Fernanda Carolina da Silva Pereira

Análise Comparativa de Métodos de Geração de Amostras Adversariais na Detecção de Ataques

Belo Horizonte, Minas Gerais
2024

Fernanda Carolina da Silva Pereira

Análise Comparativa de Métodos de Geração de Amostras Adversariais na Detecção de Ataques

Relatório apresentado como requisito da disciplina de Monografia em Sistemas de Informação I do Curso de Bacharelado em Sistemas de Informação da UFMG

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Orientador: Prof.ª Dra. Michele Nogueira Lima
Co-orientador: MSc. Mateus Pelloso

Belo Horizonte, Minas Gerais
2024

Resumo

O aprendizado de máquina adversarial (em inglês, *Adversarial Machine Learning* – AML) compreende um campo que estuda os ataques em modelos de aprendizado de máquina, bem como suas defesas. Para a aplicação desses ataques, são usadas amostras adversariais, que consistem em amostras especialmente criadas para comprometer o classificador. Apesar da existência de contramedidas aplicadas com sucesso para defesa dessa ameaça em sistemas de visão computacional, os mesmos não podem ser prontamente aplicados em contextos de cibersegurança. Esses necessitam de adaptações e melhorias específicas para este domínio, como seleção de propriedades do modelo e distribuição de dados de treinamento. Essas limitações, ainda não amplamente exploradas, geram desafios únicos, haja vista que, pequenas perturbações em pixels de uma imagem não a alteram visualmente, mas podem corromper o resultado final do modelo ou afetar seu comportamento. O estudo de AML aborda um subconjunto de técnicas de aprendizado de máquina que vem sendo empregado em diversos campos para compreender os mecanismos por trás dos ataques, como também desenvolver métodos eficazes para mitigar o impacto na detecção de ataques cibernéticos de forma automatizada. Este trabalho de conclusão de curso I estuda os conceitos sobre AML e aplica métodos de geração de amostras adversariais em modelos de classificação de ciberataques, visando avaliar seu impacto sobre o classificador. Para tal, foram estudados e reproduzidos experimentos existentes para comparar o efeito dos métodos de geração de amostras adversariais na identificação de ataques, como PartOfAHorizontalPortScan, DDoS, entre outros. Os resultados apresentados demonstram o potencial risco dos ataques adversariais contra os modelos e os aprendizados alcançados serão aplicados na segunda etapa desta monografia.

Palavras-chave: Amostras Adversariais. Aprendizado de Máquina Adversarial. Aprendizado de Máquina. Ataques. Cibersegurança. Inteligência Artificial.

Abstract

Adversarial Machine Learning (AML) encompasses a field that studies the attacks on machine learning models, as well as its defenses. To apply these attacks, adversarial examples are used, which consist on specially designed inputs to deceive classifiers. Although the most extensive studies on AML have been successfully carried out in the area of image recognition, the same could not be applied to cybersecurity systems. The last ones require specific adjustments and improvements, as well as the model properties selection and distribution of the data training. These limitations, not widely explored yet, have unique challenges, considering that adversarial perturbations on a few pixels produce apparent visual effects, however, can corrupt the model results or affect its behaviour. AML comprises a subset of machine learning techniques, currently explored attempting to understand the mechanisms behind the attacks and to develop effective methods to anticipate potential risks against data sets being corrupted, model theft, and adversarial samples, in an automated way. This final course work studies the concepts and related work related to AML and to apply adversarial sample generation technique on classification models, targeting its evaluation. For such, availabale experiments were reproduced to compare different adversarial sample generation techniques in identifying attacks, as PartOfAHorizontalPortScan, DDoS, among others. The presented results show the potential risk against machine learning models and the gained knowledge will be applied on the next stage of this work.

Key-words: Adversarial Samples. Adversarial Machine Learning. Artificial Intelligence. Attacks. Cybersecurity. Machine Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AML	Adversarial Machine Learning
ANN	Artificial Neural Network
ART	Adversarial Robustness Toolbox
ASCII	American Standard Code for Information Interchange
AT	Adversarial Training
C&C	Command and Control
C&W	Carlini and Wagner
CIC-IDS2017	Intrusion detection evaluation dataset
DDoS	Distributed Denial of Service
DoS	Denial of Service
DNN	Deep Neural Network
FGSM	Fast Gradient Signed Method
FN	Falso Negativo
FP	Falso Positivo
FTP	File Transfer Protocol
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology database
MSI	Monografia em Sistemas de Informação

NLP	Natural Language Processing
RAM	Random-Access Memory
SSH	Secure Shell Protocol
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
XSS	Cross-Site Scripting
ZOO	Zeroth-Order Optimization

Sumário

1 Introdução

Em 1959, Arthur Samuel, pioneiro no estudo de Aprendizado de Máquina (em inglês, *Machine Learning* – ML), definiu o campo como o ramo de pesquisa que confere aos computadores a habilidade de aprender sem a utilização de programação explícita [2]. Desde então, o uso crescente de soluções de ML, impulsionado por avanços significativos nas últimas décadas, possibilitou uma revolução na análise e processamento de dados em todos os âmbitos, desde a academia até o mercado. Entretanto, tal revolução foi acompanhada pelo surgimento de preocupações sobre a segurança e robustez dos modelos aplicados, principalmente em relação à sua vulnerabilidade a ataques adversariais.

Nesse sentido, uma das principais preocupações diz respeito a ataques em ambientes cibernéticos, como ataques de negação de serviço (em inglês, *Denial of Service* – DoS), ataques de injeção de *scripts* entre sites (em inglês, *Cross-Site Scripting* – XSS) e ataques adversariais. Estes últimos, buscam explorar vulnerabilidades nos modelos de ML para a inserção de anomalias nos dados. Essas anomalias, conhecidas como amostras adversariais, consistem em amostras especificamente projetadas para enganar modelos de aprendizado de máquina, gerando um comportamento não esperado no modelo treinado, fazendo-o gerar classificações incorretas [3].

Na literatura é possível encontrar tentativas de combater esse tipo de ameaça. Alguns autores aplicam a técnica de proteção de classificadores contra ataques adversariais usando modelos generativos (em inglês, *Generative Adversarial Network Defense* – Defense-GAN) [4], que consiste no treinamento do modelo para representar a distribuição de imagens não perturbadas. Outra técnica empregada na defesa contra ataques adversariais é a destilação (em inglês, *Distillation*) [5]. Neste caso, o modelo extrai conhecimento adicional a respeito dos pontos de treinamento como vetores de probabilidade de classes, produzidos por uma rede neural profunda (em inglês, *Deep Neural Networks* - DNN), que são realimentados ao treinamento.

Recentemente, uma pesquisa [6] examinou de forma mais abrangente ataques baseados em amostras adversariais contra sistemas de cibersegurança baseados em aprendizagem profunda. Foi concluído pelos autores que este domínio apresenta desafios únicos, que ainda não foram amplamente explorados, haja vista que, pequenas perturbações em *pixels* de uma imagem não a alteram visualmente [7], mas podem corromper o resultado do modelo ou afetar seu comportamento. Portanto, embora os métodos aplicados com sucesso a sistemas de visão computacional [8] inspirem soluções semelhantes para o contexto de cibersegurança, eles não necessariamente são prontamente adotados, pois necessitam de adaptações e melhorias específicas para esta área, como seleção de propriedades do modelo e distribuição de dados de treinamento [6].

Diante dessa realidade, existem soluções que aplicam amostras adversariais ao trei-

namento de modelos, como o Treinamento Adversarial (em inglês, *Adversarial Training* - AT) [9]. Sua ideia principal é desenvolver métodos de defesa em que um modelo é treinado recebendo a injeção de amostras adversariais a cada iteração do treinamento. Dessa forma, é possível minimizar o impacto dos ataques adversariais nos modelos de classificação/detecção de ataques [10].

1.1 Objetivos

1.1.1 Objetivo Geral

Sendo assim, este trabalho consiste em experimentar os avanços feitos até o momento com métodos de geração de amostras adversariais e, analisar o impacto da utilização dessas técnicas para, consequentemente, identificar as possibilidades de aprimoramento de modelos de ML para solucionar o problema investigado.

1.1.2 Objetivos Específicos

Para isso, serão explorados principalmente três métodos de geração de amostras adversariais, FGSM, L-BFGS e ZOO, examinando como se comparam em termos de eficiência e desempenho contra modelos de detecção de ataques. Através desta análise comparativa, é esperado contribuir para um melhor entendimento dos desafios enfrentados na proteção contra ataques adversariais em modelos de ML, identificando fragilidades ou pontos de melhoria para possibilitar o desenvolvimento ou aprimoramento de modelos mais eficazes para a defesa e a mitigação deste problema.

1.2 Contribuições

As contribuições deste trabalho consistem na comparação de métodos de geração de amostras adversariais, aplicados contra um modelo classificador baseado em rede neural. Sua relevância se dá ao possibilitar o entendimento do impacto na robustez do classificador, utilizando de métricas quantitativas. A investigação e os experimentos aqui aplicados evidenciam vulnerabilidades de modelos de detecção de ataques e contribuem para o desenvolvimento de técnicas de defesa contra essas ameaças.

1.3 Organização do texto

O primeiro capítulo desde trabalho, Introdução, apresenta o problema a ser investigado e sua contextualização, como também seus objetivos gerais e específicos. O segundo capítulo, Fundamentos e Referencial Teórico, se trata da apresentação dos conceitos

pertinentes para o problema e sua importância, embasando-se em referenciais teóricos da literatura. Os passos previstos para a execução do projeto e ferramentas utilizadas para abordar o problema são descritos no capítulo 3, Metodologia. O capítulo 4, Experimentos, aborda os experimentos desenvolvidos e resultados alcançados. Por fim, o trabalho é concluído no capítulo 5, demonstrando objetivos alcançados e próximos passos.

2 Fundamentos e Referencial Teórico

Este capítulo apresenta os fundamentos teóricos e as principais referências que embasam a pesquisa sobre a Análise Comparativa de Métodos de Geração de Amostras Adversariais na Detecção de Ataques. Neste caso, serão explorados conceitos fundamentais a ataques adversariais, a estrutura de classificadores baseados em redes neurais, aos métodos de geração de amostras adversariais e a aplicação desses métodos no contexto da detecção de ciberataques.

2.1 Ataques Adversariais

O termo "amostras adversariais"foi estabelecido por [11], ao perceber que aplicando perturbações imperceptíveis aos dados de entrada de um modelo classificador levavam-no a gerar classificações erroneamente. Logo, outros estudos foram desenvolvidos demonstrando o efeito da aplicação dessas amostras em modelos de detecção de ataques [12][13][14][15].

Inspirado no trabalho [16], 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 [17].

Além disso, de acordo com [18] 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. No que diz respeito ao 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 [17].

2.2 Classificador

Segundo Aurélien Géron, consultor de ML, em sua obra [19], soluções utilizando redes neurais frequentemente superam outras soluções em ML, principalmente devido ao grande montante de dados disponível atualmente para treinar esses modelos. Com algoritmos aprimorados e poder computacional suficiente para treinar grandes redes neurais em um período razoável de tempo, soluções baseadas em redes neurais, como classificadores de ataques, são cada vez mais adotadas e evoluídas. Os autores citados aqui utilizaram classificadores baseados em redes neurais, que recebem como entrada dados de rede,

contendo ataques cibernéticos, processam esses dados em camadas ocultas e produzem resultados nas camadas de saída, como mostra a Figura 1.

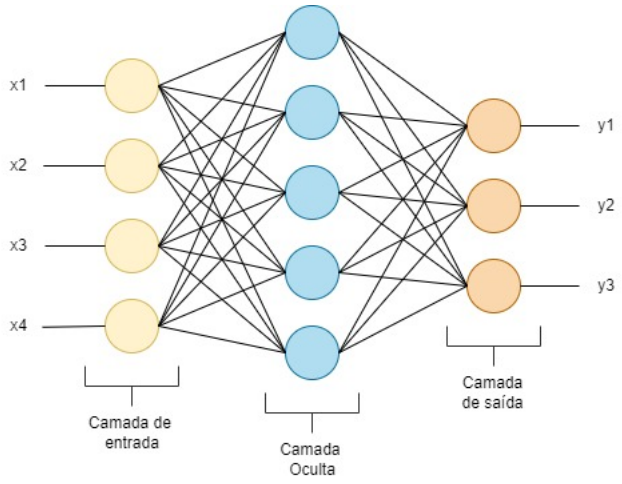

Figura 1 – Rede Neural Artificial, adaptada de [1]

Uma Rede Neural Artificial, (em inglês, *Artificial Neural Network* - ANN) contém neurônios artificiais conhecidos como nós, estruturadas em séries de camadas que constituem todo a Rede Neural. Uma camada pode conter até milhões de nós, dependendo do quão complexas serão as tarefas do modelo, usados para aprender os padrões ocultos de um conjunto de dados. Dessa forma, uma ANN 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 [20].

Uma ANN é o tipo mais simples de rede neural, que transmite a informação da camada de entrada, através das camadas ocultas, até a camada de saída [19]. Neste caso, as conexões entre os nós não formam ciclos, ou seja, não há retro propagação, (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 1 ilustra uma rede neural artificial como a utilizada neste trabalho.

2.3 Métodos de Geração de Amostras Adversariais

2.3.1 FGSM

Em [3], os autores propuseram o FGSM, (em inglês, *Fast Gradient Sign Method*) um método baseado em gradiente, simples e rápido, cuja ideia é gerar amostras adversariais. Neste caso, um vetor imperceptivelmente pequeno, cujos elementos são iguais ao sinal dos elementos do gradiente da função de custo, é aplicado aos dados de entrada do modelo,

criando perturbações nos dados. Para isso, as perturbações são aplicadas em cada *feature* do conjunto de dados.

A efetividade deste método reside na sua capacidade de criação de amostras com custo computacional comparativamente eficiente, sendo considerado simples e rápido. Dessa forma, é possível alterar a classificação de uma imagem usada como exemplo no estudo [3], demonstrando que o classificador apresenta uma taxa de erro de 99,9% com uma confiança média de 79,3% no conjunto de testes MNIST, uma rede simples totalmente conectada com uma ou mais camadas ocultas e um classificador *softmax*. Assim, utilizando um algoritmo simples e de baixo custo computacional, os autores geraram amostras classificadas erroneamente pelos modelos.

2.3.2 L-BFGS

O L-BFGS, (em inglês, *Limited-memory Broyden–Fletcher–Goldfarb–Shanno*) é um algoritmo de otimização não-linear baseado em gradiente que possui como objetivo encontrar perturbações ótimas para aplicar aos dados de entrada do modelo [11]. Seu funcionamento é baseado em encontrar o mínimo local de uma função de ativação usando uma quantidade limitada de memória no computador. Este, por sua vez, aproxima a segunda derivada para problemas em que não pode ser calculada diretamente. Assim, é necessário realizar o cálculo do inverso da matriz Hessiana, que pode ser computacionalmente intensivo. Apesar de considerado efetivo na geração de amostras adversariais, possui um custo computacional intensivo devido aos cálculos mencionados.

Em [11], os autores desenvolveram amostras adversariais utilizando o L-BFGS para aplicação contra um classificador de imagens. Assim como o FGSM, este classifica, com maior probabilidade, os resultados errados, mas não faz distinção sobre qual classe classificada erroneamente deve ser selecionada pelo modelo. Segundo os autores, ambos os métodos são suficientes para pequenos conjuntos de dados.

2.3.3 ZOO

Por fim, em [21], os autores propuseram um método chamado ZOO (em inglês, *Zeroth-Order Optimization*) permitindo estimar o gradiente de classificadores sem acessar o classificador, ideal para ataques caixa-preta (em inglês, *black-box attack*). Esta característica torna a técnica baseada em oráculo, em que não é necessário o treinamento de modelos substitutos e o atacante não necessita de informações sobre o classificador.

A técnica do ZOO, diferentemente do L-BFGS e FGSM, permite a geração das amostras sem necessitar do gradiente, manipulando o modelo para criar amostras adversariais com base em suas predições [22]. Segundo os autores, o método se mostrou eficaz ao estimar o gradiente, apesar de exigir uma quantidade abundante de consultas ao oráculo. Considerando que o método não requer gradiente, isso permite sua efetividade em ataques

caixa preta. Sua aplicação principal se dá quando o gradiente, ou demais informações sobre o classificador, são difíceis ou não podem ser encontradas.

2.4 Aplicação

O Treinamento Adversarial, em que o modelo recebe amostras adversariais injetadas ao seu treinamento, é amplamente aceito como o método mais eficaz na melhoria da robustez dos modelos contra ataques adversariais [23]. Neste caso, é esperado conhecer o desempenho do modelo no cenário de pior caso, evitando que suas vulnerabilidades sejam exploradas por ataques adversariais.

Desse modo, a relevância do AT na mitigação de ciberataques é cada vez maior, considerando o crescente emprego de ML nas soluções de detecção de ataques. Neste contexto, o domínio da geração de amostras adversariais é particularmente interessante para fortalecer modelos utilizados amplamente no espaço cibernético. Além disso, o ciberespaço está repleto de adversários (por exemplo, desenvolvedores de malware que desejam escapar de produtos antivírus e filtros de spam baseados em aprendizado profundo e de máquina) com metas e objetivos definidos [24].

É importante destacar que todos os métodos mencionados foram aplicadas com sucesso em cenários de Computação Visual ou Processamento de Linguagem Natural (em inglês, *Natural Language Processing* - NLP) [24]. Para a Cibersegurança, elas não podem ser aplicadas prontamente, sem devidas adaptações e experimentos para o contexto, demonstrando que o ambiente cibernético continua exposto a ataques adversariais. Assim, mostra-se de crucial importância treinar modelos de ML para a detecção de desses ataques.

2.5 Resumo

Neste capítulo foram explicados e embasados os conceitos de Ataque Adversarial, que objetivam confundir classificadores baseados em ML, assim como seus tipos, que abrangem ataques caixa branca, caixa cinza e caixa preta, determinados pelo nível de conhecimento do modelo por parte do atacante. Além disso, foi detalhada a estrutura do classificador, baseado em rede neural artificial, com uma camada de entrada, uma camada oculta, para aprendizado e uma camada de saída dos dados.

Da mesma forma, os métodos de geração de amostras adversariais, FGSM, L-BFGS e ZOO foram detalhados para compreensão das técnicas, em que as duas primeiras necessitam de gradiente e a última, não. Isso permitiu entender como aplicá-los, como também seus custos, computacionalmente. Por fim, abordou-se a aplicação dos métodos adotados, como no Treinamento Adversarial, principalmente, colaborando para a mitigação de ataques.

3 Metodologia

Este capítulo apresenta as etapas de desenvolvimento deste projeto, detalhando os conjuntos de dados e *features* selecionados, o processamento dos dados necessários para a aplicação aos experimentos, ataques compreendidos nos conjuntos e abordagens adotadas, como o treinamento do modelo e os métodos de geração de amostras adversariais. Além disso, também é descrito o fluxo de trabalho abordado e as métricas empregadas para avaliação dos resultados.

O trabalho referente ao TCC 1 concentrou-se na utilização da técnica FGSM como base para implementar os demais métodos de interesse, neste caso, L-BFGS e ZOO. A finalidade da comparação entre os três é identificar seu impacto quando aplicadas contra um modelo classificador de ataques baseado em rede neural, por meio da visualização das métricas de acurácia, precisão, *recall* e *f1-score*, usadas como base de comparação quantitativa do modelo.

Os métodos elencados para este trabalho são abordagens apresentadas na literatura dentre um amplo número de métodos. Logo, elas foram escolhidas de forma empírica para as comparações, considerando que é uma pesquisa ainda em estágio inicial. Além disso, ainda não foi encontrada na literatura uma avaliação direta entre elas, o que justifica a investigação.

3.1 Conjuntos de Dados

Para conduzir os experimentos, foi usado o conjunto de dados de Avaliação de Detecção de Intrusão (em inglês, *Intrusion detection evaluation dataset* - CIC-IDS2017) [25]. Esse conjunto de dados é composto por uma captura de tráfego de rede, que contém dados benignos e de ataques, com características que simulam dados verdadeiros de fluxos de rede do mundo real. Para este conjunto de dados, foi considerado o comportamento de 25 usuários, com base nos protocolos HTTP, HTTPS, FTP, SSH e e-mail. O período de coleta se deu entre a segunda-feira, 3 de julho de 2017 e finalizou na sexta-feira, 7 de julho de 2017. Além disso, os ataques do conjunto compreendem *Brute Force FTP*, *Brute Force SSH*, DoS, *Heartbleed*, *Web Attack*, *Infiltration*, *Botnet* e DDoS. Sua escolha para esse trabalho se deu por ser um conjunto de dados disponível na Internet, possibilitando aos pesquisadores verificarem seus experimentos.

As *features* presentes neste conjunto foram extraídas via CICFlowMeter, uma ferramenta de extração de dados de rede. Dentre elas, estão *SourceIP*, *SourcePort*, *DestinationIP*, *DestinationPort* e *Protocol*. Todas as 80 *features* extraídas são descritas no repositório

da ferramenta[1]. Além disso, as *features* foram rotuladas pelos autores em dados benignos e dados maliciosos [26].

Também foi empregado um segundo conjunto de dados para realizar comparações, chamado IoT-23 [27]. O IoT-13 é um conjunto de dados de tráfego de dispositivos IoT capturado na Universidade CTU, 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. Neste trabalho, a captura utilizada foi a de número 17.1 (Kenjiro), extraídos quase 5 milhões de pacotes, onde o tráfego malicioso pertence a ataques de C&C-*HeartBeat*, DDoS, *Okiru*, *PartOfAHorizontalPortScan* e *PartOfAHorizontalPortScan-Attack*.

Neste conjunto de dados, foram consideradas as 23 *features* extraídas pela ferramenta Zeek[2] de análise de dados de rede. Além disso, os dados extraídos foram rotulados em dados benignos ou dados maliciosos pelos autores [27], usando a ferramenta Flaber [3].

3.1.1 Ataques

Em ambos os conjuntos de dados adotados, alguns ataques foram abordados para classificação pelo modelo e métodos definidos. Portanto, os rótulos aplicados em cada um dos conjuntos[25][27] são brevemente descritos:

- *Brute Force FTP/SSH*: Ataque usado para quebra de senhas, como também descobrir páginas e conteúdos ocultos em uma aplicação *web*. Neste caso, o ataque é realizado para conseguir acesso não autorizado a servidores FTP/SSH;
- DoS: O objetivo deste ataque é tornar um recurso indisponível temporariamente ao sobrecarregar o serviço com solicitações supérfluas;
- *Heartbleed*: Proveniente de uma vulnerabilidade na biblioteca de criptografia OpenSSL, permitindo que o atacante tenha acesso à memória do servidor, que pode conter informações sensíveis;
- *Web Attack*: Ataques realizados às aplicações *web*, como XSS, por exemplo;
- *Infiltration*: O atacante se infiltra na rede via uma vulnerabilidade de *software*;
- *Botnet*: Dados trafegados em dispositivos conectados a internet podem ser explorados por um atacante em posse de botnets;
- DDoS: Atacantes sobrecarregam a largura de banda, gerando enorme tráfico de rede;

[1] https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt
[2] https://zeek.org/
[3] https://github.com/stratosphereips/flaber

- C&C-*Heartbeat*: Similar ao *Heartbleed*, o nome descreve tipos de ataques que exploram o mecanismo *Heartbeat* em softwares, enviando pequenos pacotes para rastrear o host infectado pelo servidor C&C;
- *Okiru*: O rótulo indica que as conexões possuem características de um *botnet Okiru*;
- *PartOfAHorizontalPortScan*: O rótulo indica que as conexões são usadas para realizar uma varredura horizontal de portas para coleta de informações para ataques adicionais.

3.1.2 Pré-processamento dos dados

Para conduzir os experimentos com os dados provenientes do conjunto IoT-23, o arquivo *conn.log.labeled* da captura Kenjiro neessitou de um 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. Para isso, foram explicitados no código os nomes das *features* esperadas e implementada a leitura dos arquivos linha a linha, ignorando os comentários (apontados com #), separando os dados a serem recebidos por cada *feature* pelo espaço em branco e reunindo em um vetor de dados. A partir dos dados capturados e *features* definidas, o conteúdo foi concatenado entre si em um *dataframe*. Além disso, os dados inválidos, rotulados com *'-'* e *'empty'*, foram padronizados como *'NaN'*.

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'*.

Após limpeza dos dados inválidos e reestruturação da *feature 'label'*, os tipos das *features* restantes foram mapeados explicitamente para *'int'* e *'string'*, dependendo do tipo, visto que, após a importação, todos assumiram o mesmo tipo, *'object'*. Por fim, os dados necessitaram de limpeza e remoção de valores inválidos, normalização e padronização de escalas, bem como balanceamento para ajuste do viés, via *undersampling*.

No contexto do conjunto CICIDS-2017, os dados foram importados de cada arquivo disponibilizado com extrações por dias da semana, do formato *csv* diretamente para dataframes, concatenados em apenas um *dataframe* abrangendo todos os registros. Assim como no conjunto anterior, os dados passaram por limpeza e remoção de dados inválidos, como vazios ou incoerentes para o contexto, possibilitando sua normalização e padronização. Por fim, foi aplicada a estratégia de *undersampling* para ajuste do viés.

3.2 Abordagens

Primeiramente, foi necessário treinar o modelo classificador com o otimizador Adam e uma função de custo apropriada, a partir das amostras limpas dos conjuntos de dados CICIDS-2017 e IoT-23.

Após o treinamento do modelo, a função *FastGradientMethod* da biblioteca *Adversarial Robustness Toolbox* (ART) foi aplicada. Esta função gera amostras adversariais com base no método FGSM, adicionando pequenas perturbações aos dados de entrada. Essas perturbações são calculadas a partir do gradiente da função de custo em relação aos dados de entrada, visando maximizar o erro de classificação do modelo.

Para a geração de amostras adversariais utilizando o método L-BFGS, duas abordagens foram exploradas: uma com a biblioteca Numpy e outra com a biblioteca Foolbox. A função *FeatureAdversaries* da biblioteca Numpy foi utilizada para gerar amostras adversariais. Os parâmetros necessários para essa função incluem vetores que representam os ataques e suas classes correspondentes. Dessa forma, a função *FeatureAdversaries* aplica o método L-BFGS para encontrar as perturbações mínimas necessárias que enganam o modelo, gerando amostras adversariais a partir dos dados de entrada.

Adicionalmente, a biblioteca *Foolbox* foi testada utilizando a função *LBFGSAttack*. A função requer como entrada o modelo classificador treinado e uma classe de probabilidades como alvo. O método L-BFGS é então aplicado para modificar os dados de entrada de maneira a maximizar a probabilidade de classificação incorreta, gerando amostras adversariais que visam enganar o modelo.

Por fim, a função *ZOOAttack* da biblioteca *Adversarial Robustness Toolbox* (ART) foi utilizada para gerar amostras adversariais utilizando o método ZOO (Zeroth Order Optimization), um ataque caixa preta que não requer conhecimento dos gradientes do modelo. A função foi configurada com os parâmetros necessários, incluindo um vetor com os valores de entrada a serem atacados e os rótulos das classes de ataque. Assim, a função ZOOAttack gera perturbações que são adicionadas aos dados de entrada, criando as amostras adversariais.

3.3 Fluxo de trabalho

A etapa de pré-processamento dos dados no conjunto IoT-23 abordou transformação para adequação do formato de arquivos na exportação, limpeza e remoção de dados inválidos, alteração de tipos, normalização de escalas e reestruturação das features de rotulação para incluir informações sobre ataques em apenas uma *feature*. Com os dados adequados ao contexto, foi aplicada a estratégia de *undersampling* para ajuste do viés.

O pré-processamento dos dados no conjunto CICIDS-2017 envolveu importação dos dados extraídos para cada dia da semana e concatenação para um *dataframe*. Isso

possibilitou a limpeza e remoção de dados inválidos, padronização e normalização de escalas e, por fim, ajuste do viés com a estratégia de *undersampling*.

A partir dos dados pré-processados, os dados foram injetados no modelo, baseado em uma rede neural artificial, a ser treinado para classificação. Após treinamento e resultados de detecção serem gerados a partir dos dados limpos, o modelo recebeu a aplicação de amostras adversariais geradas pelos métodos FGSM e ZOO, individualmente. Assim, foram gerados novos resultados de classificação e métricas para comparação. O fluxo desenvolvido é demonstrado na Figura 2.

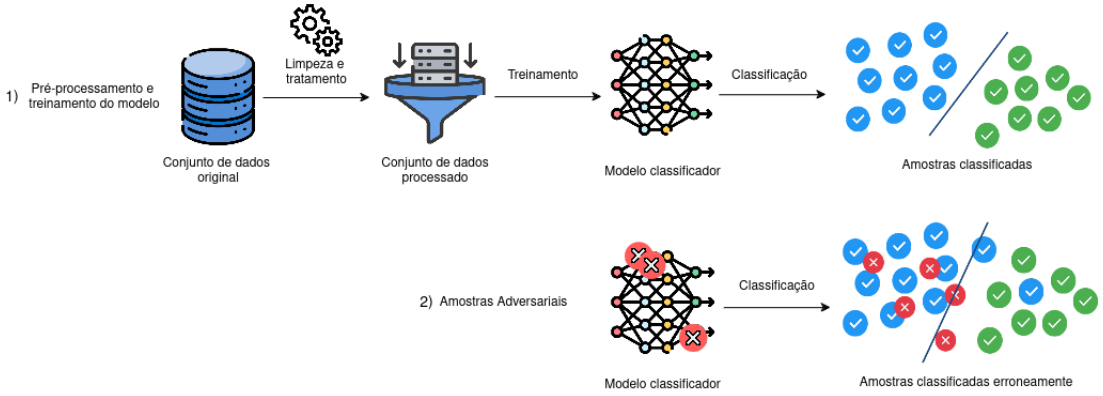

Figura 2 – Fluxo de trabalho exemplificado em parte 1 e 2

3.4 Métricas

Para a avaliação dos resultados, foram empregadas quatro métricas [19] estatísticas: acurácia, precisão, *recall* e *f1-score*. Acurácia (3.1) mede a proporção de previsões corretas (verdadeiros positivos e verdadeiros negativos) entre o número total de casos examinados. Precisão (3.2) mede a fração de previsões positivas que realmente pertencem ao conjunto de previsões positivas. *Recall* (3.3) quantifica o número de previsões positivas feitas entre todos os exemplos positivos no conjunto de dados. O *f1-score* (3.4) 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 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.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.1}$$

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{3.2}$$

$$\text{Recall} = \frac{VP}{VP + FN} \tag{3.3}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{3.4}$$

3.5 Resumo

Neste capítulo foi apresentada a metodologia adotada para o desenvolvimento dos experimentos. Primeiramente, os conjuntos de dados CICIDS-2017 e IoT-23 foram apresentados e detalhados conforme pré-processamento para aplicação no modelo. Da mesma forma, os ataques presentes em cada um dos conjuntos foram brevemente descritos. Logo, as abordagens, como treinamento do modelo e geração de amostras adversariais com os métodos FGSM, L-BFGS e ZOO foram abordadas conforme aplicação.

Além disso, este capítulo também demonstra o fluxo de trabalho envolvido, desde pré-processamento até geração das amostras adversariais e classificações do modelo. Por fim, as métricas empregadas na avaliação dos resultados são apresentadas, sendo acurácia, precisão, recall e f1-score, considerando valores de verdadeiro positivo e negativo, bem como falso positivo e negativo, que possibilitarão a análise comparativa dos métodos.

4 Experimentos

Neste capítulo serão abordadas as configurações necessárias para a condução dos experimentos deste trabalho, bem como da plataforma adotada e recursos empregados. Além disso, serão descritos os resultados obtidos com os últimos testes e uma avaliação comparativa usando as métricas quantitativas elencadas. Por fim, os valores estarão dispostos em uma tabela e dois gráficos, possibilitando comparar os resultados.

4.1 Experimentos

Para realizar os experimentos, o código utilizado foi inspirado no código disponibilizado pelo estudo [28] e disponível em um repositório público do GitHub[1]. Neste caso, modificações foram necessárias, tendo em vista que o autor do código havia implementado a configuração da rede neural para classificação dos ataques e o método FGSM para geração de amostras adversariais no conjunto de dados CICIDS-2017. Sendo assim, os métodos ZOO e L-BFGS foram implementados para este conjunto, após reproduzir os experimentos com o método FGSM. Logo, para atender ao contexto do conjunto IoT-23, os métodos FGSM, L-BFGS e ZOO foram implementados.

Durante a realização dos experimentos, testes utilizando as ferramentas encontradas para a aplicação de cada método, a fim de avaliar as abordagens citadas. No entanto, ao implementar o método L-BFGS, foram encontradas incompatibilidades que impediram a conclusão bem-sucedida do experimento utilizando esta abordagem.

Um dos experimentos envolveu a utilização da função *FeatureAdversaries* com a biblioteca Numpy. A proposta era gerar amostras adversariais para avaliar a robustez do modelo. Porém, a função de geração de amostras exigia dois parâmetros obrigatórios: *source* e *guide*. Durante a execução, não foi possível atender ao formato do parâmetro *guide* conforme esperado, impedindo a realização do experimento conforme planejado.

Por fim, tentamos aplicar o ataque LBFGSAttack, disponível na biblioteca *Foolbox*, para avaliar a vulnerabilidade do modelo a amostras adversariais. Contudo, o pacote LBFGSAttack não estava mais presente na versão atual da biblioteca *Foolbox*. Este pacote existia em versões antigas, mas foi removido nas atualizações mais recentes, impossibilitando a realização do experimento.

Os modelos foram testados utilizando a 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

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

de *back-end* do *Google Compute Engine* em Python 3, para conduzir experimentos de Aprendizado de Máquina.

4.2 Resultados

Cada uma das abordagens foi executada duas vezes para avaliação, uma vez para cada um dos conjuntos de dados escolhidos. Para tal, amostras reduzidas pelo *undersampling* foram coletadas aleatoriamente de cada um dos conjuntos, onde 70% dos dados foram utilizados para treino e 30% para testes. Além disso, o modelo classificar foi treinado utilizando 200 épocas. Por fim, as métricas geradas pelos resultados de cada um dos métodos foram extraídas e comparadas entre os diferentes conjuntos de dados.

4.2.1 Avaliação

Os resultados obtidos por cada um dos modelos podem ser visualizados na Tabela 1 e gráficos da Figura 4 e 3. Ao compará-los, é possível perceber que, em ambos os conjuntos de dados, os métodos deterioraram consideravelmente o modelo de detecção, principalmente o método FGSM.

Conjunto de Dados	Abordagem	Acurácia	Precisão	Recall	F-Score
CICIDS-2017	Sem amostras adversariais	90,96%	90,83%	90,88%	90,60%
	FGSM	**7,62**	**6,84%**	**0,97%**	**1,69%**
	ZOO	**43,11%**	**42,06%**	**43,03%**	**40,42%**
IoT-23	Sem amostras adversariais	98,25%	98,59%	98,41%	92,42%
	FGSM	**13,96%**	**14,28%**	**3,69%**	**5,87%**
	ZOO	**81,26%**	**82,78%**	**76,41%**	**79,01%**

Tabela 1 – Comparação de desempenho cada abordagem nos conjuntos de dados IoT-23 e CICIDS-2017.

Ao visualizar os resultados obtidos para o conjunto CICIDS-2017, é possível observar que, com uma acurácia de 90,96%, o modelo está acertando quase 91% de todas as instâncias. Sua precisão de 90,28%, indica que, ao realizar uma classificação, o modelo está correto em quase 91% das vezes. Isso também é demonstrado na métrica *recall*, com 90,89%, indicando que o modelo comete poucos falsos negativos. Com o *f1-score*, é possível perceber que há um equilíbrio entre a precisão e o *recall*, particularmente importante ao balancear falsos positivos e falsos negativos.

Em contrapartida, ao avaliar o desempenho do classificador ao aplicar o método FGSM, percebe-se que a acurácia e precisão caíram drasticamente para 7,63% e 6,85%, respectivamente. Isso indica que o modelo está acertando menos de 8% das instâncias e que, apenas cerca de 7% das previsões são corretas. Logo, ao apresentar um *recall* de 0,96%, o modelo está identificando menos de 1% dos ataques reais. Além disso, um *f1-score*

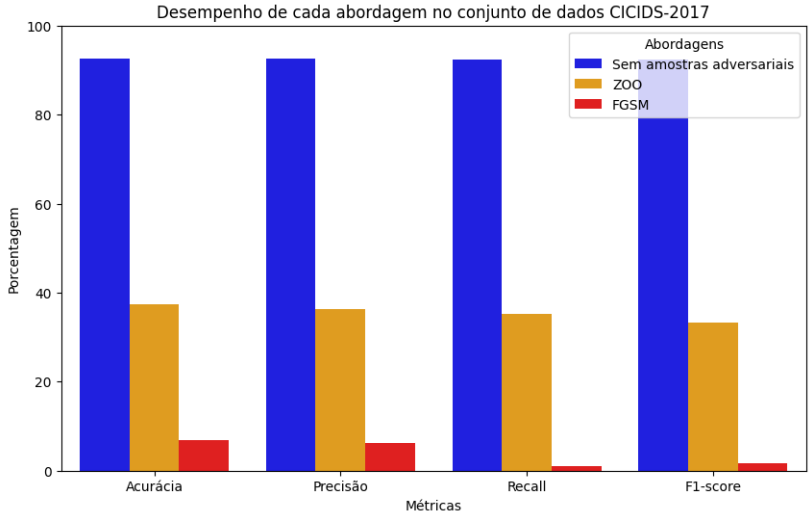

Figura 3 – Comparação de desempenho de cada abordagem no conjunto de dados CICIDS-2017.

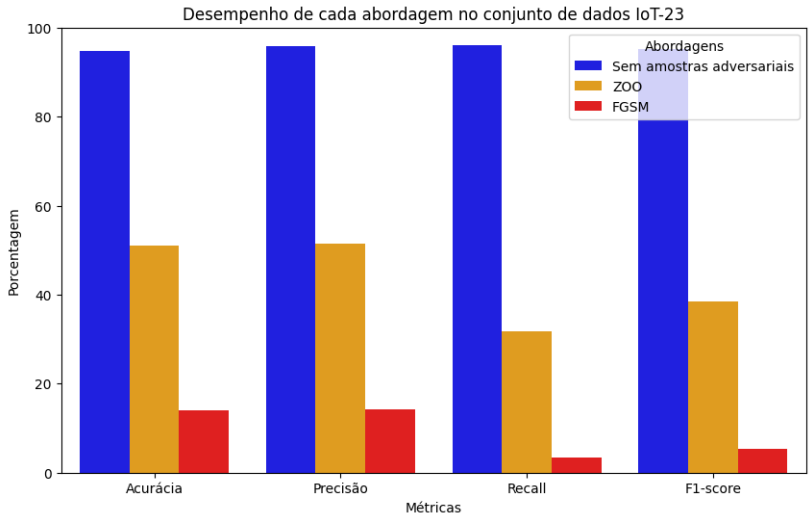

Figura 4 – Comparação de desempenho de cada abordagem no conjunto de dados IoT-23.

de 1,70% reflete um péssimo desempenho geral do modelo, falhando em fornecer um bom equilíbrio entre as métricas.

Da mesma forma, ao avaliar a aplicação do método ZOO, o desempenho observado é significativamente melhor do que com o observado com o método anterior, mas ainda representa uma queda substancial em relação ao desempenho inicial. Com uma acurácia de 43,11%, o modelo está acertando um pouco menos da metade das instâncias, indicando uma precisão semelhante, com 42,06%. Ao observar o *recall*, a taxa de 43,03% sugere que o modelo está identificando um pouco menos da metade dos ataques. Isso se reflete ao notar que o *f1-score*, com 40,42%, reflete um desempenho equilibrado entre as métricas, ou seja, o modelo está falhando em identificar ataques corretamente, bem como em evitar falsos positivos.

Ao avaliar os resultados obtidos para o conjunto IoT-23, por sua vez, nota-se que, com uma acurácia de 98,25%, o modelo acerta mais de 98% das instâncias. A alta assertividade do modelo ao realizar uma classificação também é evidenciada ao analisar a precisão, que apresenta uma taxa de 98,59%. Além disso, a métrica *recall*, indica que o modelo comete ainda menos falsos negativos que no anterior, com 98,41%. Logo, a métrica *f1-score* também reflete um grande equilíbrio entre as métricas, com 92,42%.

Bem como após a aplicação das amostras adversariais no conjunto anterior, ao avaliar o desempenho do modelo classificador após aplicar o método FGSM, é possível notar o grande impacto gerado. Nesse caso, as métricas de acurácia e precisão apresentam valores de 13,96% e 14,28%, respectivamente. Esses números indicam que o modelo acerta em torno de 14%, com apenas 14,28% de previsões corretas. Isso se torna mais claro ao observar as métricas de *recall* e *f1-score*, denotando que o modelo identifica apenas 3,69% dos ataques, bem como falha ao fornecer equilíbrio entre as métricas, com 5,87%.

Por fim, ao aplicar o método ZOO no conjunto IoT-23, o desempenho é consideravelmente maior que o anterior. Ao avaliar a acurácia do modelo neste caso, percebemos que ele acerta mais de 80% das instâncias, com uma precisão semelhante, de 82,78%. Isso sugere que o classificador está identificando ataques corretamente em mais de 75% dos casos, conforme a métrica *recall*. Além de melhores taxas quando comparado ao FGSM, é observa-se que há um equilíbrio considerável entre as métricas, representado pelo valor de 79,01% no *f1-score*.

Em resumo, os resultados mostram que, antes da aplicação das amostras adversariais, os classificadores apresentam um desempenho robusto em ambos os conjuntos de dados, com métricas de precisão, *recall* e *f1-score* altas. No entanto, após a aplicação de métodos para geração de amostras adversariais, as métricas caem significativamente, evidenciando a vulnerabilidade do classificador a tais ataques. Esta análise evidencia a necessidade de desenvolver técnicas de defesa mais eficazes para melhorar a robustez dos modelos contra ataques adversariais.

4.3 Resumo

Este capítulo abordou as configurações necessárias para a condução dos experimentos, como da plataforma Google Colab, além do código tomado como inspiração. Também foram descritos os experimentos para cada um dos métodos, FGSM e ZOO, obtendo sucesso, e L-BFGS, apresentando pendências que impediram sua aplicação como esperado. Assim, o modelo foi treinado para detecção de ataques, a partir de cada abordagem.

As métricas elencadas, acurácia, precisão, *recall* e *f1-score*, apoiaram a análise comparativa do desempenho dos métodos de geração de amostras adversariais, obtendo resultados para ambos os conjuntos de dados empregados. A tabela e os gráficos possibilitaram a visualização dos valores alcançados e sua comparação ao modelo sem amostras adversariais.

5 Desafios e Limitações

O método com maior impacto alcançado à robustez do modelo classificador foi o FGSM, além de apresentar consumo de tempo e recursos computacionais eficientes em relação os demais métodos. Porém, seu uso apresenta algumas restrições no campo de cibersegurança. Neste caso, são aplicadas perturbações em todas as *features* de entrada do modelo classificador, incluindo as não-modificáveis [29]. Ao considerar a classificação de imagens, campo de maior aplicação do método, é possível alterá-las sem impactar sua funcionalidade. Ou seja, a imagem não tem sua função principal alterada, o que não se pode afirmar sobre dados de pacotes de rede, por exemplo. Então, é necessário buscar a preservação da funcionalidade ao gerar amostras adversariais [30] almejando o Treinamento Adversarial (AT).

Com os resultados alcançados, observa-se que o método ZOO gera bastante impacto à robustez do modelo, ainda que menor que o gerado pelo FGSM. No entanto, por se tratar de um método caixa preta e não necessitar de acesso ao gradiente, é preciso estimá-lo. Para isso, o método consulta o modelo classificador inúmeras vezes. Apesar de experimentos sugerirem que o ataque é efetivo contra modelo de ML com confgurações caixa preta, consome muito mais recursos computacionais que métodos caixa branca. Além disso, seu desempenho se assemelha muito ao do método C&W (em inglês, *Carlini&Wagner*), um método caixa branca, baseado em gradiente, que consome menos recursos [29].

Ao se tratar do método L-BFGS, considerado eficiente na geração de amostras adversariais [11], também são encontras algumas limitações. No que tange a aplicação do ataque, foram encontradas dificuldades com a falta de bibliotecas amplamente disponibilizadas para sua reprodução, como também pendênciais em sua aplicação utilizando as bibliotecas encontradas, como a listagem não intuitiva dos parâmetros necessários para a geração de amostras com a função *FeatureAdversaries* da biblioteca Numpy, bem como a função LBFGSAttack, da biblioteca Foolbox, obsoleta. Além disso, o método necessita de cálculos intensivos para minimizar as perturbações, exigindo um alto custo computacional [29].

Em geral, ainda há espaço para estudos e pesquisas no campo da defesa contra ataques adversariais no contexto da cibersegurança, especialmente considerando a preservação das informações de pacotes de rede. Como apresentado nas referências e embasamentos deste trabalho, os métodos existentes para geração de amostras adversariais são amplamente aplicados na área de computação visual, que não atende aos mesmos requisitos [30].

6 Conclusões e Trabalhos Futuros

Este trabalho possibilitou avançar nos estudos sobre Aprendizado de Máquina Adversarial, assim como solidificar os conhecimentos sobre diferentes ataques cibernéticos, tais como ataques convencionais e ataques específicos de ML. Em relação aos experimentos feitos com os métodos FGSM (*Fast Gradient Sign Method*) e ZOO (*Zeroth-Order Optimization*) foi possível verificar como as amostras adversariais, quando introduzidas ao conjunto de treinamento do modelo, levam-no a detectar erroneamente um ataque. A aplicação do Treinamento Adversarial, permite conhecer o desempenho do modelo em cenários de pior caso, conhecendo pontos de vulnerabilidade dos atacantes.

Para ambos os conjuntos de dados, CICIDS-2017 e IoT-23, o classificador obtém um desempenho considerável antes da geração de amostras adversariais. Contudo, diante das amostras utilizando o método FGSM, o modelo sofre perdas drásticas de desempenho em todas as métricas, indicando uma alta vulnerabilidade a este tipo de método. Para o método ZOO, em ambos os conjuntos, o classificador sofre perdas significativas, mas não na mesma proporção sofrida com o método anterior. Ou seja, a robustez do modelo, neste caso, é consideravelmente menos afetada por este tipo de método.

Dessa forma, as métricas obtidas com os experimentos realizados elucidam a importância do desenvolvimento de defesas contra ataques adversariais, no que tange a robustez e confiabilidade da rede neural classificadora. Por fim, apesar da pendência do uso do método L-BFGS, os objetivos para a primeira etapa desse projeto foram concluídos e os aprendizados colhidos serão utilizados para o desenvolvimento do restante do projeto, buscando alcançar os objetivos gerais definidos.

Referências

[1] BAHI, M.; BATOUCHE, M. Deep learning for ligand-based virtual screening in drug discovery. In: *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. [S.l.: s.n.], 2018. p. 1–5.

[2] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 44, n. 1.2, p. 206–226, 2000.

[3] GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. *Explaining and Harnessing Adversarial Examples*. 2015.

[4] SAMANGOUEI, P.; KABKAB, M.; CHELLAPPA, R. *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*. 2018.

[5] PAPERNOT, N. et al. *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks*. 2016.

[6] MACAS, M.; WU, C.; FUERTES, W. Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. *Expert Systems with Applications*, v. 238, p. 122223, 2024. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417423027252>.

[7] AKHTAR, N.; MIAN, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *CoRR*, abs/1801.00553, 2018. Disponível em: <http://arxiv.org/abs/1801.00553>.

[8] MACHADO, G. R.; SILVA, E.; GOLDSCHMIDT, R. R. Adversarial machine learning in image classification: A survey towards the defender's perspective. *CoRR*, abs/2009.03728, 2020. Disponível em: <https://arxiv.org/abs/2009.03728>.

[9] TRAMèR, F. et al. *Ensemble Adversarial Training: Attacks and Defenses*. 2020.

[10] MADRY, A. et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. Disponível em: <https://arxiv.org/abs/1706.06083>.

[11] SZEGEDY, C. et al. *Intriguing properties of neural networks*. 2014.

[12] CHEN, L.; YE, Y. Secmd: Make machine learning more secure against adversarial malware attacks. In: PENG, W.; ALAHAKOON, D.; LI, X. (Ed.). *AI 2017: Advances in Artificial Intelligence*. Cham: Springer International Publishing, 2017. p. 76–89. ISBN 978-3-319-63004-5.

[13] BARRENO, M. et al. The security of machine learning. *Mach. Learn.*, Springer Science and Business Media LLC, v. 81, n. 2, p. 121–148, nov. 2010.

[14] CHEN, L.; YE, Y.; BOURLAI, T. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In: *2017 European Intelligence and Security Informatics Conference (EISIC)*. [S.l.: s.n.], 2017. p. 99–106.

[15] LIU, J. et al. Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems. *IEEE Communications Surveys Tutorials*, v. 24, n. 1, p. 123–159, 2022.

[16] HUANG, L. et al. Adversarial machine learning. In: *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*. New York, NY, USA: Association for Computing Machinery, 2011. (AISec '11), p. 43–58. ISBN 9781450310031. Disponível em: <https://doi.org/10.1145/2046684.2046692>.

[17] APRUZZESE, G. et al. Addressing adversarial attacks against security systems based on machine learning. In: *2019 11th International Conference on Cyber Conflict (CyCon)*. [S.l.: s.n.], 2019. v. 900, p. 1–18.

[18] BIGGIO, B. et al. Security evaluation of support vector machines in adversarial environments. *CoRR*, abs/1401.7727, 2014. Disponível em: <http://arxiv.org/abs/1401.7727>.

[19] GERON, A. *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. 2. ed. Sebastopol, CA: O'Reilly Media, 2019.

[20] YAO, X. Evolving artificial neural networks. *Proceedings of the IEEE*, v. 87, n. 9, p. 1423–1447, 1999.

[21] CHEN, P.-Y. et al. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017. Disponível em: <http://dx.doi.org/10.1145/3128572.3140448>.

[22] ZHOU, S. et al. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 55, n. 8, dec 2022. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3547330>.

[23] BAI, T. et al. Recent advances in adversarial training for adversarial robustness. *CoRR*, abs/2102.01356, 2021. Disponível em: <https://arxiv.org/abs/2102.01356>.

[24] ROSENBERG, I. et al. *Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain*. 2021.

[25] SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: INSTICC. *Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*. [S.l.]: SciTePress, 2018. p. 108–116. ISBN 978-989-758-282-0.

[26] DRAPER-GIL., G. et al. Characterization of encrypted and vpn traffic using time-related features. In: INSTICC. *Proceedings of the 2nd International Conference on Information Systems Security and Privacy - ICISSP*. [S.l.]: SciTePress, 2016. p. 407–414. ISBN 978-989-758-167-0. ISSN 2184-4356.

[27] GARCIA, S.; PARMISANO, A.; ERQUIAGA, M. J. *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. [S.l.]: Zenodo, 2020.

[28] MCCARTHY, A. et al. Defending against adversarial machine learning attacks using hierarchical learning: A case study on network traffic attack classification. *Journal of Information Security and Applications*, Elsevier, v. 72, p. 103398, 2023.

[29] DEBICHA, I. et al. *Review on the Feasibility of Adversarial Evasion Attacks and Defenses for Network Intrusion Detection Systems*. 2023. Disponível em: <https://arxiv.org/abs/2303.07003>.

[30] MCCARTHY, A. et al. Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey. *Journal of Cybersecurity and Privacy*, v. 2, n. 1, p. 154–190, 2022. ISSN 2624-800X. Disponível em: <https://www.mdpi.com/2624-800X/2/1/10>.

[31] RAGHUNATHAN, A.; STEINHARDT, J.; LIANG, P. *Certified Defenses against Adversarial Examples*. 2020.