MultiMagNet: Uma Abordagem Não Determinística na Escolha de Múltiplos *Autoencoders* para Detecção de Imagens Contraditórias

Gabriel R. Machado[1], **Eugênio Silva**[2], **Ronaldo R. Goldschmidt**[1]

[1]Seção de Engenharia de Computação (SE/8) – Instituto Militar de Engenharia (IME)
CEP.: 22290-270 – Rio de Janeiro - RJ

[2]Unidade de Computação (UComp) – Universidade Estadual da Zona Oeste (UEZO)
CEP.: 23070-200 – Rio de Janeiro - RJ

{gabriel.rmachado,ronaldo.rgold}@ime.eb.br, eugeniosilva@uezo.rj.gov.br

Abstract. *Studies reveal that machine learning algorithms can be induced to misclassify adversarial images. Recent research has created a method that incorporates a non-deterministic component to detect these images. The non-determinism hinders attackers from tracking the behavior of the defense method. However, this method has been bypassed by attacks conducted systematically, which succeeded in reproducing the defense's response. Thus, this paper aims at proposing a detection method that randomly considers multiple components, increasing the non-deterministic effect. Experimental results prove the robustness of the proposed method against the state-of-the-art adversarial attacks.*

Resumo. *Estudos mostram que os algoritmos de aprendizado de máquina podem ser induzidos a cometer erros de classificação diante de imagens contraditórias. Uma pesquisa recente criou um método que incorpora um componente não determinístico para detectar essas imagens. O não determinismo dificulta ao atacante mapear o comportamento do método. No entanto, essa abordagem tem sido superada por ataques aplicados de forma sistemática, que conseguiram abstrair a essência do comportamento de defesa. Assim, este artigo tem como objetivo propor um método de detecção que, ao considerar múltiplos componentes aleatórios, amplia o efeito do não determinismo. Resultados experimentais comprovam a robustez do método proposto frente aos ataques do estado da arte.*

1. Introdução

Os algoritmos de Aprendizado de Máquina vêm se desenvolvendo intensamente nos últimos anos e têm sido cada vez mais utilizados no apoio à decisão em tarefas de classificação e reconhecimento de imagens [Goodfellow et al. 2018]. A adoção desses algoritmos que, em alguns casos, apresentam desempenho superior ao de um humano [Karpathy 2014], vem se tornando comum na automatização de tarefas onde a segurança é um fator crucial, como o desenvolvimento de veículos autônomos, reconhecimento biométrico e sistemas de vigilância [Bojarski et al. 2016, Labati et al. 2018, Ding et al. 2018]. Entretanto, pesquisas recentes como [Szegedy et al. 2013, Goodfellow et al. 2014, Papernot et al. 2016a,

Carlini and Wagner 2017c] demonstram que o bom desempenho apresentado por esses algoritmos tem queda significativa perante imagens contraditórias (*adversarial images*). Imagens contraditórias contêm perturbações geradas maliciosamente, geralmente ínfimas e imperceptíveis ao olho humano, que induzem classificadores a cometer erros de classificação que podem acarretar em acidentes e/ou prejuízos em larga escala, como ilustrado pela Figura 1.

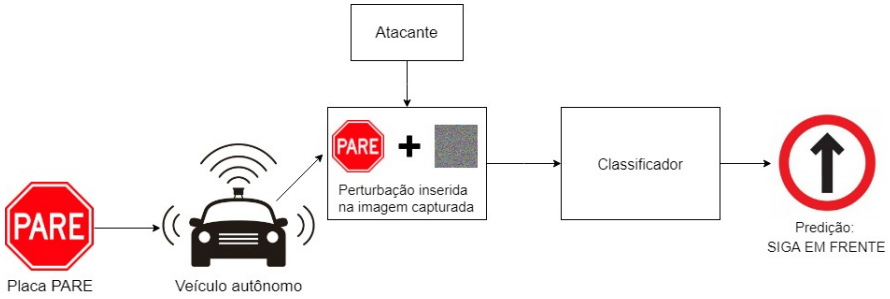

Figura 1. Exemplo de um cenário de ataque onde a geração de uma imagem contraditória por um atacante poderia causar um acidente de trânsito.

Como tentativa de mitigar os efeitos nocivos que os ataques contraditórios (ver Seção 2.3) podem causar, diversas estratégias de defesa têm sido propostas [Xu et al. 2018, Gong et al. 2017, Metzen et al. 2017]. De maneira geral, essas defesas consistem em detectar as imagens maliciosas antes que possam provocar os erros de classificação. Entretanto, [Carlini and Wagner 2017a, He et al. 2017] demonstram que essas defesas apresentam limitações importantes, principalmente porque (i) são preparadas para detectar apenas imagens contraditórias geradas por algoritmos de ataque específicos, e/ou (ii) empregam abordagens determinísticas para a detecção. Estratégias determinísticas de defesa utilizam sempre o mesmo procedimento de detecção, o que facilita o entendimento de seu *modus operandi* pelo atacante.

A fim de evitar a previsibilidade de comportamento, [Meng and Chen 2017] elaborou um método de detecção que possui um componente de comportamento aleatório que provoca o não determinismo, chamado *MagNet*. O *MagNet* introduz o não determinismo ao escolher aleatoriamente em tempo de execução, um único *autoencoder*, que baseado no erro de reconstrução, define um limiar para classificar as imagens como legítimas ou contraditórias. No entanto, [Carlini and Wagner 2017b] revela que, apesar do não determinismo, o *MagNet* pode ser superado por ataques conduzidos de maneira sistemática.

Diante do exposto, o presente trabalho levanta a hipótese de que introduzir múltiplos componentes aleatórios no método de detecção pode ampliar o efeito do não determinismo de maneira a reduzir a previsibilidade de comportamento e, consequentemente, tornar o método mais robusto aos diferentes tipos de ataques do que o *MagNet*. Assim, este trabalho propõe o *MultiMagNet*, um método de defesa que contém múltiplos componentes escolhidos aleatoriamente em tempo de execução. Para tanto, ao se inspirar no *MagNet*, o método proposto utiliza um comitê de *autoencoders* montado de forma dinâmica e aleatória para identificar a quantidade de perturbação presente nas imagens e, com base nessa informação, detectar a presença de imagens contraditórias. Os resultados dos experimentos realizados no *dataset* MNIST apontam para a validade da hipótese levantada, ao mostrar que o *MultiMagNet* superou o *MagNet* na maioria dos cenários

avaliados, sendo capaz de detectar perturbações geradas por diferentes tipos de ataques.

Este artigo está estruturado da seguinte maneira: a Seção 2 traz os conceitos básicos necessários para o entendimento do trabalho. A Seção 3 faz uma síntese das principais propostas de estratégias de defesa contra imagens contraditórias disponíveis na literatura. A Seção 4 apresenta em detalhes o método de defesa proposto. A Seção 5 descreve os experimentos realizados e discute os resultados obtidos. Finalmente, a Seção 6 traz as considerações finais, destacando as principais contribuições do trabalho e indicando alternativas de trabalhos futuros.

2. Conceitos Básicos

2.1. Redes Neurais Profundas

As Redes Neurais Profundas, que neste trabalho se limitam às Redes Neurais Convolucionais ou *ConvNets* [Lecun et al. 2010], são um tipo de arquitetura de rede neural destinada à resolução de tarefas e aplicações de visão computacional, como por exemplo a classificação de imagens. As RNPs realizam automaticamente a extração e aprendizagem de características utilizando basicamente dois grupos de camadas principais: camadas de convolução e camadas de *pooling*. Ao final, a camada densamente conectada atua de maneira semelhante a uma rede neural comum, produzindo como saída as probabilidades de a imagem de entrada pertencer a cada classe do problema em estudo. A Figura 2 apresenta um exemplo de arquitetura de uma RNP. Mais detalhes sobre o funcionamento dos componentes de uma RNP podem ser obtidos em [Goodfellow et al. 2016].

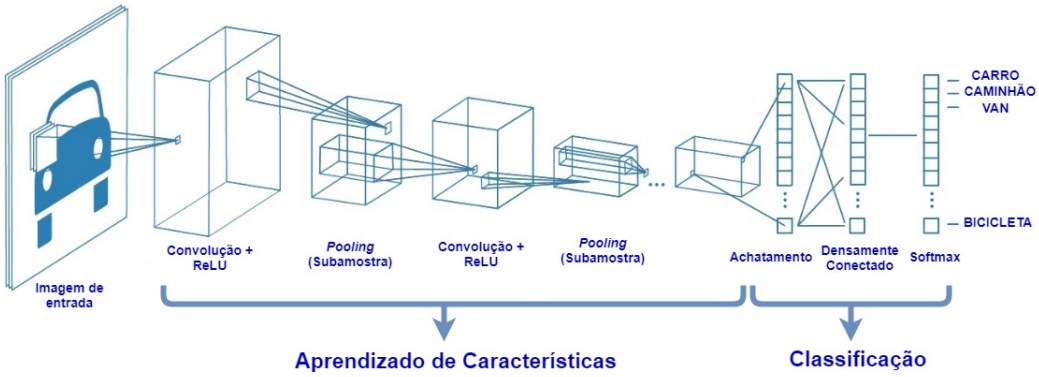

Figura 2. Exemplo de uma RNP com várias camadas convolucionais.[1]

2.2. Autoencoders

Autoencoders são redes neurais treinadas com o objetivo de reconstruir uma entrada x, gerando uma aproximação $\hat{x}$ como saída que produza o menor erro de reconstrução [Goodfellow et al. 2016]. Formalmente, um *autoencoder* $ae = d \circ e$ contém dois componentes, um *encoder* $e : \mathbb{S} \rightarrow \mathbb{H}$, e um *decoder* $d : \mathbb{H} \rightarrow \hat{\mathbb{S}}$, onde $\mathbb{S}$ é o espaço de entrada, $\mathbb{H}$ é o espaço da representação oculta aprendida pelo *autoencoder* e $\hat{\mathbb{S}}$ representa o espaço

[1]Adaptado de https://www.mathworks.com/discovery/convolutional-neural-network.html. Acessado em 23 de junho de 2018.

de entrada $\mathbb{S}$ reconstruído. O erro de reconstrução $ER_{ae(x)}$ é a diferença entre a entrada x e sua reconstrução $ae(x)$, representado pela Equação 1, onde p é a métrica de distância.

$$ER_{ae(x)} = ||x - ae(x)||_p \tag{1}$$

Autoencoders geralmente são usados como forma de redução de dimensionalidade e aprendizado de características, pois esses modelos são forçados a priorizar o que deve ser copiado da entrada, aprendendo as propriedades mais importantes dos dados [Goodfellow et al. 2016]. Nos experimentos realizados neste trabalho foram utilizados dois tipos de *autoencoders*: os *autoencoders* convolucionais, e os *denoising autoencoders*. Ambas as arquiteturas de *autoencoders* são semelhantes à arquitetura das Redes Neurais Convolucionais, entretanto os *denoising autoencoders* inserem um ruído[2] intencional nas imagens de treinamento. Esta medida evita que após o treinamento, os *autoenconders* se transformem em funções-identidade.

2.3. Imagens Contraditórias

Uma imagem contraditória é uma imagem que contém uma pequena perturbação, às vezes imperceptível aos olhos humanos, gerada por um atacante através de um algoritmo malicioso, com o objetivo de causar erros em algoritmos de classificação (ver Figura 3). Formalmente, dado um classificador F treinado com imagens legítimas x, tal que $x \in \mathbb{R}^{w \times h \times c}$, onde w e h são as dimensões da imagem e c a quantidade de canais de cor, gera-se uma imagem x', tal que $x' = x + \delta x$, onde δx é a perturbação e $F(x) \neq F(x')$. O ato de gerar imagens contraditórias e enviá-las ao classificador com o intuito de produzir erros de classificação é chamado de **ataque contraditório** (*adversarial attack)*.

Um atacante pode realizar basicamente dois tipos de ataques contraditórios, de acordo com a quantidade de informação a que tem acesso: (i) ataque de caixa-preta e (ii) ataque de caixa-branca. Em um ataque de caixa-preta, o atacante não tem conhecimento de que há uma defesa presente, e também não tem acesso aos parâmetros do classificador almejado. Em um ataque de caixa-branca, o atacante tem acesso às arquiteturas e aos parâmetros tanto da defesa quanto do classificador e usa esse conhecimento para gerar imagens contraditórias eficazes.

É importante destacar que um ataque contraditório pode ser não direcionado ou direcionado. Em um ataque não direcionado, uma imagem legítima x, pertencente a uma classe $y \in Y$, é perturbada para gerar uma imagem contraditória x', cujo objetivo é induzir o classificador a associar essa imagem a qualquer classe $y' \in Y$, desde que $y' \neq y$. Em um ataque direcionado, x' é gerada de forma a induzir o classificador a associá-la a uma classe específica $y' \in Y$, onde $y' \neq y$. Nos experimentos realizados neste trabalho foram utilizados apenas ataques não direcionados, por produzirem imagens contraditórias de maior dificuldade de detecção [Meng and Chen 2017, Carlini and Wagner 2017c].

[2]É importante ressaltar que neste trabalho, os termos **ruído** e **perturbação** designam formas diferentes de se corromper uma imagem. O ruído é qualquer distorção já naturalmente presente ou gerada em uma imagem de maneira não sistemática, *i.e.* sem utilizar um algoritmo malicioso. Uma perturbação é uma distorção sistemática gerada maliciosamente por um algoritmo de ataque, produzindo uma imagem contraditória.

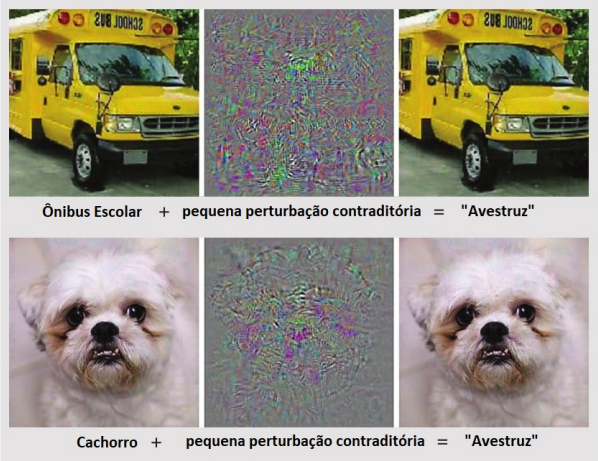

Figura 3. Perturbações maliciosas e geralmente imperceptíveis em uma imagem legítima fazem com que um modelo treinado cometa um erro de classificação. Adaptado de [Klarreich 2016].

2.4. Algoritmos de Ataques Contraditórios

Os algoritmos de ataques contraditórios são algoritmos que geram e inserem perturbações em imagens legítimas com a finalidade de enganar classificadores previamente treinados. Existem diversos algoritmos de ataques contraditórios na literatura, entretanto para os experimentos apresentados neste trabalho foram utilizados quatro, que são o estado-da-arte na geração de imagens contraditórias:

- ***Fast Gradient Sign Method* (FGSM)** [Goodfellow et al. 2014]: O FGSM é um algoritmo de ataque não-iterativo, cujas principais características são a complexidade linear e a capacidade de produzir perturbações maiores que aquelas produzidas por algoritmos iterativos. Dada uma imagem $x \in \mathbb{R}^{w \times h \times c}$, o FGSM gera uma imagem contraditória x' por meio da Equação 2:

$$x' = x - \epsilon \cdot sign(\nabla_x J(\Theta, x, y)) \tag{2}$$

  onde Θ representa os parâmetros da rede, y a classe associada a x, ϵ a quantidade máxima de perturbação que pode ser inserida na imagem x e $J(\Theta, x, y)$ a função de custo utilizada para treinamento da rede, que em um ataque de caixa-preta é um classificador auxiliar semelhante ao classificador sob ataque.

- ***Basic Iterative Method* (BIM)** [Kurakin et al. 2016a]: O BIM é semelhante ao FGSM, porém iterativo. Ao invés de executar um passo de tamanho ϵ na direção do gradiente descendente, como é feito pelo FGSM, vários passos α menores são executados e o resultado é limitado por ϵ, atuando como um teto para que a perturbação não exceda a quantidade desejada pelo atacante. Formalmente, o BIM é um método recursivo, que gera x' segundo a Equação 3:

$$x' = \begin{cases} x'_0 = 0 \\ x'_i = x'_{i-1} - clip(\alpha \cdot sign \nabla_x J(\Theta, x'_{i-1}, y)) \end{cases} \tag{3}$$

- ***DeepFool*** [Moosavi-Dezfooli et al. 2016]: A ideia do *Deepfool* consiste em encontrar a imagem legítima x que esteja mais próxima da fronteira de decisão no espaço e

atravessá-la, ou seja, perturbá-la sutilmente para que engane o classificador. Devido à alta dimensionalidade da imagem, é adotada uma abordagem iterativa por aproximação linear. A cada iteração, o *Deepfool* lineariza o modelo em torno do x' intermediário e calcula uma direção de atualização ótima no modelo linearizado. Em seguida, x' é atualizada nessa direção por um pequeno passo α.

- ***Carlini & Wagner Attack* (CW)** [Carlini and Wagner 2017c]: O CW representa o estado da arte em termos de geração de imagens contraditórias. Formalmente, o CW é um ataque iterativo em que, dada uma RNP F com a penúltima camada (*logits*) Z e uma imagem x pertencente à classe t, o ataque utiliza o gradiente descendente para resolver a Equação 4:

$$\text{minimizar } ||x - x'||_2^2 + c \cdot \ell(x') \tag{4}$$

onde a função de custo $\ell(x')$ é definida pela Equação 5:

$$\ell(x') = max(max\{Z(x')_i : i \neq t\} - Z(x')_t, -k) \tag{5}$$

onde o hiperparâmetro k se refere à taxa de confiança do ataque. Quanto maior o valor de k, maior é a transferibilidade da imagem contraditória, porém, maiores também são as perturbações.

3. Trabalhos Relacionados

Defender modelos de classificação contra imagens contraditórias não é uma tarefa trivial e várias estratégias já foram propostas na literatura. A seguir são apresentados os trabalhos mais relevantes sobre o assunto e suas respectivas abordagens.

- ***Adversarial Training*:** [Szegedy et al. 2013, Goodfellow et al. 2014, Kurakin et al. 2016b, Madry et al. 2017] propuseram o *adversarial training*, também conhecido como **otimização robusta**, que é uma forma de defesa proativa[3] que se baseia no *data augmentation*[4] para treinar classificadores com um conjunto de imagens legítimas e contraditórias, forçando o modelo a produzir as saídas corretas para as imagens maliciosas. Contudo, o *adversarial training* apresenta duas limitações: (i) é computacionalmente custoso e (ii) cria dependências do modelo com algoritmos específicos de ataque contraditórios.

- **Destilação Defensiva:** proposto por [Papernot et al. 2016b], é uma defesa proativa que treina um modelo F em um conjunto de amostras legítimas X e rótulos Y, gerando um vetor de saídas probabilísticas $F(X)$. O conjunto de rótulos Y então é substituído pelo conjunto $F(X)$, e um novo modelo F_d com a mesma arquitetura de F é criado e treinado com o mesmo conjunto de amostras X, porém com os rótulos probabilísticos $F(X)$. Após o treinamento, o modelo F_d obtido, denominado modelo destilado, produz as saídas probabilísticas destiladas $F_d(X)$. O modelo destilado

[3]Defesas contra imagens contraditórias se dividem em (i) defesas proativas, que visam tornar os classificadores mais robustos às imagens contraditórias e (ii) defesas reativas, que agem como detectores de imagens contraditórias, impedindo-as de chegar ao classificador.

[4]Procedimento realizado em um conjunto de dados para aumentar a quantidade de amostras utilizadas no treinamento de classificadores.

$F_d(X)$ é uma tentativa de ocultar o gradiente utilizado pelos algoritmos de ataque, entretanto, [Carlini and Wagner 2017a] mostraram que é possível gerar perturbações que enganem classificadores com destilação defensiva.

- ***Feature Squeezing*:** defesa reativa proposta por [Xu et al. 2018] que se baseia na hipótese de que a alta dimensionalidade da imagem contribui para um atacante gerar perturbações. Por isso, os autores utilizam basicamente dois métodos para reduzir a dimensionalidade das imagens e remover as perturbações que possam estar presentes: (i) redução dos bits de cor e (ii) suavização espacial. Através de um classificador, é realizada uma comparação entre a predição da imagem original x com a predição da versão x', reduzida pelo método (i), e com a predição de x'', reduzida pelo método (ii), utilizando um limiar pré-definido. Caso uma dessas comparações esteja acima do limiar, a imagem x é rotulada como contraditória e descartada antes de chegar ao classificador. Apesar desta defesa ter apresentado bons resultados diante do ataque CW, [He et al. 2017] mostraram que é possível evadí-la.

- ***MagNet*:** [Meng and Chen 2017] propuseram o *MagNet*, que é um método que utiliza duas camadas de defesa: (i) um detector que rejeita, utilizando um limiar pré-definido, as imagens distantes da fronteira de decisão, *i.e.* com maiores perturbações, e (ii) um reformador que, ao receber a imagem proveniente do detector, a reconstrói buscando eliminar possíveis perturbações remanescentes. Após a reconstrução, a imagem é enviada ao classificador. O *MagNet* escolhe aleatoriamente dois *denoising autoencoders* de um repositório, um para a camada de detecção e outro para a camada de reforma. Apesar do uso da aleatoriedade na escolha de ambos os *autoencoders*, [Carlini and Wagner 2017b] mostrou que é possível evadir essa defesa em um cenário de caixa-branca.

4. Método de Defesa Proposto

Denominado *MultiMagNet*, o método de defesa proposto neste artigo é uma evolução do *MagNet* [Meng and Chen 2017]. Baseia-se em uma estratégia não determinística para detecção de imagens contraditórias, que cria em tempo de execução um comitê formado por diferentes *autoencoders* escolhidos aleatoriamente. A Figura 4 ilustra graficamente as cinco etapas do processo de detecção de imagens contraditórias executado pelo *MultiMagNet*: *(i) Montar Time, (ii) Calcular Limiar, (iii) Reconstruir Imagem, (iv) Calcular Erros de Reconstrução e (v) Apurar Votos*. *Montar Time* e *Calcular Limiar* são etapas destinadas à calibração dos parâmetros da defesa. A seguir, as cinco etapas são explicadas em detalhes.

A primeira etapa, *Montar Time*, recebe como entrada um repositório S contendo n *autoencoders* com diferentes arquiteturas e hiperparâmetros, treinados previamente utilizando um conjunto de treinamento específico. Produz como saída um conjunto $R \subseteq S$ contendo m diferentes *autoencoders* escolhidos aleatoriamente, tal que $m \leq n$ e $m \bmod 2 = 1$. A restrição de que m seja ímpar tem como objetivo evitar empate no final da votação a ser realizada na etapa *Apurar Votos*.

Na etapa *Calcular Limiar*, o *MultiMagNet* recebe como entrada o conjunto R, contendo os m *autoencoders* escolhidos, e o conjunto de validação V, contendo somente imagens legítimas. Nesta etapa, as imagens de V são reconstruídas por cada *autoenco-*

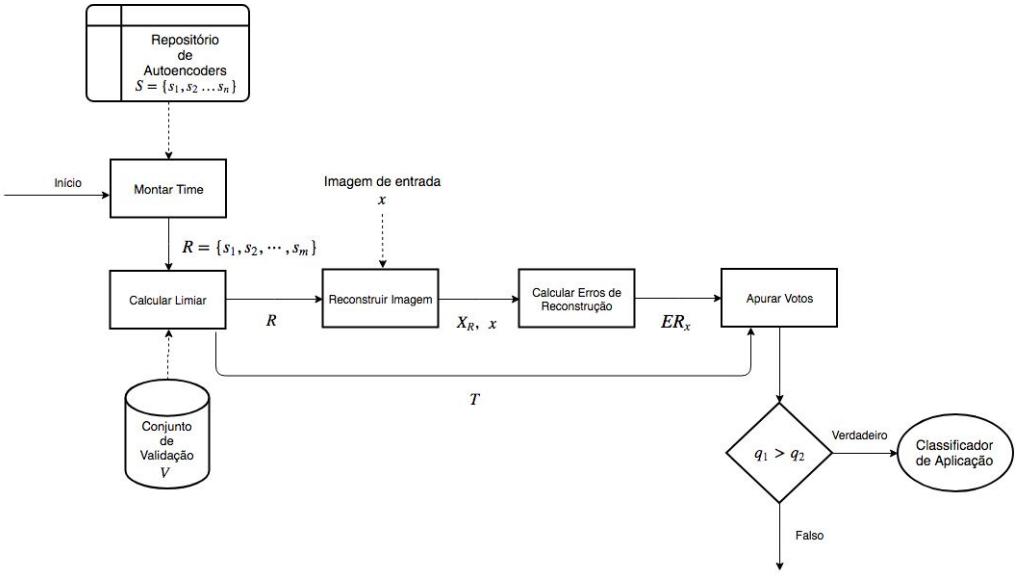

Figura 4. *MultiMagnet* - Visão macro-funcional do método de defesa proposto.

der $s_i \in R$, formando o conjunto V_{R_i}, onde $V_{R_i} = \{vr_i | vr_i = s_i(v_i), v_i \in V\}$ ($s_i(v_i)$ corresponde ao resultado da reconstrução da imagem v_i pelo *autoencoder* s_i*)*. Em seguida, são calculados os erros de reconstrução entre as imagens originais e suas versões reconstruídas por s_i, formando o conjunto $ER_i = \{||V - V_{R_i}||_p\}$, sendo p a métrica de distância definida pelo usuário. Finalmente, utilizando o parâmetro t_{fp} (também definido pelo usuário) que representa a porcentagem de falsos positivos permitida (*i.e.*, a quantidade de imagens legítimas classificadas como contraditórias), é produzido como saída o conjunto T de limiar(es), cujo conteúdo pode variar segundo duas abordagens distintas: a de (i) limiar múltiplo, onde considera-se os limiares associados a todos os *autoencoders* (i.e., $T = \{\tau_1, \tau_2, \cdots, \tau_m\}$); ou a de (ii) limiar mínimo, onde considera-se o menor entre os limiares associados a todos os *autoencoders* (i.e., $T = \{min\{\tau_1, \tau_2, \cdots, \tau_m\}\}$). Cada $\tau_i \in T$ representa o erro de reconstrução do *autoencoder* s_i, calculado a partir do conjunto ER_i em função do parâmetro de falsos positivos t_{fp}.

Uma vez calibrado pelas duas primeiras etapas, o *MultiMagNet* pode receber imagens de entrada a fim de identificar se são ou não imagens contraditórias. As próximas etapas descrevem esse processo.

A terceira etapa, *Reconstruir Imagem*, recebe como entrada uma imagem x, a ser avaliada, e o conjunto R contendo os m *autoencoders* escolhidos. Produz como saídas o conjunto X_R, formado pelas versões reconstruídas xr_i da imagem de entrada x por cada *autoencoder* $s_i \in R$.

A quarta etapa, *Calcular Erros de Reconstrução*, recebe como entradas o conjunto X_R e a imagem x, e retorna como saída os erros de reconstrução $ER_x = \{||x - xr_i||_p\}$, onde $xr_i \in X_R$, e $xr_i = s_i(x)$.

Por fim, a quinta etapa, *Apurar Votos*, recebe como entradas os conjuntos ER_x e T. Produz como saídas as variáveis q_1 e q_2, que representam as quantidades de votos que classificam x como legítima e como contraditória, respectivamente. Durante o processo, o limiar τ_k, (onde: $k = 1, ..., m$, caso a abordagem utilizada seja a de limiar múltiplo, ou $k = j$, onde $\tau_j = min\{\tau_1, \tau_2, \cdots, \tau_m\}$, caso a abordagem seja a de limiar mínimo)

é comparado com cada erro de reconstrução $er_i \in ER_x$. Caso $er_i < \tau_k$, q_1 recebe um voto; caso $er_i \geq \tau_k$, q_2 recebe um voto. Ao final do processo, a imagem x é classificada como legítima e enviada para o classificador da aplicação, se $q_1 > q_2$. Caso $q_2 > q_1$, x é considerada contraditória, sendo descartada em seguida. É importante relembrar que não há possibilidade de haver empate nessa etapa, já que a quantidade de votos é ímpar.

5. Experimentos e Resultados

O protótipo do *MultiMagNet* foi desenvolvido na linguagem Python, utilizando as bibliotecas *TensorFlow*, *Keras*, *SciKit-Learn*, *NumPy* e *SciPy*. Seu código-fonte encontra-se disponível para *download*[5]. Os experimentos foram todos realizados em uma única máquina com processador *Core* i7 3770, 8GB de memória RAM e uma placa de vídeo GTX 1060 6GB com 1280 *CUDA cores*.

O conjunto de dados escolhido para os experimentos foi o MNIST [LeCun et al. 1998], por ser um *dataset* amplamente utilizado em diversos trabalhos da área de defesa contra imagens contraditórias [Goodfellow et al. 2014, Meng and Chen 2017, Xu et al. 2018]. É formado por imagens monocromáticas legítimas de dígitos de 0 a 9 manuscritos. Todas as imagens possuem dimensões $28 \times 28 \times 1$, que representam 28 *pixels* de largura, 28 *pixels* de altura e 1 canal de cor, respectivamente, como ilustrado na Figura 4. O MNIST possui um total de 70.000 imagens distribuídas em 10 diferentes classes, cada classe representando um dígito. Nos experimentos, as 70.000 imagens foram particionadas da seguinte forma: as 51.000 primeiras imagens para treinamento dos *autoencoders*, as 9.000 imagens seguintes utilizadas para a definição do conjunto T e as 10.000 imagens restantes separadas para teste e avaliação dos modelos de defesa. Todas as imagens do MNIST foram normalizadas para terem os valores das intensidades dos seus *pixels* no intervalo $[0, 1]$, ao invés do intervalo original de $[0, 255]$. Após a normalização, os valores 0 e 1 indicam, respectivamente, *pixels* pretos e brancos. Valores intermediários indicam *pixels* em tons de cinza.

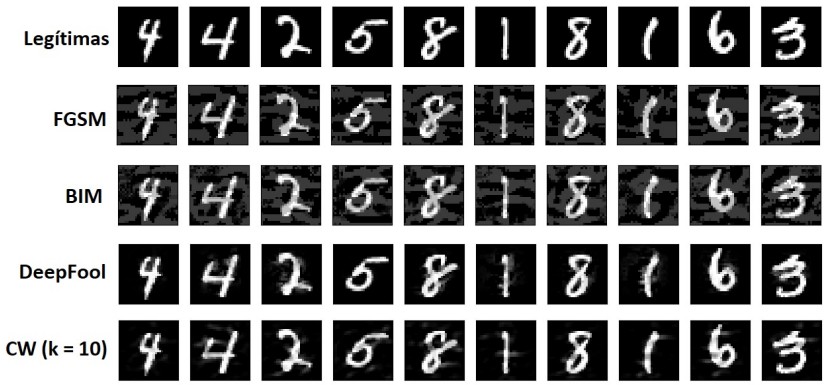

Figura 5. Exemplos de imagens do MNIST. A primeira linha contém imagens legítimas. As demais apresentam as versões perturbadas dessas imagens pelos ataques FGSM, BIM, *DeepFool* e CW, respectivamente.

Para realizar os experimentos de avaliação do *MultiMagNet*, além das imagens legítimas do MNIST, foi necessário enriquecer o conjunto de teste D com imagens geradas por diferentes métodos de ataque do estado da arte, conduzidos em um **cenário de**

[5] https://github.com/gabrielrmachado/MultiMagNet

Tabela 1. Parâmetros definidos de cada método de ataque

Ataque	Parâmetros
FGSM	$\epsilon = 0,2$
BIM	$\epsilon = 0,15, \alpha = 0,07$ e máximo de 50 iterações
DeepFool	$overshoot = 0,02$ e máximo de 50 iterações

caixa-preta. Os métodos utilizados foram: FGSM, BIM, *DeepFool* e cinco variações do método CW a partir do nível de confiança k: $CW(k = 0)$, $CW(k = 10)$, $CW(k = 20)$, $CW(k = 30)$, e $CW(k = 40)$. Para o enriquecimento, foi adotado o critério descrito a seguir. Das 10.000 imagens separadas para teste, as 2.000 primeiras foram selecionadas. Tais imagens foram rotuladas como *legítimas* e armazenadas em D_{Leg}. Em seguida, os métodos FGSM, BIM, *DeepFool* foram aplicados a cada uma das imagens de D_{Leg} e as imagens resultantes armazenadas em D_{FGSM}, D_{BIM} e $D_{DeepFool}$, respectivamente. Foram utilizadas as implementações dos referidos métodos que estão disponíveis na biblioteca *Cleverhans* [Papernot et al. 2018]. A Tabela 1 indica os parâmetros adotados na obtenção das imagens contraditórias desses métodos.

As imagens dos ataques CW foram diretamente extraídas do repositório *online* do *MagNet*[6]. Para cada variação do CW foi criado um conjunto com as 2.000 primeiras imagens associadas ao nível de confiança c correspondente: $D_{CW(k=c)}$, onde $c \in \{0, 10, 20, 30, 40\}$. Assim, ao todo, foram criados oito cenários de teste para simular oito ataques, a saber: $D_{Leg} \cup D_{ataq}$, onde $ataq \in \{FGSM, BIM, DeepFool, CW(k = 0), CW(k = 10), CW(k = 20), CW(k = 30), CW(k = 40)\}$.

O repositório S do *MultiMagNet* foi configurado com 10 *autoencoders* com diferentes arquiteturas e hiperparâmetros, previamente treinados com as imagens legítimas disponíveis no conjunto de treinamento. Dos 10 *autoencoders* presentes em S, 5 são *autoencoders* convolucionais (CAEs) e 5 *denoising autoencoders* (DAEs). O repositório do *MagNet* foi configurado com 2 *denoising autoencoders*, de acordo com o experimento apresentado em [Meng and Chen 2017] e a implementação disponibilizada *online*[7] pelos autores. Foi definido para a métrica p a norma l_1, por produzir melhores resultados no *MagNet* e *MultiMagNet*. A norma l_1 é definida como $p = ||x||_1 = \sum_i |x_i|$.

Foram feitos experimentos com as abordagens de limiar múltiplo e limiar mínimo, a fim de comparar a influência deste critério nos resultados da defesa proposta. As Tabelas 2 e 3 apresentam os melhores resultados obtidos pelo *MultiMagNet* em cada cenário de experimentação com as abordagens de limiar múltiplo e limiar mínimo, respectivamente. Em ambas, os desempenhos do *MultiMagNet* e do *MagNet* são expressos por meio da métrica de precisão em relação à classe negativa (*i.e.*, quantas são as imagens realmente contraditórias entre as imagens classificadas como contraditórias pelo método de detecção). Na Tabela 2, a segunda coluna mostra a quantidade de *autoencoders* que levou

[6] `https://www.dropbox.com/s/2x509u80g5zkuea/MagNet_support_data.zip?dl=0&file_subpath=%2Fattack_data`.

[7] Disponível em: `https://github.com/gabrielrmachado/MagNet`. Adaptado de `https://github.com/Trevillie/MagNet` para fins comparativos com a camada de detecção do *MagNet*.

o *MultiMagNet* a obter a precisão indicada. A terceira coluna, por sua vez, indica a taxa de falsos positivos t_{fp}. Por fim, a quarta e quinta coluna apresentam os resultados obtidos por meio da métrica de precisão em relação à classe negativa. Na Tabela 3, a coluna τ_{min} apresenta o valor do limiar mínimo de T, utilizado pelo *MultiMagNet* para classificar as imagens em legítimas ou contraditórias.

Tabela 2. Melhores resultados obtidos com a abordagem de limiar múltiplo

Cenário de Ataque	**# *autoencoders* escolhidos**	t_{fp}	**Precisão *MultiMagNet* (%)**	**Precisão *MagNet* (%)**
FGSM	3 e 5	0,001	100,00	100,00
BIM	3 e 5	0,001	100,00	100,00
DeepFool	3	0,1	74,30	67,80
CW (k = 0)	5	0,05	72,51	52,30
CW (k = 10)	3	0,05	91,10	69,15
CW (k = 20)	3	0,01	87,91	65,25
CW (k = 30)	3	0,01	95,25	77,75
CW (k = 40)	3	0,01	95,65	88,45

Tabela 3. Melhores resultados obtidos com abordagem de limiar mínimo

Cenário de Ataque	**# *autoencoders* escolhidos**	t_{fp}	τ_{min}	**Precisão *MultiMagNet* (%)**	**Precisão *MagNet* (%)**
FGSM	3	0,001	0,0335	100,00	100,00
BIM	3	0,001	0,3340	100,00	100,00
DeepFool	3	0,01	0,0184	84,95	32,55
CW (k = 0)	3	0,01	0,0184	80,11	29,00
CW (k = 10)	3	0,01	0,0184	91,90	47,25
CW (k = 20)	5	0,001	0,0262	96,70	40,75
CW (k = 30)	3	0,01	0,0184	98,65	77,75
CW (k = 40)	3	0,01	0,0218	100,00	88,45

Em uma comparação direta entre o *MultiMagNet* e o *MagNet*, percebe-se que o primeiro obteve resultados melhores ou iguais ao segundo em todos os cenários de todos os tipos de ataque de caixa-preta adotados, incluindo o ataque proposto por [Carlini and Wagner 2017b]. Na Tabela 2, a precisão do *MultiMagNet* ficou com média de $89,59\%$ e desvio-padrão de $10,79$ pontos percentuais, enquanto que *MagNet* obteve uma média de $77,59\%$, com desvios-padrão de $17,24$ pontos percentuais. Na Tabela 3, a precisão do *MultiMagNet* ficou com média de $94,04\%$ e desvio-padrão de $7,71$ pontos percentuais, enquanto que *MagNet* obteve uma média de $64,47\%$ e desvio-padrão de $30,26$ pontos percentuais. Tal comparação aponta para a validade da hipótese defendida neste trabalho.

Cabe ressaltar que nos cenários onde ambos os métodos obtiveram 100% de precisão as imagens foram geradas pelos ataques FGSM e BIM. Tal resultado se justifica, pois as perturbações geradas por estes ataques são maiores, e, portanto, são mais fáceis de se detectar utilizando a abordagem de cálculo do limiar utilizada por ambos os métodos.

Também vale a pena comentar que em vários cenários, a diferença entre os desempenhos do *MultiMagNet* e do *MagNet* foi superior a 40 pontos percentuais. Por exemplo, nos ataques *DeepFool* ($84,95\%$ contra apenas $32,55\%$) e CW $k = 20$ ($96,70\%$ contra apenas $40,75\%$). Tais resultados são indícios que reforçam para a validade da hipótese levantada neste trabalho.

A abordagem do limiar mínimo do *MultiMagNet*, quando comparada à sua abordagem de limiar múltiplo, proporcionou um aumento de aproximadamente 5% na média da precisão e uma redução de aproximadamente $28,5\%$ no desvio padrão. É importante mencionar também que na abordagem do limiar mínimo, houve uma diminuição dos valores da taxa de falsos positivos t_{fp} nos ataques *DeepFool* e CW, com os níveis de confiança $k = \{0, 10, 20\}$, em comparação à abordagem de limiar múltiplo. A diminuição dos valores da taxa t_{fp} e o aumento da precisão do *MultiMagNet* indicam que a abordagem do limiar mínimo torna a defesa mais precisa em detectar perturbações geradas por ataques mais elaborados, como o *DeepFool* e as variações do ataque CW.

Assim, diante dos comentários acima expostos, pode-se perceber que os resultados apresentados pelos métodos *MultiMagNet* e *MagNet* realmente apontam para a validade da hipótese levantada neste trabalho de que a introdução de múltiplos componentes aleatórios em um método de detecção pode ampliar o efeito do não determinismo de maneira a tornar esse método mais robusto aos diferentes tipos de ataques do que os métodos existentes.

6. Considerações Finais

Nos últimos anos, diversos trabalhos vêm demonstrando que os algoritmos de Aprendizado de Máquina podem ser propositalmente induzidos a cometer erros de classificação diante de imagens contraditórias (*i.e.*, imagens que contenham perturbações geradas de forma maliciosa por ataques de tipos variados). Embora várias defesas tenham sido criadas para detectar imagens contraditórias, a maioria delas têm sido superadas porque facilitam ao atacante mapear seu comportamento. A fim de evitar a previsibilidade de comportamento, uma iniciativa de pesquisa buscou criar um método de detecção não determinístico, chamado *MagNet* [Meng and Chen 2017]. Em essência, esse método possui um componente de comportamento aleatório que provoca o não determinismo. Estudos recentes revelam que, apesar do não determinismo, esse método de defesa tem sido superado por ataques que, ao serem aplicados de forma sistemática, conseguem abstrair a essência do comportamento de defesa.

Diante do exposto, o presente trabalho levantou a hipótese de que introduzir múltiplos componentes aleatórios no método de detecção pode ampliar o efeito do não determinismo de maneira a torná-lo mais robusto aos diferentes tipos de ataques do que os métodos existentes. Assim, o presente trabalho propôs o *MultiMagNet*, um método de defesa que contém múltiplos componentes escolhidos aleatoriamente em tempo de execução. Para tanto, o método proposto utiliza um comitê de *autoencoders* formado aleatoriamente em tempo de execução para identificar o nível de perturbação presente nas imagens e, com base nessa informação, detectar a presença de imagens contraditórias. Os resultados dos experimentos simulando ataques de caixa-preta em imagens do *dataset* MNIST apontaram para a validade da hipótese levantada, ao mostrar que o *MultiMagNet* superou o *MagNet*, sua versão não determinística que utiliza apenas um *autoencoder*, na maioria dos cenários avaliados, sendo capaz de detectar perturbações geradas por diferentes tipos de ataques.

Como iniciativas de trabalhos futuros, podem ser destacadas: a implementação de novas abordagens para identificar automaticamente os limiares de perturbação para cada *dataset*; a avaliação do *MultiMagNet* em outros *datasets* e em cenários de ataque de caixa-

branca, em especial o ataque proposto em [Carlini and Wagner 2017b]; a investigação de técnicas de pré-processamento para tratamento das imagens, dentre outras.

Referências

[Bojarski et al. 2016] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[Carlini and Wagner 2017a] Carlini, N. and Wagner, D. (2017a). Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *10th ACM Workshop on Artificial Intelligence and Security*, page 12, Dallas, TX.

[Carlini and Wagner 2017b] Carlini, N. and Wagner, D. (2017b). Magnet and"efficient defenses against adversarial attacks"are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*.

[Carlini and Wagner 2017c] Carlini, N. and Wagner, D. (2017c). Towards Evaluating the Robustness of Neural Networks. *Proceedings - IEEE Symposium on Security and Privacy*, pages 39–57.

[Ding et al. 2018] Ding, L., Fang, W., Luo, H., Love, P. E., Zhong, B., and Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: integrating convolution neural networks and long short-term memory. *Automation in Construction*, 86:118–124.

[Gong et al. 2017] Gong, Z., Wang, W., and Ku, W.-S. (2017). Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*.

[Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Goodfellow et al. 2018] Goodfellow, I., McDaniel, P., and Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66.

[Goodfellow et al. 2014] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR'14)*, pages 1–11.

[He et al. 2017] He, W., Wei, J., Chen, X., Carlini, N., and Song, D. (2017). Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong. In *11th USENIX Workshop on Offensive Technologies (WOOT' 17)*, Vancouver, CA.

[Karpathy 2014] Karpathy, A. (2014). What I learned from competing against a ConvNet on Imagenet. Disponível em `http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet`. Acessado em 02 de setembro de 2018.

[Klarreich 2016] Klarreich, E. (2016). Learning securely. *Communications of the ACM*, 59(11):12–14.

[Kurakin et al. 2016a] Kurakin, A., Goodfellow, I., and Bengio, S. (2016a). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

[Kurakin et al. 2016b] Kurakin, A., Goodfellow, I., and Bengio, S. (2016b). Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

[Labati et al. 2018] Labati, R. D., Muñoz, E., Piuri, V., Sassi, R., and Scotti, F. (2018). Deep-ecg: Convolutional neural networks for ecg biometric recognition. *Pattern Recognition Letters*.

[LeCun et al. 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lecun et al. 2010] Lecun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional Networks and Applications in Vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256. IEEE.

[Madry et al. 2017] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards Deep Learning Models Resistant to Adversarial Attacks. *arxiv: 1706.06083*.

[Meng and Chen 2017] Meng, D. and Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM.

[Metzen et al. 2017] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. (2017). On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

[Moosavi-Dezfooli et al. 2016] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582.

[Papernot et al. 2018] Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Juang, Y.-L., Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., and Long, R. (2018). cleverhans v2.1.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*.

[Papernot et al. 2016a] Papernot, N., Mcdaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016a). The limitations of deep learning in adversarial settings. *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, pages 372–387.

[Papernot et al. 2016b] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016b). Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 582–597.

[Szegedy et al. 2013] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. In *International Conference on Learning Representations*, pages 1–10.

[Xu et al. 2018] Xu, W., Evans, D., and Qi, Y. (2018). Feature squeezing: Detecting adversarial examples in deep neural networks. *Network and Distributed Systems Security Symposium (NDSS) 2018*.