ABSTRACT A wide variety of adversarial attacks have been proposed and explored using image and audio data. These attacks are notoriously easy to generate digitally when the attacker can directly manipulate the input to a model, but are much more difficult to implement in the real world. In this paper we present a universal, time invariant attack for general time series data such that the attack has a frequency spectrum primarily composed of the frequencies present in the original data. The universality of the attack makes it fast and easy to implement as no computation is required to add it to an input, while time invariance is useful for real world deployment. Additionally, the frequency constraint ensures the attack can withstand filtering defenses. We demonstrate the effectiveness of the attack on two different classification tasks through both digital and real world experiments, and show that the attack is robust against common transform-and-compare defense pipelines.

INDEX TERMS Adversarial machine learning, deep learning, time series, automatic speech recognition, artificial intelligence.

I. INTRODUCTION
The quantity of proposed adversarial attacks for both image and audio data is vast. Generally, these attacks are easy to create and deploy when the attacker can directly modify an image or audio recording that a model receives as an input. However, implementation is more difficult in the real world where an attacker must interfere with the data as it is collected. In the image domain, this may require printing out a patch or other object and placing it in the scene before the scene is photographed and in the audio domain this may require broadcasting an attack over-the-air while the data is recorded [1], [28].

In addition to the added cost of physically implementing these attacks, real world attacks are also constrained by physical limitations. For example, several timed series attacks implemented digitally propose computing the attack based on a signal and then mixing it back into the signal [2]. However, with real-time streaming data this is infeasible because the attack cannot be calculated until the signal is recorded, and thus the attack cannot be mixed into the signal during recording [3], [15]. Moreover, a real world attack is limited by equipment such as microphones, which may automatically remove frequencies outside of the range of human hearing or other receiver effects [7]. Finally, a real world attack must also be robust against environmental effects such as noise and reverberation [29]. Beyond these physical considerations, an effective attack should also evade common defense methods. For example, filtering may be able to remove an attack outside the frequency spectrum of the unperturbed data.

We propose a procedure to train a universal, time invariant attack, $v$, for general time-series data such that the frequency spectrum of $v$ matches the frequency spectrum of the original, unperturbed data. Given a trained model $f$, a universal adversarial attack is a single $v$ such that $f(x + v)$ fools the model for most inputs $x$ sampled from a distribution [16]. The universality of the attack does not require us to know the specific signal that we are going to attack ahead of time and allows us to efficiently add the attack to a signal. The time invariance of the attack means that we can play the attack on a loop and the effectiveness of the attack will not be sensitive to the alignment of the start of the attack and signal. Finally, the frequency constraint ensures that our attack is robust against basic filtering defenses. We demonstrate that this attack is effective on both speech data and unintended radiated emission (URE) data through digital and real world experiments.
II. BACKGROUND

Image classifiers were originally found to be vulnerable to digital adversarial attacks in Szegedy et al. [25]. That is, the addition of a small, carefully crafted perturbation to a given image can cause the classifier to incorrectly classify the image as a specified target. Moreover, these adversarial attacks often transfer between models. It was later shown by Moosavi-Dezfooli et al. [16] that universal, targeted perturbations exist. These are single perturbations that cause an image classifier to misclassify almost all inputs.

Adversarial attacks are not unique to image classifiers, and deep learning models for many different real-world tasks are vulnerable to adversarial attacks. This includes speech systems, such as speech-to-text models [2], radio signal classification models [21], and graph models [13].

It is also possible to implement adversarial attacks in the real world. Sharif et al. [22] printed a pair of eye glasses frames, that allowed people wearing them to evade recognition by a facial recognition system or impersonate another person. Eykholt et al. [6] generated stickers to cause targeted misclassification of road signs. However, while these initial works demonstrate the feasibility of attacks in the real world, they do not necessarily reflect practical constraints as both of these physical attacks require ample training data and prior knowledge of the person or sign respectively in the attack. Brown et al. [1] used the idea of a universal attack to create a universal patch that can be printed out, added to a scene, and photographed, causing a classifier to predict a targeted class. Similarly, Wu et al. [28] created wearable clothing with patterns what allowed a person to evade an object detector in the context of different scenes. These universal attacks are important for realistic physical attacks because the attacker does not need to know the exact deployment scene when generating the attack.

Similarly, for time series data, some over-the-air attacks under realistic constraints have been proposed. In the speech domain, Yakura and Sakuma [29] designed a training procedure for a targeted attack that limits the frequency range of the constructed attack so that the learned attack can be emitted from a microphone with limited distortion. Yet time series attacks must also deal with uncertainty about the data being attacked. The exact data and data length are not known until the data is recorded, but an over-the-air attack must be broadcasting during the recording. Chiquier et al. [3] proposes a predictive attack for speech data computed in real-time while Mathov et al. [15] proposes a training an untargeted, universal attack for speech data that can be replayed on a loop during recording. Sadeghi and Larsson [21] also craft an untargeted, universal attack for radio signal classifiers.

At the same time as the development of both digital and physical adversarial attacks, defenses against these adversarial attacks have been devised. Broadly, these methods include training a model to be robust against adversarial examples [14], [17] as well as methods to flag adversarial examples. One common, generally successful approach for this is to detect distribution shifts between benign and adversarial examples. In the RF domain, Kokalj-Filipovic and Miller [11] use statistical tests that rely on that fact that benign and adversarial examples have different Peak-to-Average Power Ratio distributions, as well as different distributions of softmax scores from the last later of a neural network classifier. The power statistic is in particular designed to be robust to real world considerations from an unknown and noisy environment.

A related approach in the speech domain uses a transform-and-compare pipeline to detect distribution shifts between benign and adversarial examples. Given an input, which could be benign or adversarial, the pipeline compares model predictions on the input and a transformed version of the input. Several different transformation functions have been proposed including the addition of random noise [5], [18], audio compression [19], [30], quantization [10], down-up sampling [10], and filtering [12]. Then, the distance between model predictions on transformed and original input is calculated to construct a distribution for each of benign and adversarial distances. This method exploits the fact that adversarial examples are generally less robust against perturbations than benign examples are. If the distance between the model predictions on the transformed and original input is higher than a threshold, the input is flagged as adversarial. The attack in Mathov et al. [15], is implementable in the real world, but as they point out, the attack appears to be vulnerable to a compression defense. Thus, real world adversarial attacks should not only take into account practical considerations, but should also be robust against relevant defense strategies.

III. ATTACK

A. THREAT MODEL

We assume the attacker’s goal is not necessarily to create a state-of-the-art attack, but rather to create an attack with a non-zero, but not necessarily perfect success rate, be relatively inexpensive to create and be difficult to filter through simple or moderate methods. An attack with these characteristics would mean the defender either has to expend much more energy to defend against the attack, than was used to create it, or the defender must accept the small but not zero degradation to their model’s performance.

We consider a white-box model case, in which we assume the attack has access to the defender’s model, including the architecture and all weights. In addition, we consider a limited black-box case, in which we assume the attacker has access to neither the architecture nor the weights. In both cases we assume the attacker can collect data from the defender.

B. ALGORITHM

Let $D = \{(x_i, y_i)\}_{i=0}^{n}$ be a training dataset of time-series, $x_i \in \mathbb{R}^T$ sampled at rate $f_s$ with labels $y_i$. Our proposed method learns a single attack, $v$ of length $T$. We refer to the attack in time space as $v_{\text{time}}$ and we refer to the attack in frequency space as $v_{\text{freq}}$. The attack can be converted between these representations using the fast Fourier transform (FFT)
In the proposed implementation, the attack is played repeatedly in the background. In the context of speech recognition, the time when a person starts speaking and the recording begins may not align with the beginning of the attack cycle. We design the attack so that it is time invariant, meaning that the attack remains effective regardless of what point in its cycle the recording begins at. In the URE context, the person can be replaced by a device emitting an electronic signal.

or the inverse fast Fourier transform (IFFT). Explicitly, \( v_{\text{freq}} = \text{FFT}(v_{\text{time}}) \) and \( v_{\text{time}} = \text{IFFT}(v_{\text{freq}}) \). Because our datasets are real-valued, we take \( \text{Re}(v_{\text{time}}) \) as the final universal attack.

The proposed implementation of the attack is to repeatedly play the trained attack on a loop while data is recorded intermittently as depicted in Fig. 1. In order for the attack to be effective it must therefore be time invariant, meaning that the attack remains effective regardless of what point in its cycle the recording begins at. To ensure time invariance, we advance the attack by a random time shift of \( t \) at each pass through the model during training. This is implemented in frequency space, by multiplying the attack’s Fourier coefficient for frequency \( k \), \( v_{\text{freq}}[k] \), by \( e^{i2\pi kt/T} \).

We also constrain the frequency spectrum of the attack to match the frequency spectrum of the original time series, \( \{x_i\}_{i=0}^{n} \), to ensure that the attack is not easily detectable or removed through filtering. Specifically, during the first phase of training, we require that each Fourier coefficient of \( x_i + v \) be no more than twice the corresponding Fourier coefficient of \( x_i \). We use the loss term, \( L_{\text{spectrum}}^1 \), defined in (1), to enforce this constraint. Then, during the second phase of training we replace \( L_{\text{spectrum}}^1 \) with \( L_{\text{spectrum}}^2 \), defined in (2), which compares the Fourier spectrum on a log scale, rather than linear scale. This second phase of training accounts for the different scales of the Fourier coefficients and enables the attack to better match the frequency spectrum of the training dataset than just using the first loss term alone.

\[
L_{\text{spectrum}}^1 = \sum_{i=0}^{n} \text{ReLU}(|\text{FFT}(x_i + v)|) - 2 \times |\text{FFT}(x_i)| \quad (1)
\]

\[
L_{\text{spectrum}}^2 = \sum_{i=0}^{n} \text{ReLU}\left(20 \log_{10}\left(\frac{|\text{FFT}(x_i + v)|}{2 \times |\text{FFT}(x_i)|}\right)\right) \quad (2)
\]

In Fig. 2, we show the spectrogram of an adversarial example, \( x_i + v \) trained on the Speech Commands dataset, as well as the frequency spectrum of a baseline UAP attack. The frequency spectrum of our attack is closely aligned with the frequency spectrum of the unperturbed benign example, whereas the other, baseline UAP attack has high frequency components not present in the original data. In section V-B, we demonstrate that other baseline attacks are much more vulnerable to being removed through low-pass filtering than our attack is.

The other loss term we train with, \( L_{\text{classifier}} \), is the negative cross-entropy loss on the model prediction, \( f(x_i + v) \), to ensure that the adversarial attack fools the model. The full training procedure is outlined in detail in Algorithm 1. Note that the classification model \( f \) refers to the composition of the spectrogram prepossessing step and the classifier network.

IV. EVALUATION

A. METRICS

We use the adversarial success rate (ASR) as our primary evaluation metric. The ASR of an attack is defined as the

FIGURE 1. In the proposed implementation, the attack is played repeatedly in the background. In the context of speech recognition, the time when a person starts speaking and the recording begins may not align with the beginning of the attack cycle. We design the attack so that it is time invariant, meaning that the attack remains effective regardless of what point in its cycle the recording begins at. In the URE context, the person can be replaced by a device emitting an electronic signal.

FIGURE 2. A benign example, and learned adversarial examples, \( x_i + v \), on the Speech Commands dataset. The amplitude of each learned attack has been adjusted so that the SNR is fixed at 10 dB. The spectrogram of our attack is much more similar to the spectrogram of the benign example compared to the baseline UAP attack, particularly at high frequencies.
Algorithm 1: Training Procedure for Universal FFT Attack.

**Input:** Training dataset \( D = \{(x_i, y_i)\}_{i=0}^n \) with time series \( x_i \in \mathbb{R}^T \) sampled at rate \( f_s \) with labels \( y_i \), trained classification model \( f \), desired number of training epochs \( N \)

**Output:** Attack vector \( v \in \mathbb{R}^T \)

1: Initialize the vector in the frequency domain \( v_{freq} \leftarrow 0 \)
2: while Current epoch \( < N \) do
3: for each \( (x_i, y_i) \in D \) do
4: Sample a random time shift, \( t \sim \text{Uniform}[0, \frac{T}{f_s}] \),
5: for \( k \in \left[ \frac{-N}{2}, \ldots, \frac{N}{2} - \frac{L}{f_s}, \frac{L}{f_s} \right] \) do
6: \( v_{freq}[k] = v_{freq}[k]e^{2\pi ikT} \)
7: end for
8: Transform, \( v_{time} = \text{Re}((FFT)(v_{freq})) \)
9: \( \hat{y} = f(x_i + v_{time}) \)
10: if Current epoch \( < 0.8 \times N \) then
11: \( L = \mathcal{L}_{\text{classifier}}(y, \hat{y}) + \beta \mathcal{L}_{\text{spectrum}}(x_i, x_i + v_{time}) \)
12: else
13: \( L = \mathcal{L}_{\text{classifier}}(y, \hat{y}) + \alpha \mathcal{L}_{\text{spectrum}}(x_i, x_i + v_{time}) \)
14: end if
15: Update the perturbation, \( v_{freq} \leftarrow v_{freq} + \alpha \nabla v_{freq} L \)
16: end for
17: end while
18: return \( \text{Re}(v_{time}) \), real-valued attack in the time domain

Percentage of originally correct model predictions that the attack successfully changes the prediction of. Unlike the error rate, the ASR only counts inputs where the attack changes the model prediction and does not give the attack credit for inputs the model was originally wrong on. An ASR close to one indicates a highly effective attack. More formally, for a model \( f \), attack \( v \), and dataset \( D = \{(x_i, y_i)\}_{i=0}^n \) with inputs \( x_i \) and labels \( y_i \):

\[
\text{ASR} = \frac{\sum_{i=0}^n I(f(x_i) = y_i \land f(x_i + v) \neq y_i)}{\sum_{i=0}^n I(f(x_i) = y_i)} \tag{3}
\]

A useful quantity for validation and visualization is the ASR as a function of the signal-to-noise ratio (SNR). As in [29], the SNR is \( 10 \log_{10} \frac{P_s}{P_v} \) where \( P_s \) is the power of the unperturbed input, \( \frac{1}{T} \sum_i^T x_i^2 \), and \( P_v \) is the power of the attack, \( \frac{1}{T} \sum_i^T v_i^2 \). The SNR is large when the attack is small and presumably less perceptible. The SNR is adjusted by scaling the magnitude of the attack \( v \). Equivalently, we vary \( \alpha \) so that the perturbed input is \( x_i + \alpha v \) and the SNR averaged over all data is as desired.

Additionally, we compare our models against two simple, baseline adversarial attacks: Fast Gradient Sign Method (FGSM) [8] and Universal Adversarial Perturbation (UAP) [16]. We consider these attacks as they are implementable in both the speech and URE domains considered. Moreover, we are unaware of any other attacks for general applications which are designed to be real-time implementable, time-invariant, and, robust to common filtering defenses. The attack in Mathov et al. [15] is the closest to this, however, the attack is known to be vulnerable to the defenses we consider.

In the FGSM attack, the perturbed input is \( x_i + \alpha \text{sgn}(\nabla_{x_i} \mathcal{L}_{\text{classifier}}(f(x_i), y_i)) \), where \( \alpha \) is selected to adjust the SNR. Note that this is not a universal attack, and rather a separate attack is generated for each input to the model. The UAP attack is trained as described in Algorithm 1 in Moosavi-Dezfooli et al. The perturbed input is \( x_i + \alpha v \) with \( \alpha \) adjusted to the desired SNR. Like our attack, these attacks are pre-computed before deployment. Additionally, all attacks are trainable within minutes, after appropriate hyperparameter tuning for our attack and the UAP attack.

### B. DATA

#### 1) SPEECH COMMANDS

The Speech Commands dataset is an audio dataset consisting of one-second clips of one-word commands such as ‘stop’ or ‘go’ sampled at a rate of 16 kHz [27]. For simplicity, we have removed audio clips labeled as background noise or unknown from the dataset resulting in ten classes with 30 k training examples and 3.7 k validation examples. We focus on a classification task with this dataset.

#### 2) CORONA DUFF

The Corona Duff dataset consists of unintended radiated emission (URE) data from 20 common household devices, including a desktop monitor, alarm clock, and a table fan, collected in a residential environment [26]. Voltage and current data were collected from each device over four non-consecutive ten minute runs at a sample rate of 192 kHz. The classification task is to identify the device based on a time series of its voltage data. Our training dataset consists of 10 k randomly selected 0.1 s segments of voltage data from the different devices. The validation data consists of 2 k randomly selected 0.1 s segments of voltage data, selected from different data collection runs than the training data. Fig. 10 includes a visualization of Corona Duff data.

#### 3) PREPROCESSING

For both datasets, we convert the time series to a spectrogram as a preliminary step in the model pipeline. We adjust the length of the FFT used and the step size between FFT windows for each dataset and stack the real and imaginary channels so that the resulting real-valued spectrogram has dimensions \( 2 \times 224 \times 224 \).

### V. DIGITAL EVALUATION

For digital experiments we train the attack on a ResNet18 [9] which has been pretrained on ImageNet [4] and finetuned on the relevant dataset. We additionally utilize a VGG16 [23] for black-box testing.

In Fig. 3, we plot the ASR of our attack and the baseline attacks on both datasets as a function of the SNR. These results are averaged over five attacks, each trained on a
FIGURE 3. Adversarial success rate as a function of the SNR on the Speech Commands dataset (left) and on the Corona Duff dataset (right). We evaluate each attack on a white box (WB) model, as well as on a black box (BB) model.

FIGURE 4. Adversarial success rate as a function of the time shift on the adversarial attack. At time shift \( t \), the attack used is the segment of the looped attack from time \( t \) to time \( t + T \) where \( T \) is the length of the input being attacked.

A. TIME INVARIANCE

To test the time invariance of the attack, we repeatedly play the attack on a cycle and shift the start time of the speech recording or URE data as depicted in Fig. 1. For these experiments with fix the SNR at 10 dB. We also include an ablation study, where we removed the time invariance transformation from the attack training by fixing the random training time shift to \( t = 0 \) in line 4 of Algorithm 1.

In Fig. 4 we see that both the white box and black box attacks have relatively constant ASR on both datasets (blue lines). In contrast, the white box ablation attack on the Speech Commands dataset has a high ASR of 35% with a time shift of zero at evaluation time, but for all evaluation time shifts greater than zero the ASR drops below 15%. The black box ablation attack is also sensitive to the evaluation time shift, demonstrating a greater variance in ASR as a function of time than the black box version of the attack. The Corona Duff dataset exhibits similar trends.

B. ROBUSTNESS TO FILTERING

To evaluate the robustness of the attack to filtering, we propose a set-up where a defender receives an input, which could be benign or adversarial. The defender filters the received input and then evaluates it on a model that has been trained on filtered benign data. Since our datasets are composed of mostly low frequency information, we use low-pass filtering. If the received input is adversarial and the attack is composed of high frequencies, the filtering should remove most of the attack before the model makes its prediction, thus reducing...
FIGURE 5. Adversarial success rate of each attack on a classifier that preprocesses inputs using a low-pass filter with the indicated cutoff frequency. The SNR for each attack has been adjusted so that the adversarial success rate at the highest cutoff frequency is comparable. For the Speech Commands data, the SNRs are FFT (5.5 dB), UAP (3.5 dB), FGSM (3.5 dB), FFT- No Fourier Constraint (12 dB). For the Corona Duff data, the SNRs are FFT (12 dB), UAP (5 dB), FGSM (20 dB), FFT- No Fourier Constraint (12 dB).

FIGURE 6. $L_2$ distance distributions between classifier predictions of an input, which could be adversarial or benign, and the transformed input after MP3 compression on the Speech Commands dataset. From left to right the adversarial inputs are generated using our FFT attack, FGSM, and UAP.

FIGURE 7. Accuracy of real world implementation of the untargeted attacks.

the effectiveness of the attack. If the received input is benign, the model prediction should be accurate because the model has been trained on filtered benign data.

More formally, let $f_k$ denote a classifier which has been trained on data low-pass filtered with cutoff frequency $k$. Then, $f_k(\text{lowpass}_k(x))$ is the classifier prediction on benign input $x$ and $f_k(\text{lowpass}_k(x + \alpha v))$ is the classifier prediction on adversarial input $x + \alpha v$. We evaluate the ASR of our attack and our baseline attacks on $f_k$ for a range of frequencies. Note that the attacks are the same attacks as in the previous sections. These were learned on an unfiltered model with unfiltered data and were not trained on the $f_k$ they are evaluated on.

Fig. 5 depicts the results. For reference, we also include a version of our attack trained without either $L_{\text{spectrum}}$ loss term so that the frequency spectrum of this ablation attack is not constrained to match the frequency spectrum of the benign data. We adjust the SNR of each attack so that the ASR of each attack is comparable when the cutoff frequency is half the sampling rate.

On the Speech Commands dataset with low pass filtering the ASR of our attack is greater than or equal to 32% for all frequencies tested. In contrast, for all other attacks, as lower cutoff frequencies are used, the effectiveness of the attack is reduced from an ASR of 40% at a cutoff frequency of 8 kHz to below 20% for cutoff frequencies of 500 Hz and below. As depicted in Fig. 2, the baseline UAP and FGSM attacks have high frequency components which low-pass filtering removes. Similarly, on the Corona Duff dataset our attack has an approximately constant ASR, averaging 82% across all cutoff frequencies tested. In contrast, the other attacks demonstrate a significant reduction in ASR with filtering from a 70% ASR at 192 kHz to below a 40% ASR for the baseline UAP and FGSM attacks. Thus, by restraining our attack to only use the
low frequencies present in the original data, we have designed an attack that is much more robust to this filtering test than any of the baseline attacks tested.

C. TRANSFORM AND COMPARE DEFENSES

We additionally test our attack against the transform-and-compare defense pipeline described in Section II. This pipeline is typically applied to white box models. In a speech recognition system defense pipeline, the character error rate is used to measure the distance between model predictions. However, because our models are generic classifiers rather than speech-to-text models, we use the $L_2$ distance between the model outputs of the original input and transformed input. We calculate this distance for 800 benign inputs and 800 adversarial inputs. From the resulting distributions, an optimal threshold can then be determined to use for flagging adversarial examples.

Using MP3 compression as the transformation function on Speech Commands data, we plot the distribution of distances between the original and transformed benign inputs and the distribution of distances between the original and transformed adversarial inputs for our attack (FFT) and the FGSM and UAP baselines in Fig. 6. For the FGSM and UAP attacks, the distances for adversarial data are generally larger than for benign data, making adversarial examples generated with these attacks more easily identifiable. For our attack, we find much more overlap in the distribution of benign distances and the distribution of adversarial distances, demonstrating that our attack is much more difficult to flag using this defense method.

Table 1 reports the area under the curve (AUC) that plots the true positive rate against the false positive rate for all transformations tested, averaged over 5 runs of attack training.

Full details of each transformation function, as well as the full AUC plots, are in Appendix B. On both datasets, for all transformations tested, our attack has an AUC score close to 0.50, averaging an AUC of 0.57 across all transformations on Speech Commands and 0.56 on Corona Duff. For comparison, the AUC scores of the baseline attacks are much higher with average AUC scores of 0.87 and 0.81 for the UAP and FGSM attack on Speech Commands and 0.83 and 0.72 for the UAP and FGSM attack on Corona Duff. This demonstrates that our attack is more difficult to identify than the other attacks using the transform-and-compare defense pipeline.

D. REAL WORLD EVALUATION

For real world experiments, we attack a classifier that utilizes X-vectors [24] pretrained on the Speech Commands dataset by SpeechBrain [20]. To initialize a baseline, we record a speaker reading out words in the Speech Commands dataset. Then, for each of white noise, our attack, and the baseline UAP attack, we set the SNR of each attack averaged over the Speech Commands dataset to 5 dB, and play each attack from a MacBook Pro at the same volume while a speaker reads out words from the Speech Commands dataset. Each word is recorded five times for each attack.

The data was recorded on a MacBook Pro using the Voice Memo application. We then input the recordings directly to the model. The white box attack was trained directly on the SpeechBrain model and the black box attack was trained on a ResNet. We do not include the FGSM attack in our experiments as this attack requires knowledge of the exact data it is attacking.

Fig. 7 summarizes the results averaged over all recordings and demonstrates that our attack is an effective real world attack as both a white-box and black-box attack. When we recorded the speaker reading out speech commands in the presence of no noise or Gaussian noise, the classifier achieved 100% and 82% accuracy respectively. Our attack trained on Speechbrain’s model is 10% accurate, the equivalent of a null classifier, while the baseline UAP attack is 40% accurate. The black box attacks are also effective. Additionally, there is little change to these numbers when low-pass filtering is applied to our recorded data, which is likely due to the filtering preprocessing built into the SpeechBrain model.

VI. CONCLUSION

We presented an adversarial attack for general time series data designed for real world implementation. Through digital experiments, we demonstrate that for both speech and URE data, the universal attack is time invariant, robust to filtering, and is robust to common transform-and-compare defense pipelines. We also test in the attack in the real world and demonstrate that it is effective.

It would be interesting to adapt this work to make the attack targeted. Initial experiments have demonstrated that there is a sharp trade-off between the effectiveness of a targeted attack trained using this method and its detectability. Future work to improve a targeted attack will be needed.
FIGURE 8. ROC curves for each transformation used in the defense transform-and-compare pipeline. The top row depicts results for the Speech Commands (SC) dataset, and the bottom row depicts results for the Corona Duff (CD) dataset.

FIGURE 9. Our learned universal attack on each dataset, in time space.

FIGURE 10. Sample adversarial examples on each dataset in time space. The lighter colored time series is the original, unperturbed benign example and the darker colored time series is the adversarial example with an SNR of 10 dB.

APPENDIX A

A. ATTACK VISUALIZATIONS

Below we provide further visualizations of the learned attacks on our datasets in Figs. 9 and 10.

B. DEFENSES

In this section, we provide full details of the transformation functions tested in the transform-and-compare defense pipeline. The quantization function quantizes inputs to an 8-bit representation and then dequantizes inputs. We use the TensorFlow implementation. The down-up sampling function downsamples inputs to half the original sample rate and then upsamples this sequence back to the original sample frequency. With noise flooding Gaussian noise with standard deviation 0.01 is added to the inputs. We additionally tested noise flooding specific frequency bands by filtering the Gaussian noise with a band-pass filter as in Rajaratnam and Kalita [18]. The results for noise flooding without filtering are very similar to noise flooding with band-pass filtering so we just report the scores for noise flooding without filtering.

In Fig. 8, we plot ROC curves for each transformation used. For both datasets and all transformations used, our attack is the least detectable under the transform-and-compare pipeline.

ACKNOWLEDGMENT

The research described in this paper was conducted under the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy. This effort was also enabled by utilizing Pacific Northwest National Laboratory Research Computing resources.
REFERENCES

[1] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” 2017, arXiv:1712.09665.

[2] N. Carlini and D. Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in Proc. IEEE Secur. Privacy Workshops, 2018, pp. 1–7, doi: 10.1109/SPW.2018.00009.

[3] M. Chiquier, C. Mao, and C. Vondrick, “Real-time neural voice camouflage,” in Proc. Int. Conf. Mach. Learn., 2022.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

[5] M. Dong, D. Yan, Y. Gong, and R. Wang, “Adversarial example devastation and detection on speech recognition system by adding random noise,” College Inf. Sci. Eng., Ningbo Univ., Ningbo Zhejiang, China, 2023. [Online]. Available: https://aes2.org/publications/elibrary-page/?id22029

[6] K. Eykholt et al., “Robust physical-world attacks on deep learning visual classification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1625–1634.

[7] B. Flowers, R. M. Buehrer, and W. C. Headley, “Evaluating adversarial evasion attacks in the context of wireless communications,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 1102–1113, 2020.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 770–778.

[10] S. Hussain, P. Neekhara, S. Dubnov, J. McCaulley, and F. Koushanfar, “WaveGuard: Understanding and mitigating audio adversarial examples,” in Proc. USENIX Secur. Symp., 2021, pp. 2273–2290.

[11] S. Kokalj-Filipovic and R. Miller, “Adversarial examples in RF deep learning: Detection of the attack and its physical robustness,” in Proc. IEEE Glob. Conf. Signal Inf. Process., 2019, pp. 1–5.

[12] H. Kwon, H. Yoon, and K.-W. Park, “Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system,” Neurocomputing, vol. 417, pp. 357–370, 2020[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220312960

[13] H. Liu, B. Wang, J. Wang, X. Dong, T. Yang, and J. Caverlee, “Everything perturbed at all once: Enabling differentiable graph attacks,” in Proc. ACM Web Conf., 2024, pp. 485–488.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.

[15] Y. Mathow, T. B. Senior, A. Shabtai, and Y. Elovici, “Stopbugging me! Evading modern-day wiretapping using adversarial perturbations,” Comput. Secur., vol. 121, 2020, Art. no. 102841.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 86–94.

[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2016, pp. 582–597.

[18] K. Rajaratnam and J. Kalita, “Noise flooding for detecting audio adversarial examples against automatic speech recognition,” in Proc. IEEE Int. Symp. Signal Process. Inf. Technol., 2018, pp. 197–201.

[19] K. Rajaratnam, K. Shah, and J. Kalita, “Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition,” in Proc. 30th Conf. Comput. Linguist. Speech Process., 2018, pp. 16–30.

[20] M. Ravanelli et al., “SpeechBrain: A general-purpose speech toolkit,” 2021, arXiv:2106.04624.

[21] M. Sadeghi and E. G. Larsson, “Adversarial attacks on deep-learning based radio signal classification,” IEEE Wireless Commun. Lett., vol. 8, no. 1, pp. 213–216, 2019.

[22] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1528–1540, doi: 10.1145/2976749.2978392.

[23] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[24] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, “Spoken language recognition using X-vectors,” in Proc. Speaker Lang. Recognit. Workshop, 2018, pp. 105–111.

[25] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, arXiv:1312.6199.

[26] J. M. Vann, T. P. Karnowski, R. Kerekes, C. D. Cooke, and A. L. Anderson, “A dimensionally aligned signal projection for classification of unintended radiated emissions,” IEEE Trans. Electromagn. Compat., vol. 60, no. 1, pp. 122–131, Feb. 2018.

[27] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018, arXiv:1804.03209.

[28] Z. Wu, S.-N. Lim, L. Davis, and T. Goldstein, “Making an invisibility cloak: Real world adversarial attacks on object detectors,” in Proc. Eur. Conf. Comput. Vis., 2019.

[29] H. Yakura and J. Sakuma, “Robust audio adversarial example for a crime: Real and stealthy attacks on state-of-the-art face recognition,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1528–1540, doi: 10.1145/2976749.2978392.

[30] K. Rajaratnam, K. Shah, and J. Kalita, “Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition,” in Proc. 30th Conf. Comput. Linguist. Speech Process., 2018, pp. 16–30.

[31] M. Ravanelli et al., “SpeechBrain: A general-purpose speech toolkit,” 2021, arXiv:2106.04624.

[32] M. Sadeghi and E. G. Larsson, “Adversarial attacks on deep-learning based radio signal classification,” IEEE Wireless Commun. Lett., vol. 8, no. 1, pp. 213–216, 2019.

[33] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1528–1540, doi: 10.1145/2976749.2978392.

[34] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[35] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, “Spoken language recognition using X-vectors,” in Proc. Speaker Lang. Recognit. Workshop, 2018, pp. 105–111.

[36] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, arXiv:1312.6199.

[37] J. M. Vann, T. P. Karnowski, R. Kerekes, C. D. Cooke, and A. L. Anderson, “A dimensionally aligned signal projection for classification of unintended radiated emissions,” IEEE Trans. Electromagn. Compat., vol. 60, no. 1, pp. 122–131, Feb. 2018.

[38] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018, arXiv:1804.03209.

[39] Z. Wu, S.-N. Lim, L. Davis, and T. Goldstein, “Making an invisibility cloak: Real world adversarial attacks on object detectors,” in Proc. Eur. Conf. Comput. Vis., 2019.

[40] H. Yakura and J. Sakuma, “Robust audio adversarial example for a crime: Real and stealthy attacks on state-of-the-art face recognition,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1528–1540, doi: 10.1145/2976749.2978392.

[41] K. Rajaratnam, K. Shah, and J. Kalita, “Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition,” in Proc. 30th Conf. Comput. Linguist. Speech Process., 2018, pp. 16–30.