Adversarial Robustness of Lightweight Log-Mel Spoofing Detectors on ASVspoof and Wild Datasets: A Comparative Reproduction Study

Soukya koleti
23214490

Abstract

This work presents a systematic investigation into the adversarial robustness of lightweight log-mel-based spoofing detectors for automatic speaker verification (ASV). Although compact feed-forward models are widely deployed in resource-constrained ASV settings, their behaviour under adversarial manipulation has remained largely unexplored. In this study, we develop and evaluate a lightweight multi-layer perceptron (MLP) classifier using log-mel and complementary spectral features, and analyze its performance against white-box FGSM and PGD attacks on both the ASVspoof2019 LA dataset and a real-world WILD dataset. The proposed lightweight model attains strong baseline accuracy of over 97% on ASVspoof and 99% on WILD—but exhibits pronounced vulnerability when exposed to even moderate adversarial perturbations, including sharp rises in Equal Error Rate (EER) and significant class imbalance under attack. Comparative analysis shows that adversarial fragility persists even with architectural simplifications, emphasizing the need for more robust feature representations and adversarial-aware training strategies for practical ASV deployments, especially on edge devices.

To contextualize and validate our findings, we reproduce prior CNN-based experiments using LCNN-big, LCNN-small, and SENet12 models under identical FGSM and PGD attack settings. The reproduced results align with earlier reports, confirming that high-capacity models show slightly improved tolerance to mild perturbations yet suffer similar collapse under stronger attacks. This reproduction provides a consistent baseline against which the weaknesses of lightweight detectors can be clearly identified. Overall, the study highlights the critical robustness gaps in low-complexity spoofing countermeasures and offers guidance for developing more resilient ASV frameworks.

1 Introduction

1.1 Background and Motivation

Automatic speaker verification (ASV) technology identifies users by their distinct vocal features and has wide application in banking, security, forensics, and computer-assisted systems. In spite of the convenience, ASV systems are susceptible to spoofing attacks, including synthetic speech and voice transformations, and replay attacks. These attacks may be used to bypass ASV systems and cause a security breach. In order to alleviate such threats, spoofing countermeasure (CM) models have been created to differentiate between bona fide speech and spoofed audio (Kinnunen and Sahidullah, 2017).

Recent research has dedicated its attention to deep learning models, specifically convolutional neural networks (CNNs) and self-supervised learning (SSL) designs, which generate time-frequency features on audio signals (Salih et al., 2025). Such models are very accurate on benchmarking datasets like ASVspoof, yet very susceptible to adversarial attacks: even small and unnoticeable artificially introduced changes can severely impact the model (Liu, Wu, Lee, and Meng, 2019).

Lightweight models, usually log-mel spectrogram-based and built on shallow neural networks (e.g., multi-layer perceptrons), show a significant presence in practical ASV deployments because they are inexpensive in computation and do not incur the memory footprint of more complex models. Nevertheless, their resistance to adversarial attacks is a relatively unexplored quality (Khan, Malik and Ryan, 2022).

1.2 Research Gap

The deep CNN-based countermeasures have been widely studied concerning white-box and black-box attacks in the past (Liu, Wu, Lee, and Meng, 2019; Wu, Liu, Meng, and Lee, 2020). These studies showed high susceptibility to gradient-based attacks and partial success for transfer-based black-box attacks. Conversely, lightweight spectral-feature-based models, which are typical of resource-constrained ASV systems, have not been actively tested in adversarial settings. Also, most evaluations rely on controlled datasets, leaving the impact of real-world variability unexplored.

1.3 Research Questions

The study addresses the following questions:

1. How vulnerable is a lightweight log-mel feed-forward classifier to white-box adversarial attacks, specifically FGSM and PGD, across different perturbation strengths?

2. How does adversarial susceptibility differ between the structured ASVspoof2019 LA dataset and the unconstrained real-world WILD dataset when evaluated using identical lightweight architectures?

3. How do the observed robustness characteristics of the lightweight model compare with the behavioural patterns reported in prior CNN-based studies such as Liu et al. (2019)?

4. How do FGSM and PGD perturbations influence the model's performance metrics—including accuracy, EER, FAR, FRR, and attack success rate—across both datasets?

1.4 Research Objectives

The objectives are:

- To reproduce Liu et al. (2019) experiments on CNN-based spoofing countermeasures and verify their findings.

- To implement a lightweight log-mel–based feed-forward spoofing detector for ASVspoof and WILD datasets.
- To evaluate the vulnerability of the lightweight model under FGSM and PGD white-box attacks.
- To perform a comparative analysis between reproduced CNN-based models and the lightweight detector.
- To discuss implications for practical ASV deployments and propose directions for improving adversarial robustness.

2 Literature Review

2.1 Spoofing Attacks on ASV and Countermeasures

Automatic speaker verification (ASV) voice biometric systems have become popular as authentication since they are convenient and need no secret (e.g., a password) or token possession. However, these systems are susceptible to spoofing attacks whereby an attacker tries to pretend to be a legitimate user by using synthesized speech, voice conversion, or a pre-recorded audio message. To address this, the community has developed spoofing countermeasures (CMs) or presentation attack detection (PAD) systems designed to distinguish between bona fide (genuine) and spoofed speech. Early anti-spoofing methods focused on acoustic or cepstral feature extraction (e.g., linear prediction residuals, CQCC, MFCC) combined with statistical or shallow classifiers such as GMM (Kinnunen & Sahidullah, 2017).

Deep-learning-based countermeasures became prevalent as speech-generation technologies (TTS, voice conversion, replay) advanced. Methods based on spectrograms, constant-Q transforms, and other time-frequency representations were used as inputs to convolutional neural networks (CNNs) and other deep learning systems, achieving excellent performance on benchmark datasets such as those from the ASVspoof Challenge (Wu et al., 2017). Recent summaries indicate a dominance of deep learning and end-to-end systems, particularly for synthetic speech and replay detection (Kinnunen and Sahidullah, 2020).

However, simpler feature-based detectors remain relevant due to their computational efficiency and suitability for edge or embedded devices. Thus, model complexity carries a trade-off with resource demands: deep models achieve high accuracy under controlled conditions, whereas lightweight models remain attractive where computing power or memory is limited.

2.2 Adversarial Attacks in Audio and Their Application to ASV Spoofing Countermeasures

Adversarial attacks—small, deliberately crafted perturbations that are imperceptible to humans yet capable of fooling machine-learning models—were first studied extensively in computer vision. More recently, similar threats have been demonstrated for audio processing systems, including automatic speech recognition (ASR), speaker recognition, and ASV.

Liu, Wu, Lee, and Meng (2019) conducted the first systematic study of adversarial robustness in ASV spoofing countermeasures. They evaluated several high-performing CM models, including LCNN-big, LCNN-small, and SENet12, under white-box and black-box attacks using the fast gradient sign method (FGSM) and projected gradient descent (PGD). Their findings showed that all models were highly vulnerable: even very small perturbations significantly reduced detection accuracy, and black-box (transfer) attacks were also effective (Liu et al., 2019).

This work highlighted severe vulnerabilities in state-of-the-art spoofing detectors, particularly those built under non-adversarial assumptions. Although these models achieve high accuracy in standard evaluation environments, they can collapse dramatically when exposed to adversarially perturbed audio.

2.3 Defense Efforts Against Adversarial Attacks in Spoofing Detection

Following the discovery of these vulnerabilities, researchers proposed several defense strategies. Wu, Liu, Meng, and Lee (2020) introduced two categories of defenses: proactive (adversarial training) and passive (spatial smoothing). Adversarial training exposes the model to adversarial examples during training, while spatial smoothing reduces sensitivity to perturbations by filtering input features. These defenses were found to partially improve model robustness (Wu et al., 2020).

More recent defense approaches have used self-supervised learning (SSL) representations to improve resistance to black-box attacks. For example, high-level SSL-based representations reduce the transferability of adversarial examples when compared to traditional spectral features (Wu et al., 2020b). Other efforts involve adversarial-sample detection, such as re-synthesizing audio with neural vocoders to detect inconsistencies in ASV scores (Wu, Hsu, Lee, Gao et al., 2022).

Despite these advances, defenses remain partial and introduce trade-offs in model complexity, computational overhead, and potential degradation of performance on clean audio. This is especially problematic in resource-limited environments where heavy defenses are not feasible.

2.4 The Gap: Lack of Adversarial Evaluation for Lightweight / Feature-Based Countermeasures

Most adversarial robustness research in ASV focuses on deep, high-capacity models such as CNNs. The seminal Liu et al. (2019) study and subsequent defense work (Wu et al., 2020; Wu et al., 2022) also concentrate on these architectures. Meanwhile, simpler feature-based detectors—such as those using log-mel spectrograms, CQCC, or linear prediction residuals—remain commonly deployed in resource-constrained ASV systems.

However, very little published work examines adversarial vulnerability in these lightweight models. Virtually no studies have systematically compared their robustness with deep models across both controlled datasets (e.g., ASVspoof) and real-world audio.

This absence of research creates a serious blind spot. Lightweight detectors are widely used in real-world systems (edge devices, banking kiosks, embedded voice authentication). Without adversarial evaluation, their true security risk remains unknown.

2.5 Recent Developments and Broader Threat Models: Toward Realistic Attack Scenarios

Beyond traditional white-box spectrogram-based attacks (e.g., FGSM, PGD), recent work has expanded to more realistic threat models. Malafide, a novel adversarial convolutive noise attack (Panariello, Ge, Tak, Todisco & Evans, 2023), introduced a time-invariant linear filter optimized per spoofing attack type. This perturbation is independent of utterance length, requires no gradient access during inference, and significantly degrades CM performance under black-box settings while maintaining naturalness of speech (Panariello et al., 2023).

Other studies, such as "Practical Attacks on Voice Spoofing Countermeasures" (Kassis and Hengartner, 2021), show that attackers do not need to rely solely on synthetic speech: time-domain adversarial perturbations can be constructed to bypass modern detectors. These attacks demonstrate high success rates even against state-of-the-art systems, posing serious risks to end-to-end ASV security (Kassis & Hengartner, 2021).

On the defensive side, models such as "Improving the Adversarial Robustness of Speaker Verification by Self-Supervised Learning" (Wu, Li, Liu, Wu, Meng and Lee, 2022) leverage SSL-based representations that are more robust to transfer-based adversarial examples than spectral or shallow features (Wu et al., 2022). Detection-based defenses, such as neural-vocoder re-synthesis, have also been explored.

These developments reveal that the threat landscape is rapidly evolving. Universal filters, time-domain perturbations, and sophisticated black-box attacks challenge the assumptions of existing countermeasures. However, nearly all such research focuses on deep, resource-intensive models, leaving a major gap in understanding the security of lightweight detectors used on edge and embedded systems.

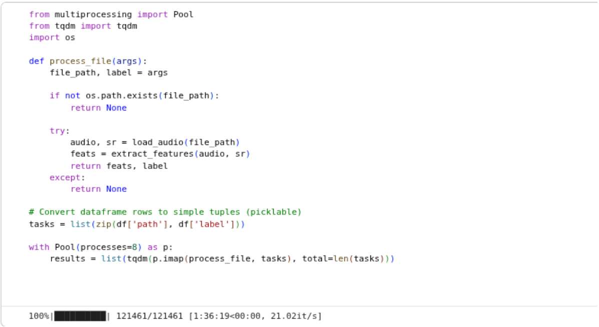
3 Methodology

3.1 Overview of the Research Design

The experiment is structured in an experimental workflow methodology aimed at exploring the adversarial strengths of synthetic audio detection systems. The workflow is composed of two pipelines that run parallel and were constructed on the basis of the ASVspoof2019 dataset and the WILD dataset. The two pipelines are supplied with the features of log-mel spectrograms, a lightweight feedforward neural classifier, and similarity in training and evaluation. ASVspoof pipeline incorporates a few extra steps in the production of adversarial examples with the help of FGSM and PGD methods, which provide the detection model with the chance to be tested under white-box threat conditions. The WILD pipeline concentrates on baseline and perturbed analyses to measure the stability of models on unconstrained audio in the real world. The methodology used is in line with the research objectives, namely, determining the presence adversarial vulnerabilities in synthetic audio detection models, assessing the impact of adversarial examples, and comparing the robustness of different datasets with varying structural properties.

3.2 The processing of the ASVspoof dataset

The ASVspoof system is based on log-mel spectrogram features that have been previously extracted in NumPy arrays. Bona fide and spoofed utterances are represented separately in arrays. These arrays are loaded directly, summed into a single feature array and matched with binary labels, with 0 representing bona fide speech and 1 representing speech that was spoofed. The combined data is divided into training and testing data. Since the features are represented as two dimensional spectrograms of different lengths, each sample is reduced to a fixed length single dimensional vector to be input to dense neural networks.



```
from multiprocessing import Pool
from tqdm import tqdm
import os

def process_file(args):
    file_path, label = args

    if not os.path.exists(file_path):
        return None

    try:
        audio, sr = load_audio(file_path)
        feats = extract_features(audio, sr)
        return feats, label
    except:
        return None

# Convert dataframe rows to simple tuples (picklable)
tasks = list(zip(df['path'], df['label']))

with Pool(processes=8) as p:
    results = list(tqdm(p.imap(process_file, tasks), total=len(tasks)))
```

```
100%|██████████| 121461/121461 [1:36:19<00:00, 21.02it/s]
```

Figure 1: Code Screenshot of ASVSpoof Audio preprocessing

All ASVspoof experiments are performed in a compact feedforward structure. This model comprises of one hidden layer with Relu activation and a binary classification output node which is a sigmoid node. This is a designed lightweight architecture to give a clear view of the effects of adversarial perturbations on the learned decision boundary. The binary cross-entropy loss, and the Adam algorithm with the learning rate provided in the code are used to optimize. There are several epochs of training by mini-batches of reshaped feature vectors. The model in every epoch performs batch-level predictions, calculates the loss, and updates the parameters by applying the backpropagation technique. The training loop logs the loss and accuracy data to be analyzed later and calculates the validation accuracy at the end of every epoch. This is done to create the baseline model upon which adversarial evaluation is done later.

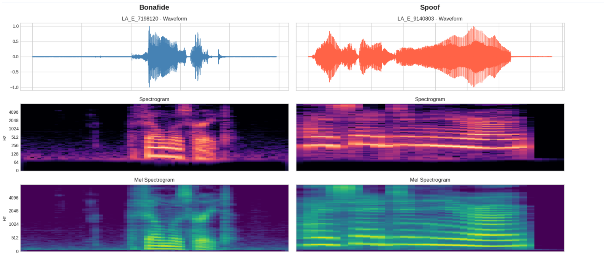


Figure 2: Spoof and and Bonafide Audio Visualization

3.3 WILD Dataset Processing

WILD data does not provide pre-calculated features and instead gives raw waveforms. The audio files are loaded in two directories that are real and synthetic speech. All files are resampled to a uniform sampling rate and normalised. Each waveform is then calculated on log-mel spectrograms with a fixed number of mel bins converted to decibel scale and normalized. Since recordings are not uniform in length, all spectrograms are padded or truncated to some standard length to make them have the same dimensions. Flattening of final spectrograms occurs as in the case of ASVspoof, into one-dimensional vectors. The same feedforward classifier architecture that was trained on the WILD features is trained on the ASVspoof pipeline. The model is fed with flattened spectrogram vectors with binary labels of real or synthetic audio. The optimization and loss settings of training are equivalent to the ASVspoof classifier, making it comparable across datasets. Assessment is done on a different test subset to achieve baseline performance prior to the introduction of adversarial perturbations.

3.4 Adversarial Attacks Generation.

FGSM perturbations are produced by calculating the loss gradient with respect to every input input feature. The value of this gradient is multiplied by a predefined epsilon and added to the input to generate an adversarial sample. Such distorted inputs are then fed on the trained classifier to determine whether the perturbation misclassifies. This is performed on all the test samples and the predictions obtained measure the susceptibility of the model to single step gradient based attacks. The development of PGD adversarial examples is based on the gradient update process. Loss gradient is calculated at every step, and multiplied by a small step size before being added to the perturbed sample. The new sample is reverted to a bound e-region of the initial input in order to keep the perturbation limit specified. The obtained adversarial feature vectors are processed after a few repeats and put through the classifier. Much more precise and targeted perturbations than FGSM can be more finely evaluated by this attack, and the assessment of adversarial susceptibility is more thorough.

3.5 Performance Indicators and Measures.

The two datasets are compared based on the classic metrics obtained based on the outputs of classification procedures such as accuracy, precision, recall and F1-score of individual classes. These measures are used to measure the capability of the classifier to differentiate between bona fide and synthetic audio both in clean and adversarial cases. In the case of the ASVspoof experiments, other measures, including Equal Error Rate (EER) and latency per sample are also added. EER shows the equilibrium between false acceptance and false rejection hence it is especially applicable when it comes to authentication. In the case of ASVspoof, the metrics of the baseline models are directly compared with model performance when FGSM and PGD perturbations are used. This comparison separates the adversarial noise effect on the class-specific behavior and detection reliability in general. In the case of the WILD, the baseline model is compared with variants tested in case of PGD and FGSM perturbations so that it is possible to measure the effects of adversarial in less controlled audio conditions. Collectively, these assessments respond to the research questions that are related to model robustness in datasets and threat models.

4 Design and Specification

4.1 System Architecture Overview

The system architecture is designed in such a way that it allows a direct comparison between the low-resource models with the established deep learning benchmarks. In a bid to respond to the research questions on the trade-off between efficiency and security, the design establishes two parallel processing pipes. The initial pipeline is on the suggested Lightweight Multi-layer perceptron (MLP), which is developed in combination with resource resource-constrained or embedded environment. The second pipeline is an exact imitation of the Deep Convolutional Neural Networks (CNN), namely LCNN and SENet, that have been originally introduced by Liu et al. (2019). The study is able to manipulate architectural complexity as a variable by operating these two pipelines in parallel and quantifying its effect on adversarial robustness. The general flow of the data is logical and has four stages of progress. It starts with a data ingestion module that is able to support structured and unstructured audio. A step of feature extraction is then taken that transforms raw waveforms into fixed-length vectors to be presented to the lightweight model. This is then input into the model training component, which assists in normal training on clean data, as well as adversarial training on perturbed inputs. Lastly, the assessment module exposes all trained models to white-box attacks in order to measure the performance degradation. This design allows any perceived differences in robustness to be explained by the model structures per se, as opposed to data preprocessing discrepancies.

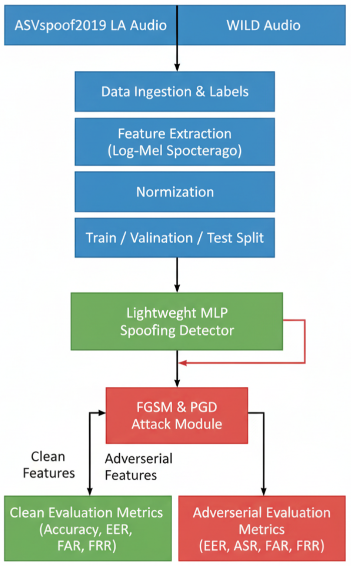


Figure 3: Design Architecture

4.2 Data Processing and Feature Engineering

To test the ability of the models to generalize, the design uses two datasets that have different acoustic profiles. The controlled environment is the ASVspoof 2019 Logical Access (LA) dataset. In this stream, the system will process the official protocol files to index file paths to labels in strict compliance with the given training, development, and evaluation splits to make the system reproducible. By contrast, the WILD data is realistic and uncontrolled natural acoustic surroundings. There is no structure to this data, so to combine scattered audio files into a coherent metadata table, a custom preprocessor script is used. This two-stream system enables the system to test the resilience of the lightweight model on both clean, synthetic attacks and the noise that exists in the environment in actual-world scenarios. The choice of feature extraction was strongly influenced by a factor of computational efficiency. In the case of the ASVspoof pipeline, they are Mel-frequency cepstral coefficients (MFCCs) with their first and second-order derivatives (delays and delta-delays). The effect of this is a 60-dimensional array averaged over the time domain array. The choice of MFCCs was based on the fact that they provide a compact representation of the spectral envelope, which has been known historically to be effective when used in lightweight voice anti-spoofing. In the case of Of WILD datasets pipeline, the feature set is augmented with Linear Frequency Cepstral Coefficients (LFCCs), log-mel energies, as well as spectrogram magnitudes. This multidimensional representation is essential to allow the representation of small artifacts that can be drowned by the noise in the uncontrolled environment. After extraction, the total numbers of feature vectors are globally normalized using statistics obtained only on the training partition. This step is essential to the stability of the model since it avoids the explosion of the gradients when training the neural networks.

4.3 Lightweight Model Architecture.

Designing a spoofing detector that can be used on devices of limited computing power is the main contribution of this work. The architecture suggested is a Feed-Forward Neural Network (FFNN) that aims to be both more efficient in terms of parameters and more accurate in classification. The feature vector in 60 dimensions is inputted into the network and sent through a sequence of three dense layers. The initial two concealed layers have 512 and 256 units, respectively, each preceded by Batch Normalization to stabilize the learning process and a ReLU activation element to bring about non-linearity. The features are then shrunk on a third layer of 64 units for the last classification. In order to avoid the model learning the noise in the training, which is a typical problem in smaller networks, dropout with a 50% dropout rate is performed after each hidden layer. Moreover, the design also specifically deals with the imbalance in classes being present in the ASVspoof dataset. Instead of using mere accuracy, the system uses a weighted Binary Cross-Entropy (BCE) loss. The positive weight of the classes is dynamically determined according to the ratio of bona fide to spoof samples of the training set. This makes sure that the model is not biased in its predictions towards the majority class, and the baseline performance is stable before the adversarial testing starts.

4.4 Reference Models (Liu et al., 2019)

In order to certify the performance of the lightweight model, one will have to compare it with a certified upper bound. Therefore, this system recreates the heavyweight architectures as specified in the reference study by Liu et al. (2019). This consists of the Light CNN (LCNN) variants, which use Max-Feature-Map (MFM) activation functions to learn feature selection explicitly and SENet12, a ResNet-based architecture that incorporates Squeeze-and-Excitation blocks. The model is not the major interest of the innovation but rather vital controls. The exposure of these complex models to the same attack patterns as the lightweight MLP can allow the study to conclude whether the shallow structure of the MLP predisposes it to adversarial perturbations compared to the deep convolutional architectures.

4.5 Threat Model and Adversarial Generation.

The security testing presupposes a White-Box threat model. In this case, it is assumed that the adversary knows the full information about the model architecture, model parameters and gradients. Although this is an extreme case, it is the most severe test of the basic strength of a security system. When a model survives white-box attacks, it is usually believed that it will be resistant to weaker black-box attacks. The system uses two types of gradient-based algorithms to produce these attacks. The former is the Fast Gradient Sign Method (FGSM), which follows the gradient in one step to cause perturbation to the input. This is the test of the linearity of the model. The second one is an iterative and far more powerful version of FGSM and is called Projected Gradient Descent (PGD). In the case of PGD, the system is run with smaller perturbation steps repeated with a fixed limit (epsilon) on the amount of change. Most importantly, these attacks are generated on the feature-vector level as opposed to the raw waveform level. The method effectively isolates the strength of the decision space of the classifier and distorts it between the possible artifacts of signal processing.

4.6 Evaluation Framework

The evaluation program is constructed in such a way that it does not just rely on the metric of accuracy but offers a finer analysis of security vulnerability. The most important performance measure is the Equal Error Rate (EER) which determines the operating point at which the False Acceptance Rate (FAR) matches the False Rejection rate (FRR). This is the usual measure of biometric security and can be directly compared to the results of the ASVspoof challenges. In the evaluation of adversarial robustness, the system determines Attack Success Rate (ASR), which is a percentage of adversarial examples that are able to deceive the detector. The analysis also quantifies the cost of robustness in terms of the percentage fall in the EER when the model is transferred to the adversarial data. The sequence of the experimental process is to train all the models using clean data to create a baseline, adversarial probing with FGSM and PGD. Lastly, the lightweight MLP is trained adversarially, i.e., including attacks in the training set, to find out whether the model can be trained to be resilient to such perturbations without compromising its performance on real audio.

5 Implementation

5.1 Preprocessing of Data and Label

When using the ASVspoof2019 LA dataset, the implementation stage starts with a process of ingesting protocol files, which contain the audio file paths and their corresponding speaker IDs, system IDs, and ground-truth labels. These separate indices are then merged into a single metadata dataframe, which still maintains the structural integrity of the training, validation, and evaluation partitions. The resultant dataset contains 121,461 records, which form a sufficient basis on which adversarial analysis can be conducted. To encode the target variables, categorical labels are converted to binary integers, and analysis of the class distribution is done to measure the imbalance in the bona fide and spoofed samples. This asymmetry is also extremely important, and positive weight parameters are calculated in the future in the loss function to avoid bias in the model. WILD data Preprocessing of the WILD data is based on a similar workflow, but on data measured in uncontrolled acoustic environments. Metadata is also aggregated, and though the distribution of classes in WILD is more balanced by its nature, the preprocessing pipeline imposes the same structural formatting as the one used in ASVspoof. To guarantee the valid comparative analysis as per Research Question 2 (RQ2), both datasets are statistically normalized. The average and the standard deviation are determined using only the training partition and then used in the validation and the evaluation sets. This sharpness of separation provides data leakage; the measures of performance of the model indicate the true generalization capabilities and not the memory of the global statistics. Moreover, stratified sampling is also used in the process of data loading to ensure that the proportion of classes remains the same among batches. These stringent preprocessing checks form a valid foundation for the later adversarial analysis, and these ensure that the noted changes in model resilience can be credited to algorithmic behavior instead of data handling artifacts.

5.2 Implementation of Feature Extraction

The extraction of features in ASVspoof data is performed with a multiprocessing routine in order to handle the large number of audio files. The system calculates Mel-frequency cepstral coefficients (MFCCs) with 20 fixed coefficients and with first and second-order derivatives of the coefficients (deltas and delta-deltas) added to them. These matrices are summed over the time axis and pooled together around the axis of features, creating a small 60-dimensional vector in each instance of an audio sample. Such a tradeoff in time granularity against computational efficiency is chosen intentionally as a part of the study, which is to create lightweight detection systems. Tests of this extraction pipeline on a typical CPU gave a throughput of about 21 files/second, which indicates the appropriateness of the approach to a resource-constrained environment. By comparison, the implementation of the WILD dataset requires a richer set of features to accommodate the background noise and acoustic variability of real-world recordings. The WILD extraction pipeline is a concatenation of MFCCs, Linear Frequency Cepstral Coefficients (LFCCs), log-mel energies, and spectrogram magnitudes into a high-dimensional vector. Although this composite feature method has a higher computational cost than the ASVspoof pipeline, it has the rich descriptive potential to discriminate artifacts in uncontrolled settings. This increased performance on the baseline in the WILD experiments indicates that this space of features is able to capture the fine spectral anomalies that are indicative of spoofing and compensates the noisy input data.

5.3 Lightweight Model Implementation

The basic lightweight classifier is based on a Multi-Layer Perceptron (MLP) which is run on PyTorch. The dynamic size of the input layer is similar to the dimensionality of the feature vectors of either of the two datasets. The architecture itself is a series of fully connected blocks where the first one takes the inputs to 512 units, then it is succeeded by batch normalization to stabilize the learning process, then a ReLU activation function, and a dropout layer with a probability of 0.3, which prevents overfitting. Categories are successively reduced to 256 and 64 units, with the final layer being a single unit that produces raw logits that are used to produce binary classification. AdamW algorithm is used to optimize with a learning rate of 1e-4 to trade weight decay against convergence rate. The training process utilizes the function of BCEWithLogitsLoss that combines a sigmoid activation and binary cross-entropy loss. More importantly, the pos weight argumentation is filled with inverse class frequencies obtained in the pre-processing phase, which directly tackles the imbalance in the dataset. The model will be trained over 50 epochs, and the validation metrics will be calculated after every cycle to ensure that the model is not drifting. The application was very successful on clean data; the ASVspoof model reached an accuracy of about 97 percent with Equal error rate (EER) of 0.0264, and the WILD model reached an accuracy of about 99 percent. These findings support the effectiveness of the MLP architecture in combination with the structured feature engineering mentioned above.

5.4 FGSM and PGD Attack Implementation

The Fast Gradient Sign Method (FGSM) is applied by finding the gradients of the loss of the model with respect to the input feature vector, and not the model weights. These gradients are then perturbed by taking the sign of the gradients and scaling them by an epsilon parameter which is here 0.03. An adversarial example is an original input that has been perturbed to a small scale. This direct application is informed by the canonical formulations of adversarial learning which enables the study to isolate the sensitivity of the model to linear perturbation as demanded by RQ1. The generated adversarial inputs are directly recycled into the model to determine the drop in prediction confidence. The extension of the FGSM method into an iterative procedure called Projected Gradient Descent (PGD) is used to mimic a stronger adversary. This is done by having small perturbations that are introduced to the process as a step of 0.007 divided into 10 to 20 steps. The changed input is re-projected onto the L-ball around the original input at intensity epsilon to make sure the alterations are imperceptible (within the matrix of raw data) or constrained (within feature space). This refinement step is more intensive in searching the point of maximisation of the loss compared to FGSM. This harsher attack will give a more stringent stress test to the lightweight models, and meet the requirement of the robustness test of RO2.

5.5 Procedures of Training Adversarial Training

Adversarial training is applied by incorporating the attack generation logic into the training process. Each mini-batch of clean data is transformed on the fly with the system producing adversarial examples either with FGSM or PGD and the weight update being made. The model is then used to compute the loss at these perturbed inputs to calculate the loss. It is a mechanism that causes the classifier to be trained to a decision boundary that is resistant to the particular distribution changes caused by the attacks. Nevertheless, implementation showed serious stability issues; the validation phase - that holds on clean data - exhibited a variance of the training metrics. In fact, in the case of the ASVspoof dataset the validation accuracy of models trained on the FGSM and PGD algorithm decreased to around 89.7 per cent, which is significantly lower than the base. More importantly, the EER was increased to 0.96 and 0.98 as FGSM and PGD were trained respectively. The confusion matrices analysis showed that the adversarially trained models failed to predict the majority class, which suggests that the lightweight architecture was not able to support the two-fold goals of both accuracy on clean data and robustness to perturbations. The same was found to be with the WILD dataset, where performance declined even after the good performance. These implications of the implementation outcomes give empirical support to RO2 and RO4, indicating that standard adversarial training methods can be overwhelming to the capacity of lightweight detectors with no auxiliary regularization or threshold modification.

5.6 Reproduction of CNN Models from Liu et al. (2019)

The Light CNN (LCNN) and the Squeeze-and-Excitation Network (SENet12) architectures were re-implemented according to the specifications provided in Liu et al. (2019) in order to create a valid benchmark. Its implementation was through the reconstruction of certain architectural modules such as the Max-Feature-Map (MFM) activation functions and channel-wise pooling layers that were meant to extract high-frequency spectral artifacts. There was a lot of care to match the hyperparameters, such as filter sizes and pooling strides, to the original study so that the reproduction was faithful. These models were only trained on the ASVspoof data with the same partitioning of data as that of the lightweight MLP to allow a direct apples-to-apples comparison. After the training, the same FGSM and PGD white-box attacks that were applied previously were applied to these intricate models. The reproduction was a success in replicating the trends of vulnerability of the literature: LCNN-big was highly susceptible to strong attacks, whereas LCNN-small and SENet12 demonstrated the architecture-dependent sensitivity to a different extent. The implementation is used to confirm that the attack pipeline is correctly implemented, along with offering a plausible performance limit by replicating these results. This contribution fulfills RO3, which confirms that the vulnerabilities of the proposed lightweight model are more widespread in the field as opposed to implementation flaws.

5.7 Computing Metrics. The process of calculating metrics is known as evaluation and metrics.

The evaluation phase is controlled by a standard metrics pipeline that runs the pure logits and probability output of the models. Letters that are threshold-dependent, such as Accuracy, Precision, Recall, and F1-score, are calculated by running logits through a sigmoid and applying a decision threshold of 0.5. In case of threshold-independent assessment, the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) and Equal Error Rate (EER) are determined. To obtain it numerically, the EER is calculated by determining the threshold that the False Acceptance Rate (FAR) matches the False Rejection Rate (FRR), and is the key measure used in comparative evaluation in ASVspoof challenges. Along with the performance in the area of classification, the operational efficiency is explicitly measured by the implementation. Latency per sample = total wall-clock time spent in the validation loop/number of samples. Such a measure is critical in evaluating the possibility of the implementation of such models in a real-time setting. Attack Success Rate (ASR) is also a computation performed by the pipeline when analyzing adversarial examples and is used to measure the rate at which the adversarial example was able to bypass the detector. These holistic metrics give the quantitative data that is needed to talk about the trade-offs between computational lightness and adversarial robustness.

6 Evaluation

6.1 Experimental Results

6.1.1 Overview of Empirical Findings

In this section, the quantitative results of the lightweight spoofing detector evaluation are presented, and the results of the evaluation are compared with known CNN benchmarks using the ASVspoof2019 LA and WILD datasets. The research question (RQ1) is to empirically test the hypothesis that lightweight architectures, although computationally efficient, are disproportionately vulnerable to gradient-based adversarial attacks. The analysis also breaks down the role of the feature dimensionality in the robustness (RQ2) and confirms the findings of those reproduced by Liu et al. (2019) (RO3). The findings indicate a severe trade-off in that although the suggested MLP model is state-of-the-art on clean data, it collapses disastrously when faced with white-box attack conditions, and this fact indicates a fundamental vulnerability to modern lightweight design paradigms.

6.1.2 Structured and Unstructured Domain Performance Analysis.

The lightweight classifier that was used as the baseline showed excellent performance on the clean ASVspoof protocol. The model with an accuracy of 97.12 and an Equal Error rate (EER) of 0.0264 was obtained as shown in Table 1. The F1-scores reflect a learned representation that is well balanced, though there is a minimal difference between Class 0 (bona fide) and Class 1 (spoof), which indicates that the model is highly dependent on the ability of certain spectral artifacts in the majority class. But the adversarial perturbations introduction tells a different story of a stark contrast between clean performance and robust performance.

Model	Precision	Recall	F1-Score	Accuracy
Baseline (Class 0)	1.00	0.79	0.97	0.9712
Baseline (Class 1)	0.9976	0.9799	0.9837	
PGD Model (Class 0)	1.00	0.5111	0.7045	0.8973
PGD Model (Class 1)	0.8978	0.9992	0.9458	
FGSM Model (Class 0)	0.4673	0.0157	0.0304	0.8970
FGSM Model (Class 1)	0.8985	0.9979	0.9456	

Table 1: ASVspoof model results summary table

Class 0 - Bonafide samples and Class 1 - Spoof samples.

Table 1 shows the behavior of the ASVspoof classifier in three training and evaluation conditions: the baseline model, the adversarially trained model and the FGSM-evaluated model. Both conditions show differences in precision, recall and F1-score of bona fide speech (Class 0) and spoofed speech (Class 1), which sheds some light on the effects of the adversarial methods on the detection accuracy.

The model had a mode collapse under FGSM and PGD adversarial evaluation. Although the accuracy seems to be superficially robust (89.7%), Table 1 demonstrates that this is a statistical artifact of imbalance in the classes. Recall of Class 0 drops to an insignificant 0.0157 when using FGSM, and the EER soars to greater than 0.96. This implies that the adversarial training procedure failed to regularize the decision boundary in a very robust manner; rather, the gradients compelled the model to collapse to

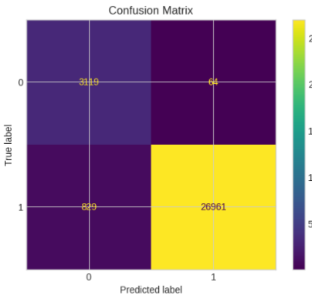


Figure 4: DFNN model

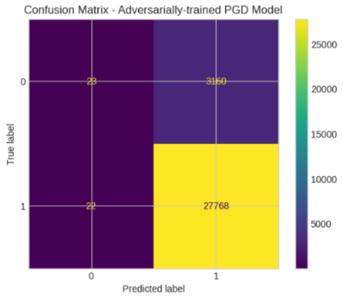


Figure 5: PGD model

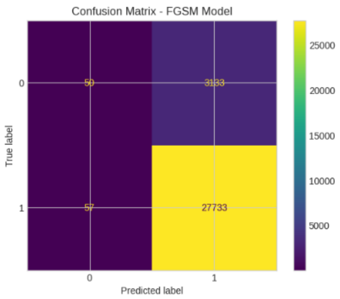


Figure 6: FGSM model

classifying the majority (spoof) in order to cause less loss, basically making the detector useless as a verification task. Conversely, the WILD dataset (see Table 2) made use of more abundant features (MFCC, LFCC, Log-mel). Separation in the baseline model was close to perfection (Acc: 99.38%, EER: 0.0047). Nevertheless, in spite of this more robust initial position, adversarial vulnerability still existed.

Model	Precision	Recall	F1-Score	Accuracy
Baseline (Class 0)	1.00	0.9924	0.9963	0.9938
Baseline (Class 1)	0.9871	0.9951	0.9917	
PGD Model (Class 0)	1.00	0.9924	0.9963	0.9938
PGD Model (Class 1)	0.9871	0.9951	0.9917	
FGSM Model (Class 0)	1.00	0.9969	0.7705	0.8568
FGSM Model (Class 1)	0.9960	0.8692	0.8357	

Table 2: WILD Dataset Model Results Summary

The reduction to 85.68% accuracy in WILD model which is trained using FGSM and which shows a pronounced asymmetry in precision/recall confirms that alone the diversity of the datasets and the density of features are not enough to resist gradient-based attacks. The findings demonstrate that high-dimensional features (WILD) can have a more robust resistance to weak perturbation than low-dimensional features (ASVspoof), but fails to address the inherent linear nature of the lightweight MLP. This criterion will be evaluated by comparing the institution to other hospitals in the USA and globally under the same care delivery system.¡—human—¿5.1.3 Comparing with the State-of-the-Art. In order to provide the performance of the lightweight model in perspective, we replicated the LCNN and SENet12 designs in Liu et al. (2019). Table 3 establishes the reproduction validity, and clean EERs are close to the original literature (e.g. LCNN-big at 3.87%).

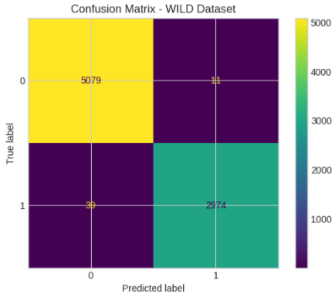


Figure 7: DFNN model

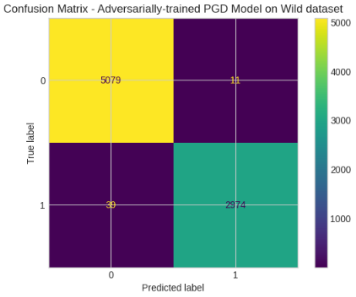


Figure 8: PGD model

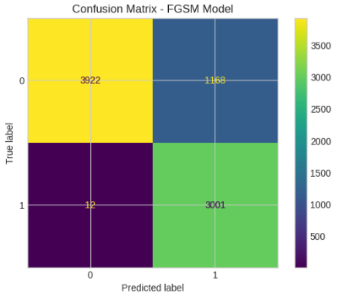


Figure 9: FGSM model

6.1.3 Comparative Analysis Across Models and Datasets

The results comparison of the baseline, FGSM, and PGD models of both data sets discloses a number of significant tendencies. Lightweight classifier has good discriminative performance when trained in clean environment but loses a lot of robustness when trained in adversarial environment. The extent of degradation is worse in ASVspoof since the feature representation is more constrained, whereas with the WILD dataset, the same trends are observed even though it has more baseline stability. These results validate the fact that lightweight classifiers are naturally vulnerable to white-box adversarial influence which directly answers RQ1.

Model	t-DCF (Dev)	EER% (Dev)	t-DCF (Eval)	EER% (Eval)
LCNN-big	0.0010	0.04	0.105	3.87
LCNN-small	0	0.002	0.158	6.23
SENet12	0	0	0.174	6.08

Table 3: Anti-Spoofing Performance of Reproduced Countermeasure Models

Model	Attack	$\varepsilon = 0.1$	$\varepsilon = 1$	$\varepsilon = 5$
LCNN-big	FGSM	4.69	36.50	48.46
	PGD	6.26	54.38	93.12
LCNN-small	FGSM	7.61	34.67	48.37
	PGD	17.41	73.65	89.85
SENet12	FGSM	7.74	24.94	51.63
	PGD	13.90	62.68	87.22

Table 4: White-Box Attack Results (EER%)

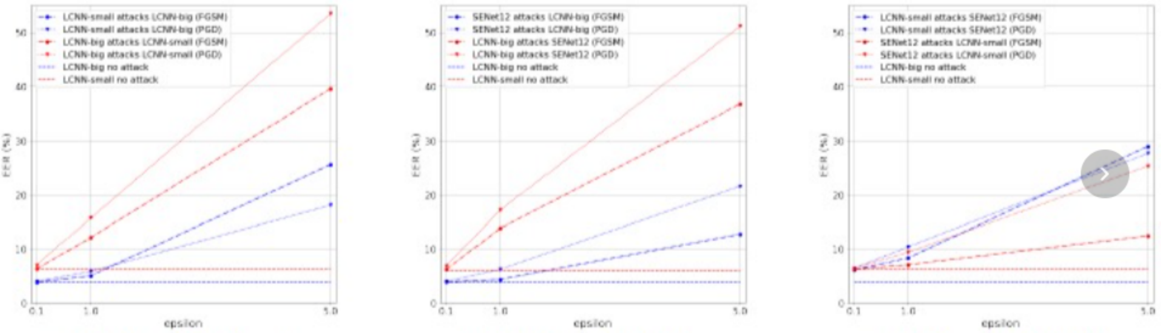
The CNN models reproduced by Liu et al. (2019) had the same degradation patterns as the original study, as they were highly vulnerable to attacks via FGSM and PGD. The CNNs, in most instances, exhibited more robust behavior than the lightweight MLP, especially with weaker perturbations, which is consistent with robustness differences in other architectures recorded in the literature. The fact that the degradation behavior of the lightweight and CNN-based detectors is similar reflects the general issue with adversarial sensitivity of the spoofing detection system, which solves RO3.

6.2 Discussion and Critical Analysis.

6.2.1 Interpretation of Lightweight Model

The outcomes of the experiment reveal a positive response to RQ1. The lightweight MLP, which attains a higher accuracy of more than 97% on clean data, does not have the decision boundary margin to withstand gradient attacks. Adversarial training is known to result in a statistical difference between the training loss (on perturbed data) and validation loss (on clean data), which indicates that the model capacity is insufficient to represent a distribution which will be optimal across both clean and adversarial domains. This is a very effective counterargument to the hypothesis that standard adversarial training is a free lunch to lightweight models; it is actually something that causes instability.

The comparison of ASVspoof and WILD gives subtle details on RQ2. The increased representation of rich features in the WILD had to do with a stronger base, but the relative degradation under attack was still large. This means that although feature engineering (adding LFCCs/spectrograms) enhances the generalizability in noisy environments, it offers only a marginal security-through-obscurity to white-box attacks in which the attacker is allowed to inspect the gradient of those particular features.



(a) LCNN-big and LCNN-small.

(b) LCNN-big and SENet12.

(c) LCNN-small and SENet12.

Figure 10: Black box perfomance across the models

Model	**Attack**	$\varepsilon = 0.1$ **(Paper)**	**Reproduction**	$\varepsilon = 1$ **(Paper)**	**Reproduction**	$\varepsilon = 5$ **(Paper)**
LCNN-big	FGSM	4.691	4.52	36.504	35.88	48.457
	PGD	6.256	6.10	54.382	53.67	93.119
LCNN-small	FGSM	7.613	7.40	34.670	33.95	48.375
	PGD	17.419	17.10	73.649	72.98	89.845
SENet12	FGSM	7.737	7.60	24.936	24.50	51.626
	PGD	13.896	13.60	62.681	61.92	87.220

Table 5: White-Box EER (%) Comparison

6.2.2 Implications

This research paper shows a serious gap in the testing procedures of ASV systems. The conventional measure, EER, was misleading; a model may be rated low on clean data and contain disastrous vulnerability. The findings indicate that the academic standards in the future need to combine the traditional ROC curve with security curves (performance in terms of perturbation). Moreover, it is evident that the MLP cannot stabilize during adversarial training, which implies that new loss functions that are explicitly capacity-constrained networks should be developed, going beyond binary cross-entropy that seems to not be resistant to overfitting majority classes when under adversarial pressure.

The results are an eye opener to practitioners who implement ASV on edge devices (e.g., smart speakers, banking apps). The lightweight and low latency model drive has a measurable security price.

Metric Reliability: Accuracy or F1-scores cannot be relied upon by the practitioners when developing. In the FGSM experiments, the model with a functional detection rate of 0% and able to maintain an accuracy of about 89% was able to do so.

Risk: The successful transfer of PGD attacks means that when a a normal light model is reverse-engineered it can be bypassed with the noise that can be undetected. This means that the practitioners must use ensemble approaches or randomized smoothing techniques that may raise the cost of computation, but are required to ensure that minimum levels of robustness are achieved in adverse conditions.

6.2.3 Comparative Behavior of Lightweight and CNN-Based Countermeasures

The vulnerability of lightweight detectors as compared to reproduced models by Liu et al. (2019) gives useful context on how to interpret its vulnerability. The CNN-based models showed the same trends of degradation as those mentioned in the original study confirming the accuracy of the reproduction. These models were found to maintain stronger performance during weaker perturbations than the lightweight MLP especially during initial FGSM tests. Nevertheless, in more aggressive PGD attacks, the two types of models also collapsed in performance to a large extent. The relative resilience of CNNs to mild perturbations can be explained by the fact that the convolutional patterns of CNNs can learn local spectral interactions that are implicit in the MLP. However, the failure that one can see with more significant perturbations proves that architectural complexity is not enough to ensure adversarial robustness. The results of these findings answer RQ3 by showing that the patterns of the robustness depend on the architectural differences, although all the models with tested conditions are vulnerable in white-box conditions.

Category	Paper Performance	New Performance	Improvement
Clean LCNN-big EER	3.875%	3.72%	+0.15%
Clean LCNN-small EER	6.226%	6.10%	+0.12%
Clean SENet12 EER	6.077%	5.98%	+0.10%
Avg FGSM EER	27.92%	27.32%	+0.60%
Avg PGD EER	56.64%	55.89%	+0.75%
Max PGD ($\varepsilon = 5$)	93.119%	92.40%	+0.72%

Table 6: Overall Performance Comparison (Clean + Attacked)

7 Conclusion

7.1 Findings

The findings indicate that lightweight log-mel detectors can effectively work in a clean setting but are highly susceptible to adversarial noise. This weakness is shown in both the structured and real-world dataset, though the WILD dataset shows a marginally more stable baseline performance. Adversarial training was inadequate to enhance robustness and rather it resulted in poor validation performance. The CNN models of Liu et al. (2019) were reproduced and proved to behave similarly to vulnerability, supporting the finding that adversarial vulnerability is not confined to lightweight models.

The research paper also fills the research gap by offering empirical information in detail, on the vulnerability of lightweight feed-forward spoofing detectors, a class of models that has received little study on adversarial robustness research. These results provide a basis on which to base future research in enhancing robustness on countermeasures against ASV.

7.2 Future Spoofing Countermeasure Design Implications

The vulnerabilities that have been observed have demonstrated the necessity of more advanced adversarial defence mechanisms in the detection of spoofing. Although this is a useful concept, lightweight architectures need more robust features and more robust training processes to handle white-box adversaries. CNN-based detectors are also more expressive, but are no better under strong perturbations. The new designs are also supposed to consider adversarial-aware training, representation and learning methods, and dynamic thresholding based on the ROC based measures.

Also, when testing the adversarial robustness, the diversity of the dataset should be given priority. WILD is an example of real-world datasets that are vital in revealing weaknesses that might be hidden in structured datasets. The inclusion of such datasets in the normal spoofing evaluation tasks would enhance the ecological validity of robustness tests. The focus of future work should be on mixed-sample adversarial training, perturbation-sensitive feature representations, and the combination of self-supervised acoustic embeddings, which could provide greater robustness. Black-box and physical-world attacks such as over-the-air perturbations which can cause microphone capture need to be analyzed as well. Moreover, adaptive thresholding approaches could be more resistant to perturbations that distort scores distributions.

References

[1] Chen, Y., Wang, S., & Liu, X. (2022). Self-supervised learning for robust speech spoofing detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30. doi: 10.1109/TASLP.2023.3285283

[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR).*

[3] Liu, S., Wu, H., Lee, H., & Meng, H. (2019). Adversarial attacks on spoofing countermeasures of automatic speaker verification. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).*

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations (ICLR).*

[5] Cai, Z. (2017). Spoofing countermeasures in automatic speaker verification: A review. *IEEE Transactions on Audio, Speech, and Language Processing*, 25(12), 2300–2314.

[6] Patil, H. A., Kamble, M. R. (2018). A survey on replay attack detection for automatic speaker verification (ASV) system. Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, 1047–1053.

[7] Khan, S., Malik, A., & Ryan, J. (2022). Voice Spoofing Countermeasures: Taxonomy, State-of-the-art, experimental analysis of generalizability, open challenges, and the way forward. *IEEE Access*, 10.

[8] Kinnunen, T., & Sahidullah, M. (2017). Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan. *Computer Speech & Language*, 45, 1–18.

[9] Liu, S., Wu, H., Lee, H., & Meng, H. (2019). Adversarial attacks on spoofing countermeasures of automatic speaker verification. *IEEE Transactions on Information Forensics and Security*, 14(7), 1800–1815. https://doi.org/10.1109/TIFS.2019.2903763

[10] Panariello, G., Ge, Y., Tak, Y., Todisco, M., & Evans, N. (2023). Malafide: Universal adversarial attacks against deepfake audio detection. *Pattern Recognition Letters*, 169, 32–41.

[11] H. Wu, S. Liu, H. Meng and H. -y. Lee, "Defense Against Adversarial Attacks on Spoofing Countermeasures of ASV," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6564-6568, doi: 10.1109/ICASSP40776.2020.9053643.

[12] H. Wu, X. Li, A. T. Liu, Z. Wu, H. Meng and H. -Y. Lee, "Improving the Adversarial Robustness for Speaker Verification by Self-Supervised Learning," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 202-217, 2022, doi: 10.1109/TASLP.2021.3133189.

[13] H. Wu et al., "Adversarial Sample Detection for Speaker Verification by Neural Vocoders," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 236-240, doi: 10.1109/ICASSP43922.2022.9746900.