A Generative Adversarial Network-based Attack for Audio-based Condition Monitoring Systems

Abdul Rahman Ba Nabila, Eduardo K. Viegas, Abdelrahman Almahmoud, Willian T. Lunardi
Secure Systems Research Center, Technology Innovation Institute (TII)
United Arab Emirates, Abu Dhabi
{abdulrahman.banabila, eduardo, abdelrahman, willian}@ssrc.tii.ae

2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) | 978-1-6654-9734-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCNC51644.2023.10060644

***Abstract*—Over the last years, several machine learning techniques have been proposed for the condition monitoring of physical assets based on audio. As a result, adversaries have been trying to circumvent the reliability of deployed systems, typically through the generation of maliciously altered audio samples that are subsequently introduced as input by the model. However, altering the input in production settings is not always feasible, on the contrary, samples are often collected through a microphone, significantly increasing the attack execution effort. In this paper, we propose a realistic generative adversarial network attack for an audio-based condition monitoring system. We first train a generator and a discriminator with a joint objective of generating audio samples corresponding to the difference between the two classes, e.g., normal and faulty. Additionally, we test our approach by overlapping our generated audio on the samples collected by the microphone. Our main goal is the proposal of a GAN-based attack capable of generating audio samples that when overlaid with the original microphone-captured audio may induce misclassification given a target class. Experiments performed through our captured audio dataset from normal and broken unmanned aerial vehicle propellers show that the proposed attack achieved a mean success rate of 40%, decreasing the F-measure concerning random noise by 13.3%, 20%, and 37.8% for ResNet-18, AlexNet, and DenseNet-169 models, respectively.**

***Index Terms*—Generative Adversarial Networks; Adversarial Example; Audio Generation.**

I. Introduction

Condition Monitoring Systems (CMSs), aligned with cyber-physical systems, play an essential role in monitoring and diagnosing various physical assets such as motors, pumps, and fans. In recent years, several techniques have been proposed for condition monitoring. Machine learning (ML) techniques play a critical role in performing intelligent diagnostics from data collected from various sensors such as cameras, accelerometers, gyros, and microphones [1], e.g., microphones can be used for fault detection of motors, engines, and even propellers [2]. Due to the high criticality and financial costs of monitored assets, adversaries are highly motivated to circumvent the reliability of ML-based CMSs.

Proposed attack schemes on CMSs typically rely on generative adversarial networks (GAN) [3]. The GAN [4] framework establishes a min-max adversarial game between a generative model G and a discriminative model D. The discriminator $D(x)$ computes the probability that a point x in data space is a sample from the data distribution rather than a sample from the generative model. The generator $G(z)$ maps samples z from the prior $p(z)$ to the data space. $G(z)$ is trained to maximally confuse the discriminator into believing that the samples it generates come from the data distribution. The process is iterated, leading to the famous minimax game between G and D [4]. Therefore, given the data used to train an ML-based CMS (or even the model itself), one may easily employ a GAN-based attack strategy to generate samples that can induce misclassification on the CMS. However, the attacker cannot easily manipulate the sensor signal. For instance, to evade an audio-based CMS, the attacker must be able to introduce tailored audio artifacts that, when overlaid with the original audio from the monitored asset and captured by the microphone, affect the CMS' performance, thus, significantly increasing the attack execution efforts [5]. Surprisingly, the vast majority of GAN-based techniques assume that the CMS's input signal can be manipulated as needed, despite the challenges related to the execution of such attacks in real-world settings.

In this paper, we propose a more realistic GAN-based attack for audio-based CMSs where the attacker aims to fool the CMS into misclassifying collected audio samples. The attack is based on the generated audio samples that, when overlapped with the original audio produced by the monitored physical asset, will result in poisoned audio that will be classified as a given target class, e.g., a normal audio sample being classified as faulty. In summary, the main contributions of this paper are as follows:

- A GAN-based attack for audio-based CMSs that generates audio samples that, when overlaid with the original microphone-captured audio, may induce misclassification with a mean success rate of 40%.
- An extensive evaluation of audio-based CMSs for fault detection in an unmanned aerial vehicle (UAV), more specifically, UAV broken propeller detection. Experiments have shown that current approaches are reliable to external noises that may be captured by the system microphone. Regardless, with an SNR of 10, our proposed attack decreased F-measure concerning the random noise by 13.3%, 20%, and 37.8% for ResNet-18, AlexNet, and DenseNet-169, respectively.

The remainder of the paper is organized as follows. Section II provides the necessary background concerning audio-based CMSs and adversarial machine learning. Section III reviews

and discusses previous relevant works on GAN-based attack schemes. Section IV describes our proposed attack scheme, model architecture, loss functions, and adversarial training strategy. Section V presents the experimental analysis. Finally, Section VI concludes this paper.

II. Preliminaries

This section further describes the application of audio-based techniques for condition monitoring tasks and how adversaries can explore such properties to circumvent the reliability of developed approaches.

A. Audio-based Condition Monitoring

Audio-based techniques for condition monitoring of physical assets are typically implemented through four sequential modules [6], namely data acquisition, pre-processing, classification, and alert. First, the data acquisition module continuously collects audio samples, often divided into a predefined time window. The built audio samples are then used as input to a pre-processing module. The goal is to extract a representation that can be used for classification tasks, such as spectrograms considering an audio-base CMS case. The pre-processed samples are then used as input by a classification module, typically composed of a DL model, which performs feature extraction, and finally classifies it as either normal or abnormal.

B. Adversarial Machine Learning

In recent years, several works have been proposed to circumvent the reliability of ML systems, wherein proposed schemes are implemented following two main strategies [7]: (i) *causative*, and (ii) *exploratory*. Causative attacks poison the training data to introduce patterns that may induce the built model to incorrectly classify the attacker desired events. As a result, it assumes that the attacker has access to the training dataset, significantly increasing the attack execution efforts. Exploratory attacks perturb the system input to evade the deployed ML model. Consequently, the attacker only needs to be able to affect the ML model input, significantly easing the attack execution efforts.

In light of this, several approaches have been proposed for *exploratory* attacks in ML systems, wherein authors typically rely on GAN techniques [3]. To achieve such a goal, the trained model generates "adversarial examples" that can be used to evade ML-based systems. Despite promising results, most proposed schemes assume that attackers can change the CMS' input as needed. In contrast, in production settings, the system input is strongly bound to the used sensor (e.g., microphone); thus, the attacker must be able to first introduce the needed artifacts to the sensor collected values to adequately produce the ML desired input. Surprisingly, only recently related works have considered such a challenge, such as proposing printing physical patches to evade a camera-based ML system [5].

III. Related Work

Exploratory adversarial attacks on ML systems have been a widely explored topic over the last few years. For instance, Liu et al. [8] proposed a GAN-based technique to attack data-driven strategies with a success of 70%. The authors assume structured data and knowledge of the input shape. The attack is not on the sensor level, and the adversarial strategy implements multiple generators for each class. Bai et al. [9] proposed a GAN approach that generates more training samples to enhance the performance of membership inference attacks. The model increases efficiency by 23% by generating samples that are used to train an attacking model while not generating additive injections. Jia et al. [10] proposed a GAN-based strategy that successfully generates new video streams to fool gait recognition systems. While this is an excellent example of the application of GANs as an attacking tool, this method requires access to the data pipeline and swapping the original input video stream for the generated video.

Usama et al. [11] proposed a GAN-based attack strategy to alter network traffic data. The generated data only modifies the non-functional contents of the network traffic. Their proposed attack succeeds in lowering the accuracy of a black box intrusion detection system. While they assume a black box model, their approach completely swaps the generated content for the original content. Abdullah et al. [12] proposed a perturbation engine to fool voice processing systems (VPSs). They managed to generate samples that get accepted and transcribed incorrectly by 7 VPSs. In some situations, samples were rejected for being distorted beyond recognition. In the context of speech-to-text transcription, Carlini and Wagner [13] proposed a method for generating small perturbations to the original audio to cause misclassification. They successfully performed an attack on DeepSpeech in multiple scenarios. Their work assumes white-box settings.

Alzantot et al. [14] explored gradient-free approaches for generating adversarial examples. The proposed method succeeds in generating samples successfully to fool the target model. While their approach assumes a black box setting, it requires access to the outputs of the specific model they are trying to attack. Also, this work does not overlap noise on top of the original samples but instead generates a new audio sample. Our considered pipeline is practical since we do not have to tamper with the collected samples but rather perform the attack on the sensor level before data acquisition. Several works have tried to exploit ML capabilities in generating their adversarial examples. Most of the works were directed toward the field of computer vision [15], while most audio-based attacks assumed either white-box settings or access to the model's outputs/predictions. Our work generates realistic noise that can be overlapped physically with actual audio samples with no knowledge of the underlying model performing the classification task.

IV. Methodology

The proposed model considers an adversary whose goal is to circumvent the reliability of an ML-based CMS used to

monitor a given target physical asset, e.g., the detection of broken UAV propellers. The CMS analyzes the audio samples from the physical asset through a DL model, which signals undesired asset conditions accordingly. In this context, the attacker aims to make the deployed DL model misclassify the collected audio sample. The attacker must be able to generate audio samples that, when overlaid with the original audio, will result in poisoned audio that will be classified according to the attacker's desired class, e.g., the normal audio sample being classified as faulty. Our proposed scheme extends Fre-GAN [16], a GAN originally designed to reconstruct audio samples from their perspective spectrogram.

Our proposed model considers a realistic attack scenario for audio-based CMSs. More specifically, we consider an attacker with the following capabilities: (1) the CMS internals is not known to the attacker (*black-box*), including the used feature extraction algorithm and the ML model weights; (2) the attacker can continuously collect the audio sounds emitted from the monitored physical asset; (3) the attacker can generate audio sounds through an audio reproduction device, e.g., stereo-speaker; (4) the audio sounds emitted by the attacker audio reproduction device will be captured by the CMS microphone and overlaid to the audio sounds emitted by the monitored physical asset; (5) the attacker has audio samples corresponding to both states of the targeted system.

A. Audio Pre-processing

We consider a CMS with binary DL classifiers or anomaly-based approaches [17] that categorizes audio from UAV propellers as either normal or broken/faulty. Based on the assumptions above, we can access the drone's audio samples corresponding to both classes. Therefore, we consider audio samples recorded with a sampling rate of 16kHz, where each audio sample is a second long. The mel-spectrogram is produced with a window size of 1024, a hop size of 256, 1024 fast Fourier transforms, and a total of 80 mel-banks. We pad some zeros to the end of the audio segment to ensure no sample loss in the conversion back and forth.

B. Model Architecture

In our proposal, the generator and discriminators are trained on a min-max adversarial game. The discriminator computes the probability of a point in the data space (the difference between normal and faulty) rather than the samples produced by the generative model. The generator maps a normal audio sample into the data space. It is iteratively trained to maximally confuse the discriminator into believing that the samples it generates come from the data distribution. Our proposed model extends Fre-GAN [16]. Figure 1 shows an overview of the proposed approach. In our approach, a forward pass uses three inputs—the waveform, the mel-spectrogram of a base class, and a waveform of a target class. The base class can be either of the two classes, while the target class will always be the opposite of the base class. Our proposed approach aims to generate noise samples that best represent the difference between the target and base class's audio.

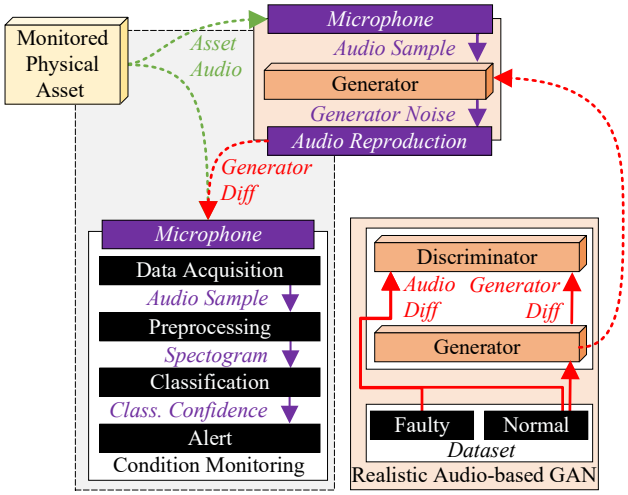


Fig. 1: Proposed realistic generative adversarial network attack for audio-based CMSs.

We implement two families of discriminators: (i) multi-period discriminator (MPD) and multi-scale discriminator (MSD). The goal of the discriminators is to maximize the distance measured between the generated audio and the ground truth. The two discriminators aim to learn periodic and sequential features in the passed audio samples. For the MPD, the audio samples are reshaped into a 2D representation. It constitutes five periodic discriminators with periods of $[2, 3, 5, 7, 11]$. On the other hand, the RSD contains three scale discriminators that process the whole audio at its original sampling rate, a 2x downsampled audio, and finally, 4x downsampled audio, respectively. Both discriminators use discrete wavelet transforms (DWT) to account for higher and lower frequencies and implement residual connections. Note that our architecture follow very closely those introduced by Kim et al. [16], see (Kim et al. [16], §2) for more details.

C. Advesarial Training

Given the target audio sample x^{T} and the base class audio x^{B}, our proposed training objective for the discriminator and generator are respectively given by

$$\begin{aligned}\mathcal{L}_D = &\sum_{n=0}^{4} \mathbb{E}\big[\|D_n^P(x^{\mathrm{T}} - x^{\mathrm{B}}) - 1\|_2] + \|D_n^P(\hat{x})\|_2\big] \\ &+ \sum_{m=0}^{2} \mathbb{E}\big[\|D_m^S(\phi^m(x^{\mathrm{T}} - x^{\mathrm{B}}) - 1)\|_2 \\ &+ \|D_m^S(\phi^m(\hat{x}))\|_2\big],\end{aligned} \tag{1}$$

and

$$\begin{aligned}\mathcal{L}_G = &\sum_{n=0}^{4} \mathbb{E}\big[\|D_n^P(\hat{x}) - 1\|_2 + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}}(G; D_n^P)\big] \\ &+ \sum_{m=0}^{2} \mathbb{E}\big[\|D_m^S(\hat{x}) - 1\|_2 + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}}(G; D_n^P)\big] \\ &+ \lambda_{\text{mel}}\mathcal{L}_{\text{mel}}(G),\end{aligned} \tag{2}$$

where $\hat{x}$ denotes the generated audio, D^P and D^S denotes the RPD and RSD discriminators, and ϕ^m represent the m-level discrete wavelet transform. The hyper-parameters λ_{fm} and λ_{mel} (fixed to $\lambda_{\text{fm}} = 2$ and $\lambda_{\text{mel}} = 45$ as suggested in [16]) controls the balance between the feature matching loss $\mathcal{L}_{\text{fm}}$ and the mel-spectrogram loss $\mathcal{L}_{\text{mel}}$, given by

$$\mathcal{L}_{\text{fm}}(G; D_k) = \mathbb{E}\Big[\sum_{i=0}^{Q-1} \frac{1}{N_i} \|D_k^{(i)}(x^{\text{T}} - x^{\text{B}}) - D_k^{(i)}(\hat{x})\|_1\Big], \quad (3)$$

where Q denotes the number of layers in the discriminator. $D_k^{(i)}$ refers to the i^{th} layer feature map of the k^{th} sub-discriminator, N_i is the number of units in each layer, and

$$\mathcal{L}_{\text{mel}}(G) = \mathbb{E}\Big[\|\psi_k^{(i)}(x^{\text{T}} - x^{\text{B}}) - \psi_k^{(i)}(\hat{x})\|_1\Big]. \quad (4)$$

where ψ is the STFT function to convert raw audio to the corresponding mel-spectogram.

The generator's loss is composed of three main elements: (i) first takes the discriminator's feedback score into account (ii) the second takes into consideration discrepancies in feature maps; (note that the discriminators generate 12 feature maps in total, and that ensures that the same low and high-level audio features are present in the generated sample) (iii) the last element considers differences in the spectrograms of the generated samples and that of their ground truths. By minimizing those three elements, the generator's sample will be as close to the ground truth's nature as possible. On the other hand, The discriminators will produce a real/fake score. They will maximize the distance between those scores for ground truth and the corresponding generated image. This is also equivalent to minimizing equations 1.

D. Model Execution

The deployment of our previously described model (see Section IV-B) considers an attacker equipped with an external sound device (e.g., stereo-speaker) used to affect the audio samples collected by the system. To achieve such a goal, the attacker collects the monitored physical asset sound through a microphone and uses the collected sample as input to a previously trained generator. The generator outputs an audio difference that can be used by the attacker's sound device to affect the audio sample collected by the CMS.

V. Experimental Evaluation

This section evaluates the effectiveness of our proposed GAN-based attack for audio monitoring systems. The considered test-bed and UAV propeller audio-based dataset and implementation details are described in Section V-A. Section V-B investigates how naturally occurring noises affect the performance of audio-based classification methods applied for condition monitoring. Finally, Section V-C assesses the approach's effectiveness and how it affects the classification accuracy of CMSs.

(a) UAV Propellers (b) Holybro X500 UAV

Fig. 2: UAV configuration used in our testbed. The broken region of the deffective UAV propeller is highlighted with a red circle.

A. Broken Propeller Dataset and Implementation Details

This work considers an audio-based CMS designed to detect physical faults in Unmanned Aerial Vehicles (UAVs). To reproduce such a scenario, we set up a controlled testbed with a Holybro X500 UAV equipped with a PX4 flight controller. The UAV carries a mission computer that runs a UP Xtreme i7 8665UE with a Seeed Studio ReSpeaker Mic Array for audio data collection for condition monitoring purposes. The testbed was executed for a total of 4 hours, with several execution rounds, wherein two hours were related to *normal* UAV conditions and two hours with *faulty* UAV conditions. In a *normal* UAV scenario, all of the 4 UAV propellers were undamaged. In contrast, from one to four UAV propellers are damaged in a *faulty* UAV scenario. Figure 2 shows a normal and a faulty UAV propeller used in the testbed for the dataset generation. For each testbed execution, the UAV autonomously flies ($\approx$ 5 minutes) in an eight or four-shape configuration, as controlled by the PX4 flight controller. The audio was collected with a sampling rate of 16kHz and a 16 bit integer representation. The collected audio is divided into 1-second long audio samples. Over 432 thousand samples of 1 second long audio were collected. Finally, the inputs are represented by an 80-band mel-spectrogram transformed with 1024 of window size, 256 of hop size, and 1024 points of Fourier transform built over the 1 second long audio sample, making use of the Python package *librosa* 0.9.2.

Three widely known neural network models were evaluated to play the role of the DL-based CMS, namely *ResNet-18*, *AlexNet*, and *DenseNet-169*. Each model was implemented using PyTorch 1.8 and trained for 1000 epochs using Adam optimizer with default parameters and a learning rate of 0.001. The models' input is given by the pre-processing described in Section IV-A. The dataset described above was randomly split into *training*, *testing*, and *validation* datasets, each composed of 60%, 30%, and 10% of samples, respectively. We use F-Measure, false-positive rate (FPR), and false-negative rate (FNR) as evaluation metrics. The FPR denotes the ratio of *normal* UAV audio samples incorrectly classified as *faulty*. In contrast, the FNR denotes the ratio of *faulty* audio samples incorrectly classified as *normal*.

TABLE I: The impact on the audio-based CMS's performance when subject to noise.

Deep Learning	Scenario	F-Measure	FPR	FNR
ResNet-18	Noise-free	0.96	0.058	0.0167
	Added Noise	0.9	0.16	0.04
AlexNet	Noise-free	0.96	0.06	0.03
	Added Noise	0.9	0.15	0.05
DenseNet-169	Noise-free	0.97	0.04	0.02
	Added Noise	0.9	0.13	0.08

TABLE II: The impact on the audio-based CMS' performance when subject to our proposed generated attacks.

Deep Learning	Scenario	F-Measure	FPR	FNR
ResNet-18	FPR Increase	0.75	0.18	0.29
	FNR Increase	0.86	0.25	0.06
	Error Increase	0.78	0.15	0.27
AlexNet	FPR Increase	0.68	0.94	0.01
	FNR Increase	0.79	0.04	0.32
	Error Increase	0.72	0.08	0.39
DenseNet-169	FPR Increase	0.53	0.22	0.56
	FNR Increase	0.73	0.06	0.39
	Error Increase	0.56	0.08	0.57

B. Robustness of Audio-based UAV CMS

Our first experiment aims to evaluate the impact of additional noises on the performance of the selected techniques, e.g., in a city environment, the UAV may be exposed to significantly loud noises such as drilling or car engines. Recall that the audio data collected was not subject to various natural noises that may occur in the real world. Therefore, we evaluate the robustness of audio-based condition monitoring when exposed to naturally occurring audio (such as voices, cars, and engines) from the Microsoft Scalable Noisy Speech (MSNS) dataset [18]. We overlay on the UAV audio samples audio segments of noise from the MSNS dataset to achieve an SNR of 10. Table I shows the classification accuracy of the selected techniques when subject to production settings audio noises given the added noise. It is possible to note that the selected approaches are robust to the added audio noises, as commonly experienced in production settings. More specifically, the added noise scenario decreased the F-Measure by up to 0.08 for the DenseNet-169 DL model. Results show that traditional audio-based approaches are robust to production environment settings, presenting similar accuracy rates even when subject to additional audio noises that the microphone may capture.

C. Attacking Audio-based UAV CMS

Here we analyze the effectiveness of our proposed approach to circumvent the reliability of audio-based CMSs in a realistic setting. To achieve such a goal, we consider an adversary that reproduces generated audios with a SNR $\approx$ 20. More specifically, we consider a CMS which collects the monitored physical asset sound $\approx$ 20 louder than those generated by an attacker reproduction device. We consider three attack scenarios, as follows:

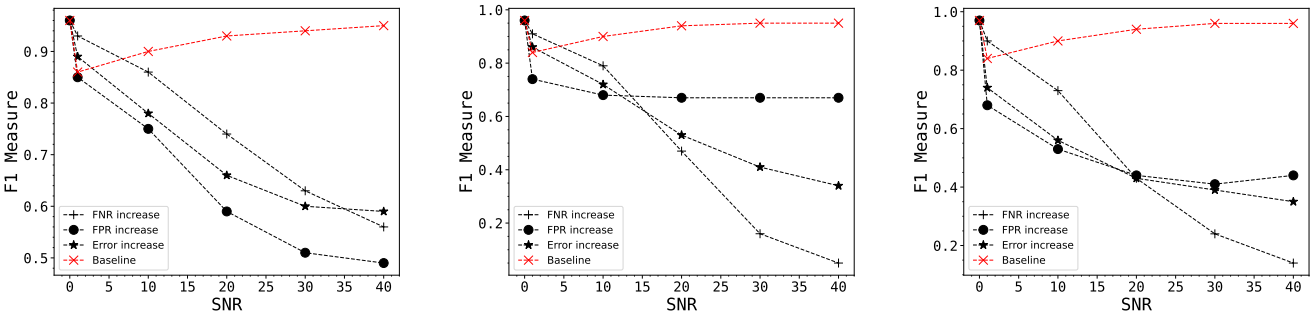
- **FPR Increase**. Overlaid generated samples lead normal audio samples to be misclassified as faulty.
- **FNR Increase**. Overlaid generated samples lead to faulty audio samples being misclassified as normal.
- **Error Increase**. Overlaid generated samples target both misclassifications, e.g., normal to be misclassified as faulty, and faulty to be misclassified as normal.

Under each attack mode described above, we trained and generated malicious samples to evaluate the effectiveness of our approach given each setting. As previously mentioned, the generator's samples exhibit specific characteristics, such as frequency and amplitude, that it deems most effective as additive noise. For comparison, we overlay the generated samples at a 10 SNR (analogous to the noise level presented in Table I). Table II shows the impact on the audio-based CMS' performance when subject to our proposed scheme. Our proposed model significantly decreased the accuracy of the selected techniques compared to the added noise from the MSNS dataset. For instance, given the *FPR increase* scenario, the proposed attack was able to increase the FPR rate to up to 0.36% for the ResNet-18 model, a further degradation of 0.3% when compared to the natural added noise presented in Table I. In contrast, in the *Error Increase* case, our proposed attack significantly affects the F-measure, decreasing it by up to 0.33 for the DenseNet-169 model. One of the drawbacks of our proposed approach is that by adding noise to the original sample, frequency bins in a spectrogram representation end up having higher energy levels, allowing the detection of such attacks by examining energy levels in spectrogram bins against a threshold. Typically in audio pre-processing, as in most ML applications, inputs are often normalized, or features are scaled to enhance the performance of gradient descent. This approach is considered a viable attack in a typical audio pre-processing pipeline.

We further examine the impact of the proposed attack on the CMS' F-measure given generated attack audio samples with SNR in $[0, 40]$. Figure 3 shows the impact of the proposed attack on the considered metric given different SNRs, where an SNR of 0 refers to the original audio. The baseline (denoted by a red line) shows the CMS' performance under overlaying of noise from the MSNS dataset. The three black lines correspond to the overlap of generated audio given the three attack modes. Our proposed attack significantly degrades the CMS' performance compared to the baseline. For the *Error Increase* scenario with an SNR of 10, the random noise case degrades the CMS' F-measure by 6.25%, 6.25%, 7.21% for the ResNet16 and AlexNet, and DenseNet-169 respectively. On the other hand, our proposed attack degrades it by 18.75%, 25%, and 42.26%. This is an improvement of 13.33%, 20%, and 37.8% for each considered model.

VI. Conclusion

Over the last few years, several works have proposed techniques for evading ML-based CMSs. Proposed schemes are inadequate for real-world settings, negligently assuming that generated data can be indiscriminately introduced to



(a) ResNet-18 (b) AlexNet (c) DenseNet-169

Fig. 3: CMS' F-measure as a function of SNR. Higher SNR means lower attack sound volume.

the CMS' input. This paper has proposed a new generative adversarial network for audio-based attacks on CMSs. The proposed scheme can generate audio samples that can be used to overlay with the original audio produced by the monitored physical asset and evade the deployed audio-based ML model responsible for classification in the CMS. Experiments performed using audio samples for fault detection of UAVs have shown the proposal's feasibility. The proposed attack achieved a mean success rate of 40%, decreasing the F-measure for "random noise" experiments by 13.3%, 20%, and 37.8% for ResNet-18, AlexNet, and DenseNet-169, respectively.

REFERENCES

[1] P. A. Higgs, R. Parkin, M. Jackson, A. Al-Habaibeh, F. Zorriassatine, and J. Coy, "A survey on condition monitoring systems in industry," in *Engineering Systems Design and Analysis*, vol. 41758, 2004, pp. 163–178.

[2] A. Altinors, F. Yol, and O. Yaman, "A sound based method for fault detection with statistical feature extraction in UAV motors," *Applied Acoustics*, vol. 183, p. 108325, Dec. 2021.

[3] J. Viola, Y. Chen, and J. Wang, "FaultFace: Deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method," *Information Sciences*, vol. 542, pp. 195–211, Jan. 2021.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[5] Y.-C.-T. Hu, B.-H. Kung, D. S. Tan, J.-C. Chen, K.-L. Hua, and W.-H. Cheng, "Naturalistic physical adversarial patch for object detectors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 7848–7857.

[6] R. R. dos Santos, E. K. Viegas, A. O. Santin, and V. V. Cogo, "Reinforcement learning for intrusion detection: More model longness and fewer updates," *IEEE Transactions on Network and Service Management*, pp. 1–17, 2022.

[7] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[8] Z. Liu, Q. Wang, Y. Ye, and Y. Tang, "A gan based data injection attack method on data-driven strategies in power systems," *IEEE Transactions on Smart Grid*, 2022.

[9] Y. Bai, D. Chen, T. Chen, and M. Fan, "Ganmia: Gan-based black-box membership inference attack," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[10] M. Jia, H. Yang, D. Huang, and Y. Wang, "Attacking gait recognition systems via silhouette guided gans," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 638–646.

[11] M. Usama, M. Asim, S. Latif, J. Qadir *et al.*, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *2019 15th international wireless communications & mobile computing conference (IWCMC)*. IEEE, 2019, pp. 78–83.

[12] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," *arXiv preprint arXiv:1904.05734*, 2019.

[13] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE security and privacy workshops (SPW)*. IEEE, 2018, pp. 1–7.

[14] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[16] J.-H. Kim, S.-H. Lee, J.-H. Lee, and S.-W. Lee, "Fre-GAN: Adversarial frequency-consistent audio synthesis," in *Interspeech 2021*. ISCA, Aug. 2021.

[17] W. T. Lunardi, M. A. Lopez, and J.-P. Giacalone, "ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection," *arXiv preprint arXiv:2205.01432*, 2022.

[18] C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," *Proc. Interspeech 2019*, pp. 1816–1820, 2019.