Avaliação das CGANs na Robustez de Modelos de Classificação de Ataques em IoT

Fernanda C. S. Pereira[1], **Mateus Pelloso**[2,3], **Michele Nogueira**[1,2]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)

[2]Departamento de Informática
Universidade Federal do Paraná (UFPR)

[3]Departamento de Ensino, Pesquisa e Extensão
Instituto Federal Catarinense (IFC)

{fernanda.pereira,michele}@dcc.ufmg.br, mpelloso@inf.ufpr.br

Abstract. *This paper evaluates the use of Conditional Generative Adversarial Networks (CGANs) to classify cyberattacks on IoT networks. Unlike standard GANs, CGANs incorporate labels – such as network flow information – into the sample generation process, allowing for more targeted and realistic adversarial samples. These generated samples mimic real data and are used to test the resilience of attack classifiers by attempting to mislead them. The study examines how these adversarial examples affect model robustness and proposes an adversarial training approach that incorporates CGAN, generated samples during training, to improve classifier performance. Experiments are conducted using two widely adopted IoT datasets, IoT-23 and TON-IoT, with evaluation based on metrics such as accuracy and precision.*

Resumo. *Este artigo avalia o uso de Redes Geradoras Adversárias Condicionais (do inglês, Conditional Generative Adversarial Networks – CGANs) na classificação de ciberataques em redes da Intenet das Coisas (IoT). Diferentemente das GANs tradicionais, as CGANs incorporam rótulos — como informações de fluxo de rede — no processo de geração de amostras, permitindo a criação de amostras adversariais mais direcionadas e realistas. Essas amostras geradas imitam os dados reais e são utilizadas para testar a resiliência de classificadores de ataques, ao tentar induzi-los ao erro. O estudo analisa como essas amostras adversariais impactam a robustez dos modelos e propõe uma abordagem de treinamento adversarial, na qual as amostras geradas pelas CGANs são integradas ao processo de treinamento para melhorar o desempenho do classificador. Os experimentos foram realizados com dois conjuntos de dados amplamente utilizados na área – IoT-23 e TON-IoT – e avaliados com base em métricas como acurácia e precisão.*

1. Introdução

O uso crescente e cada vez mais amplo da Internet das Coisas (do inglês, *Internet of Things* - IoT) tem sido acompanhado por desafios na cibersegurança. A Internet das Coisas integra diversos tipos de dispositivos, tais como os vestíveis que monitoram a saúde

do usuário, assistentes virtuais, *Smart Speakers* e até veículos. Uma vez conectados à Internet, esses dispositivos estão sujeitos a ataques mais elaborados, como ataques baseados em Inteligência Artificial (IA, do inglês, *Artificial Intelligence*) [Kemmerer 2003]. Isso exige defesas inovadoras, pois evoluem constantemente e são mais eficazes. No contexto de IA, destacam-se os ataques adversariais [Ayub et al. 2020], uma vulnerabilidade progressiva decorrente da aplicação de técnicas de IA neste contexto em que o atacante manipula os dados de entrada para levar os modelos a classificações incorretas.

Este trabalho foca na perda de robustez dos modelos de classificação de ataques em IoT. Esses modelos estão sujeitos a perturbações geradas por amostras adversariais, ou seja, os denominados de ataques adversarias. Essas amostras têm a capacidade de alterar os dados de entrada do modelo de forma imperceptível para os humanos, mas que levam os modelos a classificações incorretas. Uma técnica adotada para a geração dessas amostras é a Rede Adversarial Generativa Condicional (em inglês, *Conditional Generative Adversarial Network* – CGAN). Essa estrutura possibilita a criação de dados condicionados a informação dos rótulos de ataque, gerando registros cada vez mais parecidos com dados reais. Como investigado em [Szegedy et al. 2014], os classificadores de ameaças são vulneráveis se expostos a amostras adversariais, quando não há um treinamento prévio adequado do modelo. Portanto, o objetivo deste estudo é avaliar como as CGANs afetam a robustez desses modelos na classificação de ataques em dados IoT.

Diante dessa realidade, uma estratégia sendo estudada é a aplicação das técnicas usadas para a geração de amostras adversariais no processo de treinamento dos modelos. O objetivo é tornar os modelos de aprendizado de máquina (em inglês, *Machine Learning* – ML) mais robustos a ataques adversariais por meio de abordagens como o Aprendizado de Máquina Adversarial (em inglês, *Adversarial Machine Learning* – AML) ou Treinamento Adversarial (em inglês, *Adversarial Training* - AT), em que o modelo é treinado recebendo a injeção de amostras adversariais geradas por Redes Adversariais Generativas (em inglês, *Generative Adversarial Networks* – GANs) a cada iteração do treinamento. Recentemente, uma pesquisa [Dunmore et al. 2023] foi realizada para examinar de forma mais abrangente a aplicação dessa abordagem em Cibersegurança. Os autores concluíram que esta abordagem é efetiva ao prevenir os ataques em cibersegurança, que evoluem constantemente devido a novas ameaças e adaptações dos atacantes.

Assim, este artigo considera os estudos feitos até o momento avaliando sistematicamente a proposta de geração de dados sintéticos condicionados em cenários de treinamento distintos, proporcionando sugestões de estratégias defensivas embasadas nas descobertas obtidas. A análise dos resultados ocorre com base nas métricas de acurácia, precisão, *recall* e *f1-score*, empregando os conjuntos de dados IoT-23 e TON-IoT. Esses conjuntos representam capturas de tráfego de dispositivos IoT e possibilitam a análise de um contexto multiclasse, com o IoT-23, que apresenta múltiplos ataques, além de dados benignos, e, de um contexto binário, com o TON-IoT, que aponta a presença de ataques ou não. Com este trabalho, é reforçado que a aplicação de técnicas adversariais e de métodos de geração de dados sintéticos desempenham um papel relevante na busca por modelos mais resilientes a ataques. Embora haja ganhos práticos e teóricos, ainda há muito espaço para aprimoramentos, indicando que a investigação acerca da robustez e eficácia de classificadores em diferentes cenários de IoT permanece em aberto para novas descobertas e inovações. Finalmente, os resultados mostram que os ataques adversariais gerados pela

CGAN deterioram a capacidade de generalização do modelo. Foi possível perceber melhoria na robustez contra os ataques após o treinamento adversarial para o conjunto de dados multiclasse. O mesmo não foi observado para o conjunto binário.

O restante deste artigo é organizado como segue. A Seção 2 resume os trabalhos relacionados. A Seção 3 traz a fundamentação teórica, abordando os conceitos essenciais de ataques adversariais, CGANs, do classificador adotado e do treinamento adversarial. A Seção 4 detalha a metodologia da avaliação das CGANs na robustez de modelos de classificação de ataques em IoT, bem como os resultados alcançados. Por fim, a Seção 5 conclui o trabalho, sintetizando as principais descobertas e as direções futuras.

2. Trabalhos Relacionados

Esta seção aborda os trabalhos relacionados ao contexto deste estudo, apresentando o desempenho de modelos classificadores quando submetidos a ataques adversariais gerados por diferentes GANs, bem como variados conjuntos de dados. Estes ataques demonstram sua efetividade ao deteriorar a robustez dos modelos classificadores ou detectores de ataques, provocando a redução de suas respectivas acurácias e demais indicadores.

Em [Lin et al. 2022], os autores propõem um *framework* baseado em GANs, denominado IDSGAN, para geração de ataques adversariais contra sistemas de detecção de intrusão. Neste caso, o objetivo foi gerar *features* maliciosas de um tráfego malicioso que conseguisse enganar os sistemas de defesa e detecção para provocar ataques de evasão. Os autores demonstraram que a IDSGAN reduziu as taxas de detecção adversarial para DoS sob todos os algoritmos de detecção diminuem notavelmente de cerca de 80% para menos de 1%. As baixas taxas de detecção e as altas taxas de aumento de evasão obtidas em várias categorias de ataque e vários algoritmos de detecção de intrusão, indicam a grande eficácia e generalização da IDSGAN em ataques adversariais para evadir sistemas de detecção de intrusão (em inglês, *Intrusion Detection Systems* - IDS).

No estudo [Randhawa et al. 2021] foi proposta uma técnica para geração de dados realistas de *botnet* usando GANs, a fim de melhorar a decisão de classificadores ao detectar amostras potenciais de evasão. A metodologia do trabalho consistiu em usar amostras de teste para verificar a qualidade dos dados de tráfego gerados, em que o gerador é avaliado com a métrica *recall*. Os valores obtidos por essa métrica caem excessivamente para técnicas baseadas em GAN, inferindo que as amostras geradas por GAN podem escapar dos classificadores mais do que as técnicas de sobreamostragem por pares, técnica comparada no trabalho [Randhawa et al. 2021]. Os autores concluem que um modelo GAN com hiperparâmetros apropriados podem gerar amostras ainda mais realistas.

No estudo apresentado em [Ullah and Mahmoud 2021], foi desenvolvido um conjunto de modelos para detecção de anomalias em redes IoT, baseados em redes adversariais generativas condicionais. Entre os modelos propostos, destacam-se o ocGAN, voltado para conjuntos de dados desbalanceados, e o bcGAN, focado no aumento de dados. Ambos foram avaliados utilizando sete conjuntos de dados em cinco diferentes cenários. Os resultados demonstraram um desempenho expressivo, com o modelo bcGAN alcançando uma precisão de 98,10% no conjunto de dados KDD99. Logo, a arquitetura proposta mostrou-se robusta para a tarefa de detecção de anomalias em IoT.

A pesquisa desenvolvida em [Chauhan and Heydari 2020] apresentou a implementação de ataques DDoS polimórficos, que apresentam mudanças contínuas nas

assinaturas de um *malware*, usando uma variação das Redes Adversariais Generativas (GANs). O propósito do estudo foi entender melhor a capacidade do IDS de detectar ataques DDoS adversariais, como também fornecer melhor treinamento para IDS defensivo baseado em IA. Os resultados encontrados pelos autores indicam que os ataques polimórficos adversariais gerados pela GAN podem escapar do IDS baseado em ML enquanto mantêm uma taxa de falso positivo muito baixa.

Estudos anteriores exploraram a geração de amostras adversariais utilizando estruturas baseadas em GANs, com o objetivo de provocar evasão em classificadores ao enganar os modelos discriminadores [Lin et al. 2022, Randhawa et al. 2021, Ullah and Mahmoud 2021, Chauhan and Heydari 2020]. Neste contexto, a análise conduzida neste trabalho expande esses estudos ao propor uma avaliação que transforma o modelo gerador para essa finalidade. Além disso, os resultados obtidos fundamentaram o desenvolvimento de uma metodologia de treinamento defensiva contra ataques adversariais a partir do treinamento adversarial, demonstrada e validada por meio de dois conjuntos de dados relevantes no domínio, contribuindo para o avanço das abordagens existentes.

3. Fundamentação

Esta seção apresenta os fundamentos teóricos que embasam este trabalho. Inicialmente, são explorados conceitos fundamentais sobre o classificador adotado e dos ataques adversariais, apresentando suas características e impactos na robustez de modelos de ML. Em seguida, é explorada a estrutura da CGAN, detalhando seu funcionamento, principais componentes e variantes, bem como sua aplicação no treinamento adversarial no contexto da classificação de ataques em redes IoT, destacando os desafios e benefícios dessa abordagem no aprimoramento da segurança cibernética.

3.1. Ataques Adversariais

O termo 'amostras adversariais' foi estabelecido em [Szegedy et al. 2014] ao perceber que aplicando perturbações imperceptíveis aos dados de entrada de um modelo classificador, levavam-no a gerar classificações equivocadas. Logo, outros estudos foram desenvolvidos demonstrando o efeito da aplicação dessas amostras em modelos de detecção de ataques [Chen et al. 2017][Ayub et al. 2020][Kuppa and Le-Khac 2021]. Esses efeitos são demonstrados na Figura 1, em que o modelo é levado a gerar classificações incorretas.

Inspirado no trabalho de [Apruzzese et al. 2019], os ataques adversariais são divididos em subconjuntos de ataques com base em duas propriedades, influência e violação. A influência determina se o ataque é aplicado em tempo de treinamento ou em tempo de teste e a violação representa o tipo de violação de segurança que afeta a disponibilidade ou integridade do sistema. É necessário considerar algumas características para compreender os ataques, como o objetivo do atacante, o conhecimento do modelo atacado, as ações que os atacantes podem executar e, sua estratégia para alcançar seu objetivo. Acerca do conhecimento do modelo atacado, os autores distinguem os ataques em ataques caixa-branca (em inglês, *white box attacks*), em que o atacante possui conhecimento completo do modelo atacado, ataques caixa-cinza (em inglês, *grey box attacks*), que abrangem conhecimento parcial do modelo por parte do atacante, e ataques caixa-preta (em inglês, *black box attacks*), em que não há nenhum conhecimento sobre o modelo.

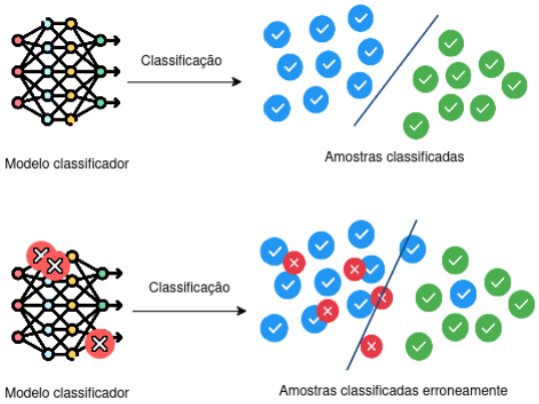


Figura 1. Ataques adversariais

3.2. CGANs

Uma Rede Adversarial Generativa Condicional, (em inglês, *Conditional Generative Adversarial Network* – CGAN) é um modelo proposto em [Mirza and Osindero 2014] como uma variante das GANs (Generative Adversarial Networks), onde o gerador e o discriminador são condicionados a informações adicionais, como rótulos ou atributos. Isso significa que, além de gerar dados, a rede é capaz de gerar dados específicos de uma classe ou categoria. Uma CGAN é composta por dois componentes principais: o **Gerador (G)** e o **Discriminador (D)**. Ambos são redes neurais treinadas de forma adversarial, visando gerar dados específicos baseados em uma condição predefinida.

Gerador (G): O Gerador é responsável por gerar dados sintéticos a partir de um vetor de ruído aleatório z e uma condição c, que pode representar uma classe específica ou atributo dos dados desejados. A entrada do Gerador consiste em dois componentes: o vetor de ruído z e a condição c, concatenados para formar a entrada final para a rede. Essa entrada combinada é processada pelo Gerador para produzir dados que imitam as características dos dados reais associados à condição c.

$$\text{Entrada para o Gerador} = \{z, c\}$$

Discriminador (D): O Discriminador, por sua vez, é uma rede treinada para distinguir entre dados reais e gerados, recebendo tanto os dados quanto a condição c associada. Ao receber esses *inputs*, o Discriminador tenta determinar se a amostra fornecida x é real ou gerada, considerando a relação entre a condição e os dados. Dessa forma, o Discriminador aprende a identificar as características que tornam os dados gerados plausíveis, a partir da condição dada, possibiltiando a geração de dados mais realistas.

$$\text{Entrada para o Discriminador} = \{x, c\}$$

3.3. Classificador

Uma Rede Neural Artificial, (em inglês, *Artificial Neural Network* - ANN) contém neurônios artificiais conhecidos como nós, estruturadas em séries de camadas que constituem toda a Rede Neural. Uma ANN, tipo mais simples de rede neural, contém uma camada de entrada, que recebe os dados a serem analisados, que vão ser transmitidos por

uma ou mais camadas ocultas, que transformarão os dados de entrada em dados úteis, por meio do aprendizado, para a camada de saída, que retornará como resposta aos dados aplicados na entrada [Géron 2022]. Neste caso, as conexões entre os nós não formam ciclos, ou seja, não há retropropagação, (em inglês, *backpropagation*), como nas redes mais complexas. Em suma, o modelo é uma simples rede neural artificial, projetada para tarefas básicas de classificação e detecção, com duas camadas ocultas e uma camada de saída *softmax*. A Figura 2 ilustra uma rede neural artificial como a utilizada neste trabalho.

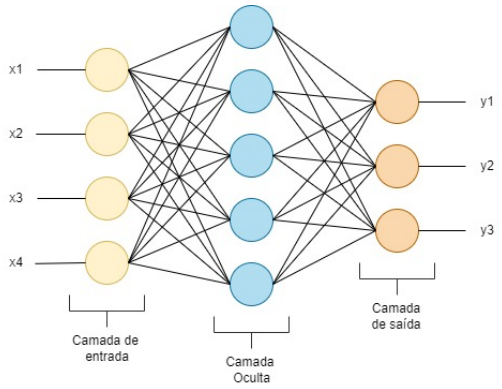


Figura 2. Rede Neural Artificial, adaptada de [Géron 2022]

3.4. Treinamento Adversarial

O Treinamento Adversarial consiste em uma prática para treinamento de modelos de ML que agrega amostras adversariais e amostras originais, proposta pelos autores do trabalho [Szegedy et al. 2014], demonstrando que seria possível incrementar seu desempenho de alguma forma. Baseando-se no *framework* proposto, outros autores [Madry et al. 2019] incrementaram a técnica para cenários e necessidades específicas, com foco principalmente no domínio de imagens. Dessa forma, o AT se tornou uma das estratégias mais efetivas para proporcionar mais robustez contra ataques adversariais.

A relevância do treinamento adversarial na mitigação de ciberataques é progressiva, considerando o crescente emprego de ML nas soluções de detecção de ataques. O domínio da geração de amostras adversariais é particularmente interessante para fortalecer modelos utilizados amplamente no espaço cibernético, visto que há no ciberespaço atacantes (por exemplo, desenvolvedores de *malware*) que buscam continuamente escapar de antivírus e filtros de *spam* baseados em aprendizado profundo e ML. Assim sendo, as amostras adversariais são relevantes no treinamento dos modelos de classificação, dado que contribui para sua resiliência, restringindo o espaço de ação dos atacantes.

4. Avaliação

Esta seção apresenta uma avaliação das Redes Generativas Condicionais (CGANs) na robustez de classificadores de ataques em IoT, que sofrem os ataques a partir do treinamento convencional dos modelos. A avaliação compreende a implementação de um modelo classificador treinado com amostras originais e atacado com amostras geradas pela CGAN. A partir dos resultados alcançados, é sugerida uma abordagem de treinamento adversarial que aplica as CGANs almejando recuperar a robustez dos classificadores. O processo de treinamento e validação dos modelos ocorre com base em dois conjuntos de

dados, TON-IoT e IoT-23, empregados também na geração das amostras pela CGAN. O primeiro conjunto, TON-IoT, abrange dados de ataque ou normais, considerado binário, ao contrário do segundo, IoT-23, que aborda múltiplas classes de ataques além dos dados benignos. O desempenho do modelo para as técnicas aplicadas foi avaliado considerando as métricas de precisão, acurácia, *recall* e *f1-score*, além de matrizes de confusão associando as classificações de dados originais e adversariais após cada tipo de treinamento. Dessa forma, este estudo ressalta o impacto da aplicação de amostras adversariais geradas pela CGAN contra o modelo treinado originalmente e uma alternativa para a recuperação da robustez a partir de amostras também geradas pela CGAN.

A Subseção 4.1 detalha a metodologia para o desenvolvimento deste trabalho. A Subseção 4.2 apresenta as as métricas utilizadas na avaliação e a Subseção 4.3 os conjuntos de dados e *features* selecionados, o processamento dos dados necessários para a aplicação aos experimentos e os ataques compreendidos nos conjuntos. A Subseção 4.4 detalha as abordagens adotadas, como o tipo de classificador empregado, sua arquitetura e o treinamento do modelo. Além disso, a Subseção 4.5 descreve o fluxo de trabalho abordado, o método de geração de amostras adversariais e os resultados obtidos, que, por conseguinte, são analisados a partir das métricas elencadas para avaliação.

4.1. Metodologia

A avaliação da aplicação de CGANs na robustez de modelos classificadores de ataques em IoT iniciou-se com a seleção da técnica geradora, CGAN, pois possibilita o uso dos rótulos de cada registro no aprendizado e geração de novas amostras, bem como a definição de uma rede neural simples, mas eficiente na classificação dos ataques. A partir disso, foram selecionados os conjuntos de dados TON-IoT e IoT-23. A etapa de pré-processamento dos dados nos conjuntos abordou transformação para adequação do formato de arquivos na exportação, limpeza e remoção de dados inválidos, ajustes dos tipos de dados, normalização de escalas e reestruturação das *features* de rotulação para incluir informações sobre ataques em apenas uma *feature*.

A partir dos dados pré-processados, o modelo é treinado com os dados resultantes, processo exemplificado na etapa 1 da Figura 3. Para atacar o modelo classificador, foi necessário gerar amostras usando a CGAN. Portanto, na segunda etapa, a CGAN é desenvolvida e treinada. Após construídos ambos os modelos, gerador e discriminador, o modelo gerador recebe um ruído de entrada e gera amostras a partir dele, que o discriminador definirá como reais ou falsas. Esse processo se repete para todas as amostras geradas, enquanto o gerador tenta gerar amostras falsas que enganam o discriminador. Na terceira etapa, após treinar a CGAN, o ruído e os rótulos falsos são dados como entrada para o gerador gerar amostras, que, na etapa 4, são usadas para atacar o modelo classificador treinado com as amostras originais. Logo, as amostras geradas são integradas aos dados originais, resultando em um novo conjunto de treinamento, etapa 5 da Figura 3. Esse novo conjunto é a entrada para o treinamento do modelo classificador, que é novamente treinado com amostras adulteradas, processo chamado de treinamento adversarial, conforme etapa 6. Dessa forma, ao sofrer ataques adversariais, conforme exemplificado na etapa 7, o classificador perde menos robustez ao fazer a classificação de ataques.

No pré-processamento dos conjuntos de dados e treinamento da CGAN, foi usada uma máquina virtual que dispõe de 24 GB de memória RAM, armazenamento de 300 GB

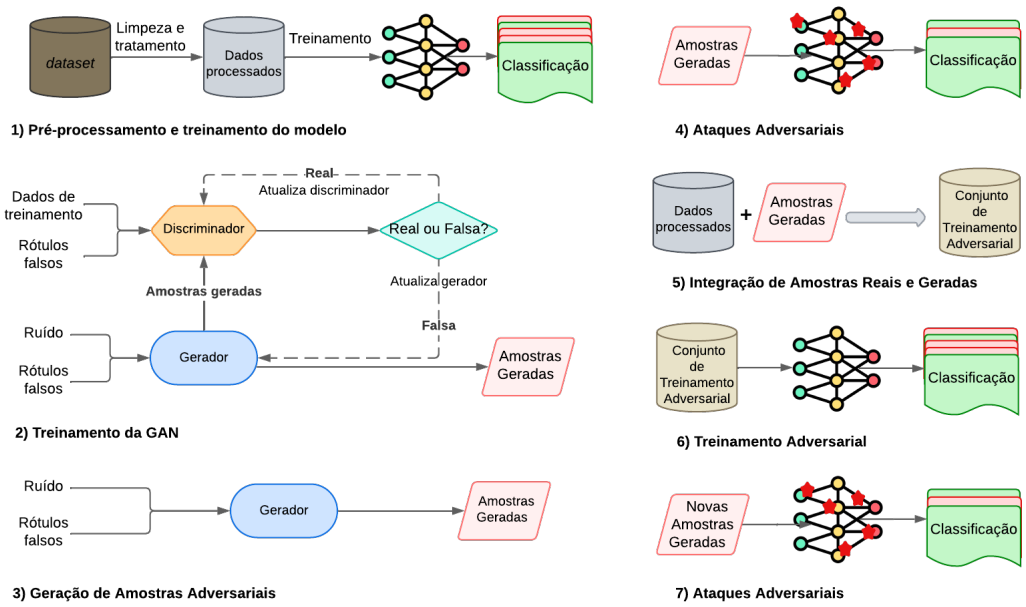


Figura 3. Fluxo de trabalho exemplificado em etapas

e 24 núcleos virtuais de CPU. Para treinar os modelos, o código implementado foi inspirado no código disponibilizado pelo estudo [McCarthy et al. 2023][1]. Adicionalmente, a parametrização destes modelos sofreu ajustes de acordo com as necessidade impostas pelas características disponíveis nos conjuntos de dados avaliados pelos experimentos.

Os modelos foram testados através da plataforma Google Colaboratory, um serviço do Jupyter Notebook hospedado que não requer configuração para uso e oferece acesso gratuito a recursos de computação. A ferramenta disponibiliza uma máquina virtual em nuvem, contendo 12,7 GB de memória RAM, 107,72 GB de armazenamento e uma GPU de *back-end* do Google Compute Engine em Python 3, para conduzir experimentos de Aprendizado de Máquina. Além disso, algumas bibliotecas são necessárias para processamento e manipulação dos experimentos, como: Pandas, Numpy, Matplotlib, Sklearn e, em especial, Keras, aplicadas em atividades de pré-processamento, representação de dados e implementação dos modelos de rede neural.

4.2. Métricas de avaliação

Para a avaliação dos resultados, foram empregadas quatro métricas [Géron 2022] estatísticas: acurácia, precisão, *recall* e *f1-score*. Acurácia (1) mede a proporção de previsões corretas (verdadeiros positivos e verdadeiros negativos) entre o número total de casos examinados. Precisão (2) mede a fração de previsões positivas que realmente pertencem ao conjunto de previsões positivas. *Recall* (3) quantifica o número de previsões positivas feitas entre todos os exemplos positivos no conjunto de dados. O *F1-Score* (4) é uma pontuação única que combina precisão e *recall*, definida como a média harmônica da precisão e do *recall* de um modelo. Para o cálculo dessas métricas, consideram-se os seguintes valores: Verdadeiro Positivo (VP) é a amostra de tráfego malicioso classificado

[1]https://github.com/mccarthyajb/HL-NTAC

como malicioso, Verdadeiro Negativo (VN) é a amostra de tráfego normal classificado como normal, Falso Positivo (FP) é a amostra de tráfego normal classificado como malicioso, e Falso Negativo (FN) é a amostra de tráfego malicioso classificado como normal.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2}$$

$$\textit{Recall} = \frac{VP}{VP + FN} \tag{3}$$

$$\textit{F1-Score} = \frac{2 \cdot \text{Precisão} \cdot \textit{Recall}}{\text{Precisão} + \textit{Recall}} \tag{4}$$

4.3. Conjuntos de dados

Os experimentos foram conduzidos aplicando o conjunto de dados IoT-23 [Garcia et al. 2020] e o conjunto TON-IoT [Alsaedi et al. 2020]. O IoT-13 é um conjunto de dados que contém tráfego de dispositivos IoT capturado na Universidade Técnica Checa, República Tcheca, no período de 2018 a 2019. Este conjunto possui diversas capturas de tráfego de dispositivos IoT reais, além de tráfego normal (benigno) e consiste em 23 capturas (chamadas de cenários) de diferentes amostras. Este conjunto abrange tráfegos de rede agrupados por dispositivos IoT, onde o tráfego malicioso pertence a ataques como C&C-*HeartBeat*, DDoS, *Okiru* e *PartOfAHorizontalPortScan*. A descrição da distribuição dos rótulos presentes neste conjunto é apresentada na Tabela 1.

Para conduzir os experimentos com os dados provenientes do conjunto IoT-23, o arquivo *conn.log.labeled* da captura Kenjiro necessitou de pré-processamento. Este arquivo foi disponibilizado em formato *ASCII text*, portanto, foi necessário aplicar regras específicas para a importação adequada do conteúdo, considerando que a importação padrão de dados *csv* não se adequou ao contexto. As *features* elencadas para os experimentos com esse conjunto foram id_orig_p, id_resp_p, duration, orig_bytes, resp_bytes, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes e label. Para a importação, os nomes foram explicitados no código e a leitura dos arquivos foi implementada linha a linha, separando os dados a serem recebidos por cada *feature* e reunindo em um vetor de dados, em formato *dataframe*. Neste caso, os arquivos foram rotulados aplicando uma *feature* chamada *'label'* e outra *'detailed-label'*, em que *'label'* aponta se o registro é benigno ou maligno e *'detailed-label'*, o tipo de ataque, em caso de dado maligno. Com isso, foi aplicada uma regra a cada registro do novo *dataframe*. Logo, a cada registro maligno, o nome do ataque foi recebido para sobrescrever o rótulo maligno em *'label'*, permitindo descartar a *feature 'detailed-label'*. Essa operação possibilitou acessar os tipos específicos de ataques para experimentos multiclasse. Por fim, os dados necessitaram de limpeza e remoção de valores inválidos, normalização e padronização de escalas.

O conjunto TON-IoT, por sua vez, inclui fontes de dados heterogêneas coletadas de conjuntos de dados de telemetria de sensores de IoT e IIoT, conjuntos de dados de sistemas operacionais do Windows 7 e 10, bem como conjuntos de dados de tráfego de rede e TLS do Ubuntu 14 e 18. Os conjuntos de dados foram coletados de uma rede realista e

de grande escala projetada no Cyber Range e IoT Labs, na Escola de Engenharia e Tecnologia da Informação, UNSW Canberra the Australian Defence Force Academy. Este conjunto de dados também apresenta várias capturas de tráfego dos dispositivos mencionados para cada cenário, IoT, Linux, Windows e dados de rede, contendo diferentes tipos de dados benignos e de ataque em cada um, neste caso, *Scanning*. A descrição da distribuição dos rótulos presentes neste conjunto é apresentada na Tabela 2.

O pré-processamento foi semelhante ao do conjunto IoT-23. Neste contexto foi adotada a captura 1 para os dados de rede, que contém dados benignos e dados de ataque *Scanning*, em formato *csv*, não necessitando de adequações. As *features* elencadas para esse conjunto foram duration, dst_bytes, missed_bytes, src_pkts, src_ip_bytes, dst_pkts, dst_ip_bytes, label, type. A partir disso, os tipos de dados foram mapeados explicitamente para *'int'* e *'string'*, dependendo do tipo. Da mesma forma, os processos de limpeza, remoção de valores inválidos, normalização e padronização de escalas foram aplicados.

A escolha das *features* para cada conjunto de dados, bem como dos próprios conjuntos foi motivada pelos seguintes fatores: (i) a presença de ataques rotulados, facilitando a análise e interpretação dos dados; (ii) o uso do formato de arquivo *pcap* e *csv*, amplamente adotados por ferramentas de redes, o que permite o manuseio dos registros; (iii) sua ampla utilização em pesquisas anteriores, possibilitando comparações e validações; e (iv) sua disponibilidade pública na Internet, garantindo a reprodutibilidade dos resultados.

Tabela 1. Distribuição dos rótulos - IoT-23

Rótulo	**Número de Fluxos**
PartOfAHorizontalPortScan	27.311.187
Okiru	13.655.215
DDoS	13.655.172
C&C-HeartBeat	6.834
Benign	31.438
Attack	4
PartOfAHorizontalPortScan-Attack	5
Total	54.654.855

Tabela 2. Distribuição dos rótulos - TON-IoT

Rótulo	**Número de Fluxos**
Scanning	791.321
Normal	208.679
Total	1.000.000

4.4. Classificador

No contexto deste estudo foram adotados modelos baseados em aprendizado de máquina supervisionado, ou seja, há rótulos para os dados aplicados como teste e validação. Essa adoção possibilita avaliar corretamente a classificação do fluxo de dados pelo modelo. Logo, foram usados dois classificadores com arquiteturas específicas para cada conjunto de dados: IoT-23 e TON-IoT, conforme ilustrado nas Tabelas 3 e 4. Ambos os modelos seguem uma estrutura sequencial composta por três camadas densas, seguidas de uma camada de ativação. A principal diferença entre os classificadores reside na quantidade de

neurônios por camada e no número total de parâmetros ajustáveis, refletindo a adaptação das arquiteturas às características de cada conjunto de dados.

Ao lidar com tarefas de classificação, manipulação de conjuntos de dados de tamanhos variados e capacidade de generalização, a decisão se deu por redes neurais, que tem se mostrado mais adequadas ao contexto [Géron 2022]. Sendo assim, cada rede neural foi treinada por 200 épocas, com os dados previamente pré-processados, normalizados e divididos em conjuntos de treinamento (70%) e teste (30%). O otimizador Adam foi utilizado durante o treinamento. O desempenho dos modelos foi avaliado com base nas métricas acurácia, precisão, *recall* e *f1-score*.

Tabela 3. Arquitetura RN - TON-IoT

Modelo: sequencial		
Camada (tipo)	**Formato de Saída**	**Parâmetro #**
dense (Densa)	(None, 32)	256
dense_1 (Densa)	(None, 16)	528
dense_2 (Densa)	(None, 2)	34
activation (Ativação)	(None, 2)	0
Total de parâmetros:		818
Parâmetros treináveis:		818
Parâmetros não-treináveis:		0

Tabela 4. Arquitetura RN - IoT-23

Modelo: sequencial		
Camada (tipo)	**Formato de Saída**	**Parâmetro #**
dense (Densa)	(None, 32)	352
dense_1 (Densa)	(None, 16)	528
dense_2 (Densa)	(None, 7)	119
activation (Ativação)	(None, 7)	0
Total de parâmetros:		999
Parâmetros treináveis:		999
Parâmetros não-treináveis:		0

4.5. Resultados

Nesta subseção, são apresentados e analisados os resultados obtidos a partir dos experimentos desenvolvidos. São discutidos os impactos das amostras adversariais geradas pela CGAN quando aplicadas contra um modelo treinado convencionalmente. Como proposta de defesa, são analisados também os impactos do treinamento adversarial usando a CGAN. Os resultados obtidos para cada um dos conjuntos de dados podem ser visualizados nas Tabelas 5 e 6 e na Figura 4. Ao avaliar os resultados, é possível observar que, em ambos os conjuntos, a CGAN deteriorou a robustez do classificador de ataques IoT.

Conforme apresentado na Tabela 5 para o conjunto TON-IoT, após o treinamento convencional, o modelo obteve uma acurácia de 66,26% e precisão de 66,67%, com *recall* de 77,89% e *f1-score* de 62,66% quando não foram utilizadas amostras adversariais. Esses valores indicam que o modelo consegue recuperar boa parte dos ataques (*recall*), bem como que os resultados apresentados são razoavelmente balanceados para os dados originais injetados no modelo, o que se reflete em um *f1-score* moderado. Ao avaliar a aplicação de amostras geradas por CGAN antes do treinamento adversarial, os resultados

Tabela 5. Resultados - ToN-IoT

Métrica	Treinamento Convencional		Treinamento Adversarial	
	Sem Adv.	**CGAN**	**Sem Adv.**	**CGAN**
Acurácia	66,26%	50,07%	50,52%	49,37%
Precisão	66,67%	66,67%	51,15%	50,02%
Recall	77,89%	77,89%	74,05%	62,18%
F1-Score	62,66%	62,66%	35,59%	33,11%

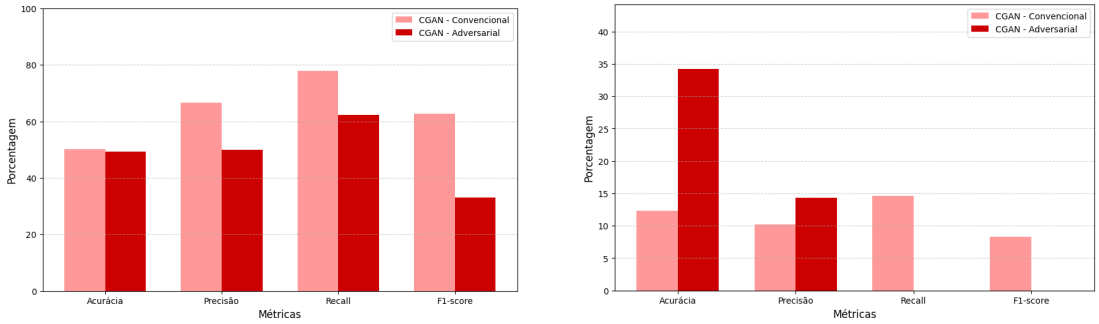
Tabela 6. Resultados - IoT-23

Métrica	Treinamento Convencional		Treinamento Adversarial	
	Sem Adv.	**CGAN**	**Sem Adv.**	**CGAN**
Acurácia	99,98%	12,28%	99,97%	34,22%
Precisão	85,70%	10,20%	71,42%	14,28%
Recall	85,63%	14,57%	71,32%	0%
F1-Score	85,66%	8,29%	71,37%	0%

mostram uma acurácia de 50,07%, e curiosamente, os mesmos valores de precisão, *recall* e *f1-score* obtidos sem as amostras adversariais. Isso sugere que os dados gerados sinteticamente podem ter confundido o modelo, mas não a ponto de alterar todos os indicadores de maneira uniforme. Além disso, é possível notar que o cenário sem amostras adversariais para o treinamento adversarial apresentou valores menores de acurácia (50,52%) e *f1-score* (35,59%) quando comparado ao mesmo cenário de treinamento convencional. Isso pode ocorrer porque o treinamento adversarial tende a penalizar o modelo por memorizar padrões muito específicos dos dados originais, fazendo com que ele seja, em geral, menos sobreajustado (*overfitted*). Por outro lado, os ataques com a CGAN após o treinamento adversarial resultaram em uma queda de desempenho (49,37% de acurácia e 33,11% de *f1-score*), possivelmente porque as amostras sintéticas geradas não seguiram um padrão de perturbação sistemático, dificultando a adaptação do modelo ou confundindo a rede de forma mais aleatória.

Como apresentado na Tabela 6 para o conjunto de dados IoT-23, as métricas apresentam resultados extremos em cada cenário. Após o treinamento convencional e sem amostras adversariais, o modelo atinge 99,98% de acurácia e 85,70%, em que um *f1-score* de 85,63% indica um bom aprendizado do modelo para os dados originais, com equilíbrio entre as demais métricas. Ao aplicar ataques com a abordagem CGAN, a acurácia sofre redução para 12,28%, com *f1-score* de 8,29%. Para o treinamento adversarial, o modelo mantém a acurácia em 99,97% e valores para demais métricas em torno de 71% no cenário sem amostras adversariais, repetindo o comportamento do treinamento convencional. Após ataques gerados com a CGAN, a acurácia cai para 34,22%, e o *recall* é 0%, resultando em *f1-score* também de 0%. Apesar da queda, é importante ressaltar que, após o treinamento adversarial, o modelo apresenta melhoria de 22% na acurácia e 4% na precisão de classificação quando exposto a amostras geradas pela CGAN, quando comparado aos resultados obtidos para a avaliação após o treinamento convencional.

Conforme retratado pela Figura 4, os dados sintéticos gerados por CGANs afetam a robustez do classificador. Antes do treinamento adversarial, as amostras geradas con-



(a) TON-IoT (b) IoT-23

Figura 4. Desempenho das CGANs

tribuíram para confundir o modelo, resultando em uma queda significativa das métricas de desempenho. Ao usar a CGAN como etapa do treinamento adversarial, é possível, em alguns casos, melhorar a robustez do modelo, como para o conjunto IoT-23, mas não garante proteção ampla contra todos os tipos de ataques. Neste conjunto, apesar do desempenho ser ótimo antes dos ataques, o ataque com amostras geradas pela CGAN expõe fragilidades do modelo. Esse contraste reforça a importância de avaliar a estratégia de geração de amostras adversariais (tipo de ataque, intensidade das perturbações e características do *dataset*) e de adotar mecanismos de defesa adicionais, como regularizações específicas, para lidar com diferentes ameaças no contexto de detecção de ciberataques.

5. Conclusão

A geração de dados com distribuição de dados próxima à de ambientes reais é um desafio em cibersegurança devido à complexidade dos cenários. Este trabalho avaliou o uso de CGANs na robustez de classificadores de ataques no contexto da Internet das Coisas, mostrando que amostras artificiais degradam o desempenho quando o treinamento é convencional. Para mitigar esse efeito, utilizou-se o treinamento adversarial com CGANs, resultando na recuperação parcial da robustez em dados multiclasse. Os resultados indicam o potencial das CGANs na defesa contra ataques adversariais, embora desafios como a fidelidade dos dados sintéticos e a configuração dos modelos ainda exijam investigação.

Agradecimentos

Os autores agradecem o apoio do IFC, UFPR e da UFMG e o auxílio financeiro da FAPESP #2018/23098-0 e do CNPq #444824/2024-3, #440553/2024-5 e #313844/2020-8.

Referências

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., and Anwar, A. (2020). Ton-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8:165130–165150.

Apruzzese, G., Colajanni, M., Ferretti, L., and Marchetti, M. (2019). Addressing adversarial attacks against security systems based on machine learning. In *International Conference on Cyber Conflict (CyCon)*, volume 900, pages 1–18.

Ayub, M. A., Johnson, W. A., Talbert, D. A., and Siraj, A. (2020). Model evasion attack on intrusion detection systems using adversarial machine learning. In *annual conference on information sciences and systems (CISS)*, pages 1–6. IEEE.

Chauhan, R. and Heydari, S. S. (2020). Polymorphic adversarial ddos attack on ids using gan. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE.

Chen, L., Ye, Y., and Bourlai, T. (2017). Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 99–106.

Dunmore, A., Jang-Jaccard, J., Sabrina, F., and Kwak, J. (2023). A comprehensive survey of generative adversarial networks (GANs) in cybersecurity intrusion detection. *IEEE Access*, 11:76071–76094.

Garcia, S., Parmisano, A., and Erquiaga, M. J. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic.

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. "O'Reilly Media, Inc.".

Kemmerer, R. A. (2003). Cybersecurity. In *International Conference on Software Engineering, 2003. Proceedings.*, pages 705–715. IEEE.

Kuppa, A. and Le-Khac, N.-A. (2021). Adversarial xai methods in cybersecurity. *IEEE Transactions on Information Forensics and Security*, 16:4924–4938.

Lin, Z., Shi, Y., and Xue, Z. (2022). *IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection*, page 79–91. Springer International Publishing.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2019). Towards deep learning models resistant to adversarial attacks.

McCarthy, A., Ghadafi, E., Andriotis, P., and Legg, P. (2023). Defending against adversarial machine learning attacks using hierarchical learning: A case study on network traffic attack classification. *Journal of Information Security and Applications*, 72:103398.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Randhawa, R. H., Aslam, N., Alauthman, M., Rafiq, H., and Comeau, F. (2021). Security hardening of botnet detectors using generative adversarial networks. *IEEE Access*, 9:78276–78292.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks.

Ullah, I. and Mahmoud, Q. H. (2021). A framework for anomaly detection in iot networks using conditional generative adversarial networks. *IEEE Access*, 9:165907–165931.