11 (digits)	26%
SpearPhone [3]	Accelerometer	Smartphone loudspeaker	Isolated Words	58 (keywords)	67%
AccelEve [6]	Accelerometer	Smartphone loudspeaker	Isolated Words	36 (10 digits + 26 alphabets)	55%
iSpyU	**Accelerometer + Gyroscope**	**Smartphone loudspeaker**	**Continuous Speech (ASR)**	**9950 words**	**53.3%**
iSpyU+	**Accelerometer + Gyroscope**	**Smartphone loudspeaker**	**Continuous Speech (ASR)**	**2000 words**	**59.9%**

alphabets, or certain words spoken in isolation, the privacy threat was not considered serious enough. In contrast to detecting tens of digits or keywords spoken in isolation, *iSpyU* spies on naturally spoken continuous speech with a dictionary size $\approx$ 10000 words. Although iSpyU is still comparable to their respective previous work setting, we believe the ability to perform ASR on a large vocabulary is a new contribution in iSpyU that has not been explored before. With this capability, an adversary can potentially launch a massive attack. For example, insurance companies could infer health status of several of their customers in automated ways, eliminating the need to manually process each client's data. Advertising companies can similarly learn customer interests in automated ways and place targeted advertisements on the site massively. We believe such an eavesdropping of real-world conversations is of critical concern when speech privacy is of interest.

To summarize, *iSpyU* differs from prior works in the following ways (overview in Table 1): (i) In contrast to detecting isolated keywords or digits, *iSpyU* performs ASR on motion sensor data to convert natural spoken languages into text – first such attempt to our best knowledge. Since humans mostly communicate in sentences instead of single words, we believe *iSpyU* performs a more realistic attack than prior works.
(ii) In contrast to 50-100 specific keywords used in prior works, *iSpyU*'s dictionary space is much larger: $\approx$ 10000 words [57]. This can cover a majority of words used in natural conversation, thereby making the attack stronger.
(iii) In contrast to using accelerometer and gyroscope separately, *iSpyU* fuses them together and evaluates the capabilities offered by individual sensors, and with fusion.

Performing an end-to-end ASR on motion sensors is challenging for many reasons: (i) The sensor sampling rate is low, whereas human speech frequencies can range upto 8kHz. (ii) Unlike online audio datasets for training speech-based ASR, there is no large-scale training data available for developing ASR models for motion sensors. (iii) Therefore, *iSpyU* generates synthetic training data from online speech datasets. This introduces a domain adaptation problem due to differences in distribution between synthetic and real motion sensor data. (iv) Sentences in a natural conversation include words that blend seamlessly into each other, thereby making it challenging to identify them using motion sensor data. (v) The sensor data is inherently noisy. The spectral distribution differs from that of typical audio.

Towards handling the above challenges, *iSpyU* exploits many opportunities. (i) *iSpyU* fuses accelerometer and gyroscope data together towards harvesting the best information possible from a limited sampling rate. (ii) *iSpyU* creates synthetic motion sensor data for training by performing signal processing transformations on speech samples from large corpus of online speech datasets [57] (iii) *iSpyU* then performs fine-tuning of ASR models trained with synthetic data to systematically handle the residual differences in distributions of synthetic and real data. In particular, deep-learning based ASR models have millions of parameters. Fine-tuning these layers directly requires a lot of training data for convergence. Therefore, *iSpyU* introduces tunable Linear Hidden Networks (LHN) layers into the ASR model with a fewer set of parameters capable of effective fine-tuning. (iv) *iSpyU* designs machine learning models based on attention-mechanism, and performs weakly supervised learning using

an appropriate sequential probability thus obviating the need to segment words in a sentence. Furthermore, language models are incorporated that can fill in the gaps in sensing information based on the context from surrounding words. (v) Preprocessing techniques like spectral subtraction are exploited to handle noisy data.

Fig. 1 depicts the overall architecture. The ASR model is trained on synthetic motion sensor data generated from online speech datasets [57]. The synthetic model thus created (shown in red dotted box) is fine-tuned using small scale real sensor data from a smartphone to generate an ASR model (shown in green dotted box) suitable for converting motion sensor data into text. The attacker does not need any training data from the victim's smartphone but generates their own training data for fine-tuning.

iSpyU is implemented on OnePlus 9 Pro, Samsung Galaxy S20, and Huawei P20 smartphones and tested over various surfaces used in daily life such as table, carpet, hand (even while walking), sofa, floor, etc. Because the speaker and IMU share the same motherboard, there is a strong direct channel between the two. Therefore, the impact of dampening by external surfaces in contact with the phone is negligible.
The ASR models were implemented using PyTorch and trained on a Nvidia Quadro RTX 8000 GPU. The training data is based on synthetic datasets derived from the popular LibriSpeech dataset. Given that LibriSpeech data uses sophisticated vocabulary than a typical everyday conversation, it is known to be a popular benchmark for testing ASR applications including Baidu's Deep Speech [1] and voice assistants such as Amazon Alexa [72]. Testing with an independent and diverse set of users and sentences was conducted. The accuracy at the word level varies between $53.3 - 59.9\%$ over a dictionary of $2000 - 9950$ words. At the character level, the accuracy can be higher ($70.0 - 74.8\%$) since incorrect words can be close matches to the ground truth. While the accuracy is not ideal for a usability related ASR application such as a voice assistant, it might be of critical concern when privacy is of interest. For instance, these levels of accuracy are known to be sufficient enough to even hold a basic conversation by inferring missing words from context [39, 40, 42, 43], thus potentially revealing sensitive information about the victim's location, mood, health status, political inclination, etc. Inspection of raw sentence decodings (examples in Sec. 7) suggests that sensitive information about the context of communication can be inferred thus compromising privacy. Furthermore, some of the wrong words are close matches to correct words (Friday was decoded as Friday's). While decoding in *iSpyU* does not cross boundaries of sentences, we sketch ideas that exploit Natural Language Processing (NLP) in Sec. 8 which could potentially exploit context across sentences for enhancing the accuracy in the future.

Considering the above possibilities, the contributions in *iSpyU* can be summarized as follows:
(i) *Design of an eavesdropping attack with motion sensors to spy on continuous speech on phone loudspeakers.* In contrast to prior work, our dictionary size is 100 fold increase from less than 100 words in prior work to 10000 in our paper. (ii) *Design of synthetic training datasets to efficiently train deep learning based ASR models.* The strategy adopted by iSpyU in using a combination of large quantities of synthetic training data plus small quantity of real world data achieves a sweet-spot in the trade-off between training overhead and accuracy. (iii) Fusion of attention mechanisms, weakly supervised learning, and language models for performing ASR on motion sensor data with insufficient information. Thus we can focus on continuous speech and obviate the need to segment a sentence into individual words during training or inference. (iv) *Efficient domain adaptation of ASR models trained with synthetic data for improving the accuracy of inferences on real sensor data.* iSpyU introduces tunable Linear Hidden Networks (LHN) layers into the ASR model with a fewer set of parameters capable of effective fine-tuning. (v) *Implementation and evaluation on off-the-shelf smartphones.*

2 RELATED WORK

Table 1 provides an overview of key related work, the details are elaborated below.

2.1 Side-Channel Attacks on Mobile Sensors

Mole [89] uses smartwatch accelerometer to spy on contents of a user's typing. S3 [18] detects drawings on a tablet using an apple pencil by exploiting variations in magnetic fields sensed by the magnetometer. Accelerometer sensors are also known to reveal passwords as entered on the touchscreen of a phone [56]. The smartphone magnetometers are even shown to be capable of identifying the operating systems and the pattern of applications in a nearby desktop by monitoring the spinning of hard-drives which are made of magnetic materials [9]. More recently, magnetometer sensors are exploited to spy on applications on a smartphone [30]. In contrast to these works, *iSpyU* performs an attack on spying speech contents from the accelerometer signals.

2.2 Spying on Speech Content

Gyrophone [47] detects speech content from an external loudspeaker (subwoofer) using gyroscope sensors placed on the same surface (for example, shared table). Classification of 11 digits (0-9 and "oh") is shown with the best accuracy under speaker-independent case being 26%. Similarly, Speechless [2] shows the sensitivity of smartphone accelerometers to loudspeakers but does not perform word classification or ASR. Spearphone [3] classifies 58 words spoken in isolation with 67% accuracy using accelerometers. AccelEve[6] can classify 10 digits and 26 alphabets with 55% accuracy. AccelWord [95] shows the feasibility of detecting the wakeup keywords of voice commands such as "Okay Google", and "Hi Galaxy" using accelerometers. PitchIn [23] shows the feasibility of eavesdropping ambient speech by fusing data from multiple non-acoustic sensors (accelerometers, gyroscope, geophone, etc). Evaluated over 40 words, enhancing the sampling rate to 8 kHz by above fusion is needed to reach an accuracy of 50%, whereas the accuracy is around 5% even with a sampling rate of 1000 Hz. Work in [5] shows the feasibility of detecting digits and four specific phrases using motion sensors. In contrast to such works, to our best knowledge, *iSpyU* is the first work to show the limits of performing end-to-end ASR on continuous speech over a large dictionary of $\approx$ 10000 words. Since humans communicate in sentences instead of single words, we believe *iSpyU* performs a more realistic attack over a large dictionary of words. In addition to smartphones, other forms of attacks have been explored. LidarPhone [69] uses lidar sensors on vacuum cleaning robots to spy on speech within a room. Lamphone[53] analyzes vibration on a light bulb due to sound pressure variations using electro-optical sensors. Visual Microphone [15] records sound vibrations on objects in the environment using cameras. In contrast, *iSpyU* shows the feasibility of spying on natural speech content on a smartphone loudspeaker.

2.3 ASR Models

HMMs were popular in the early days of ASR [64]. Since the rise of deep learning, hybrid DNN-HMM models have emerged that use a DNN encoder to extract features and a HMM decoder for outputting labels [38]. Connectionist Temporal Classifcation (CTC) [21] revolutionized ASR since it provides a way of weakly supervised learning with dramatically low overhead of labeling or segmenting the speech data. It also relaxes independence assumptions made by HMM models to provide robust recognition. Recently, attention-based models have been proposed for vision and NLP tasks [11, 46, 88] that exploit stronger spectro-temporal relationships not only between various parts of the audio input, but also between inputs and currently decoded outputs. *iSpyU* incorporates such an attention-based ASR model but customizes it for working with motion sensor data.

2.4 Domain Adaptation

Transfer-learning-based domain adaptation is popular in vision and speech processing. For example, AlexNet model [36] pretrained on ImageNet database [16] was fine-tuned for classifying images in medical domain[96], remote-sensing [24] and breast cancer [54]. Similarly, a pre-trained BERT language model [17] was fine-tuned for tasks such as text-summarizing [94], question answering [63], etc. This significantly reduces the burden of training for a new task. In a similar spirit, we use a pretrained model from synthetic motion sensor data. While this provides a good enough base model to begin with, we adapt the model with real sensor data. As discussed in Sec. 5, our domain adaptation trains only a few parameters to significantly decrease the overhead of training data generation.

3 OVERVIEW OF MOTION SENSORS

We provide a brief overview of the motion sensors.

3.1 Motion Sensors and Vibrations

3.1.1 Accelerometer. A micro electro mechanical system (MEMS) capacitive accelerometer is depicted in Fig. 2(a). A *proof-mass* is suspended by a spring system in between an array of fixed electrodes. Each mass provides the moving plate of a variable capacitance formed by an array of interlaced *fingers* [45]. When the accelerometer is in motion, the *proof-mass* resists the motion due to inertia as shown in Figure. 2(b). This causes variation in relative distances between the *proof-mass* and fixed electrodes. This displacement induces a differential capacitance

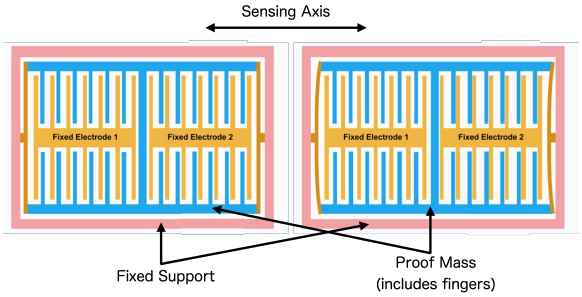


Fig. 2. A capacitive accelerometer (a) Accelerometer in rest (b) Accelerometer under Motion.

between the moving and fixed silicon *fingers* which is proportional to the applied acceleration.

3.1.2 Gyroscope. Fig. 3 depicts the high-level overview of a MEMS gyroscope. A proof mass is suspended between fixed electrodes and set to vibrate in a specific direction as indicated in the figure. When the gyroscope undergoes rotational motion, this introduces a Coriolis force [14] proportional to the angular velocity. This force is in a direction perpendicular to both the direction of vibration and the axis of rotation as indicated in the figure. This force moves the mass as shown in Figure 3(b) relative to the fixed electrodes. By measuring the change in capacitance, the Coriolis force, and hence the angular velocity can be estimated.

3.2 Impact of Speech on Motion Sensors

Accelerometer and gyroscope sensors are key components of applications in motion tracking including virtual and augmented reality, localization, sports analytics, etc [51, 91]. *iSpyU* performs a side-channel attack on these sensors. When a speech content is played by a smartphone speaker, this will induce vibrations in the sensors. Fig. 4 shows an example of the accelerometer and gyroscope readings when an audio content saying "A golden

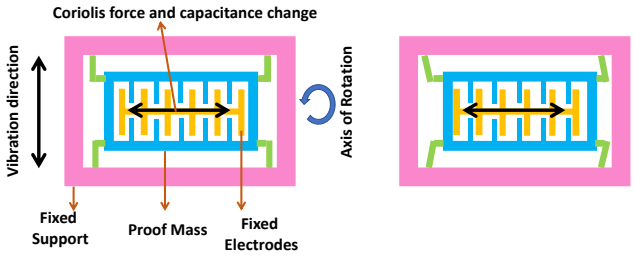


Fig. 3. (a) Components of a capacitive gyroscope (b) Gyroscope under motion

fortune and a happy life" is played on the speaker. Evidently, the accelerometer and gyroscope sensors are able

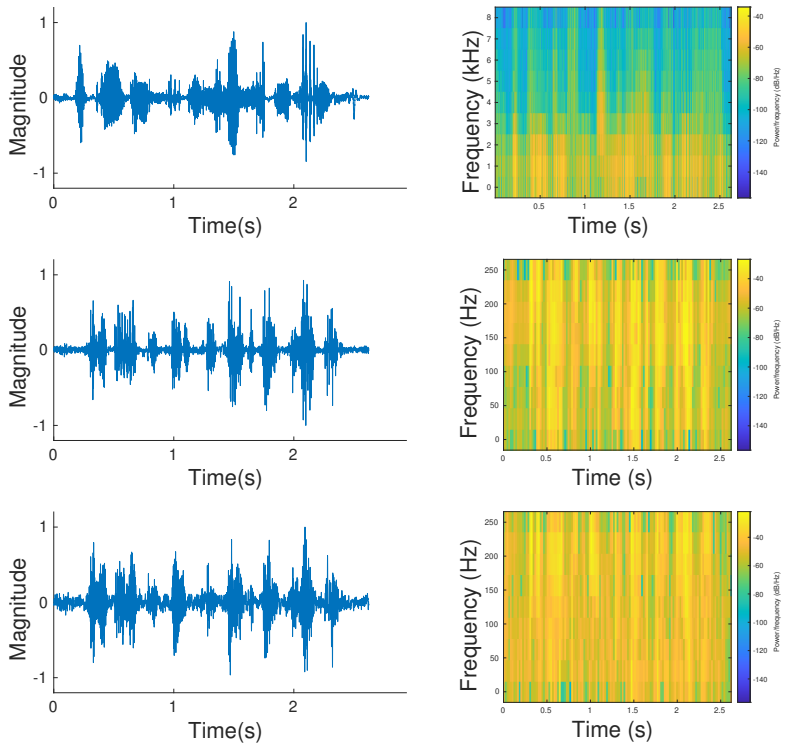


Fig. 4. (Top) Audio content for "A golden fortune and a happy life" and the spectrogram. (Middle) Corresponding accelerometer signal and the spectrogram (Bottom) Corresponding gyroscope signal and the spectrogram.

to capture the vibrations. *iSpyU* shows the limits and bounds of decoding speech content using such vibration leakage.

In addition to the sensitivity of the motion sensor hardware to vibrations, the sampling rate is another critical factor that determines the feasibility of decoding speech content. Table 2 summarizes the sampling rates available

Table 2. Android sampling rates in different settings

Setting	Delay	Sampling rate
SENSOR_DELAY_NORMAL	200ms	5 Hz
SENSOR_DELAY_UI	20ms	50 Hz
SENSOR_DELAY_GAME	60ms	16.7 Hz
SENSOR_DELAY_FASTEST	-	CPU dependent

in Android. Specifically, under the setting of *SENSOR_DELAY_FASTEST*, recent smartphones such as OnePlus 9 Pro, Samsung Galaxy S20, Huawei P20, Google Pixel 4, etc, can support a high sampling rate upto 500*Hz* [4, 6]. A natural question is: *Is it possible to capture human speech with 500 Hz?* The fundamental frequency of vocal cord vibrations for a male speaker varies between 85-180 Hz, whereas, for a female speaker, it varies between 165-255 Hz [7]. While the 500 Hz sampling rate sufficiently covers this frequency range (0-250 Hz), thus opening up opportunities, it is not sufficient to provide perfect intelligibility [78]. Fig. 5 shows the importance of various frequencies in the intelligibility of speech signals. Evidently, the higher frequency components that involve the use of consonants are critical for higher intelligibility. Unfortunately, with a sampling rate of 500 Hz, the higher frequency components can only be sensed in aliased form on the motion sensors. *iSpyU*'s ASR models (Sec. 5) will propose ideas for extracting speech content from such aliased signals.

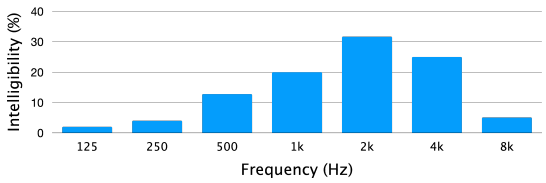


Fig. 5. Relative importance of speech frequencies [78]

4 THREAT MODEL

We elaborate on the threat model used in *iSpyU*. The overall threat model (depicted in Fig. 6) is a stronger version of the threat model considered in prior works such as SpearPhone [3], GyroPhone [47], and AccelEve [6]. *iSpyU*'s threat model differs from prior works with the capability to spy on natural conversations with upto 10000 words in the dictionary. Also, it has been shown in prior works that an adversary can disguise an app (for example, as a fitness tracking app) for eavesdropping on speech content [3, 6, 47, 89], which will be used for performing the attack. Unlike microphones, the motion sensors are *zero permission* sensors (even with latest Android 12), thus allowing the spying app to capture the data without alerting the user. The spying app can have access to the motion sensors with sampling rates up to 500 Hz on recent phones [6] without special permissions from the user. While this covers a key range of human speech frequencies (discussed in Sec. 3), with the increasing trend in CPU speed, we expect that the sampling rates may go up in the future, thus making the attack stronger [49, 50]. With these capabilities, an attacker can potentially spy on speech contents of a remote caller during video/audio conferencing (Zoom, Skype, etc) with the victim. With the advent of COVID-19 pandemic, and a thrust towards permanent remote working jobs in many companies including Facebook, Twitter, etc [66], we believe this is of critical concern because most users rely on online video/speech conferencing for conducting daily work activities. Smartphones/tablets are among the popular devices for participating in online conferencing. In addition, with the rising popularity of voice assistants (Google Assistant, Siri, etc) on smartphones, the responses of the smart

assistant during interaction with the user can also be a target of the spying attack. The above cases involve communication in a natural language such as English. The attacker will develop ASR models for using the motion sensor data to convert such spoken natural language content into text. In developing the ASR model, the attacker does not need labeled training samples from the victim's smartphone. The attacker will synthesize training data (details in Section 5) from online speech datasets to create a basic ASR model. The attacker will then fine-tune the model with small amounts of labeled sensor dataset from their own phone. Finally, the ASR model thus developed is used to launch the attack by converting the motion sensor data into text.

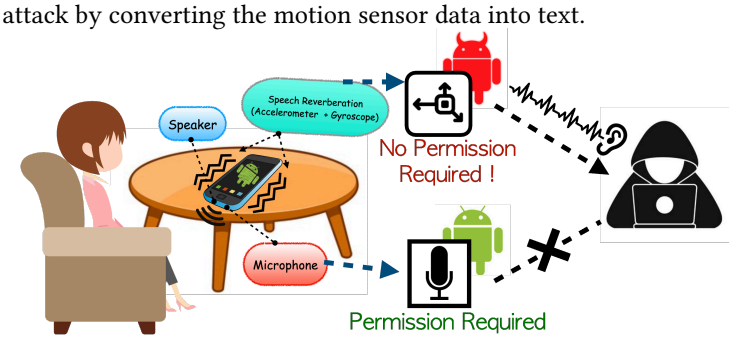


Fig. 6. Threat model of *iSpyU*.

5 ASR WITH MOTION SENSORS

We elaborate on various modules in the architecture of *iSpyU* as depicted in Fig. 1.

5.1 Synthetic Training Data Generation

The robustness of ASR models depend heavily on large-scale training datasets with diversity in speakers, genders, accents, etc. Unlike speech domain, there is a dearth of large-scale training data for motion sensors. Thus, we design synthetic training data as follows.

5.1.1 Subsampling. LibriSpeech audio data is sampled at 16 kHz. However, the sampling rate of the motion sensors is only 500 Hz. The high-frequency data superimposes onto lower frequencies in aliased form. Towards emulating this, *iSpyU* subsamples the speech data at 500 Hz before feeding it to the ASR model. While subsampling loses information, there are spectral dependencies across speech components [33] that can be leveraged to infer speech content from such lossy information.

5.1.2 Synthetic Noise Modeling. Although we employ preprocessing techniques based on filtering, and spectral subtraction to eliminate noise in the sensor data (detailed in Sec. 5.3), a residual noise still persists. Towards the creation of synthetic data that best matches the nature of real data, we add systematic noise to the synthetic data based on the distribution of the residual noise in the real sensor data. We first measure the noise distributions for accelerometer and gyroscope data. We then measure the signal strength of sound vibrations when speech is played on the phone. Using this, the signal-to-noise ratio (SNR) for the sensor data is computed. Finally, we add noise to the synthetic data such that the SNR of the synthetic data matches with the real data. Fig. 7 shows an example of synthetic accelerometer data and the corresponding real data.

5.2 Speech Detection

We first show the feasibility of distinguishing speech segments from intervals where there is silence or human motion activities like walking, running, etc. Evidently, based on results in Fig. 4, the occurrence of speech signal

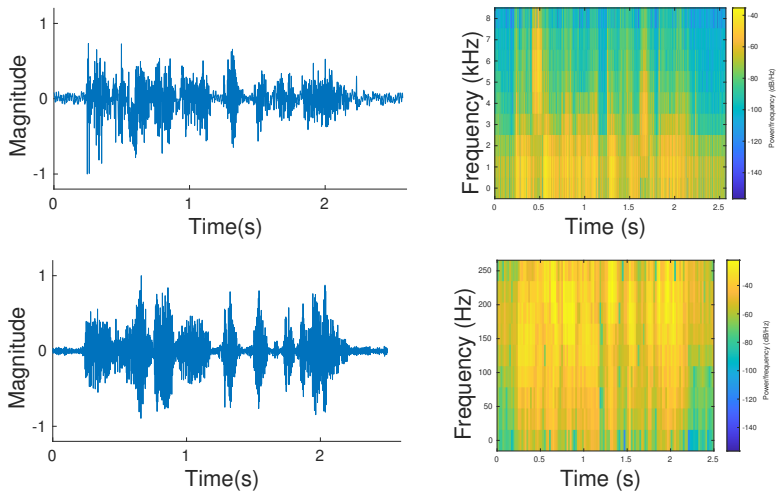


Fig. 7. (Top) Synthesized accelerometer data. (Bottom) Real Accelerometer data (noise subtracted).

is very apparent not only because of a high SNR but also because of a rich spectro-temporal pattern. We build a shallow neural network to classify the following three activities - *silence, motion activity (walking, running, etc), speech*. The data is processed in chunks of 50ms with a windowed approach with sliding length of 20ms (60% overlap across successive chunks). We achieve a perfect accuracy in speech detection because of the following reasons: (i) Unlike microphone which can be polluted by ambient noise, the IMU sensors are not affected by ambient noise through air medium, but only influenced by loudspeaker vibration which is transferred to IMU via shared motherboard (solid medium). Therefore, it is easier to distinguish speech from the hardware noise and there is no influence of ambient noise. In addition, recent conferencing apps like Skype and Zoom incorporate sophisticated signal processing algorithms for background noise elimination and speaker voice isolation [55, 86] thereby enhancing the speech quality. (ii) A typical spoken sentence is atleast 2-3 second long thereby making it an extremely rare event to have false negatives continuously over this range. (iii) The human motion activities tend to occur in the lower frequencies below 30 Hz [62], thus making it easier to identify them. This facilitates identification of speech occurrences in the motion sensor data. *iSpyU* is only activated whenever speech segments are detected. Furthermore, once speech segments are identified, *iSpyU* applies a high pass filter at 80 Hz as described in Section 5.3 because eliminating these frequencies removes some low frequency noise and human motion activities without affecting speech recognition performance and intelligibility [78] (results discussed with Fig. 11(c)).

5.3 Preprocessing and Noise Subtraction

We perform the following preprocessing techniques on the sensor data for improving the robustness of ASR models.

5.3.1 Spectral Subtraction. We perform background noise elimination using spectral subtraction techniques popular in speech processing [10]. At a high-level, the average signal spectrum and the average noise spectrum are first estimated and then subtracted from each other, which is shown to eliminate additive stationary noise [69, 84]. Fig. 8 shows an example of the accelerometer signal before and after spectral subtraction. Evidently, the signal appears cleaner after spectral subtraction.

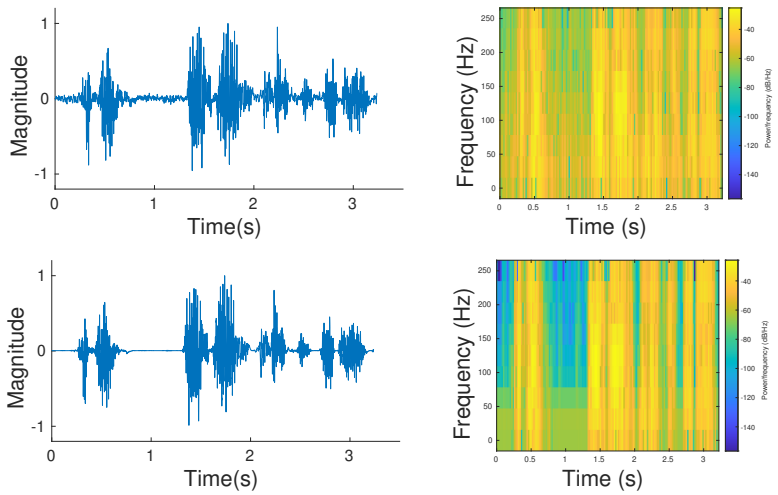


Fig. 8. (Top) Raw accelerometer signal. (Bottom) Spectral subtracted signal.

5.3.2 High Pass Filtering. The fundamental frequency of the voiced speech of a typical adult male and female will vary from 85-180 Hz, 165-255 Hz respectively . Thus, we apply a high pass filter at 80 Hz to eliminate the DC offsets and low-frequency noise without affecting ASR. This will also eliminate effects on the sensor data due to human motion – particularly if the phone is in the hand.

5.4 Feature Extraction

In the context of audio signals, extraction of rich spatio-temporal features before performing deep learning has shown to create robust models with smaller training data [19]. The popular Mel-Frequency Cepstral Coefficients (MFCC) [34] features are derived from *filter banks* [19]. More recently, the direct use of *filter banks* instead of MFCC is gaining in popularity because of a number of advantages. MFCC attempts to decorrelate features through a process of whitening so as to make them more suitable for conventional machine learning algorithms based on Gaussian Mixture Models (GMM), and Hidden Markov Models (HMM). Such a decorrelation step with Discrete Cosine Transforms (DCT) also results in loss of information. With the advent of deep learning, and the ability to handle correlated information, *filter banks* are gaining in popularity in most deep learning based ASR systems. We briefly discuss *filter banks* here as well as adapting the filters in the context of *iSpyU* for performing ASR with motion sensor data with a smaller frequency range.

In conventional ASR, the audio data is divided into frames of sizes $25ms$. However, in the context of motion sensor data, we use a frame size of $50ms$, so as to increase the resolution of the FFT stages to be discussed later. This is particularly important since the motion sensor data has a much smaller range of frequencies (0-250Hz) in comparison to speech data (0-16KHz). For each such $50ms$ frame, a hamming window w is then applied as depicted below.

$$w(n) = 0.54 - 0.46cos\frac{2\pi n}{N-1} \tag{1}$$

where $0 \leq n \leq N-1$, N is the size of the window. A N-point FFT (also called as Short Term Fourier Transform (STFT))is now performed on each frame. The power spectrum (periodogram) is computed next based on the

below equation.

$$P = \frac{|FFT(x_i)|^2}{N} \tag{2}$$

where x_i is the i^{th} frame of the signal.

Finally, the *filter banks* are computed from the periodograms. Here, the linear frequency scale is first converted to a logarithmic scale (*Mel* scale). The below equation captures this.

$$m = 2595 log_{10}(1 + \frac{f}{700}) \tag{3}$$

Using such a non-linear scale, several filters with varying center frequencies, and width are used to extract *fbank features* as shown in Fig. 9. The equations depicting these filters are enumerated below.

$$f = 700(10^{m/2575} - 1)$$

$$H_m(k) = \begin{cases} 0 \text{ for } k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} \text{ for } f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} \text{ for } f(m) \leq k \leq f(m+1) \\ 0 \text{ for } k > f(m+1) \end{cases} \tag{4}$$

Here, $H(k)$ denotes the response of the k^{th} filter bank. Each fbank filter integrates the energy within its respective frequency range as a single feature value. While conventional ASR typically uses 40 such *fbank features* to cover a larger frequency range, *iSpyU* adapts the sizes and widths of these filters appropriately to focus on the 250Hz of the motion sensor data. *iSpyU* uses 10 filters as shown in the figure. These *fbank features* form the input to the ASR models.

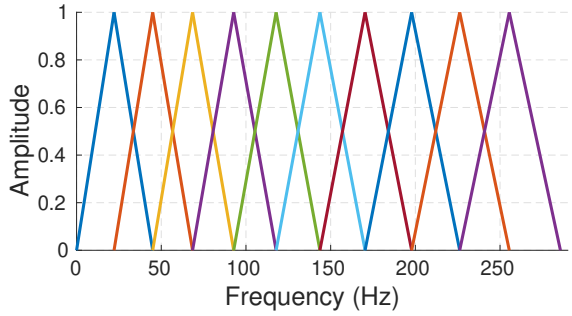


Fig. 9. Fbank filters extracting features from the motion sensor spectrum.

5.5 ASR Model

Fig. 10(a) shows the high-level overview of the ASR model designed with *attention mechanism* popular in computer vision and speech processing [11, 46, 59, 85]. The input to the model includes a sequence of *fbank features* extracted from the motion sensor data (which includes gyroscope and accelerometer data), denoted as: $\mathbf{x} = \{x_1, x_2, ..x_i..x_T\}$. Here, x_i denotes the *fbank features* extracted at the i^{th} time step where the fbank features from accelerometer and gyroscope are fused in concatenated form as a vector. The output is a sequence of words

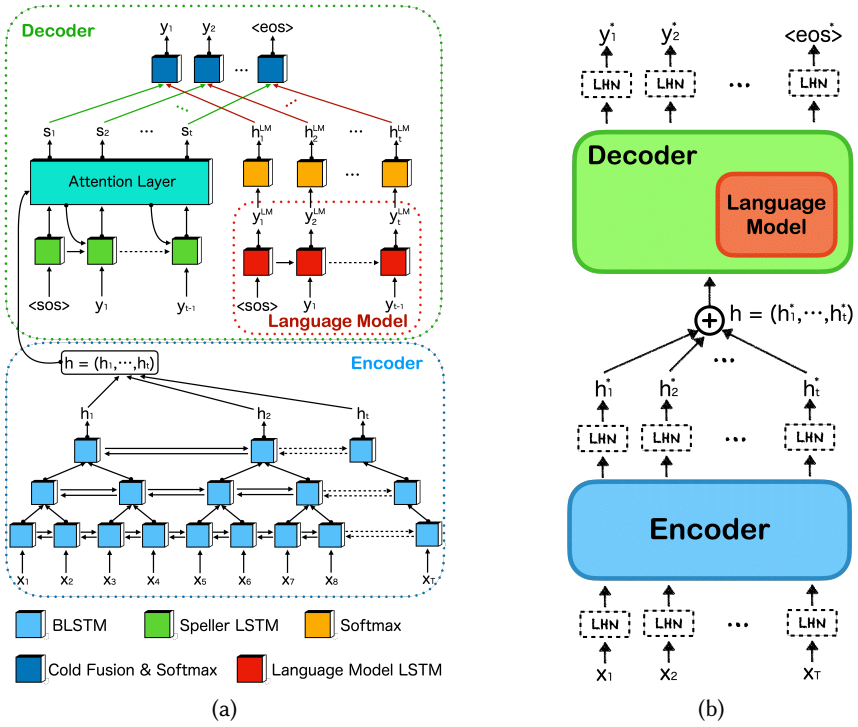


Fig. 10. (a) *iSpyU*'s ASR model. (b) LHN layers cause efficient domain adaptation

$\mathbf{y} = \{y_1, y_2, ...y_k\}$ with $k \leq T$. Given the time-step is very small to accommodate a full spoken word [35, 77], it is a common convention in ASR literature to note that the number of output words is less than the number of time-steps [11, 13]. While phonemes and graphemes are other possibilities for modeling the output unit instead of words, modeling directly based on words has attracted recent attention due to the simplified ASR pipeline by avoiding additional pronunciation lexicon. This aids in a faster decoding process with competitive performance [27, 37, 93]. The model computes a probability of i^{th} word as a function of the entire input and the previously decoded words in the sentence, denoted as $p(y_i|x, y_{\leq i})$. We now elaborate on the various components of this model.

5.5.1 Encoder. The encoder processes the *fbank features* using *Bidirectional Long Short-Term memory* (BLSTM) [70] layers to convert the input audio into compact feature representations that capture the rich spatiotemporal relationships in the motion sensor input. The pyramidal architecture with subsampling layers is included with the following benefits: (i) Avoids overfitting. (ii) Allows efficient training with fewer parameters. (iii) The fewer parameters via sub-sampling provide leverage for increasing the depth of the network. The deeper architecture allows the learning of complex spatiotemporal relationships. We have 3 layers in the model with a subsampling factor of two at each layer, thus reducing the size of the input by a factor of 8. The output of the i^{th} time step at the j^{th} BLSTM layer can be represented as:

$$h_i^j = BLSTM(h_{i-1}^j, [h_{2i-1}^{j-1}, h_{2i}^{j-1}]) \tag{5}$$

We now elaborate on the decoder that converts the encoder outputs from the final layer to a sequence of words.

5.5.2 Decoder. We first explain the action of the decoder without the language model, and fuse the language model into the decoder in the next subsection. The decoder uses an attention-based *Long Short-term Memory* (LSTM) network as depicted in Fig. 10(a). At each step, the decoder produces the conditional probability distribution of the output word dependent on all the previously decoded words. The conditional probability of i^{th} word, y_i depends on the decoder state s_i, decoder context, c_i, the entire encoder output $\mathbf{h}$, and all previously decoded words $y_{\leq i}$. Mathematically, this can be represented as :

$$\begin{aligned} c_i &= AttentionContext(s_i, \mathbf{h}) \\ s_i &= LSTM2(s_{i-1}, y_{i-1}, c_{i-1}) \\ p(y_i|\mathbf{x}, y_{\leq i}) &= WordDistribution(s_i, c_i) \end{aligned} \tag{6}$$

At each of the time steps, the context vector produced by the $AttentionContext$ function extracts the motion sensor vibration content from the encoder output needed for decoding the next word. The context vector has access to the entire encoder output, and thus has the ability to exploit rich relationships across time but ultimately narrows down the focus to a small part of the encoder output relevant for decoding the next word. The $AttentionContext$ function is further elaborated below:

$$\begin{aligned} e_{i,u} &=< \phi(s_i), \psi(h_u) > \\ \alpha_{i,u} &= \frac{exp(e_{i,u})}{\sum_{u'} exp(e_{i,u'})} \\ c_i &= \sum_u \alpha_{i,u} h_u \end{aligned} \tag{7}$$

h_u denotes the encoder output at the u^{th} time-step. $LSTM2$ denotes a two-layer LSTM network. ϕ and ψ denote multilayer perceptron (MLP) networks.

5.5.3 Language Model. *iSpyU* incorporates a *Recurrent Neural Network Language Model* (RNNLM) [29] into the ASR training framework as shown in the decoder part in Fig. 10(a). The RNNLM outputs the probability of the word y_i^{LM} given all previous words in the sentence denoted by - $p(y_i^{LM}|y_0^{LM}, y_1^{LM}, ...y_{i-1}^{LM})$. The RNN consists of an input layer and a hidden recurrent layer. The output layer computes softmax probabilities of the next word given all previously seen words in the sentence. The RNNLM is first trained using sentences in the LibriSpeech dataset. After this, during the training of the ASR model, the trained RNNLM is included so as to achieve robustness in prediction, through a process called *cold-fusion* [80].

$$\begin{aligned} h_i^{LM} &= softmax(y_i^{LM}) \\ g_i &= \sigma(W[s_i, h_i^{LM}] + b) \\ s_i^{CF} &= [s_i; g_i \circ h_i^{LM}] \\ p(y_i|\mathbf{x}, y_{\leq i}) &= softmax(s_i^{CF}) \end{aligned} \tag{8}$$

Here s_i^{CF} denotes the counterpart to s_i from Equation 6 with the *cold-fusion* update.

Loss Function: The model parameters are trained to maximize the log probability of correct sentences, based on the following loss function where $\tilde{y}_{i-1}$ represents the ground truth of previous words. This loss function jointly

$$\tilde{\theta} = \max_\theta \sum_i logP(y_i|\mathbf{x}, \tilde{y}_{<i}; \theta) \tag{9}$$

with the ASR model exploits advances in representation learning and weakly supervised learning, thus obviating the need to segment a sentence into individual words during training or inference. This dramatically decreases

the overhead of generating training labels. The loss function defined in Equation 9 processes a whole sentence for recognition, which is popular in many SOTA works in ASR on sentences with speech data [52]

Decoding and Inference: During inference, we maximize the probability of the most likely sequence of words $\hat{\mathbf{y}}$.

$$\hat{\mathbf{y}} = arg \max_{y} logP(\mathbf{y}|\mathbf{x}) \tag{10}$$

To counter the bias in the model for short sentences, the scores for a hypotheses $s(\mathbf{y}|\mathbf{x})$ is normalized by the length of each sentence $|y|_c$ as follows:

$$s(\mathbf{y}|\mathbf{x}) = \frac{logP(\mathbf{y}|x)}{|y|_c} \tag{11}$$

Beam search [82] is adopted where top-K (with K=5) sentences are explored at each step of a word decoding.

5.6 Domain Adaptation

The synthetic training data does not completely model the real-world motion sensor data because of residual errors in modeling related to sub-sampling and noise distribution estimations. Nevertheless, the ASR model trained on synthetic data provides a sufficiently reasonable base model (evaluated in Section 7) to bootstrap the process of training. This model is then *fine-tuned* with small-scale training data from real-world motion sensors to further improve the robustness of the model.

Fine-tuning the ASR model directly is not feasible because millions of parameters are combined into each layer, thus leading to convergence issues when fine-tuned with small-scale real-world sensor datasets. Therefore, *iSpyU* adds small layers of Linear Hidden Networks (LHN) at the input, encoder, and decoder levels of the original ASR model as depicted in Fig. 10(b). Such layers have fewer parameters and they can efficiently capture the difference in distribution between the synthetic and real-world motion sensor data. To be more specific, we introduced the LHN layers at three places: (i) Before the encoder (ii) Between the encoder and decoder (iii) At the end of the decoder before final inference. The intuition was to adapt the distributions before the encoder input, decoder input, and the final output layers to a distribution that each of these layers are expecting. However, we settled upon current design based on cross-validation approach by enabling/disabling LHN at the three places identified above. The LHN layers use a linear feed forward layer. The LHN layers are in the shape of a square matrix U with dimensions of 120×120 (before the encoder), 512×512 (between encoder/decoder), and 9950×9950 (at the end of the decoder respectively). This is a more effective strategy than retraining a few layers because each of the layers have large number of parameters entailing high training overhead [20, 25]. Depending on the *domain shift*, this strategy is capable of capturing accurate information for domain adaptation with minimal training.

6 EVALUATION METHOD

6.1 Implementation

iSpyU is implemented on a combination of desktop and smartphone devices. The ML model is implemented with PyTorch packages and the training is performed on a desktop with Intel i7-8700K CPU, 16GB RAM memory, and NVIDIA Quadro RTX 8000 GPU. We use Adam optimizer with a learning rate of 10^{-3} with decay starting from 10 epochs and the decay rate is 0.85. To avoid overfitting, we apply L2 regularizer with a parameter 10^{-6}, and dropouts with a parameter of 0.4. The model is first trained with synthetic motion sensor data generated from LibriSpeech [57] (discussed next) which took 55 hours on the GPU. The model is then finetuned using motion sensor data collected from smartphones which took 0.43 hours on the GPU. By setting the sensor delay as *SENSOR_DELAY_FASTEST* in the Android SensorManager API [4], we extract data at a sampling rate of 500Hz

without any special permission from the user. With the improvement in CPU and battery performance, higher sampling rates might be possible in the future, thus increasing the privacy threat [6].

6.2 Dataset

iSpyU builds on LibriSpeech dataset [57], consisting of 983.1 hours of speech data sampled at 16 kHz. This includes 292367 sentences over a dictionary of 9950 words as spoken by 1201 female and 1283 male speakers with 8-30 minutes per speaker. The sentences vary in length from 2-61 words. The dataset is derived from about 8000 audiobooks of the LibriVox [31] project. This includes a broad variety of topics in fiction, history, crime, adventure, politics, religion, etc. Thus, we expect that a model trained from this dataset will capture a variety of sensitive contexts in human communication.

6.3 Data for Training

Approximately 98% of the entire LibriSpeech data is converted into synthetic motion sensor data (as discussed in Sec. 5.1) to bootstrap the training process. This includes 2338 speakers, 1210 males, 1128 females, and a total of 281241 sentences. Given that LibriSpeech includes more sophisticated vocabulary than a typical everyday conversation, it is considered a popular benchmark for many ASR applications [58] including Baidu's DeepSpeech [1] and voice assistants such as Amazon Alexa [72].

6.4 Data for Domain Adaptation

About 1.5% of the LibriSpeech dataset is used for domain adaptation. This includes 104 speakers, 53 males, 51 females, and a total of 7681 sentences. We play the audio samples corresponding to this data on the smartphone and record the measurements from the motion sensors. This generates labeled training dataset with motion sensor recordings and their respective textual transcriptions. Evaluated in Fig. 12(a), increasing the size of data for domain adaptation beyond this point has diminishing returns.

6.5 Data for Testing

For testing, we use 16 new speakers, 8 males, 8 females, that spoke 498 new sentences of length 3-40 words on various topics including daily conversation, history, politics, sports, religion, hobbies, etc. The speakers are native with an average speaker rate of 152 words per minute. Our study protocol complies with the local IRB at our institute. The experiments are conducted under noisy conditions with a typical indoor noise level between 50-60 dB (shared office) both at the speaker's side and the receiver's side. We also make the following observations regarding the experimental setting: (i) Noise at the receiver end does not have any impact on performance because the main source of vibration in the motion sensors is the motherboard connecting the speakers and IMU. The air channel has negligible impact on the motion sensors, which is also consistent with the observation in prior-work [3, 6]. (ii) Modern conferencing tools such as Skype and Zoom already incorporate sophisticated noise suppression techniques [55, 86] to enhance the speech quality of the speaker. The motion sensor data was recorded during these Skype sessions for spying on speech content. The data is simultaneously collected on different phones by having all phones participate in a Skype conference call. Therefore, the data from all volunteers are collected on all phones. Accuracy across users, sentences, qualitative decoding, etc, are discussed here.

6.6 Metrics of Evaluation

To validate the accuracy of *iSpyU*, we use the standard *WER* metric [92] that is popular in ASR systems, as outlined below:

Table 3. Representative examples from *iSpyU*'s recognition of continuous speech.

#	**Ground Truth**	**Decoding by *iSpyU***
1	he ONLY got mild fever	he FINALLY got mild fever
2	very carefully my mom removed this powder placing it ALL TOGETHER in a dish WHERE she mixed it with a spoon	very carefully my mom removed this powder placing it *** ALTOGETHER in a dish SO she mixed it with a spoon
3	A KICK from the tall boy behind urged stephen to ask A difficult question	THE TOOK from the tall boy behind urged stephen to ask WHAT difficult question
4	well what can't be done by main focus in exam must BE DONE by circumvention	well what can't be done by main focus in exam must ** PRESENT by circumvention
5	beware of making that mistake	beware of making that mistake
6	I HAD THE FAITH in me *** THAT CLIMB MOUNTAINS	I WAS A FACE in me BUT MOVED NOT IN
7	IF YOU dressed in SILK and *** GOLD FROM TOP TO TOE YOU could not LOOK any NICER THAN IN YOUR little RED CAP	WAS HE dressed in SICK and BOY AND SAT FOR A PLACE AND could not HAVE any ***** MERCY BUT BEYOND little *** REDS
8	on FRIDAY party WILL be held all THE afternoon * AFTER LUNCH	on FRIDAY'S party WOULD be held all THAT afternoon I COULD BE
9	I have A conference tomorrow and I need all the documents ready	I have THE conference tomorrow *** I need *** the documents ready
10	He IS IN bad mood these days and DESPISES his country life	HIS ** ** bad mood these days and DESPISE his country life

$$WER = \frac{S + I + D}{N}$$
$$WAcc = 1 - WER \tag{12}$$

A *decoded* sentence by *iSpyU* is compared with the original *reference* sentence (ground truth). Here, *Substitution* (S) denotes the number of cases where a word in the *reference* sentence was replaced by another word in the *decoded* sentence. *Insertion* (I) denotes the number of new words inserted into the *decoded* sentence which do not appear in the *reference* sentence. *Deletion* (D) denotes the number of words in the *reference* sentence that do not appear in the *decoded* sentence. Finally, $N = S + C + D$, is the number of words in the *reference* sentence, where *Correct* (C) denotes the number of words that appear in both the *reference* and *decoded* sentences. We define the *word accuracy* (WAcc) of *iSpyU* to be $1 - WER$. Similar to WER, the character error rate (CER) is defined based on equation 12, but the insertions, substitutions, deletions, and correct words are computed at the character level instead of the word level. We define the *character accuracy* (CAcc) of *iSpyU* to be $1 - CER$.

7 PERFORMANCE RESULTS

We present a systematic evaluation. Various special cases are discussed under appropriate subsections. We compare three versions of *iSpyU*. (i) *iSpyU*: Uses a combination of large scale synthetic data and small scale real data for developing the ASR model from Sec. 5. (ii) *Top-5*: A version of *iSpyU*, where during each stage of the beam search decoding, top-5 most probable words are considered, and the decoding is counted in the C metric of equation 12 as long as the any of these top-5 words can be classified as C. (iii) *iSpyU+* : A version of *iSpyU* where we assume a smaller dictionary size of 2000 words at test time for decoding each sentence. Note that the ASR model is still trained with 9950 words since cutting down the dictionary size at training time dramatically cuts down the number of available sentences for training. Prior research has shown the ability to detect the topic of the conversation (sports, politics, religion, etc) by analyzing a sequence of sentences [28, 61]. Knowing the topic allows us to narrow down the search space of the dictionary leading to higher accuracies. While detecting the topic with a sequence of sentences recorded by motion sensors is outside the scope of this work, we evaluate the gain in accuracy if this was possible.

7.1 Qualitative Results

We begin by providing representative examples of sentences decoded by *iSpyU* as shown in Table 3. The ground truth and the inferred sentences are aligned [12] with each other so as to compute errors and mismatches. The capitalized words indicate a mismatch between the ground truth and the inference by *iSpyU*, whereas, "****" indicates a missing word during the alignment. Words that match correctly appear in lower cases. We note that the context of the message is clear for most of these sentences. In some cases, the mismatches are very close

(for example, Friday → Friday's, Red → Reds, All Together → Altogether, etc). Sentence 5 is an example of a perfect inference whereas sentences 6 and 7 are examples that are very erroneous. Nevertheless, we believe there is sufficient leakage of sensitive context from most sentences which poses significant privacy threat.

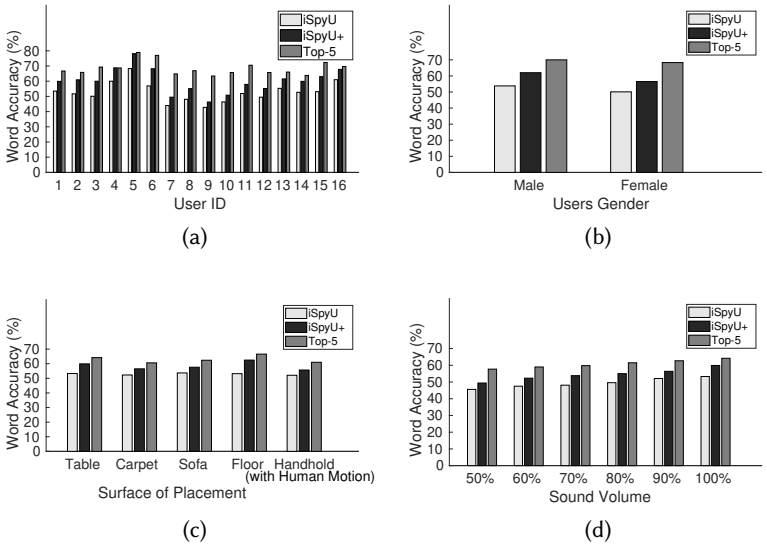


Fig. 11. Accuracy variation in *iSpyU* over (a) Users (b) Gender (c) Surface of phone placement (d) Sound Volume

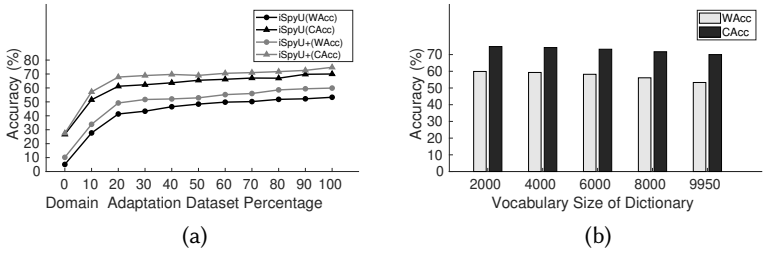


Fig. 12. Accuracy variation in *iSpyU* over (a) Size of domain adaptation dataset (b) Dictionary size

7.2 Accuracy over Users

Fig. 11(a) shows the *WAcc* for *iSpyU* as a function of different users. The average *WAcc* is 53.3%. Given that the model has been trained from a diverse distribution of users including thousands of males and females, the model is robust across a variety of users. The accuracy is also consistent across genders (Fig. 11(b)).

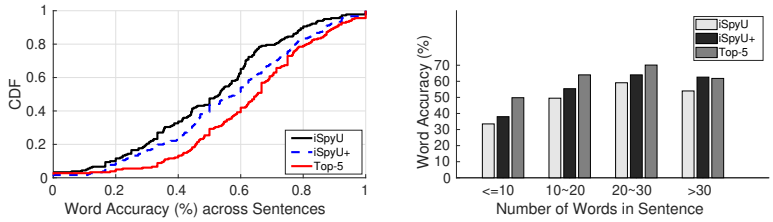
7.3 Expected Improvement based on Context

Fig. 11(a) depicts the accuracy of *iSpyU+*. With a smaller dictionary, *iSpyU+* can enhance the *WAcc* to 59.9%. Fig. 11(a) also depicts the top-5 accuracy (*Top-5*) which can be higher, 64.2%. This suggests scope for further

optimizations. For example, if prior information about the context of the sentence was available, *iSpyU* could refine the probabilities of the decoded words to extract more accurate decodings.

7.4 Accuracy over Sentences

Fig. 13(a) shows the distribution of accuracy across sentences and Fig. 13(b) shows the variation as a function of the length of the sentence. The accuracy does not degrade with the increasing length of the sentence. Longer sentences have slightly higher accuracy because the attention-based models are able to better exploit context within the same sentence. The normalization step introduced in Equation 11 also helps achieve consistent accuracy for longer sentences.



(a) (b)

Fig. 13. (a) Distribution of accuracy across sentences (b) Accuracy variation across number of words in sentences

7.5 Accuracy vs Surface of Phone Placement

We evaluate *iSpyU* on different surfaces, depicted in Fig. 11(c). When the phone is in the hand, we ask the user to walk and perform simple motion activities. Note that such human motion does not impact the results under handhold setting due to the use of the high pass filter (Sec. 5.3). We find that the accuracy on carpet/hand is slightly lower than floor/table mainly because of lightly damped vibrations from a soft surface. Because the speaker and motion sensors share the same motherboard, they have a strong channel between them, therefore the dampening effect due to external surfaces only induce minor variations. There is consistent leakage of information across all common surfaces.

7.6 Accuracy over Sound Volume

Fig. 11(d) depicts the accuracy of *iSpyU* as a function of sound volume. The SNR of motion sensor data captured at different volume levels on average are as follows: $50\% - 3.62dB$, $60\% - 4.33dB$, $70\% - 4.72dB$, $80\% - 5.52dB$, $90\% - 6.06dB$, $100\% - 6.92dB$. Because of decreasing SNR with decreasing volume, there is a graceful degradation in accuracy over decreasing volume. Evidently, the threat is maximum at full volume, typically used in video conferencing calls like Skype and Zoom for clear and comprehensible speech [87]. However, there is a non-trivial leakage of information across other volume levels.

7.7 Accuracy vs Size of Training Data

Fig. 12(a) depicts the accuracy as a function of the size of real motion sensor data used for domain adaptation. Evidently, the *WAcc* quickly jumps to 27.7% from 5.1% with only 10% of the fine-tuning dataset. Beyond that, the accuracy reaches close to 50.2% at 70% of the fine-tuning dataset and saturates thereafter. The *CAcc* follows a similar trend. This demonstrates the ability in *iSpyU* to enhance the accuracy of the model trained by synthetic datasets with small-scale real-world datasets.

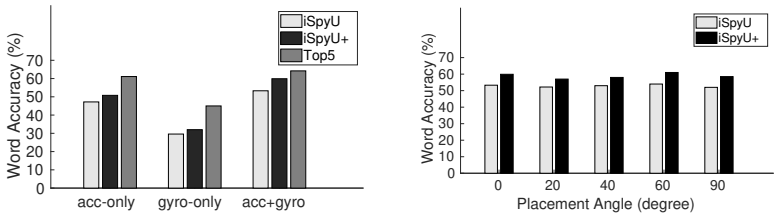


Fig. 14. (a) Individual accuracies of accelerometer and gyroscope in comparison to the accuracy with fusion of the two sensors (b) The smartphone placement angle doesn't impact the performance of *iSpyU* because the speaker and IMU are housed on the same motherboard hence and their relative orientation does not change with the phone orientation.

7.8 Accuracy vs Size of Dictionary

With access to prior information about the context (topic of conversation in politics, religion, etc), the accuracy can improve. Fig. 12(b) shows the accuracy over the size of the dictionary. The original *iSpyU* ASR model is unchanged (trained with 9950 words) but we choose smaller dictionary search-spaces (2000-8000) during test time that include all words in a given test sentence. Narrowing down the search space enhances the accuracy.

7.9 Accelerometer vs Gyroscope

Fig. 14a shows the accuracy when only accelerometer (*acc-only*) or only gyroscope (*gyro-only*) is used for the attack. The accuracy when both sensors are used is also shown (*acc + gyro*). The individual sensors can still sustain reasonable accuracy levels. The accelerometer has a slightly higher accuracy (*WAcc*, 47.2%) than the gyroscope (*WAcc*, 29.6%) because accelerometer signals have a higher SNR. Nevertheless, the gyroscope signal contains non-trivial information associated with the vibration leakage, which when combined with the accelerometer data results in an overall higher accuracy (*WAcc* - 53.3%).

7.10 Impact of Placement Angle

Fig. 14b depicts the performance of *iSpyU* as a function of the smartphone placement angle with respect to the horizontal surface. Evidently, the orientation of the smartphone does not affect the accuracy. To further validate this claim, Fig. 11(c) (last bar) evaluates a situation where the user holds the phone in the hand in an arbitrary orientation and walking while participating in a conference call. The orientation of the phone changes naturally during this experiment. Even under these conditions, the accuracy of *iSpyU* is consistent. This is because the IMU and the speaker are housed on the same motherboard, and their relative orientation is the same even when the smartphone is leaning with arbitrary orientation, thus having negligible impact on the performance.

7.11 Accuracy vs Sampling Rates

Similar to 500 Hz, a separate training and domain adaptation was done at 200 Hz to create an ASR model at 200 Hz. Fig. 15(a) depicts the comparison between the two. As expected, the accuracy is higher at 500 Hz. Given the trend in increased CPU speed and battery performance of mobile OS [49, 65], a higher accuracy of the attack is expected at higher sampling rates in the future. We believe this is certainly worthy of investigation from a security perspective. We also note that at a lower sampling rate of 200 Hz, the attack is much weaker. While switching to a lower sampling rate might be one possible defense, there are certain applications that benefit from higher sampling rates. The tradeoffs are discussed in Sec. 8.

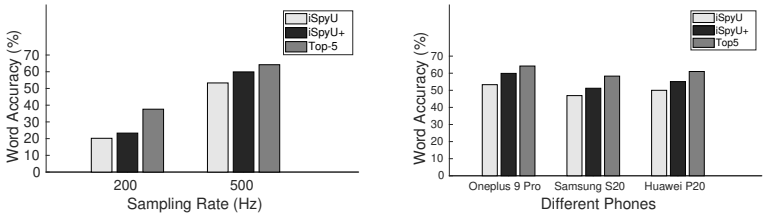


Fig. 15. (a) As expected, the accuracy increases with sampling rate. (b) The attack is feasible on multiple smartphones.

7.12 Accuracy over Phone Models

Towards evaluating the threat posed by *iSpyU* on different phone models, we compare the performance difference across different phones. The results are depicted in Fig. 15(b). The variation across phones are indicative of the differences in SNR between the speaker and the motion sensors of the phone models. Nevertheless, there is non-trivial leakage of information across phone models, which suggests significant risk to a wider community of users.

7.13 Multiphone Domain Adaptation

Figure 16 depicts the performance when the model developed for one phone is domain adapted for inference on a different phone. Evidently, this process improves the overall accuracy on average by 3.2% (overall accuracy

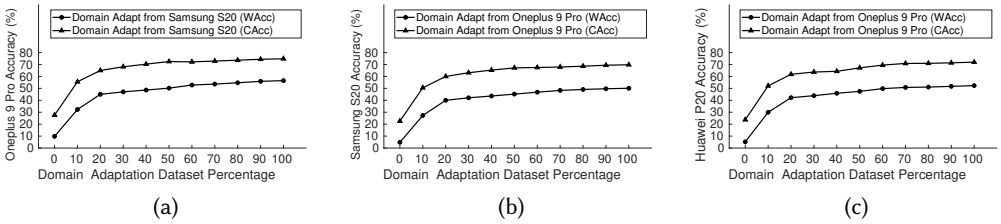


Fig. 16. Model developed for one phone is domain adapted for inference on a different phone: This process only requires 5 hours of training data to achieve an accuracy similar to domain adapting from a model trained from synthetic IMU data. It decreases the overhead of domain adaptation by 50%.

$\approx$ 56.5%) in comparison with domain adaptation over model trained entirely on synthetic data. Also, domain adapting the model from one phone for inference on a different phone requires only 5 hours of data to achieve a similar level of performance as domain adapting from a model trained on synthetic data. This decreases the overhead of collecting domain adaptation data by 50%. We believe this decreases the barrier for attack on different phones since only 5 hours of training data is required on each phone which can be collected relatively easily. In contrast, collecting 960 hours of data on multiple different phone models can make the bar for attacking higher since newer smartphones are replacing older ones in hundreds of millions each year [73, 74].

7.14 The Role of the Language Model

Described in Section 5.5, *iSpyU* incorporates a RNN-based language model during the training process of the ASR model via the principle of *cold-fusion*. The goal is to enhance the accuracy of the inference by exploiting the

Table 4. The Role of the Language Model

	Audio	**IMU (*iSpyU*)**
w/o lang. model	83.2%	51.1%
w lang. model	84.6%	53.3%

context of the previously occurring words in the sentence. Table 4 analyses the performance of the language model and the acoustic model separately and when acting in conjunction. The language model accuracy is 55.98 ppl (ppl = perplexity as defined in [60] is a measure of the predictive power of a language model), whereas the accuracy of the acoustic model alone without the language model is 51.2%. The accuracy when both models are used in conjunction is 53.3%. In contrast, for performing ASR with audio, the accuracy without the language model is 83.2%, and the accuracy with the language model is 84.6%. We note that the accuracy boost offered by the language model is slightly higher for *iSpyU* than normal audio, because a typical audio already captures a wide range of the acoustic spectrum that contains sufficient information for performing ASR.

7.15 The Role of Synthetic Data and Domain Adaptation

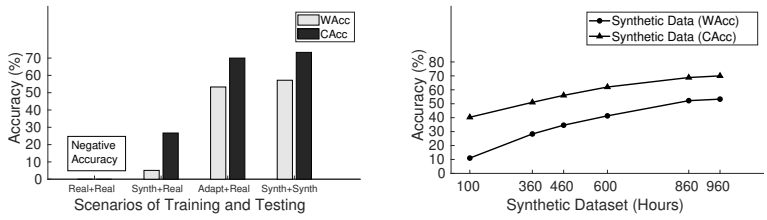


Fig. 17. (a) Synthetic data substantially decreases training overhead (b) The accuracy as a function of the size of synthetic training data

Fig. 17a depicts the accuracy under the following four cases of training and testing. (i) *Real + Real*: We begin with a model trained entirely using small-scale real data, and tested with real data. Such a model outputs random words for any input. This results in erroneous insertions and deletions that make the WER in equation 12 go above 100% thus resulting in a negative accuracy ($WAcc$). Small-scale training data is not sufficient to create a stable model that generalizes well. (ii) *Synth + Real:* This is the other extreme where a model trained entirely using synthetic training data is tested with real data. This provides a stable model with low but non-trivial accuracy. The $WAcc$ is 5.1%, and the $CAcc$ is higher at 26.7%. Given 9950 words in the dictionary, this clearly indicates a performance that is significantly better than random guessing. The synthetic data generates a model that is stable enough to bootstrap the training process for domain adaptation. (iii) *Adapt + Real:* This denotes a model trained with synthetic data and fine-tuned with small-scale real data, and tested with real data. The model achieves a dramatically higher accuracy ($WAcc$ = 53.3%, $CAcc$ = 70.0%) than the two extremes in previous settings. (iv) *Synth + Synth:* Finally, this denotes the accuracy of a model trained and tested with synthetic data ($WAcc$ = 58.2%, $CAcc$ = 73.3%). This is the upper bound of achievable accuracy where the distributions of training and test data match perfectly. Evidently, with only 1.5% of real data in comparison with original LibriSpeech data, *iSpyU* in case (iii) above, achieves an accuracy close to this upper bound. The character level accuracies $CAcc$ are higher than the corresponding word-level accuracies because some of the incorrect words could be close matches to the correct words. Finally, we also show the accuracy in *iSpyU* as a function of the size of synthetic training data in

Fig. 17b. Evidently, close to 460 hours of training data is still required to achieve atleast 30% of accuracy, and appears to taper off with an accuracy of 53.3% when the size of training data is 960 hours. These results indicate the critical role of synthetic training data and the effectiveness of domain adaptation with small-scale training data in achieving a good tradeoff between training overhead and performance.

Table 5. Various combinations of data for training, domain adaptation, and testing. Negative accuracies indicate the model did not converge with $WER > 100\%$

Training Data	**Domain Adapt Data**	**Accuracy (%) Audio Test Data**	**Accuracy (%) IMU Test Data**
Synth IMU (960 hr)	No Domain Adapt	Random (−1.9%)	5.1%
Synth IMU (960 hr)	Real IMU (10 hr)	Random (−3.2%)	53.3%
Audio (960 hr)	No Domain Adapt	85.8%	Random (−2.2%)
No Pretraining	Real IMU Data (10 hr)	Random (−24.6%)	Random (−19.3%)
Audio (960 hr)	Real IMU Data (10 hr)	Random (−9.5%)	1.8%

7.16 Performance Gap between Audio Based and Motion Sensor Based ASR

Table 5 depicts the ASR performance on audio and motion sensor data for various combinations of training and domain adaptation data. The performance of the ASR model trained and tested on audio data is 85.8%, which is the upper bound of achievable performance in *iSpyU* since the synthetic IMU data for training is derived from the same audio dataset (LibriSpeech). In contrast, *iSpyU* achieves a best performance of 53.3% when trained with synthetic IMU data and domain adapted with small quantities of real IMU data. Even though the performance is lower than pure audio based ASR we believe this is non-trivial since the motion sensor data has a much lower sampling rate than audio. Moreover, we believe the accuracy is still worthy of concern when privacy is of interest since it can still leak sensitive context of the communication (Examples in Table 3). Many of the other combinations of training and testing resulted in random accuracy (negative) as indicated in the table because the model did not converge. The only other combinations that yielded a somewhat non-random accuracy (considering a dictionary size of 10000 words where a random accuracy is 0.01%) include: (i) Training with Synthetic data and no domain adaptation yields an accuracy of 5.1%. This provides a basic model to bootstrap the process of domain adaptation. (ii) Training with audio data and domain adaptation with small amount of real data yields an accuracy of 1.8%. Under the constraints of performing ASR with limited quantities of real IMU data, we believe the overall results validate the critical importance of both synthetic IMU data and the domain adaptation with real IMU data as that is the only condition under which the accuracy is reasonable enough to decode meaningful sentences.

7.17 Comparison with Prior Work

Table 6 depicts the comparison of *iSpyU* with prior work. Given that *iSpyU* is designed to perform ASR on large vocabulary whereas many of the prior work perform recognition on a smaller vocabulary (1-100), and at a lower sampling rate, we downscale *iSpyU* to similar sampling rates (120 Hz) and vocabulary sizes for a fair comparison with such prior works. The new experimental results are summarized in Table 6. While *iSpyU* is still comparable to prior work in their respective settings, we believe the ability to perform ASR on large dictionaries is a new contribution in *iSpyU* that has not been explored before. Equipped with such an ability, an adversary can potentially launch large scale attacks. For example, insurance companies could infer health status of several of their customers in automated ways without manually processing each customers data. Advertising companies can similarly learn customer interests in automated ways and launch targeted ads on a massive scale. Therefore, our hope is to shed light on such vulnerabilities via ASR as a part of this work. Finally we note that reconstruction of speech like AccelEve and AccEar [26] needs manual recognition of reconstructed audio as admitted by authors

Table 6. Experimental Comparison with Prior Work. The star * marked results of *iSpyU* were conducted at a sampling rate of 120 Hz for a fair comparison with SpearPhone

System	Digits (10 Classes)	Keywords (58 Classes)	Digits + Alphabets (36 classes)	Automatic Speech Recognition (ASR) (9950 classes)	Speech Reconstruction for Manual Recognition
SpearPhone [3]	75.8%	69.5%	-	-	-
Gyrophone [47]	28.7%	-	-	-	-
AccelEve [6]	76.3%	-	61.7%	-	Yes
AccEar [26]	-	-	-	-	Yes
iSpyU (This paper)	80.1%*	71.6%*	93.2%	53.3%	-

of these papers. Based on experiments, performing ASR on reconstructed speech resulted in a poor performance for the task of ASR. This is because of domain shift between typical audio and the speech extracted using deep learning on motion sensor data, which still exhibit differences in distributions with some residual errors in the process of speech recovery with a low sampling rate motion sensor data. A similar observation regarding the general problem of domain shift in speech recognition been made in [44]. Towards handling such challenges, *iSpyU* designs an end-to-end pipeline by solving challenges of limited training data using synthetic data creation, signal processing, domain adaptation, etc.

7.18 Comparison between Attention-based BiLSTM and Transformer

Fig. 18 depicts the performance between attention-based BiLSTM as currently implemented in *iSpyU*, and an upgraded version of *iSpyU* that uses state of the art Transformers (*iSpyU-trans*) for performing ASR. The

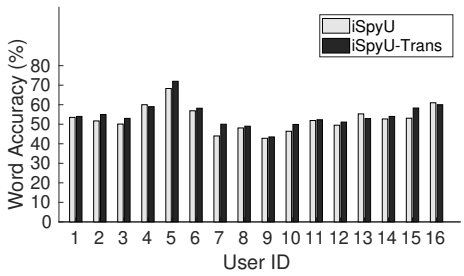


Fig. 18. Comparison between attention-based BiLSTM and transformer

architecture of the transformer model is depicted in Fig. 19. On average, transformers enhance the performance in *iSpyU* by 1.6% which is consistent with performance improvement achieved for speech recognition [79].

7.19 Resource Consumption - Power Consumption, CPU, and Network

The ML model is resource intensive and might raise the suspicion about the attack. Therefore, the adversary can choose to offload the relatively low bandwidth sensor data to perform the analysis offline[1]. This leaves the basic attack footprint low because the adversary only needs to collect and offload the sensor data. Given this is a side-channel attack and not a usability application like a voice-assistant, we believe there is no requirement of real-time latency. Therefore, the adversary can offload the data at a time of his convenience (for example they could camouflage with legitimate app data). Accordingly, we profile the *iSpyU* app as shown in Fig. 20 on

[1]Offline execution of the ML model takes 5.72s seconds for 1 minute of speech on average on the desktop configuration defined in Section 6

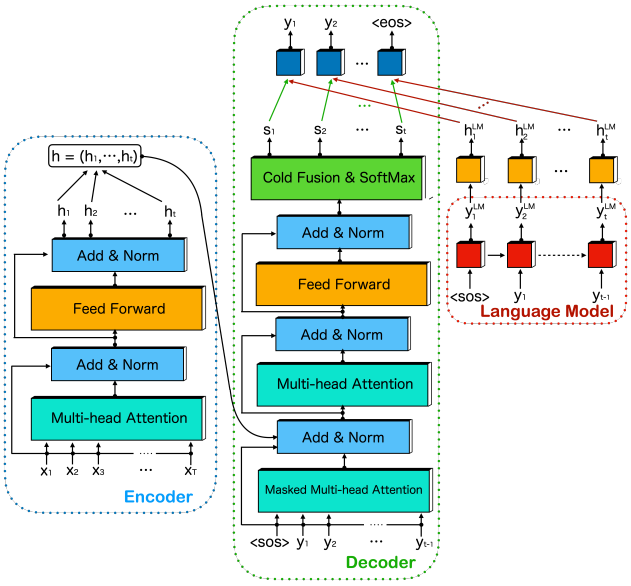
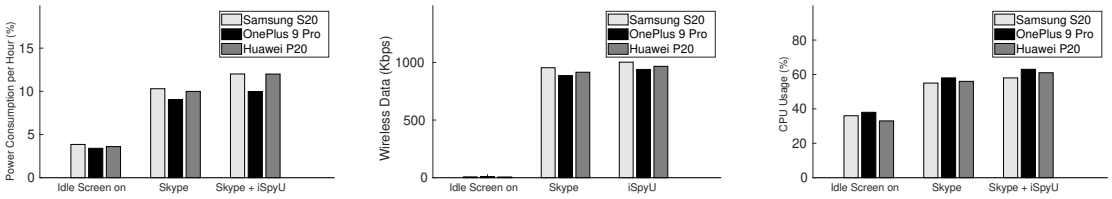


Fig. 19. *iSpyU*-trans is an upgraded version of *iSpyU* that uses the Transformer based architecture for performing ASR. The notations of variables in the model is same as the definition in Section 5.5. For domain-adaptation, we introduce LHN layers before encoder, between encoder and decoder, and after decoder in a similar manner to Fig. 10(b).



(a) (b) (c)

Fig. 20. (a) Power Consumption (b) Network Usage (c) CPU Activity

various smartphones. We first characterize the power consumption during collecting and offloading of sensor data. For profiling the power on a smartphone, we use Batterystats and Battery Historian [8] tools. We compare the difference in power between three states: (i) The device is idle with screen on. (ii) The device is running a Skype video conferencing app. (iii) The device is running the Skype app with *iSpyU* in the background for spying on the Skype call by collecting and offloading sensor data to the server. The idle display-screen on discharge rate is 3.63% per hour while the discharge rates for various modes are shown in Fig. 20(a). Evidently, *iSpyU* adds little to the power consumption when compared with an already running Skype app being used for the conferencing call while the attack is happening. Similarly, Fig. 20(b) shows the network activity with WiFi where *iSpyU* adds negligible traffic to an already running video/audio conferencing call. Finally, Fig. 20(c) shows the CPU activity, where *iSpyU*'s share is negligible. Therefore, we believe that *iSpyU*'s resource consumption maintains a low profile.

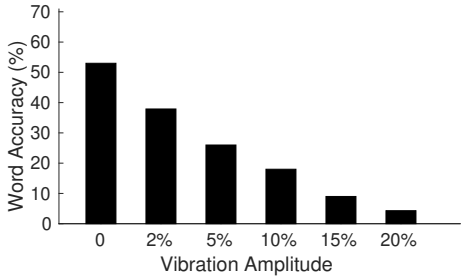


Fig. 21. Smartphone vibrator can induce noise for defending and mitigate the attack threat from *iSpyU*

7.20 Defense Strategies

A potential defense strategy is to exploit the vibrator in the smartphone to add small amounts of noise to confuse the ASR model. Prior works have shown that a vibration motor of a smartphone can produce frequencies upto 250 Hz [68]. This can potentially jam most of the frequencies in the motion sensor data. Fig. 21 provides a performance evaluation regarding the same. Even with lower amplitudes of vibrations, we can observe that the word accuracy is sufficiently degraded to mitigate the attack from *iSpyU*. We discuss these results and other potential strategies for defense in Section 8.

8 DISCUSSION AND FUTURE WORK

8.1 Summary of Results

(i) Visual inspection of qualitative results (samples in Table 3) suggest leakage of sensitive context of communication. (ii) The accuracy varies between 53.3% − 59.9% at the word level and 70.0 − 74.8% at the character level. (iii) The accuracy is consistent across users and genders. (iv) The attack is feasible on commonly used surfaces of phone placement: handheld, table, carpet, sofa, floor, etc. Because the speaker and IMU share the same motherboard, they have a strong channel between them. Thus any dampening due to external surfaces is negligible. (v) In the handheld setting, the accuracy is immune to human motion activities because *iSpyU* uses a high pass filter (Section 5.3) with the motion sensor data. (vi) The accuracy with fusion of accelerometer and gyroscope is better, even though each sensor in isolation provides non-trivial leakage of information. (vii) *iSpyU* has been evaluated on multiple smartphones: Samsung S20, OnePlus 9 Pro, Huawei P20, with consistent leakage across all platforms.

8.2 Implications on Privacy Leakage

iSpyU achieves an accuracy of 53.3 − 59.9% on detecting words in continuous speech over a dictionary of ≈ 10000 words. While the accuracy can be considered low for applications like voice assistants (Amazon Alexa, Siri, etc.), we believe these levels of accuracy are of critical concern when privacy is of interest. For example, even experienced lip readers from the deaf community can only detect about 30 − 40% of words correctly. However they can still sustain a basic level of communication by inferring the incorrect words from context [39, 40, 42, 43]. Therefore, even if the sentences are not fully decoded correctly, we believe the attacker can gain sensitive information such as location (sentence 9 in Table 3), health status (sentence 1), emotional state (sentence 10), political inclination, etc., of the victim with accuracy levels supported by *iSpyU*.

8.3 Scope for Enhancing the Attack Accuracy

While the ML models in *iSpyU* consider sentences one at a time, recent NLP research suggests opportunities in improving the accuracy by looking at multiple sentences together instead of one sentence at a time by incorporating contextual information across sentences, speaker, gender, etc [32]. *iSpyU* plans to incorporate such optimizations in future.

8.4 Defense Strategies

The OS can impose stricter access control to the motion sensors or alert the user when an application is requesting access to motion sensors. However this can also affect usability with applications like secure NFC communications, touch location sensing for UI, etc, that rely on high-frequency vibrations recorded by motion sensors [41, 67, 90]. An alternative to restricting free access to the motion sensor data could be the following. The vibration sensor can be used to produce noisy vibrations [67] such that the accuracy of the ASR model in *iSpyU* is reduced. We evaluate this strategy in Fig. 21 which indicates that even small amounts of vibrations can effectively mitigate the threat posed by *iSpyU*. Another possibility for defense would be to use speaker isolation pads between the loudspeaker and the motion sensors [3, 75, 76]. We believe the tradeoffs between usability and security need to be carefully considered while designing a defense.

8.5 Unsupervised Domain Adaptation

iSpyU only needs 1.5% of labelled real training data. We believe this is not a big overhead in the context of a security application, particularly because there is no need for any training data from the victim's phone. However, we will explore unsupervised domain adaptation to customize a pretrained model without requiring any labelled training data. This will make the attack easier as well as rapidly extensible to multiple languages. Adversarial domain adaptation [83] is of interest. Here, an unsupervised game theoretic strategy is used to transform the distribution of the feature representations from one domain into the distribution of the source domain where the model was trained. If successful, the model trained on the source domain (synthetic data) is directly useful for performing inferences on the target domain (real data). Similarly, other architectures for learning such feature transformations have been proposed [81] which are relevant for future investigation.

9 CONCLUSION

This paper shows the feasibility of attacking motion sensors for eavesdropping speech content of a phone speaker. *iSpyU* incorporates a fusion of techniques from automatic speech recognition, domain adaptation, synthetic training data generation, and signal processing to deal with challenges of low-training datasets, low-sampling rate, and noisy data to achieve an ASR accuracy of 53.3 – 59.9% over a dictionary of 2000-9950 words. The evaluation results demonstrate robustness across different users, surfaces, and gender. The raw decodings reveal information about the context. We believe this is of critical concern when privacy is of interest. In addition to an application in security, the proposed techniques in *iSpyU* will be useful for performing motion sensor based ASR in smart-earphones for futuristic applications.

ACKNOWLEDGMENTS

We sincerely thank the editors and reviewers for their comments and feedback. This research was partially supported by NSF grants: CNS-2008384.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International*

conference on machine learning. PMLR, 173–182.
[2] S Abhishek Anand and Nitesh Saxena. 2018. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1000–1017.
[3] S Abhishek Anand, Chen Wang, Jian Liu, Nitesh Saxena, and Yingying Chen. 2019. Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers. *arXiv preprint arXiv:1907.05972* (2019).
[4] Android Sensors 2022. Sensors Overview. https://developer.android.com/guide/topics/sensors/sensors_overview.
[5] Safaa Azzakhnini and Ralf C Staudemeyer. 2020. Extracting speech from motion-sensitive sensors. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 145–160.
[6] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based practical smartphone eavesdropping with built-in accelerometer. In *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*. 23–26.
[7] Ronald J Baken and Robert F Orlikoff. 2000. *Clinical measurement of speech and voice*. Cengage Learning.
[8] Batterystats 2022. Profile battery usage with Batterystats and Battery Historian. https://developer.android.com/topic/performance/power/setup-battery-historian.
[9] Sebastian Biedermann, Stefan Katzenbeisser, and Jakub Szefer. 2015. Hard drive side-channel attacks using smartphone magnetic field sensors. In *International Conference on Financial Cryptography and Data Security*. Springer, 489–496.
[10] Steven Boll. 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing* 27, 2 (1979), 113–120.
[11] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4960–4964.
[12] Panagiotis Charalampopoulos, Tomasz Kociumaka, and Shay Mozes. 2020. Dynamic string alignment. In *31st Annual Symposium on Combinatorial Pattern Matching (CPM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
[13] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4774–4778.
[14] Coriolis Force 2022. Coriolis force. https://en.wikipedia.org/wiki/Coriolis_force.
[15] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Fredo Durand, and William T Freeman. 2014. The visual microphone: Passive recovery of sound from video. (2014).
[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[18] Habiba Farrukh, Tinghan Yang, Hanwen Xu, Yuxuan Yin, He Wang, and Z Berkay Celik. 2021. S3: Side-Channel Attack on Stylus Pencil through Sensors. *arXiv preprint arXiv:2103.05840* (2021).
[19] Haytham M. Fayek. 2016. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html
[20] Roberto Gemello, Franco Mana, Stefano Scanzio, Pietro Laface, and Renato De Mori. 2007. Linear hidden transformations for adaptation of hybrid ANN/HMM models. *Speech Communication* 49, 10-11 (2007), 827–835.
[21] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.
[22] Gyroscope 2010. New Gyroscope Design Will Help Autonomous Cars and Robots Map the World. https://tinyurl.com/4s464sy8.
[23] Jun Han, Albert Jin Chung, and Patrick Tague. 2017. Pitchln: eavesdropping via intelligible speech reconstruction using non-acoustic sensor fusion. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 181–192.
[24] Xiaobing Han, Yanfei Zhong, Liqin Cao, and Liangpei Zhang. 2017. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing* 9, 8 (2017), 848.
[25] Michael Hentschel, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2018. Feature-Based Learning Hidden Unit Contributions for Domain Adaptation of RNN-LMs. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 1692–1696.
[26] Pengfei Hu, Hui Zhuang, Panneer Selvam Santhalingam, Riccardo Spolaor, Parth Pathak, Guoming Zhang, and Xiuzhen Cheng. 2022. AccEar: Accelerometer Acoustic Eavesdropping with Unconstrained Vocabulary. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1530–1530.
[27] Hirofumi Inaguma, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2018. Improving OOV detection and resolution with external language models in acoustic-to-word ASR. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 212–218.

[28] Hamed Jelodar, Yongli Wang, Rita Orji, and Shucheng Huang. 2020. Deep sentiment classification and topic discovery on novel coronavirus or covid-19 online discussions: Nlp using lstm recurrent neural network approach. *IEEE Journal of Biomedical and Health Informatics* 24, 10 (2020), 2733–2742.
[29] Shihao Ji, SVN Vishwanathan, Nadathur Satish, Michael J Anderson, and Pradeep Dubey. 2015. Blackout: Speeding up recurrent neural network language models with very large vocabularies. *arXiv preprint arXiv:1511.06909* (2015).
[30] Xiaoyu Ji, Yushi Cheng, Wenyuan Xu, Yuehan Chi, Hao Pan, Zhuangdi Zhu, Chuang-Wen You, Yi-Chao Chen, and Lili Qiu. 2021. No Seeing is Also Believing: Electromagnetic-emission-based Application Guessing Attacks via Smartphones. *IEEE Transactions on Mobile Computing* (2021).
[31] Jodi Kearns. 2014. LibriVox: Free public domain audiobooks. *Reference Reviews* (2014).
[32] Suyoun Kim, Siddharth Dalmia, and Florian Metze. 2018. Situation informed end-to-end asr for chime-5 challenge. In *CHiME5 workshop*.
[33] Wooil Kim and John HL Hansen. 2010. Missing-feature reconstruction by leveraging temporal spectral correlation for robust speech recognition in background noise conditions. *IEEE transactions on audio, speech, and language processing* 18, 8 (2010), 2111–2120.
[34] Tomi Kinnunen, Evgenia Chernenko, Marko Tuononen, Pasi Fränti, and Haizhou Li. 2007. Voice activity detection using MFCC features and support vector machine. In *Int. Conf. on Speech and Computer (SPECOM07), Moscow, Russia*, Vol. 2. 556–561.
[35] Ettien Koffi. 2021. A Comprehenisve Review of the Acoustic Correlate of Duration and Its Linguistic Implications. *Linguistic Portfolios* 10, 1 (2021), 2.
[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).
[37] Jinyu Li, Guoli Ye, Amit Das, Rui Zhao, and Yifan Gong. 2018. Advancing acoustic-to-word CTC model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5794–5798.
[38] Longfei Li, Yong Zhao, Dongmei Jiang, Yanning Zhang, Fengna Wang, Isabel Gonzalez, Enescu Valentin, and Hichem Sahli. 2013. Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition. In *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 312–317.
[39] Lip Read 2015. This Is What It Really Feels Like To Lip Read. https://www.bustle.com/articles/131261-how-accurate-is-lip-reading-this-is-how-it-feels-to-depend-on-it-every-day.
[40] Lip Read Learning 2020. How To Learn To Lip Read. https://www.connecthearing.com/blog/hearing-loss/lip-reading/.
[41] Jian Liu, Yingying Chen, Marco Gruteser, and Yan Wang. 2017. Vibsense: Sensing touches on ubiquitous surfaces through vibration. In *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.
[42] BjÖRn Lyxell and Jerker Rönnberg. 1987. Guessing and speechreading. *British Journal of Audiology* 21, 1 (1987), 13–20.
[43] BjÖRn Lyxell and Jerker Rönnberg. 1989. Information-processing skill and speech-reading. *British Journal of Audiology* 23, 4 (1989), 339–347.
[44] Akhil Mathur, Anton Isopoussu, Fahim Kawsar, Nadia Berthouze, and Nicholas D Lane. 2019. Mic2mic: using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems. In *Proceedings of the 18th international conference on information processing in sensor networks*. 169–180.
[45] MEMS Accelerometers 2017. MEMS Accelerometers | Silicon Sensing. https://www.siliconsensing.com/technology/mems-accelerometers/.
[46] Haoran Miao, Gaofeng Cheng, Pengyuan Zhang, and Yonghong Yan. 2020. Online hybrid CTC/attention end-to-end automatic speech recognition architecture. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 1452–1465.
[47] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 1053–1067.
[48] Mobile OS 2022. Mobile Operating System Market Share. https://gs.statcounter.com/os-market-share/mobile/.
[49] Mobile Processor 2020. Mobile Processor Crosses 3 GHz CPU Clock Speed Mark. https://tinyurl.com/2p8ya97w.
[50] Moore's Law 2020. Does Moore's Law still apply to smartphones in 2020? https://tinyurl.com/2p8su4cb.
[51] N Mostofi, M El-Habiby, and N El-Sheimy. 2014. Indoor localization and mapping using camera and inertial measurement unit (IMU). In *Proceedings of IEEE/ION PLANS 2014*. 1329–1335.
[52] Tomohiro Nakatani. 2019. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Proc. Interspeech*.
[53] Ben Nassi et al. 2020. Lamphone: Real-Time Passive Sound Recovery from Light Bulb Vibrations. Cryptology ePrint Archive.
[54] Wajahat Nawaz, Sagheer Ahmed, Ali Tahir, and Hassan Aqeel Khan. 2018. Classification of breast cancer histology images using ALEXNET. In *International conference image analysis and recognition*. Springer, 869–876.
[55] Noise Cancellation 2021. Microsoft adds AI-enabled noise cancellation feature to Skype: Here's how you can enable it. https://indianexpress.com/article/technology/social/skype-ai-enabled-noise-cancellation-feature-how-to-enable-7230339/.
[56] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: password inference using accelerometers on smartphones. In *proceedings of the twelfth workshop on mobile computing systems & applications*. 1–6.

[57] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.
[58] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).
[59] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International conference on machine learning*. PMLR, 4055–4064.
[60] Perplexisty 2020. Perplexity in Language Models. https://towardsdatascience.com/perplexity-in-language-models-87a196019a94.
[61] Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream nlp applications. *arXiv preprint arXiv:1710.06632* (2017).
[62] Xin Qi, Matthew Keally, Gang Zhou, Yantao Li, and Zhen Ren. 2013. AdaSense: Adapting sampling rates for activity recognition in body sensor networks. In *2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 163–172.
[63] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 1133–1136.
[64] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.
[65] Ranking 2021. Best Mobile Processor Ranking List 2021. https://www.techcenturion.com/smartphone-processors-ranking.
[66] Remote Work 2010. After embracing remote work in 2020, companies face conflicts making it permanent. https://tinyurl.com/57yavem8.
[67] Nirupam Roy and Romit Roy Choudhury. 2016. Ripple {II}: Faster communication through physical vibration. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 671–684.
[68] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. 2015. Ripple: Communicating through physical vibration. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 265–278.
[69] Sriram Sami, Yimin Dai, Sean Rui Xiang Tan, Nirupam Roy, and Jun Han. 2020. Spying with your robot vacuum cleaner: eavesdropping via lidar sensors. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 354–367.
[70] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.
[71] Sensors 2014. Sensors are Fundamental to Industrial IoT. https://tinyurl.com/54pj3268.
[72] Ilya Sklyar et al. 2021. Streaming multi-speaker asr with rnn-t. In *IEEE ICASSP*.
[73] Smart Phone Market 2022. Global Smartphone Market Share: By Quarter. https://www.counterpointresearch.com/global-smartphone-share/.
[74] Smartphone Sales 2021. Gartner Says Worldwide Smartphone Sales Grew 10.8 percent in Second Quarter of 2021. https://www.gartner.com/en/newsroom/press-releases/2021-09-01-2q21-smartphone-market-share.
[75] Sound Vibration 2022. Vibration: Origin, Effects, Solution. https://www.gcaudio.com/tips-tricks/vibration-origins-effects-solutions/.
[76] Sound Vibration Proof 2016. Practical Sound and Vibration Proofing via Speaker Isolation. https://www.andrehvac.com/blog/vibration-control-products/practical-sound-vibration-proofing-speaker-isolation-pads/.
[77] Speaking 2022. Speaking. http://www.psy.vanderbilt.edu/courses/psy216/SPEAKING.html.
[78] Speech 2021. FACTS ABOUT SPEECH INTELLIGIBILITY. https://tinyurl.com/38j2bjbt.
[79] Speech Recognition 2022. Speech Recognition on LibriSpeech test-clean. https://paperswithcode.com/sota/speech-recognition-on-librispeech-test-clean.
[80] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426* (2017).
[81] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*. Springer, 443–450.
[82] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215* (2014).
[83] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.
[84] Navneet Upadhyay and Abhijit Karmakar. 2015. Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study. *Procedia Computer Science* 54 (2015), 574–584.
[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[86] Video Conferences 2021. How Zoom leverages AI to provide the best videoconferencing experience. https://digital.hbs.edu/platform-digit/submission/how-zoom-leverages-ai-to-provide-the-best-videoconferencing-experience/.
[87] Volume Booster 2019. Volume Booster Tips for Smartphones and Tablets. https://www.lifewire.com/boost-volume-on-phone-and-tablet-4142971.

[88] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[89] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. 2015. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 155–166.

[90] Wei Wang, Lin Yang, and Qian Zhang. 2016. Touch-and-guard: secure pairing through hand resonance. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 670–681.

[91] Yan Wang, Tianming Zhao, Fatemeh Tahmasbi, Jerry Cheng, Yingying Chen, and Jiadi Yu. 2020. Driver Identification Leveraging Single-turn Behaviors via Mobile Devices. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.

[92] Word Error Rate 2019. Word Error Rate Mechanism, ASR Transcription and Challenges in Accuracy Measurement. https://tinyurl.com/4229u6a3.

[93] Chunlei Zhang, Chengzhu Yu, Chao Weng, Jia Cui, and Dong Yu. 2018. An exploration of directly using word as acoustic modeling unit for speech recognition. In *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 64–69.

[94] Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243* (2019).

[95] Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. 2015. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 301–315.

[96] Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. 2017. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7340–7351.