

Title	Detection of wake word jamming
Authors	Sagi, Prathyusha;Sankar, Arun;Roedig, Utz
Publication date	2024-11-22
Original Citation	Sagi, P., Sankar, A. and Roedig, U. (2024) 'Detection of wake word jamming', Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy, Salt Lake City, UT, USA, October 14-18, pp. 134-141. https://doi.org/10.1145/3690134.3694825
Type of publication	Conference item
Link to publisher's version	10.1145/3690134.3694825
Rights	© 2024, the Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/ - https://creativecommons.org/licenses/by/4.0/
Download date	2026-10-28 21:16:52
Item downloaded from	https://hdl.handle.net/10468/16843

Detection of Wake Word Jamming

Prathyusha Sagi
University College Cork
Cork, Ireland
p.sagi@cs.ucc.ie

Arun Sankar
South East Technological University
Carlow, Ireland
arun.sankar@setu.ie

Utz Roedig
University College Cork
Cork, Ireland
u.roedig@cs.ucc.ie

Abstract

Personal Voice Assistants (PVAs) such as Apple’s Siri, Amazon’s Alexa and Google Home continuously monitor the acoustic environment for a *wake word* to start interaction with the user. However, the wake word detection is susceptible to disruptions caused by acoustic interference. Interference might be by noise (e.g. background music, chatter, engine sounds) or a targeted jamming signal designed to disrupt PVA operations. As PVAs are increasingly used for critical applications such as medicine and the military, it is necessary to identify an attack. In this work, we re-design the wake word detection algorithm such that it is not only robust against an attack but is also able to identify an attack. Only if it is possible to identify an ongoing attack it is possible to employ appropriate countermeasures, i.e. remove the attacker. We modify the wake word detection model to function as a three-class classifier that accurately differentiates between clean wake words, wake words mixed with jamming signals and non-wake words. We further improve the classification results by examining the Direction of Arrival (DOA) and the Short Time Energy (STE) of the audio signal. DOA and STE information is usually available on off-the-shelf PVA which enables implementation of the proposed methods on existing devices.

CCS Concepts

• Security and privacy → Denial-of-service attacks.

Keywords

Personal Voice Assistant (PVA), Automatic Speech Recognition (ASR), Wake Word Recognition, Acoustic Denial of Service (DoS), Acoustic Jamming, Adversarial Training, Short Time Energy (STE), Direction of Arrival (DOA)

ACM Reference Format:

Prathyusha Sagi, Arun Sankar, and Utz Roedig. 2024. Detection of Wake Word Jamming. In *Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy (CPSIoTSec’24)*, October 14–18, 2024, Salt Lake City, UT, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3690134.3694825>

1 Introduction

A PVA continuously listens for a specific *wake word*, e.g. ‘Alexa’ in case of the Amazon Echo. Wake word detection is performed by a simple ASR model within the PVA. Upon detection, the PVA records the subsequent audio, which is sent to a cloud-based back end where advanced ASR is employed to extract and process user

commands. The wake word is used to start audio processing on a PVA and failure to detect a wake word will prevent usage of the PVA. Wake word detection can be interrupted either by background noise or by deliberate jamming signals that represent a DoS attack. Existing research has explored continuous and selective jamming [1, 6, 11, 13].

DoS attacks can exploit noise signals to jam the wake word detection process. Crafty attackers aim to carefully design effective jamming signals to be just strong and long enough to interfere with wake word recognition but avoid detection. These attacks strategically target sensitive areas within the wake word. By exploiting these areas, attackers can effectively disrupt the functioning of the PVA, posing a significant challenge to its reliability and usability [11]. To counter such deliberate jamming attacks, existing research suggests adversarial training. This involves training a wake word model with data that includes wake word samples superimposed with jamming signals only in the identified sensitive areas of the word. By subjecting the model to these adversarial conditions during training, it becomes robust to jamming attacks [10]. While this approach makes the wake word model resilient to jamming, merely detecting wake words in the presence of jamming signals is not sufficient. It is also crucial to detect the jamming attack to prompt appropriate action. In this work we address this issue by enhancing the existing wake word model proposed by Sagi et al. [10]. Our objective is not only to robustly detect the wake word but also to detect the presence of a jamming attack. By simultaneously focusing on detecting the wake word and spotting jamming signals, we provide a comprehensive solution to combat wake word jamming.

Our work extends the robust wake word detection model detailed by Sagi et al. [10], which is adversarially trained to resist jamming attacks. We transform the model from merely resisting jamming to also detecting the attack itself. We expand this model to a three-class classifier using transfer learning, enabling it to distinguish between wake words, non-wake words, and wake words mixed with jamming signals. Adversarial training might introduce a problem of false positives, to reduce such false classifications, we incorporate two additional techniques:

- Direction of Arrival (DOA) is employed to identify the angle of direction of arrival of wake word and jamming signals. Jamming signals usually originate from a different location than the wake word.
- Short Time Energy (STE) is used to detect abrupt changes in the energy pattern of an audio signal. Identifying such changes in a wake word audio suggests the presence of short, high-intensity jamming signal.

The specific contributions of this work are:



This work is licensed under a Creative Commons Attribution International 4.0 License.

CPSIoTSec’24, October 14–18, 2024, Salt Lake City, UT, USA

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1244-9/24/10

<https://doi.org/10.1145/3690134.3694825>

- (1) Adaptation of a wake word recognition model to function as a three-class classifier that enables it to accurately differentiate among clean wake words, wake words mixed with jamming signals (indicating potential attacks), and non-wake words. This adaptation enhances the model's capability to detect wake words even in the presence of jamming interference while also providing indications regarding attacks.
- (2) We use DOA and STE to further validate the model's predictions. These techniques help to reduce false alarms, thereby enhancing the reliability of the classifier.

2 Related Work

Existing work has examined the use of jamming for good (e.g. privacy preservation), for attacks (e.g. DoS) and existing work has provided some countermeasures. No existing work has investigated jamming detection in the context of wake word recognition.

Cheng et al. [1] describe using a Protection Jamming Device (PJD) to prevent a PVA from recording nearby conversations. The PJD monitors the acoustic environment for the wake word and emits a jamming signal when detected, effectively blocking wake word recognition. Sun et al. [13] propose Micsheild, a device that continuously emits white noise (0-4kHz) to prevent speech recording by always-on microphones. It stops jamming upon detecting a wake word. Li et al. [6] demonstrated a DoS attack on PVAs using an adversarial music synthesizer. Their experiments showed that jamming signals above 40dBA can disrupt wake word detection.

Sagi et al. [11] identified sensitive regions in the wake word where a short duration and low energy jamming signals could perform successful interruption. In this work, the authors focus on how an attacker might create a short and efficient signal that can cause interruption to wake word detection while remaining unnoticeable by the users. The use of inaudible sound sources is also explored by Guo et al. [4].

To improve resistance against background noise, Lim et al. [7] proposed cross-informed domain adversarial training with noisy wake word samples. Gao et al. [3] used mixed-condition training, and Yu and Zhou [15] employed random masking of audio data to enhance wake word detection under noisy conditions. Work by Sagi et al. [10] considers DoS attacks in the form of deliberate jamming and proposes adversarial training to make the wake word model more robust. This involves incorporating wake word samples along with jamming signals present exclusively in sensitive areas. The contribution detailed in this paper extends this contribution by Sagi et al. [10].

The work presented in this paper also makes use of DOA and STE to validate detection outputs. DOA is a technique used to determine the direction in which a sound signal arrives at a microphone array for sound source tracking [5]. Existing work has used microphone arrays and Deep Neural Networks (DNNs) to estimate the direction of arrival of an audio signal [5, 8]. We use this technique to track the DOA of the wake word audio to identify any changes indicating a potential attacker. STE analyzes changes in the energy of an audio signal over time. Previous work has utilized STE in telephone conversations to differentiate speech from background noise, optimizing resource usage in communication systems [2]. Here, we use STE to detect noise pulses in an audio signal.

3 Threat Model

A PVA should be available at all times and provide a reliable service to the user. We assume that attackers primary aim is to compromise the availability of these devices by exploiting vulnerabilities in the acoustic processing chain. Specifically, we assume the attacker aims to prevent correct functioning of the wake word detection models using acoustic jamming techniques. We assume attackers can emit sound near the PVA, either by controlling existing audio-capable devices such as smart TVs or home alarm systems or by planting dedicated attack devices. Attackers use a wake word detection model similar to the one used by the target PVA and emit a jamming signal upon detecting the wake word. Research indicates that attackers can only target sensitive areas within the wake word for successful jamming. Sensitive areas depend on the wake word. However, such regions have a duration of tenth of milliseconds and are located several milliseconds from the start of the wake word, giving attackers sufficient time to act [10, 11]. An attacker can detect a wake word early and place the jamming signal appropriately.

Our system acknowledges a significant challenge: while commercial PVAs might use sophisticated noise cancellation algorithms, they can only deal with background noise to some extent but are not sufficiently prepared to deal with targeted jamming attacks.

4 Attack: Wake Word Jamming

Wake word detection algorithms are simpler than backend speech recognition, to activate and process requests. These algorithms often use DNNs like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or *C-RNNs*, common in devices like Amazon Echo, Google Home, and Apple products.

For our work, we use the wake word detection model from Sagi et al. [10], trained to detect "Activate", we refer to this model as *LSTM_WD*. We focus on selective jamming to cause DoS attacks while staying undetected. The jamming signal ranges from 0kHz to 8kHz, lasting 2ms to 256ms, starting at an Signal to Noise Ratio (SNR) of 0dB.

Jamming is implemented by superimposing the jamming signal with the wake word, sliding it in 1ms steps over the entire length of the wake word. For each jamming position, we evaluate the output of the *LSTM_WD* algorithm (1375 probability scores). If all of the 1375 scores are below the threshold of 0.5, jamming was successful. Jamming using low-energy signals can be successful if the attacker targets specific areas of the wake word, called sensitive areas.

We use an experimental setup similar to the one described by Sagi et al. [10]. An office 3.8m x 3.4m is used for our setup including two Universal Serial Bus (USB) speakers and a *ReSpeaker Mic Array v2.0* equipped with four microphones. One speaker plays the wake word, while the other emits jamming signals. The *ReSpeaker Mic Array* records the audio signals emitted by two USB speakers, combines them into a single channel, and passes the audio to the *LSTM_WD* model for wake word detection. Both USB speakers are placed equidistantly, 24cm from the microphone array on a table. The jamming signal, with varying duration, power, and frequency bandwidth, is played with a 1ms incremental delay from the start of the wake word to cause interference.

To capture the results of jamming experiments, we use heatmaps. Figure 1 presents a heatmap showing wake word detection success

under interference with a 128ms duration ping noise. The x-axis represents time (duration of the wake-word), the y-axis shows the frequency range of the interference signal used. The color intensity shows the intensity of the jamming signal required to interfere successfully. The heatmap reveals that when the noise is introduced approximately between 85ms and 135ms from the start of the wake word, the model fails to detect the wake word across most frequency bands and noise power levels. Instead of continuous jamming, an attacker can focus on this sensitive area of the wake word to disrupt the wake word detection using a low-power jamming signal thus achieving a DoS attack.

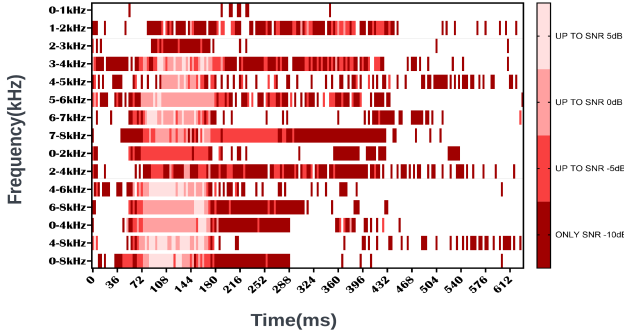


Figure 1: Wake word detection success under interference by a ping noise of 128ms duration.

5 Attack Detection

As shown by Sagi et al. [10] these DoS attacks can be countered by adversarial training. The wake word model from [10] is converted into a multiclass classifier. The modified model can categorize input audio as either *Wake Word*, *Non-Wake Word* or as *Attack* (when the wake word contains a jamming signal in the sensitive areas). As with any detection system it is necessary to reduce false alarms as much as possible. This can not only be achieved by improving the classifier but also by combining it with additional methods. When the classifier predicts a wake word under attack the result can be further validated.

One way to validate the classifier result is to analyse the DOA of the incoming wake word audio signal. PVAs usually employ a microphone array and it is possible to use it to obtain a direction estimation of the incoming sound. The attacker and user are unlikely to be co-located. Thus, when the attacker starts a jamming signal mid wake word the DOA briefly changes as it locks temporarily onto the jamming signal. If this happens in the sensitive region (see previous Section) of a wake word it is a strong indicator that the classifier result of detecting an attack was correct.

Another way to validate the classifier result is to check for sudden changes in the signal energies, specifically in the identified sensitive areas of the wake word using STE analysis. Sudden spikes in this area are a strong indicator that the classifier result of detecting an attack was correct.

Alerting users in case of an attack may be achieved by using the PVA's built-in notification systems. However, a detailed discussion of the alert mechanisms falls beyond the scope of this paper.

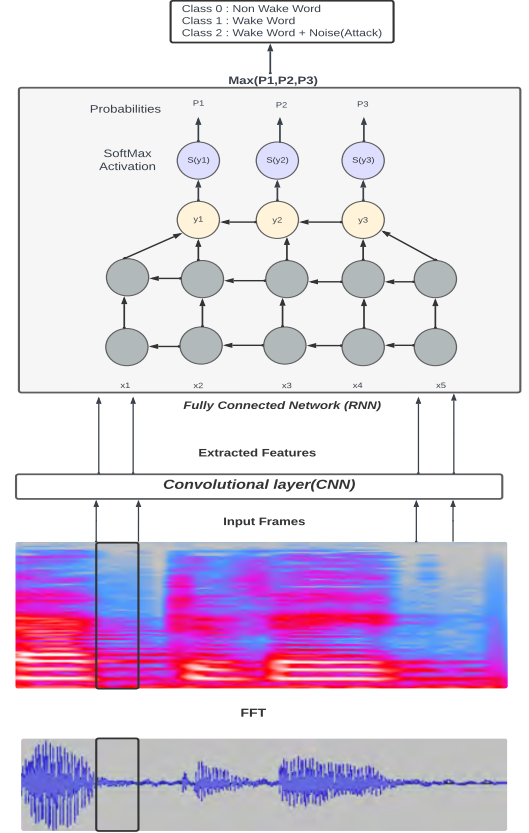


Figure 2: The wake word detection process of *LSTM_MCC*. The audio signal (bottom) is divided into frames of 10ms and features from each spectrogram are extracted and fed into an RNN. The softmax function is applied to the outputs and the maximum probability of the output is considered to make the categorization decision.

5.1 Training Data

To generate training samples, we use a similar approach as outlined by Sagi et al. [10]. We use the wake word 'Activate' spoken by thirteen different people. The training samples are created by superimposing jamming signals only over the sensitive areas of the wake word. The sensitive regions are identified experimentally as described by Sagi et al. [11]. We observe that the sensitive region of the wake word 'Activate' is always approximately between 80ms and 200ms of the wake word for different speakers. So we choose this region of the wake word spoken by all thirteen users for superimposing with jamming signals that differ in frequency bandwidth, duration and power. The jamming signal used for adversarial training is babel noise. The training data is labeled such that the *Wake Word* is identified as *Class-1*, *Non-Wake Word* is identified as *Class-0* and *Wake Word + Jamming Signal* is identified as *Class-2*.

5.2 Training Process

The pre-trained LSTM_WD model which was trained only for wake words and non-wake words is taken as a starting point and all its

layers are retrained. For the LSTM_WD model, the last layer contains a sigmoid activation function that classifies whether spoken audio contains a wake word or not. Since it is a multiclass classification task, the activation function used in the last layer is changed to the softmax activation function. The model architecture for the multiclass classifier is as shown in Figure. 2. The training samples are split into 80% for training and 20% for validation. The model is trained with a categorical cross-entropy loss function for 500 epochs with a batch size of 50. The model generated from the last epoch is chosen to perform the multiclass classification. We refer to the multiclass classifier as *LSTM_MCC*

5.3 Jamming After Adversarial Training

The LSTM_MCC model is tested for its resilience against jamming after undergoing adversarial training using the same experimental setup as described in Section 4. The jamming signal used is a ping noise of duration 128ms on the wake word 'Activate'. The model recognizes the input as *Class-2*(Wake Word+Jamming Signal) and the resulting heat map is shown in Figure. 3. It can be observed that the sensitive areas are absent in the heatmap.

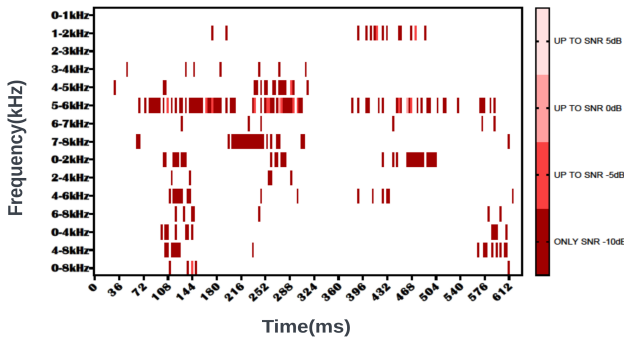


Figure 3: Wake word detection success under interference by a ping Noise of 128ms duration after adversarial training.

Adversarial training has made the LSTM_MCC model robust against jamming. It can now detect wake words even when jamming signals are present in sensitive areas. Jamming in areas other than sensitive regions is not relevant as it does not constitute a successful attack. We have used here one specific wake word algorithm. However, Sagi et al. [10] has shown that adversarial training can be applied to other models too and therefore we assume that the aforementioned technique can also be applied to other models.

5.4 Direction of Arrival (DOA)

For experimentation purposes, we used the Respeaker-Mic Array to estimate the angle of arrival. The device executes the direction of arrival algorithm for the incoming audio signals and estimates the angle of the source. The implementation details of the DOA algorithm of the Respeaker-Mic Array are not publicly available; however, the generic concept behind the DOA is as follows. DOA is a concept used in signal processing to determine the angle from which an audio signal arrives. Multiple microphones are positioned in a specific geometric arrangement on a microphone array. When

a sound wave reaches the array, each microphone detects the wave at slightly different times due to their varying distances from the sound source. This Time Delay of Arrival (TDOA) between the microphone channels can be measured. One common technique to estimate TDOA is through generalized cross-correlation or least squares methods. These techniques analyze the recorded signals from different microphones to determine the time difference among them accurately. Once the TDOA is known for each pair of microphones, the DOA of the sound wave can be calculated. This calculation exploits the array's geometric structure and the TDOA information to determine from which direction the sound wave originated [5].

5.5 Short Time Energy (STE)

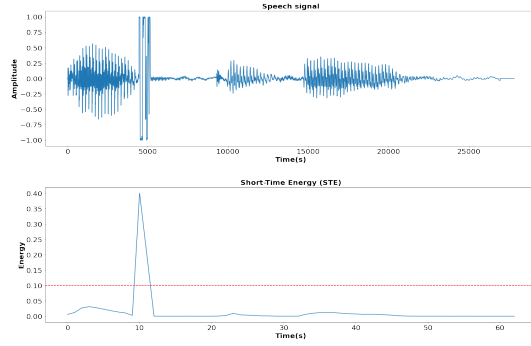
STE is used to identify how the energy of the audio signal changes over time. Instead of looking at the whole audio signal at once, STE looks at energy changes in smaller slices of time. If there is any sudden burst of noise in the audio, it will likely show a spike in the audio signal's energy within the time window [9]. We start by slicing the audio signal into windows or frames, each of size 20ms and compute STE for each time window. The detection threshold set for the STE peak is always at 0.1. Threshold value is used to identify frames where there is a significant change in energy. After conducting a series of empirical tests, we determined that a threshold value of 0.1 was optimal. If the STE of the audio signal shows any spikes in the sensitive areas of the wake word then, we are certain that there was an attempt to interrupt the wake word detection by an attacker.

6 Evaluation

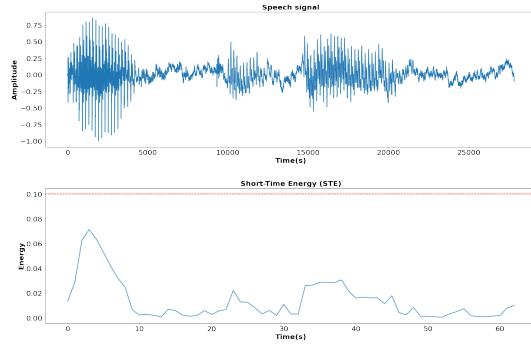
The LSTM_MCC model, which is able to detect an attack, is evaluated in various scenarios by conducting a series of experimental runs replicated 50 times. We compare the performance of this model with the basic LSTM_WD model that cannot identify an attack. The experiments are repeated 50 times under different conditions, such as different noise types and jamming signals, and at various positions of the wake word. We summarise all evaluated scenarios in Table 1 shown in the Appendix. In the following paragraphs, we detail the experimental setup and then discuss the results of the 8 different experiments sequentially.

The experiments vary in the type of word fed to the system ('wake word' or 'non-wake word'), the presence and type of background noise, and the noise's temporal position relative to the wake word. The noise types used for the evaluation are cafe background noises sourced from [12] and ambience-coffee-shop noise sourced from [12]. The jamming signals used for evaluation are manually generated ping noises, limited to a 0-8kHz bandwidth with an SNR of 0dB, and varying lengths of 2ms, 16ms, 64ms, and 128ms. We also used other types of jamming signals, such as keyboard noise and white noise. However, we do not include these results as the ping noise was the most effective signal. The non-wake words are sourced from [14].

The table displays the model's predictions and it also shows the number of times an attack was successful for each model. The classification performance was measured by the number of successful attacks. Additionally, the DOA position is analyzed, indicating the



(a) Short time energy graph of wake word in the presence of 16ms ping noise showing a peak in the sensitive area



(b) Short time energy graph of the wake word in the presence of cafe noise with no peaks

Figure 4: Comparison of STE graphs.

direction of the incoming audio. If the wake word has a jamming signal arriving from a different direction, the DOA position changes accordingly. The table also shows the STE peak position, which indicates the temporal energy peak in the audio signal, providing insights into the position of the jamming signal.

The LSTM_WD model is divided into two categories and shows the number of times the model correctly identified the input audio as either a wake word (WW), a non-wake word (NWW). The LSTM_MCC model which was adapted into a multi-class classifier shows the number of times the model correctly identified the input audio as either a wake word (WW), a non-wake word (NWW) or an attack signal(WW+Jammer). The numbers in the table reflect the count of each type of classification made by the models over 50 experiment runs.

6.1 Experiment 1

Experiment 1 is used to establish a baseline. No attack is present. A wake word is fed 50 times to both models. Both models classify the input as wake word and STE and DOA do not register.

6.2 Experiment 2

Next we test the models' ability to detect the wake word in a noisy environment. We use a cafe background noise mixed with the wake word. Again no attack is present. The models are expected to remain

unaffected by such background noises, however, when the wake word is played along with random cafe background noises, the LSTM_WD model failed to detect the wake word in 10 out of 50 trials. The LSTM_MCC model identifies all 50 trials as wake word. This shows that training to cope with attacks also helps in noisy environments. However, it mistakenly identified the background noise as a potential attack (wake word with jamming noise) in 8 out of the 50 trials. To mitigate this issue of false positives and to avoid raising a false alarm, either DOA position or STE peak position can be considered. The DOA position showed that the change in direction happened after 240ms of the wake word which does not fall in the sensitive area of the wake word. The change in the angle could be because the background noise was more perceivable to the microphones than the wake word at 240ms. This position is a silent area of the wake word and an attacker would not be successful. The STE of the signal does not exhibit a peak in this scenario, in particular not within the sensitive region of the wake word. An example is shown in Figure 4b. After applying either DOA or STE the 8 attack classifications of the LSTM_MCC model can be reduced to 0. This conclusion can be supported by the number of successful attacks column presented in the table.

6.3 Experiment 3

From Experiments 3 to 6, the models are tested by injecting different jamming signals into the wake word recognition. The jamming signals used here are ping noises in the frequency range 0-8 kHz with the same power as the wake word signal(SNR 0dB) with duration 2ms, 16ms, 64ms and 128ms. In Experiment 3, the jamming signal of 64ms duration is played at the start of the wake word and this had no impact on the wake word detection for LSTM_WD and LSTM_MCC models as the jamming signal is not placed within the sensitive areas of the wake word. As shown in Table 1 all 50 samples are classed as wake word by both models. The LSTM_MCC does not identify this scenario as an attack. While an attack is present, it is not a successful one and therefore the LSTM_MCC's output is as desired.

6.4 Experiment 4

In Experiment 4, the jamming signal of 128ms is played with a delay of 100ms after the start of the wake word (in the sensitive area of the wake word), and the delay is increased by 1ms each for 50 iterations. The LSTM_MCC model detected the jamming signal in the wake word and identified the input signal as an attack signal. Despite the attack, the wake word is still recognized and the user request is processed. Whereas the LSTM_WD model failed to detect the wake word exposing the PVA to a DoS attack.

The DOA position shows that there was a change in direction around approximately 100ms-150ms after the start of the wake word for all 50 cases. Figure 5 shows the change in the direction of arrival when a ping noise of 128ms was injected in the sensitive area of the wake word. The STE peak position also shows that there was an energy peak found in the sensitive areas of the wake word. This indicates that there was a deliberate jamming signal inserted by an attacker in the sensitive areas and the model's predictions about the attack are true. Considering the DOA positions and STE peak positions, attack classifications of the LSTM_MCC are confirmed.

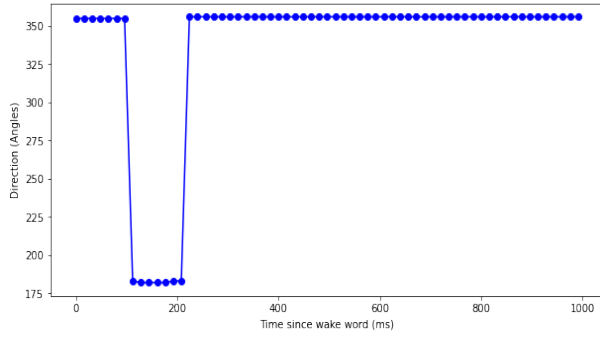


Figure 5: Direction of arrival when wake word audio, was mixed with ping noise of 128ms long, displaying a change in the angle of direction in the sensitive area if the wake word.

6.5 Experiment 5

In Experiment 5 the jamming signal of 16ms duration is used and the LSTM_MCC model identified the input as a wake word 8 times and as an attack signal 42 times for a total of 50 trials. In the same scenario, the LSTM_WD model identified the input as a wake word only 6 times, and for the remaining, the input was classified as a non-wake word leading to an attack in all 44 cases.

The DOA position shows a change in the direction approximately around 100ms-150ms after the start of the wake word and the STE peak position shows that there is a sudden change in energy around 100-150ms after the start of the wake word across all 50 experimental runs. Figure. 4a illustrates the STE graph when a jamming signal of 16ms was introduced exactly at 110ms of the wake word. The figure shows a sudden spike in the audio signal's energy when injected with a jamming signal. Since the DOA and STE results align with model's predictions about the attack, we can be sure that the input audio indeed had a jamming signal. Hence the number of successful attacks using LSTM_MCC model are zero.

6.6 Experiment 6

In Experiment 6, the jamming signal duration is reduced to 2ms. The LSTM_MCC model identifies the input as a wake word 43 times and as an attack signal 7 times while the LSTM_WD model identifies it as a wake word 41 times and as a non-wake word for the rest of the trials. In this scenario, the present attack is only identified in a rare 7 cases. While this is low, it has to be noted that the attack is not very successful to begin with. The standard LSTM_WD model is only impacted in 9 out of 50 cases. It can be argued that unsuccessful attack attempts should not raise an alarm. In cases where the LSTM_MCC model identifies an attack, either the DOA position or the STE peak position can be considered as both are displaying a possibility of attack as per the results noted in the table.

6.7 Experiment 7

In Experiment 7, the models were tested with only cafe background noise to determine if adversarial training caused the LSTM_MCC model to mistakenly identify random noises as attacks, given the use of noises during training. The results indicate that the model

successfully distinguished between jamming signals and random background noises. All 50 samples are classified as non-wake words.

6.8 Experiment 8

Experiment 8 demonstrates the models' performance with non-wake words. As expected, the models were able to distinguish non-wake words across all 50 samples. We do not demonstrate continuous jamming here because it can clearly interrupt wake word detection and is easily noticeable to users as a jamming attack.

7 Discussion

Wake word jamming attacks can be detected and mitigated by transforming a wake word model using adversarial training into a multi-class classifier along with techniques such as DOA and STE. However, it is important to decide which techniques work best in combination with the wake word detection model to detect jamming efforts. As per the results, there are few instances where the multi-class classifier identifies an input as an attack while it's just some background noise. To avoid this DOA can be used as an additional check. DOA is a simple and computationally efficient technique since most Personal Voice Assistants (PVAs) are equipped with microphone arrays that can calculate the angle of arrival. However, relying solely on DOA might lead to incorrect jamming detection because it is sensitive to noise.

To ensure the accuracy of attack detection, especially for PVAs without a microphone array, such as those in mobile phones with only a single microphone, an STE estimation can be considered. STE, which is also computationally simpler, detects changes in signal energy over time and shows peaks during abrupt energy changes. Using STE along with the multi-class classifier's predictions improves jamming detection accuracy, as STE does not show peaks for random background noises as shown in Figure. 4b. If an STE peak appears in the sensitive areas of the wake word, it indicates an attack. Therefore, combining STE with the classifier's predictions is a more reliable method for detecting jamming attacks on a wake word model.

8 Conclusion

PVAs have become very popular with their increase in application in critical sectors. So it is important to consider the security of these devices. It is easy for an attacker to use the acoustic channel of a PVA to execute DoS using wake word jamming techniques, once they have obtained control of any nearby device with a speaker. We have shown that the wake word detection algorithm can be modified such that jamming attacks are detected and the user can be alerted. To avoid false positives, we have shown that multi-class classification along with STE and DOA can be used. We further evaluated the most effective combination between DOA and STE and discovered that STE is more suitable to use alongside the model's predictions.

ACKNOWLEDGMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 19/FFP/6775 and 13/RC/2077_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

- [1] Peng Cheng, Ibrahim Ethem Bagci, Jeff Yan, and Utz Roedig. 2018. Towards reactive acoustic jamming for personal voice assistants. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*. Toronto, ON, Canada, 12–17.
- [2] Dong Enqing, Liu Guizhong, Zhou Yatong, and Cai Yu. 2002. Voice activity detection based on short-time energy and noise spectrum adaptation. In *6th International Conference on Signal Processing, 2002.*, Vol. 1. 464–467 vol.1. <https://doi.org/10.1109/ICOSP.2002.1181092>
- [3] Yixin Gao, Yuriy Mishchenko, Anish Shah, Spyros Matsoukas, and Shiv Vitaladevuni. 2020. Towards data-efficient modeling for wake word spotting. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain, 7479–7483.
- [4] Jinxi Guo, Ken'ichi Kumatani, Ming Sun, Minhua Wu, Anirudh Raju, Nikko Strom, and Arindam Mandal. 2018. Time-Delayed Bottleneck Highway Networks Using a DFT Feature for Keyword Spotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*. IEEE, Calgary, Canada, 5489–5493. <https://doi.org/10.1109/ICASSP.2018.8462166>
- [5] C Junu Jahana, MS Siniith, and PP Lalu. 2021. Direction Of Arrival Estimation using Microphone Array. In *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*. 1–6. <https://doi.org/10.1109/ICMSS53060.2021.9673617>
- [6] Juncheng Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J. Zico Kolter, and Florian Metze. 2019. Adversarial Music: Real world Audio Adversary against Wake-word Detection System. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.)*. Vancouver, Canada, 11908–11918.
- [7] Hyungjun Lim, Younggwan Kim, and Hoirin Kim. 2020. Cross-Informed Domain Adversarial Training for Noise-Robust Wake-Up Word Detection. *IEEE Signal Process. Lett.* 27 (2020), 1769–1773. <https://doi.org/10.1109/LSP.2020.3026947>
- [8] Wolfgang Mack, Julian Wechsler, and Emanuel A.P. Habets. 2022. Signal-aware direction-of-arrival estimation using attention mechanisms. *Computer Speech & Language* 75 (2022), 101363. <https://doi.org/10.1016/j.csl.2022.101363>
- [9] Selma ÖZAYDIN. 2019. Examination of energy based voice activity detection algorithms for noisy speech signals. *Avrupa Bilim ve Teknoloji Dergisi* (2019), 157–163.
- [10] Prathyusha Sagi, Arun Sankar, and Utz Roedig. 2024. Adversarial training to prevent wake word jamming in Personal Voice Assistants. (2024).
- [11] Prathyusha Sagi, Arun Sankar, and Utz Roedig. 2024. Personal Voice Assistant Wake Word Jamming. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 19–24. <https://doi.org/10.1109/PerComWorkshops59983.2024.10503529>
- [12] stephan. 2022. *The Babel Noise Generator*. <https://pixabay.com/sound-effects/search/coffee-shop/>
- [13] Ke Sun, Chen Chen, and Xinyu Zhang. 2020. "Alexa, Stop Spying on Me!": Speech Privacy Protection against Voice Assistants. In *SenSys '20: The 18th ACM Conference on Embedded Networked Sensor Systems, Virtual Event, 2020*. ACM, New York, United States, 298–311. <https://doi.org/10.1145/3384419.3430727>
- [14] Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *CoRR abs/1804.03209* (2018). arXiv:1804.03209 <http://arxiv.org/abs/1804.03209>
- [15] Qiuchen Yu and Ruohua Zhou. 2022. Wake Word Detection Based on Res2Net. *arXiv preprint arXiv:2209.15296* (2022).

Appendix

Table 1: Performance validation of the LSTM_MCC model using wake word (WW), non-wake word (NWW), and wake word with noise (WW+Jammer) samples. Additionally, the model is compared with the LSTM_WD model to demonstrate its superior performance.

Experiment	Word	Noise	Position of Noise Over Wake Word(ms)	Expected Output	LSTM_WD Outputs		LSTM_MCC Outputs			DOA Positions	STE Peak Position	Number of Successful Attacks Considering DOA and STE	
					WW	NWW	WW	NWW	WW+Noise			LSTM_WD	LSTM_MCC
1	Wake Word	No Noise	NA	WW	50	0	50	0	0	No Change in Direction	No Energy Peak	0	0
2	Wake Word	50 Different Cafe Background Noises	0 (At the Start of Wake Word)	WW	40	10	42	0	8	~at 240 ms (Changes Direction in the Silent Regions of the Wake Word)	No Energy Peak	10	0
3	Wake Word	PN SNR0 BW 0-8000 64ms	0 (At the Start of Wake Word)	WW	50	0	50	0	0	~at 0 ms (At the Start of the Wake Word)	0 (At the Start of Wake Word)	50	0
4	Wake Word	PN SNR0 BW 0-8000 128ms	100-150	WW+Jammer	0	50	0	0	50	~around 100ms-150ms (Sensitive Area)	100ms-150ms (Sensitive Area)	50	0
5	Wake Word	PN SNR0 BW 0-8000 16ms	100-150	WW+Jammer	6	44	8	0	42	~around 100ms-150ms Sensitive Area	100ms-150ms (Sensitive Area)	44	0
6	Wake Word	PN SNR0 BW 0-8000 2ms	100-150	WW+Jammer	41	9	43	0	7	~around 100ms-150ms Sensitive Area	100ms-150ms (Sensitive Area)	9	0
7	Non Wake Word	Ambience-coffee-shop(2sec)	NA	NWW	0	50	0	50	0	No Change in Direction	No Energy Peak	Not an Attack	Not an Attack
8	Non Wake Word	50 Different Non-Wake Words	NA	NWW	0	50	0	50	0	No Change in Direction	No Energy Peak	Not an Attack	Not an Attack