MÉTODOS ROBUSTOS CONTRA ATAQUES ADVERSÁRIOS NA CLASSIFICAÇÃO DE EMOÇÕES EM ÁUDIO

Victor Piotrovski Begha (Universidade Estadual de Ponta Grossa) E-mail: victorbegha@outlook.com
Alceu de Souza Britto Júnior (Universidade Estadual de Ponta Grossa) E-mail: alceubritto@gmail.com

Resumo: Ataques adversários podem ter um efeito prejudicial no desempenho de redes neurais, e há pouco estudo no comportamento específico desses padrões na classificação de emoções em áudio. Este artigo apresenta, analisa e compara vários métodos de defesa contra ataques adversários comuns. Resultados experimentais mostram que, através dessas técnicas, é possível mitigar o impacto de ataques que normalmente diminuiriam acurácia em mais de 75%, para apenas baixar a acurácia em 10% após sua aplicação. Além disso, no artigo é descrito como isso não é apenas relevante para robustez contra ataques individuais, mas também para a robustez do sistema contra pequenas mudanças no geral.

Palavras-chave: redes neurais, ataques adversários, classificação, áudio, robustez.

ROBUST METHODS AGAINST ADVERSARIAL ATTACKS IN CLASSIFICATION OF EMOTIONS IN AUDIO

Abstract: Adversarial attacks can have a harmful effect on the performance of neural networks, and there is little study on the specific behavior of these patterns on audio emotion classification. This paper presents, analyzes and compares several defense methods against common adversarial attacks. Experimental results show that, through these techniques, it is possible to alleviate the impact of attacks that would normally lower accuracy by over 75%, to only lowering accuracy by less than 10% after applying the method. Furthermore, in the paper it is described how this is not only relevant for robustness against individual attacks, but also for the robustness of the system against small changes in general.

Keywords: neural networks, adversarial attacks, classification, audio, robustness.

1. Introdução

O uso de técnicas de aprendizagem profunda e redes neurais convolucionais possibilita um alto desempenho para classificação, mas apesar de sua acurácia em situações normais, existem certos tipos de perturbação ou ruído que atuam de forma a maximizar o erro da rede sem, contudo, mudar perceptivelmente o sinal em si. Esses exemplos de alterações sutis que causam um impacto significativo no desempenho da classificação são comumente denominados como ataques adversários.

Na essência, um ataque adversário pode ser considerado como uma pequena perturbação adicionada ao sinal de entrada que provoca erros na classificação pelo modelo, mas cujo efeito é imperceptível para um ser humano. Esse ruído não é aleatório: na maioria dos casos, os ataques usam cálculos específicos com base no sinal de entrada e nas características da rede para aumentar a probabilidade de ocorrer um erro.

Existem vários estudos relacionados ao comportamento de perturbações adversárias, mas poucos especificamente sobre como agem na classificação de emoções em áudio, ou emoções na fala humana. Nesse tipo de aplicação, é comum que o áudio usado para o treinamento dos modelos seja passado para uma representação bidimensional, como um espectrograma, por exemplo (KOERICH et al, 2019). A classificação de emoções é também suscetível a ser afetada por ruídos que, propositalmente ou não, têm um impacto negativo em seu funcionamento.

É importante ressaltar o motivo da relevância desse tipo de experimentação. Mesmo se as situações que podem possivelmente existir em uma aplicação real não sejam tão impactantes quanto as amostras de ataque adversário, pode-se pensar que o exemplo adversário é a alteração que causa maior erro com o mínimo de alteração no espectrograma. Por isso, métodos de proteção contra esses ataques podem ter algum grau de transferibilidade para outros casos de ruído.

Deste modo, encontrar métodos para melhorar o desempenho das redes na ocorrência desses ataques não é relevante apenas para que sejam resistentes especificamente contra eles, mas também que sejam mais robustas contra variações sutis de maneira geral.

Este trabalho busca testar o efeito desses ataques na classificação de emoções em áudio e verificar quais modificações feitas na rede (ou nos parâmetros do treinamento, ou no modo em que a rede é utilizada) conseguem melhorar sua resiliência contra eles.

2. Trabalhos relacionados

Dentro da bibliografia referenciada existem algumas publicações que tratam do funcionamento de ataques adversários e o comportamento de classificadores frente a eles. Uma visão geral é apresentada por Chakraborty et al. (2018), cujo enfoque é compilar alguns dos modelos de ataque mais comuns e os métodos que aplicam, assim como listar possíveis defesas contra os mesmos.

No caso de ataques em espectrogramas de sinais de áudio, Koerich et al. (2019) experimentam sobre o efeito de ataques em redes neurais convolucionais com representação 2D, tal qual o que é estudado nesse trabalho. Sua contribuição principal, utilizando classificação de gêneros musicais como exemplo, é mostrar como essas alterações são pouco perceptíveis apesar de seu grande efeito negativo no desempenho na rede, e mostra também que se os áudios forem reconstruídos a partir desses espectrogramas perturbados, as faixas continuam também sendo muito semelhantes às originais, provando que essas perturbações na representação bidimensional também são imperceptíveis na representação como áudio.

Sobre como a classificação pode ser protegida, Esmaeilpour, Cardinal e Koerich (2019) estudam a resiliência de um classificador SVM (*support vector machine*) para classificação de espectrogramas com exemplos adversários, sugerindo técnicas possíveis, cuja mais relevante aqui é a possibilidade de pré-processar os espectrogramas antes da classificação, alterando as informações de cor presentes na imagem. Esse tipo de pré-processamento afeta a intensidade dos valores dos pixels e altera as cores presentes na imagem, enquanto mantém a forma da figura principal. Na prática, como a perturbação atua alterando uma quantidade pequena de pixels em locais específicos da imagem, é possível que os que têm um impacto significativo em um certo espaço de cores não tenha tanto efeito em outro, diminuindo suas consequências.

Perspectivas adicionais são vistas por Akhtar e Mian (2019), que além de descrever possíveis ataques e soluções, demonstram como podem existir na prática e por que é possível que essas alterações sutis possam ter o efeito que tem. Entre as soluções se destaca o método de treinar a rede usando exemplos adversários além dos normais, na esperança de que isso tornará a rede mais preparada para receber dados alterados no momento do teste/utilização.

Yuan et al. (2019) resumem os métodos de geração de ataques, trazendo algumas definições de tipos e nomenclatura. Por fim, Sun, Tan e Zhou (2018) demonstram alguns exemplos existentes de aplicação de ataques em situações reais e as consequências que podem ter no modelo.

3. Ataques analisados

A existência de ataques adversários não é um comportamento necessariamente inesperado dentro do funcionamento das redes neurais. Modelos profundos não "enxergam" em um conjunto de dados as mesmas características que um ser humano percebe; assim, o que para nós aparenta ser uma perturbação insignificante e imperceptível pode ser uma mudança considerável dentro dos padrões que a rede usa para avaliar as mesmas informações. Ainda assim, essas perturbações podem ocasionar uma série de erros, e portanto, devem ser tratadas de alguma forma. Como na aprendizagem profunda as características utilizadas são definidas ao longo do treinamento, encontrar maneiras de mitigar esse problema é um desafio complexo.

Existem vários tipos de perturbações que podem ser utilizadas de forma adversária na entrada das redes. Dentre os analisados aqui, a característica que possuem em comum é que não são ruídos aleatórios, mas sim construídos propositalmente para aumentar o erro. Por consequência, a construção desses exemplos requer conhecimento da rede - são considerados ataques de "caixa branca" (AKHTAR; MIAN, 2019) em razão disso.

Pode-se pensar que o ataque é como um "pior caso" de perturbações pequenas que causam erro na maior parte das situações.

3.1 Fast Gradient Sign Method (FGSM)

Um ataque comum e efetivo é o FGSM ou *Fast Gradient Sign Method*, que utiliza os gradientes da rede para encontrar o padrão que resultará na maior chance de erro para uma dada imagem de entrada (CHAKRABORTY et al, 2018). Aqui nota-se que a predição original da rede é usada para encontrar os valores das perturbações, significando que, como mencionado anteriormente, não é um ruído aleatório, e sim um ruído construído para maximizar a chance de erro.

Na Figura 1 é possível observar à esquerda um espectrograma normal, gerado a partir do áudio; à direita, uma perturbação gerada através do *Fast Gradient Sign Method*, que, com base na imagem do espectrograma normal, corresponde à modificação que, ao ser sobreposta com opacidade reduzida à essa imagem, deve causar o erro na classificação. Na Figura 2 é possível ver o resultado disso, isto é, o exemplo adversário.

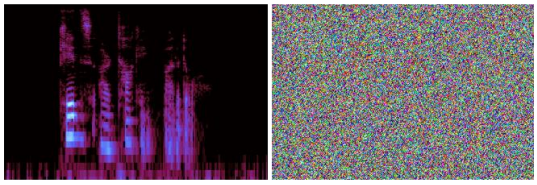

Figura 1 – Espectrograma normal e perturbação com FGSM

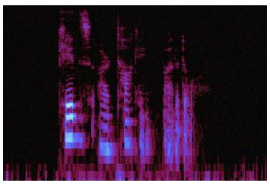

Figura 2 – Espectrograma adversário resultante

Essas duas figuras tornam evidente a imperceptibilidade da perturbação adversária, representando aparentemente o mesmo espectrograma, e em uma visão inicial, as duas imagens parecem ser idênticas. Um ser humano tipicamente não enxergaria a diferença sutil nos pixels da imagem, e ainda assim, a segunda imagem muito provavelmente seria classificada incorretamente pela rede.

O gradiente em si deriva a função de perda em relação aos parâmetros, isto é, é um vetor que mostra em qual direção a função de perda aumenta mais rapidamente; isso possibilita criar um objeto de perturbações que, somado ao espectrograma original, vai resultar em uma perda significativa. Assim, para uma dada amostra, é necessário ter a predição do modelo (e probabilidade percentual dada para cada classe).

3.2 Basic Iterative Method (BIM)

O BIM ou *Basic Iterative Method* é uma variação do FGSM, de modo que aplica os mesmos princípios de criar perturbações a partir do gradiente, sendo a principal diferença o fato que é aplicado repetidas vezes, mas em passos mais sutis. Em outras palavras, é como se fosse aplicado um FGSM sobre a imagem do espectrograma, mas com uma modificação pequena; em seguida, sobre a imagem já alterada, o processo é repetido, novamente com uma variação pequena em relação à anterior, e assim sucessivamente até se completarem os passos (CHAKRABORTY et al, 2018).

Visualmente, se fosse observada a diferença entre a figura do espectrograma antes e depois de aplicar a perturbação, as imagens teriam a mesma forma das figuras mostradas para o FGSM: dois espectrogramas aparentemente idênticos a uma visão inicial, mas com um ruído tênue que faz a amostra ser classificada incorretamente.

A distinção nesse caso é o fato de que o gradiente é calculado novamente sobre cada iteração de modo que cada uma individualmente produz uma perturbação que maximiza o erro. Isso significa que o resultado da classificação é testado em cada passo para gerar o erro com base na modificação anterior. Mesmo seguindo princípios semelhantes, o BIM tende a ser um ataque ligeiramente mais impactante do que o FGSM.

4. Método Proposto

O método consiste em determinar as diferenças de desempenho da classificação em diversas situações. Nesse caso, a informação relevante é a perda de desempenho entre um teste com exemplos normais (dados sem perturbações) e com exemplos adversários (dados com perturbações), de modo que a sequência dos testes é da seguinte forma:

a) Avaliar o desempenho da rede original com exemplos normais;

b) Avaliar o desempenho da rede original com exemplos adversários, e verificar a diferença no percentual de acerto;

c) Aplicar a modificação na rede que pode diminuir a suscetibilidade aos exemplos adversários com base na literatura relevante (essa modificação pode ser tanto nos dados de entrada, como no modo que é feito o treinamento);

d) Avaliar novamente o desempenho, com exemplos normais e adversários, e verificar a perda no percentual de acerto quando usados exemplos adversários;

e) Comparar a perda antes e depois da modificação.

Esse processo é repetido para cada método de defesa testado, a fim de comparar o acerto antes da modificação (para exemplos normais e adversários) e o acerto após a modificação (também em normais e adversários). Como a representação estudada é a forma bidimensional, isto é, os espectrogramas das amostras de áudio, esses métodos são técnicas que agem sobre imagens.

Além de testar com exemplos com perturbações adversárias, serão também feitos testes em exemplos com ruídos aleatórios. O objetivo disso é avaliar a transferibilidade das técnicas de proteção contra ataques adversários para outros casos de perturbações, que não são necessariamente direcionados; isso é uma forma de verificar se, de fato,

aumentar a robustez da rede contra exemplos adversários também aumenta a robustez do sistema de maneira geral, com outras situações de ruído.

4.1 Treinamento Completo com Exemplos Adversários

O modo mais simples e objetivo de proteção contra perturbações adversárias é gerar os próprios exemplos adversários na base de treinamento e usar esses dados acrescidos dos exemplos da base original (AKHTAR; MIAN, 2019).

O desafio neste caso é que é necessário criar esses exemplos, para que existam tanto os exemplos normais como os adversários equivalentes de toda a base de treinamento. Feito isso, o procedimento em si consiste em simplesmente unir a base de treinamento ordinária (X_{train}) e a base de treinamento com perturbações adversárias (X_{train_adv}) em um único vetor, e treinar a rede com esse conjunto, denominado "conjunto completo".

Como em qualquer caso isso aumenta o número de amostras da base de dados, esse método pode também ser considerado uma forma de *data augmentation*, ampliando a quantidade de dados disponíveis para o treinamento.

4.2 Pré-processamento Espectrogramas

Uma maneira de tornar as perturbações menos "visíveis" pela rede é mapear o espectrograma previamente em múltiplos espaços de cor diferentes (ESMAEILPOUR; CARDINAL; KOERICH, 2019). No caso deste trabalho, o espectrograma original é passado para três espaços de cor diferentes. Estando originalmente em BGR (*Blue-Green-Red*, e transformado em HSV (*hue-saturation-brightness*), XYZ, e YCrCb (luminância R-Y e B-Y). Somando esses três espaços ao original, são totalizados quatro espectrogramas provenientes de cada sinal de áudio original.

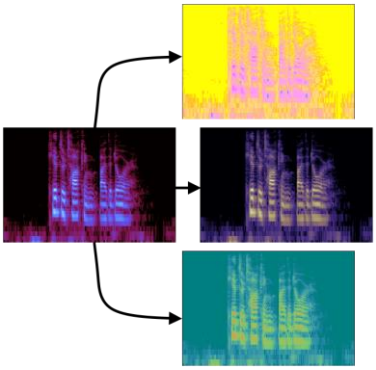

Figura 3 – Pré-processamento espectrogramas

Dessa forma, a rede é treinada com todos os quatro conjuntos de espectrogramas. Ao testar, o espectrograma na entrada é convertido para os outros três, e cada um é classificado pela rede, fazendo então um voto com base nas probabilidades encontradas para cada imagem a fim de determinar a classe escolhida para essa amostra.

4.3 Feature Squeezing

Uma característica marcante de perturbações adversárias é o fato de serem alterações sutis – caso fossem muito significativas e perceptíveis, o dado de entrada já se tornaria completamente diferente e, por consequência, seria outro som ao invés de simplesmente um sinal normal acrescido de ruído.

Com isso em mente, um modo de tornar as redes menos sensíveis a esses ruídos é torná-las menos sensíveis de maneira geral. Isto é, se a quantidade de características presentes no espectrograma que podem ser usadas pela rede for menor, é possível que ela também

seja menos afetada por exemplos adversários (CHAKRABORTY et al, 2018).

Naturalmente, isso pode acarretar em uma perda de desempenho nos exemplos normais, visto que teoricamente a rede terá menos características para basear seu processo de classificação. Ao mesmo tempo, se o processo tiver um efeito substancial na mitigação de exemplos adversários, isso pode ser suficientemente vantajoso para compensar mesmo se houverem perdas no desempenho normal, visto que isso significaria uma certa robustez frente a certas condições de ruído que podem existir.

Como os espectrogramas são essencialmente imagens, isso pode ser feito de algumas formas, por exemplo diminuindo a profundidade de cores nos pixels da imagem, ou usando um filtro de suavização/desfoque na mesma. Tal técnica causa uma perda de informação na representação do sinal – treinando a rede dessa maneira, o objetivo é que a rede seja menos sensível e "ignore" detalhes menores.

Assim, a abordagem aqui será a aplicação de um filtro gaussiano, de modo que o que antes seriam múltiplas informações na entrada acabam sendo combinados devido ao desfoque. Para os testes será variado o tamanho do kernel gaussiano (ex. 3 pixels, 5 pixels, e assim sucessivamente).

4.4 Desfoque Somente Antes da Utilização

O estudo do *feature squeezing* trouxe reflexões sobre como diminuir a sensibilidade da rede pode ser um princípio útil, e se existiriam outros modos de aplicar essa ideia. Como o intuito do processo é fazer com que as redes não "enxerguem" as pequenas variações que os ataques causam, surgiu a hipótese de não alterar o treinamento, mas aplicar um desfoque apenas antes da utilização/teste. Ou seja, treinar a rede com exemplos normais, mas quando ela for usada, os espectrogramas são desfocados.

O princípio por trás desse novo método é analisar a possibilidade de tirar parte da informação útil das imagens somente no teste e, se isso tiver um efeito positivo, mostraria que não é necessariamente obrigatório fazer modificações no modo que a rede é projetada para ainda assim fornecer algum grau de resistência contra perturbações adversárias. A rede é treinada normalmente, porém ao utilizá-la a imagem é alterada antes da entrada do classificador. A área de ação está fora da rede – o desfoque seria feito pelo sistema que a está aplicando, antes de começar a classificação em si.

Uma vantagem desse método seria que, por não exigir que a rede seja retreinada, os custos computacionais para aplicar a alteração são mínimos. Em uma aplicação real, pode-se facilmente ajustar o desfoque até alcançar um parâmetro que seja aceitável, sem a necessidade do esforço para alterar a rede ou os dados e então treiná-la novamente.

5. Resultados experimentais

Como parâmetro de comparação, é necessário primeiro verificar o funcionamento de uma rede classificando emoções em áudio em condições normais, sem aplicar nenhum tipo de ruído adicional na entrada. A base utilizada é a RAVDESS (*Ryerson Audio-Visual Database of Emotional Speech and Song*), que contém um total de 1440 amostras de áudio, representando oito possíveis emoções: neutralidade, calma, felicidade, tristeza, raiva, medo, nojo, e surpresa (LIVINGSTONE; RUSSO, 2018).

Alguns parâmetros devem ser definidos para realizar com precisão as comparações necessárias. Primeiramente, as várias classes de emoções serão divididas em somente duas: **positivas** e **negativas**. As faixas neutras, portanto, não são usadas. Em segundo lugar, deve-se definir o protocolo experimental em relação à divisão das amostras. Os áudios da base RAVDESS são interpretados por 24 atores e atrizes diferentes. Com o

intuito de promover uma análise independente do ator, todas as faixas de um único indivíduo são usadas somente no treinamento ou no teste – isto é, se um ator é escolhido para fazer parte do treinamento, todos os áudios desse ator são usados exclusivamente para treinamento, e se é escolhido para teste, suas faixas somente serão usadas no teste.

Assim, os 24 atores são divididos de forma aleatória com 70% deles sendo usados somente no treinamento (16 atores) e 30% usados no teste (8 atores).

5.1 Classificação em condições normais

A classificação é feita através de uma rede neural convolucional, aceitando a imagem bidimensional do espectrograma gerado por transformada rápida de Fourier a partir do sinal de áudio. O espectrograma completo é usado, ou seja, uma imagem individual para cada faixa de áudio; não são divididos em vários espectrogramas para evitar que isso seja um fator que interfira na análise principal, que é o efeito da perturbação adversária.

Mais detalhadamente, a rede apresenta arquitetura descrita na Tabela 1, a seguir. Os espectrogramas na entrada possuem altura de 288 pixels e largura de 432 pixels.

Tabela 1 – Arquitetura da CNN para testes

Tipo da camada	**Forma da saída**	**Nº params.**
Conv2D	(288, 432, 28)	784
Ativação	(288, 432, 28)	0
Normalização	(288, 432, 28)	112
Conv2D	(288, 432, 28)	7084
Ativação	(288, 432, 28)	0
Normalização	(288, 432, 28)	112
MaxPooling2D	(144, 216, 28)	0
Dropout	(144, 216, 28)	0
Conv2D	(144, 216, 64)	16192
Ativação	(144, 216, 64)	0
Normalização	(144, 216, 64)	256
Conv2D	(144, 216, 64)	36928
Ativação	(144, 216, 64)	0
Normalização	(144, 216, 64)	256
MaxPooling2D	(72, 108, 64)	0
Dropout	(72, 108, 64)	0
Conv2D	(72, 108, 128)	73856
Ativação	(72, 108, 128)	0
Normalização	(72, 108, 128)	512
Conv2D	(72, 108, 128)	147584
Ativação	(72, 108, 128)	0
Normalização	(72, 108, 128)	512
MaxPooling2D	(36, 54, 128)	0
Dropout	(36, 54, 128)	0
Achatamento	(248832)	0
Densa	(2)	497666
		Parâmetros totais: 781854
		Parâmetros treináveis: 780974
		Parâmetros não-treináveis: 880

Para o objetivo dessa pesquisa, os resultados da rede em condições típicas não precisam ser particularmente altos, sendo que o enfoque está na **diferença** entre o desempenho com exemplos normais e com exemplos adversários. Mais relevante que o resultado absoluto é a mudança no acerto percentual nas diferentes condições testadas.

Assim, quanto menor for essa diferença menor o efeito negativo das perturbações adversárias sobre a rede, e maior a sua robustez contra eles.

5.2 Resultados FGSM

Nesta seção são apresentados os resultados do sistema com exemplos adversários gerados pelo *Fast Gradient Sign Attack*. As condições na tabela são:

a) Inicial: Desempenho normal da rede sem modificações. Esse é o parâmetro de comparação: o objetivo dos métodos é fazer com a diferença entre o desempenho normal e adversário seja, no mínimo, menor que nessa condição. Esse desempenho é calculado através do percentual de acerto da rede na base de testes (30% da base total);

b) TCEA: Treinamento completo usando exemplos adversários;

c) PPE: Pré-processamento dos espectrogramas;

d) FS-k: *Feature Squeezing*, onde k é o tamanho do kernel do desfoque gaussiano;

e) DSAU-k: Desfoque somente antes da utilização, onde k é novamente o tamanho do kernel do desfoque gaussiano utilizado.

Dessa forma, são apresentados, para o ataque FGSM, o percentual de acerto da rede com exemplos normais, o percentual com exemplos adversários, e a diferença entre esses dois acertos, que representa o que o sistema perde ao aplicar ataques adversários.

Tabela 2 – Resultados FGSM

Condição	Normais	Adversários	Diferença
Inicial	77,67%	2,23%	**-75,44%**
TCEA	79,01%	51,33%	**-27,67%**
PPE	65,62%	34,37%	**-31,25%**
FS-3	76,33%	69,86%	**-6,47%**
FS-5	74,33%	72,99%	**-1,33%**
FS-7	74,10%	71,65%	**-2,45%**
FS-9	72,32%	72,22%	**-0,09%**
FS-15	70,08%	69,42%	**-0,67%**
DSAU-3	75,44%	57,58%	**-17,85%**
DSAU-5	72,54%	61,83%	**-10,71%**
DSAU-7	72,54%	64,28%	**-8,25%**
DSAU-9	72,09%	65,17%	**-6,92%**
DSAU-15	69,64%	65,84%	**-3,79%**

Todos os métodos obtiveram sucesso em diminuir a diferença entre o desempenho com amostras normais e o desempenho com amostras adversárias, porém em alguns deles isso veio com o custo da diminuição de desempenho em exemplos normais.

O treinamento completo com exemplos adversários e normais tem uma característica interessante, que é uma leve melhora no desempenho com exemplos normais. Com exemplos adversários, ele também possui um bom desempenho: a diferença foi de apenas (aproximadamente) 27%, o que é um resultado positivo comparado à diferença substancial de 75% nas condições normais.

O pré-processamento dos espectrogramas em diferentes espaços de cor tem como ponto

negativo uma diminuição no desempenho de exemplos normais. Contudo, este ainda conseguiu diminuir, de certa forma, o impacto dos ataques: a diferença foi de 31%, pior que outros métodos, mas ainda melhor que os 75% originais. É possível que as representações de cores escolhidas não tenham sido ideais, o que pode ter ocasionado uma perda de informação maior que a esperada.

O *feature squeezing* teve o melhor resultado no quesito de diminuir a diferença entre o desempenho com exemplos ordinários e com exemplos alterados. Porém, diminuir a sensibilidade do modelo tem o custo de um desempenho geral pior, gradativamente diminuindo a taxa de acerto à medida que a simplificação era maior. Caso fosse utilizado em uma aplicação prática, seria necessário encontrar um equilíbrio entre a perda de performance da classificação e a proteção contra as perturbações. Com essas considerações em mente, é uma técnica com efeitos positivos.

Por fim, o desfoque nos espectrogramas antes da utilização não se mostrou vantajoso, mas pode ter seus usos. Ele possui uma perda gradativa de desempenho normal à medida que o desfoque é mais intenso, tal qual os casos de *feature squeezing*, mas não diminuiu o impacto dos exemplos adversários tanto quanto eles. Ainda assim, a maior diferença desse método (aproximadamente 18%) é menor que os 75% das condições normais. Caso não exista tempo hábil para retreinar a rede, é possível aplicar esse tipo de desfoque nos espectrogramas de forma a ter uma "solução rápida", sem esforços computacionais significativos ou tempo para desenvolver um recurso melhor.

5.3 Resultados BIM

Na sequência serão apresentados os resultados utilizando exemplos adversários gerados através do *Basic Iterative Method*. A nomenclatura das condições permanece a mesma da seção anterior. Novamente, a condição inicial é a que serve de comparação para com todas as outras, sendo que este é o comportamento normal da rede sem qualquer tipo de modificação com intuito de aumentar sua robustez.

Assim, seguem as taxas de acerto do modelo com exemplos normais e adversários em cada caso de teste, e a respectiva diferença entre elas. Essa diferença pode ser entendida como o impacto do ataque adversário, ou seja, a medida de quanto o sistema perde em virtude da aplicação do ataque.

Tabela 3 – Resultados BIM

Condição	Normais	Adversários	Diferença
Inicial	77,67%	0,22%	**-77,45%**
TCEA	78,34%	51,11%	**-27,23%**
PPE	65,62%	10,04%	**-55,58%**
FS-3	76,33%	72,76%	**-3,57%**
FS-5	74,33%	71,42%	**-2,90%**
FS-7	74,10%	72,66%	**-1,44%**
FS-9	72,32%	71,52%	**-0,79%**
FS-15	70,08%	69,64%	**-0,44%**
DSAU-3	75,22%	55,80%	**-19,42%**
DSAU-5	72,54%	61,16%	**-11,38%**
DSAU-7	72,32%	64,06%	**-8,25%**
DSAU-9	72,32%	64,73%	**-7,58%**
DSAU-15	69,42%	64,73%	**-4,68%**

É possível observar que, em linhas gerais, os resultados com o BIM apresentam um comportamento semelhante aos resultados do FGSM. Por ser um ataque mais robusto que o FGSM normal, seu impacto tende a ser mais significativo, o que pode ser observado pela performance dos exemplos adversários na condição inicial, onde o acerto do modelo cai para quase 0%.

Ainda assim, os métodos continuam se mostrando geralmente efetivos em diminuir o impacto dos ataques. Uma discrepância notável nos comportamentos é no caso do pré-processamento dos espectrogramas: enquanto no FGSM o pré-processamento dos espaços de cor do espectrograma tiveram um efeito razoável, baixando a diferença de aproximadamente 75% para 28%, no caso do BIM a diferença continua alta, em 52%.

Quanto aos demais métodos, o treinamento completo com exemplos adversários e normais continua sendo uma forma de aumentar o desempenho do sistema de forma geral. Ao mesmo tempo que o impacto do ataque adversário caiu para aproximadamente 27%, o acerto em exemplos normais também é superior ao acerto com exemplos normais da rede original. O *feature squeezing* continua tendo um *trade-off* entre a intensidade da simplificação das características e a taxa de acerto, mas em todos os casos consegue mitigar substancialmente o efeito dos ataques. É possível observar que, quanto maior a simplificação das imagens, menor o desempenho, mas também diminui cada vez mais a diferença ocasionada pelo ataque adversário, até o momento que tende a próximo de zero. O desfoque antes da utilização, novamente, tem um efeito pior que o *feature squeezing*, mas ainda carrega a característica de poder ser aplicado facilmente em qualquer sistema sem exigir novo treinamento. Essa facilidade é algo que poderia ser um diferencial em certas situações onde o tempo para implementação é um fator.

5.4 Transferibilidade de técnicas contra perturbações adversárias em casos de ruído aleatório

Um dos possíveis benefícios do estudo de ataques adversários é tornar a rede mais robusta em casos gerais. Para verificar se os métodos aplicados possuem alguma efetividade com exemplos além das perturbações adversárias testadas, foram realizados experimentos também com casos de ruído aleatório.

Para tanto, as técnicas utilizadas são exatamente as que foram construídas para funcionar com os exemplos adversários – por exemplo, a condição "TCEA-FGSM" é a condição de teste do treinamento completo com exemplos adversários gerados via FGSM, e a condição "TCEA-BIM" é a rede do treinamento com exemplos adversários gerados por BIM. Nos casos de *feature squeezing* e desfoque antes da utilização, as condições de treinamento da rede eram iguais em ambos os casos, mudando apenas seu desempenho contra exemplos com FGSM e com BIM; assim, eles são mostrados somente uma vez na nova bateria de testes. Agora, essas condições serão testadas com outro tipo de alteração nos espectrogramas. Para tal, será aplicada sobre as imagens uma sobreposição de ruído *salt-and-pepper* construído de forma aleatória.

Embora aqui não se espera que o ruído seja tão impactante – por não utilizar nenhuma informação da rede e por não ser gerado para causar erro – essa é uma comparação interessante por dois motivos. Primeiro, para evitar tendenciosidade nos experimentos: é necessário verificar que o comportamento dos métodos funciona em diferentes casos de alterações inesperadas sobre a imagem, e não somente os exemplos adversários especificamente gerados para o teste. Segundo, como exemplos adversários tendem a criar os "piores casos" de erro sem alterar fundamentalmente os dados de entrada, nem sempre as situações reais de operação terão perturbações tão problemáticas, então é válido provar que as técnicas podem aperfeiçoar a performance também nesses casos.

Tabela 4 – Desempenho com Ruído Aleatório

Condição	Normais	Aleatórios	Diferença
Normal	77,67%	69,64%	**-8,03%**
TCEA-FGSM	79,01%	74,99%	**-4,02%**
TCEA-BIM	78,34%	74,10%	**-4,24%**
PPE	65,62%	62,94%	**-2,67%**
FS-3	76,33%	65,40%	**-10,93%**
FS-5	74,33%	64,90%	**-9,42%**
FS-7	74,10%	65,41%	**-8,69%**
FS-9	72,32%	65,40%	**-6,92%**
FS-15	70,08%	66,83%	**-3,25%**
DSAU-3	75,44%	72,54%	**-2,90%**
DSAU-5	72,54%	71,42%	**-1,11%**
DSAU-7	72,54%	70,08%	**-2,45%**
DSAU-9	72,09%	69,64%	**-2,45%**
DSAU-15	69,64%	66,74%	**-2,90%**

Como esses ruídos não são gerados de forma a especificamente induzir o erro, a diferença no caso normal é relativamente pequena, mas ainda assim, é possível ver que a maioria ainda consegue diminuir a diferença entre o desempenho em casos normais e casos com perturbação, à exceção dos dois primeiros casos de *feature squeezing*. Isso demonstra que técnicas de proteção contra exemplos adversários também possuem um efeito em outros casos diferentes de alteração sobre o espectrograma original.

Novamente, se destaca o treinamento completo com exemplos adversários e normais, tanto o que foi treinado com amostras de FGSM como o que foi treinado com amostras de BIM. Não somente ambos tiveram um desempenho superior à rede normal como também reduziram a perda ocasionada por efeitos com ruído.

6. Conclusão

A análise do comportamento de redes neurais frente a exemplos adversários não é relevante somente em casos em que esses ataques são especificamente gerados para causar o maior erro possível, ou em casos onde existe alguma intenção externa de quebrar o funcionamento da classificação. Esse estudo é também uma forma de explorar a robustez geral do sistema e os fatores que afetam seu desempenho. Existem inúmeras condições de ruído e de alterações que as amostras de áudio podem sofrer em uma situação real. Se o classificador está preparado para ter um desempenho aceitável mesmo no "pior caso" de exemplos adversários criados especificamente para quebrar sua funcionalidade, sua tendência é também ser mais robusto sob uma perspectiva geral.

Os experimentos realizados na classificação de emoções em áudio possibilitam várias reflexões sobre o efeito que as perturbações podem ter no funcionamento do sistema. Seu impacto em condições ordinárias é substancial: mesmo sem uma diferença visível no espectrograma do áudio, fazem com que as taxas de acerto caiam para próximas de zero. Das técnicas estudadas, algumas mitigam esse impacto de forma maior, e outras possuem um custo na performance geral, mas todas conseguem pelo menos diminuir a diferença entre as taxas com exemplos normais e as taxas com exemplos adversários.

Algo observado nos testes com ambos os tipos de ataque é a efetividade do treinamento completo com exemplos normais e adversários. Embora vários dos métodos

conseguiram reduzir o erro causado pelas perturbações, fazer uso de exemplos adversários durante o treinamento foi o único que não somente manteve o desempenho original da rede nos exemplos normais, como o melhorou. Como esse método possui um efeito positivo tanto no sentido de proteção contra ataques adversários como no sentido de *data augmentation* e aumentar o número de exemplos que a rede usa para treinar, é uma técnica que pode ser extremamente útil.

O pré-processamento de espectrogramas em vários espaços de cor não teve um desempenho ideal, mas isso pode ser em decorrência dos espaços de cor específicos escolhidos. Trabalhos futuros podem tentar usar outras condições de cor no teste, ou ainda acrescentar mais tipos, verificando se o uso de mais espaços de cor além dos quatro testados terá uma efetividade melhor.

A simplificação de características usadas pela rede é uma técnica efetiva, e que segue uma certa lógica considerando o conceito de ataques adversários como sendo perturbações que são, de fato, sutis e imperceptíveis. Se isso é verdadeiro, então tornar a rede menos sensível, e portanto menos influenciada por pequenos detalhes na imagem do espectrograma, é algo que realmente pode diminuir o efeito dessas pequenas alterações, e os resultados encontrados apoiam essa ideia. Uma linha de pensamento semelhante levou ao teste de uma nova técnica, em diminuir a qualidade das imagens através do desfoque gaussiano apenas no momento de utilizar a rede (e não de treinar), cujo objetivo era verificar se era possível implementar algum grau de defesa no sistema sem precisar retreinar a rede. Embora seu desempenho não foi tão alto quanto ao *feature squeezing*, esse método ainda conseguiu diminuir o efeito dos ataques, o que o torna uma opção possível em casos que não é possível facilmente retreinar o modelo ou aplicar alguma outra técnica para aumentar sua robustez.

Trabalhos futuros podem testar a combinação desses diferentes métodos e seu comportamento frente a outros ataques adversários diferentes. Verificar a performance de um modelo de classificação de emoções em diferentes condições de alterações e ruídos que podem existir nos espectrogramas é importante para torna-lo mais robusto, assim como um estudo essencial em outros tipos de classificação.

Referências

CHAKRABORTY, A. et al. *Adversarial Attacks and Defenses: A Survey*. ACM Computing Surveys, 2018.

KOERICH, K. M. et al. *Cross-Representation Transferability of Adversarial Perturbations: From Spectrograms to Audio Waveforms*. IEEE International Joint Conference on Neural Networks, 2019.

ESMAEILPOUR, M.; CARDINAL, P.; KOERICH, A.L. *A Robust Approach for Securing Audio Classification Against Adversarial Attacks*. IEEE Transactions on Information Forensics and Security, novembro de 2019.

AKHTAR, N.; MIAN, A. *Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey*. IEEE Access, fevereiro de 2019.

YUAN, X. et al. *Adversarial Examples: Attacks and Defenses for Deep Learning*. IEEE Transactions on Neural Networks and Learning Systems, setembro de 2019.

SUN, L.; TAN, M.; ZHOU, Z. *A survey of practical adversarial example attacks*. Springer Open Access, setembro de 2018.

LIVINGSTONE, S. R.; RUSSO, F. A. *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English*. 2018. Disponível em: https://doi.org/10.1371/journal .pone.0196391