Journal of Technology & Information

Aplicação de Redes Adversárias Generativas na Detecção e Prevenção de Ataques Cibernéticos

Application of Generative Adversary Networks in the Detection and Prevention of Cyber Attacks

Recebido/*Received*: 01/02/2025 | Revisado/*Revised*: 08/02/2025 | Aceito/*Accepted*: 10/02/2025 | Publicado/*Publish*: 12/02/2025
https://www.doi.org/10.5281/zenodo.14849898

Brunno Mendonça
Fatec Santana de Parnaíba
https://orcid.org/0009-0006-4186-8991
brunnoitm@gmail.com

Erick Santos
Fatec Santana de Parnaíba
https://orcid.org/0009-0005-5479-5894
erisk_snt@icloud.com

Guilherme Dionysio
Fatec Santana de Parnaíba
https://orcid.org/0009-0001-5195-5451
guidionysio@outlook.com

Eugenio Bittencourt
Fatec Santana de Parnaíba
https://orcid.org/0009-0003-4535-3379
eugenio.bittencourt@fatec.sp.gov.br

Resumo

Este trabalho investiga o uso de *Generative Adversarial Networks* (*GANs*) como ferramenta para fortalecer a segurança cibernética, especificamente na detecção de acessos maliciosos. Utilizando um estudo de caso único em ambiente controlado, analisando a eficácia das *GANs* para distinguir entre acessos legítimos e tentativas de intrusão, com foco na proteção da integridade dos sistemas de informação. A metodologia de estudo de caso, inclui coleta de dados de um sistema de rede, construção e treinamento de modelos *GAN* e *Machine Learning (ML)*, e análise de hiperparâmetros essenciais, como função de perda e taxa de aprendizado, para otimização do desempenho do modelo. Os resultados mostram o potencial das *GANs* em complementar sistemas de segurança, reforçando sua robustez contra-ataques de *brute force* e outras ameaças. Este estudo contribui para a pesquisa sobre a aplicação de *GANs* na Segurança da Informação, destacando o papel da inteligência artificial na defesa de redes corporativas.

Palavras-Chave: *GANs*; *Machine Learning*; *dataset*; treinamento de modelos; Segurança da Informação.

Journal of Technology & Information

Abstract

This paper investigates the use of Generative Adversary Networks (GANs) as a tool to strengthen cyber security, specifically in the detection of malicious access. Using a single case study in a controlled environment, we analyze the effectiveness of GANs in distinguishing between legitimate accesses and intrusion attempts, with a focus on protecting the integrity of information systems. The methodology casestudy, included collecting data from a network system, building and training GAN and Machine Learning (ML) models, and analyzing key hyperparameters, such as loss function and learning rate, to optimize model performance. The results show the potential of GANs to complement security systems, reinforcing their robustness against brute force attacks and other threats. This study contributes to research into the application of GANs in information security, highlighting the role of artificial intelligence in defending corporate networks.

Keywords: *GANs; Machine Learning; dataset; model training; Information security.*

1. Introdução

Com os avanços tecnológicos, é possível observar um evidente aumento da presença das *Machine Learnings (*ML*)* no nosso cotidiano. No entanto, com base nessa evolutiva, surgiram necessidades de promover uma atenção maior para os aspectos de Segurança da Informação sobre os dados que trafegam nas ML.

As *Generative Adversarial Networks* (GANs) se destacam cada vez mais como uma ferramenta para a geração de dados sintéticos a partir de amostra de dados reais (GOODFELLOW *et al.,* 2014). Embora sua aplicação mais conhecida seja na geração de imagens, elas também possuem um papel fundamental na Segurança da Informação podendo ser usadas para aperfeiçoar a detecção de anomalias e sistemas contra-ataques/defesas adversárias.

Este estudo propõe uma imersão na ligação entre GAN e Segurança da Informação, buscando explorar os fundamentos teóricos sobre o modelo e sua aplicação no combate contra-ataques cibernéticos. A partir de um ambiente controlado, foi realizado um estudo de caso único que integra a GAN e ML, em que ambas trabalharam em conjunto no processo de aprendizado com dados reais e gerados. O sistema foi projetado para rotular automaticamente essas atividades como suspeitas ou não, contribuindo para a implementação de políticas e ajustes na configuração dos sistemas de segurança de uma corporação.

Utilizando de uma abordagem transversal, este estudo contribui para o avanço do conhecimento em segurança cibernética, dando ênfase na importância das GANs para a proteção das *ML* e na integridade dos sistemas da informação (MIRZA & OSINDERO, 2014).

O objetivo deste estudo limita-se a explorar o uso do modelo GAN como o método de aprimoramento de sistemas de defesa baseados em ML, analisando sua capacidade para a geração de dados sintéticos realistas. Com isso, este estudo buscou a responder à seguinte pergunta: como as GANs podem ser aplicadas para fortalecer a defesa de sistemas de informação?

A pesquisa está organizada da seguinte maneira: a seção 2 está subdividida, contendo o referencial teórico aplicado durante o trabalho; a seção 3 aborda sobre a metodologia utilizada nesse estudo; já a seção 4 apresenta o desenvolvimento e os resultados obtidos; por fim, a seção 5 traz a conclusão deste trabalho e nossas sugestões para pesquisas futuras.

2. Referencial Teórico

2.1. *Machine Learning*

O *Machine Learning*, traduzido para o português como Aprendizado de Máquina, é um subconjunto da área de Inteligência Artificial (*Artificial Intelligence* - AI), e sua principal característica é o desenvolvimento de técnicas computacionais sobre o aprendizado e a capacidade de adquirir conhecimento de forma autônoma (MONARD & BARANAUSKAS, 2003).

Em diversos estudos, o ML se destaca por propor soluções inovadoras a desafios complexos. Com suas técnicas de aprendizado, ele ganha destaque no treinamento de algoritmos para entender a relação entre entradas e saídas de dados, também nas inter-relações das características de agrupamento de dados (SIEMURI *et al*., 2022).

Os métodos de aprendizado de uma ML podem ser divididos em aprendizado supervisionado e não supervisionado. No aprendizado supervisionado, o algoritmo é treinado utilizando conjuntos de exemplos contidos em um *dataset* rotulado, ou seja, um *dataset* com saídas pré-definidas, permitindo que a máquina aprenda a relacionar entradas e saídas específicas. Enquanto o aprendizado não supervisionado trabalha com dados sem rótulos buscando identificar padrões ou agrupamentos de dados (MONARD & BARANAUSKAS, 2003).

Para avaliar a eficácia desses algoritmos, métricas como *Recall*, que mede a capacidade de identificar corretamente os positivos reais, e o *F1-scor*e, que combina a precisão e o *recall* em uma única métrica harmônica, são utilizadas frequentemente para garantir que o modelo seja executado de forma robusta, principalmente em cenários onde os dados estão desbalanceados (POWERS, 2020).

2.2. *Deep Learning*

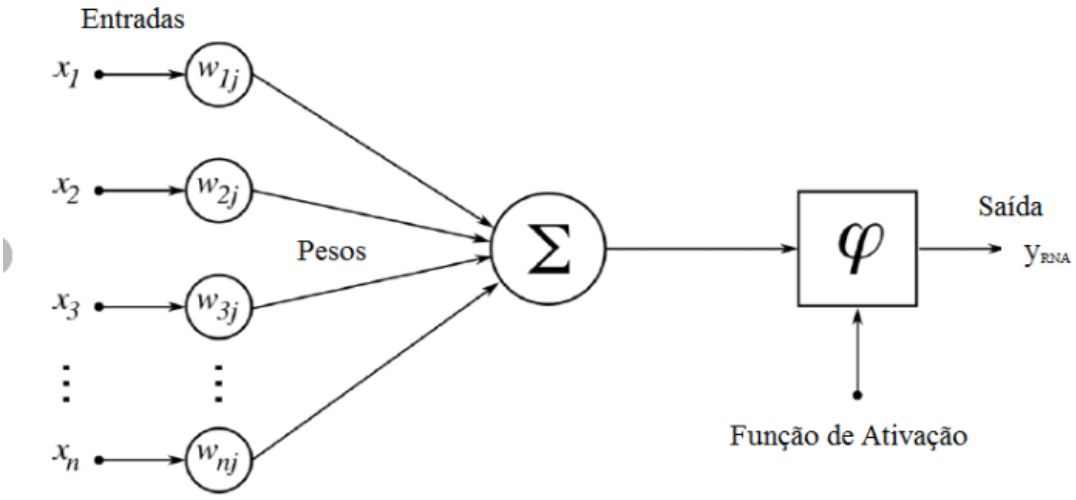
*Deep Learning (*DL*)* ou Aprendizado Profundo é um subconjunto de ML que utiliza redes neurais artificiais profundas focando no reconhecimento de padrões em diferentes contextos (HOSAKI & RIBEIRO, 2021). Porém para entender seu funcionamento, é requerido conhecimentos prévios sobre aprendizado supervisionado e redes neurais de múltiplas camadas neurônios (HOSAKI & RIBEIRO, 2021).

A inspiração para criação das Redes Neurais Artificiais (RNA) surgiu com base nos neurônios biológicos e são formados por unidades simples, chamados neurônios artificiais (FONTANA & RONCALLI, 2023).

O RNA é composto por neurônios artificiais interconectados, através dessa conexão tornam-se capazes de processar informações, transmitir sinais e realizar cálculos por meio de conexões ponderadas (FONTANA & RONCALLI, 2023). Isso pode proporcionar soluções de problemas complexos, pois há uma enorme gama de neurônios trabalhando em conjuntos (ALMEIDA, 2019).

Um neurônio artificial é fundamental para uma RNA. Para a base do modelo neural, existem três elementos cruciais: Conjunto de sinapse, também conhecido como "entradas", cada entrada deve ser caracterizada por um peso e por fim uma função de ativação para somar os pesos das entradas e assim restringir a amplitude da saída de um neurônio (FONTANA & RONCALLI, 2023). A Figura 1 evidência os elementos citados.

Figura 1 – Estrutura de um Neurônio Artificial



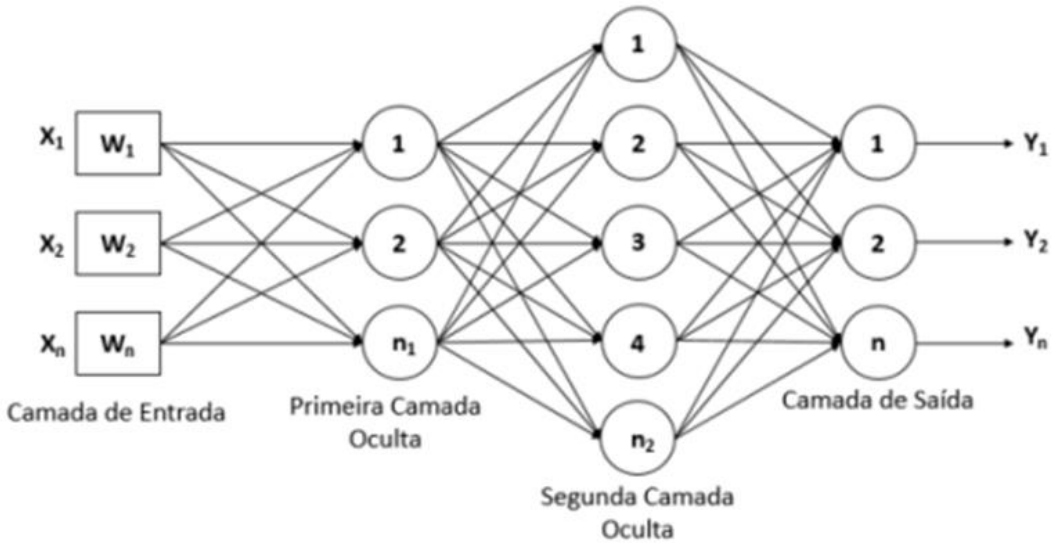
Fonte: Adaptado de MARTINIANO *et al.*, 2016.

O que difere uma rede neural de camada simples para uma de múltiplas camadas é a sua arquitetura que contém uma ou mais camadas ocultas que também são conhecidas como neurônios ocultos. Sua função é intervir entre a camada externa e a saída da rede de uma maneira útil, o que permite uma aprendizagem de forma mais complexas (HAYKIN, 1994).

Segundo NUNES *et al*. (2023), as camadas são conectadas através de nós e cada camada herda o resultado da camada anterior e produz a saída para a camada seguinte. A primeira camada é denominada camada de entrada, as camadas intermediarias são denominadas camadas ocultas e a última camada é a camada de saída (Figura 2). Cada neurônio das camadas ocultas é interconectado a todos os outros neurônios das camadas

anteriores e posteriores, formando assim as redes neurais multicamadas (NUNES *et al*., 2023).

Figura 2 – Redes Neurais com Camadas Ocultas



Fonte: Fontana, Guilherme; Roncalli, Felipe., 2023.

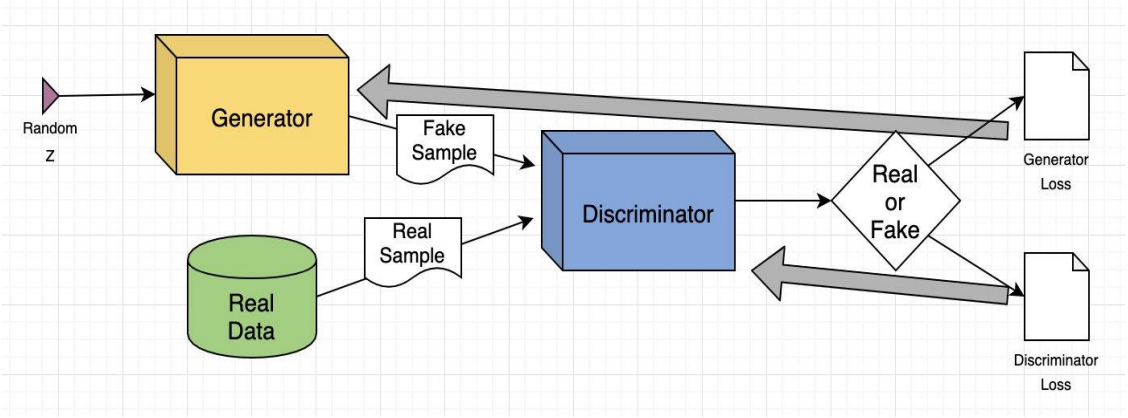
Segundo GOODFELLOW *et al* (2017), os métodos de *DL* visam descobrir um modelo a partir de um conjunto de dados e diretrizes para o aprendizado de GANs, permitindo que o algoritmo descubra padrões complexos sem intervenção manual significativa.

2.3. *Generative Adversarial Networks*

As Redes Adversárias Generativas também conhecida como *Generative Adversarial Networks* são classificadas como um tipo de rede neural artificial, desenvolvida e aplicada para tarefas de geração de dados sintéticos (GOODFELLOW *et al*., 2014; ARORA & SHANTANU, 2019). O modelo foi proposto por GOODFELLOW em 2014 em um artigo chamado "G*enerative Adversarial Nets*".

De acordo com ALQAHTANI (2019), o treinamento de uma *GAN* envolve uma dinâmica de competição entre dois modelos: o Gerador (G) e o Discriminador (D), conforme ilustrado na Figura 3, onde um sempre tenta se sobressair ao outro. G recebe um vetor de ruído (*Random Z*), permitindo que produza amostras de dados variadas a cada ciclo do processo para criar amostras sintéticas o mais próximo possível de dados realistas. D, por sua vez, recebe tanto amostras reais quanto geradas e tenta classificá-las como reais ou falsas (*AWS*, 2023).

Figura 3 – Funcionamento de uma *GAN*



Fonte: aws.amazon.com

No artigo "*Are* GANs *Created Equal? A Large-Scale Study*", de LUCIC (2019), é defendido que esses ajustes, como o vetor de ruído, se tornam essenciais para equilibrar a interação de G e D sem criar uma instabilidade no treinamento de uma GAN. Durante o treinamento, hiperparâmetros podem ser utilizados com o objetivo de maximizar o desempenho dos modelos de MLs. Numa *GAN*, hiperparâmetros comuns são: Função de perda (*loos function)*, que avalia a precisão do modelo, a taxa de aprendizado (*learning rate*), que controla a velocidade de ajuste dos pesos, tamanho de lote (*batch size*), definindo quantas amostras são processadas antes de cada atualização (WANG *et al.*, 2019).

Segundo *Data Science Academy*, no capítulo 28 do livro *Deep Learning Book* (2022), as épocas (*epochs*), que são outro tipo de hiperparâmetro, definem quantas vezes o algoritmo de treinamento percorrerá o conjunto de dados completo. Em outras palavras o modelo vai analisar todas as amostras de dados disponíveis antes de iniciar uma nova época. O controle de épocas é necessário para evitar problemas como *underfitting (*subajuste), ou seja, o modelo se torna muito simples por não aprender o suficiente com o conjunto de dados disponível, enquanto o *overfitting* ou sobreajuste, acontece quando o modelo se torna muito complexo, memorizando os dados de treinamento ao invés de aprender os padrões (*Deep Learning Book,* 2022).

2.4. Segurança da Informação

A Segurança da Informação desempenha um papel vital na preservação da integridade, confidencialidade e disponibilidade dos dados, e que os incidentes de segurança podem resultar em sérios danos financeiros e prejudicar a reputação de uma organização (GALEGALE *et al*., 2017)

De acordo com SCARFONE e MELL (2007), a adoção de ferramentas, conceitos e boas práticas é fundamental para garantir a segurança de redes empresariais. Entre os recursos destacados pelos autores, os firewalls desempenham um papel essencial no monitoramento do tráfego de rede, contribuindo para a proteção e integridade dos sistemas. O uso do aprendizado de máquina na segurança, a definição de horários críticos de acesso, documentação de processos, monitoramento em tempo real do tráfego de rede, gerenciamento de volume de dados não pode ser deixados de lado pois também desempenham papel importante para um ambiente robusto (SOMMER & PAXSON, 2010).

Segundo NAJAFABADI, *et al* (2014), em redes de computadores, um dos ataques predominantes são ataques de Força Bruta (*brute force)*. Habitualmente as senhas escolhidas por humanos são institivamente frágeis porque são selecionadas de um domínio de conhecimento superficial (NAJAFABADI *et al*., 2014).

A maioria dos ataques se iniciam através de um ataque de *brute force* em aplicativos de fácil acesso a pessoas públicas (KASPERSKY, 2022). Considerando que o *brute force* é o ataque inicial, nos últimos anos houve uma crescente no desenvolvimento de aplicações em *ML* para detecção desses ataques (KHARISMADHANY *et al.*, 2022).

2.5. *Brute Force*

O ataque de Força Bruta (*Brute Force*) é uma técnica onde invasores tentam acessar sistemas sem autorização, testando uma vasta quantidade de combinações de senhas até encontrar a correta (NAJAFABADI *et al*., 2014). A utilização de sistemas de detecção baseados em *ML* vem sendo utilizados para detectar esse tipo de ataque, pois conseguem identificar padrões de comportamento suspeitos e bloquear tentativas automatizadas em tempo real, elevando a segurança dos sistemas (*AWS*, 2023).

De acordo com SALAMA, *et al*. (2023), esses ataques podem ser realizados em grande escala em ambientes de nuvem, explorando a capacidade computacional elástica dessas infraestruturas para acelerar o processo de tentativa e erro. Além disso, a detecção e a mitigação de ataques de *brute force* exigem o uso de contramedidas eficazes, como a limitação de tentativas de *login* e o uso de técnicas avançadas de monitoramento e autenticação multifatorial, que podem impedir a exploração de vulnerabilidades (KHARISMADHANY *et al.*, 2022).

3. Metodologia

A metodologia aplicada neste estudo foi o estudo de caso único, fundamentado na abordagem proposta por Yin (2021). Essa escolha deve-se à necessidade de compreender profundamente a aplicação das redes adversárias generativas (GANs) em um contexto específico e controlado. O estudo adota um método qualitativo (Theophilo & Martins, 2016), ideal para explorar fenômenos complexos no campo da segurança cibernética.

Na Tabela 1 é possível observar as características utilizadas para realização desse estudo.

Tabela 1 – Características do estudo

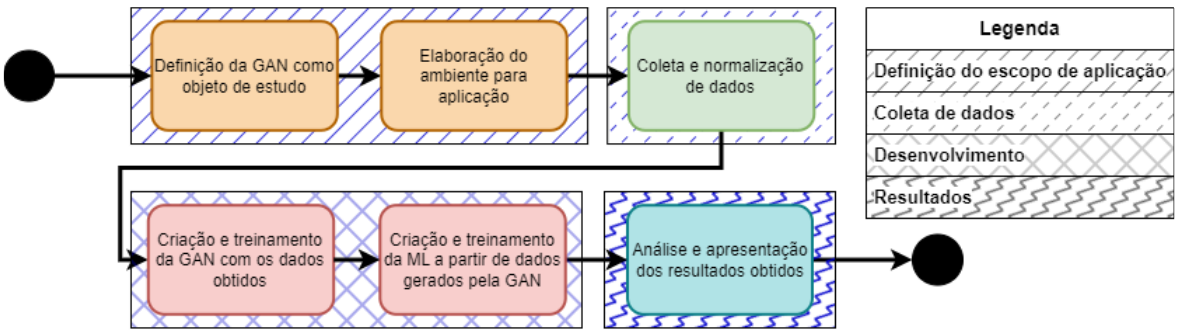
Item	Descrição	Autor(es)
Questão de Pesquisa	- Como as GANS podem ser aplicadas como método de defesa no mundo cibernético?	
Natureza	- Qualitativa	Theophilo & Martins (2016)
Metodologia	- Estudo de caso único.	Yin (2021)
Coleta de Dados	- Análise experimental	Gil (2022)
Unidade de análise	- Ambiente controlado	

Fonte: Elaborado pelos autores.

3.1. Processo Metodológico

O processo metodológico aplicado neste estudo, possui seis etapas e é dividido em quatro seções, conforme demonstrado na Figura 4.

Figura 4 – Etapas do processo metodológico



Fonte: Elaborado pelos autores.

1. **Definição da GAN como objeto de estudo**: A GAN foi escolhida para explorar a criação de dados sintéticos que possam ser aplicados à segurança cibernética.
2. **Elaboração do ambiente para aplicação**: Foi configurado um ambiente controlado, simulando a topologia de uma empresa para a reprodução de ataques *brute force*.
3. **Coleta e normalização de dados**: A retira dos logs do *firewall* foram tratadas a partir de um algoritmo e complementadas por meio de uma API de geolocalização de *IPs*, formando um *dataset* normalizado.
4. **Criação e treinamento da GAN com os dados obtidos**: Foi definido a linguagem de programação e bibliotecas paro o desenvolvimento da GAN. O treinamento baseou-se no uso dos dados do *dataset* normalizado.
5. **Criação e treinamento da ML a partir de dados gerados pela GAN**: Foi definido a linguagem de programação e bibliotecas paro o desenvolvimento da ML. O treinamento baseou-se no uso dos dados sintéticos gerados pela GAN.
6. **Análise e apresentação dos resultados obtidos**: Foi documentado o desempenho do modelo, destacando a aplicação da GAN como método de aprimoramento de sistemas de segurança de redes baseados em ML.

4. Análise e Interpretação dos Resultados

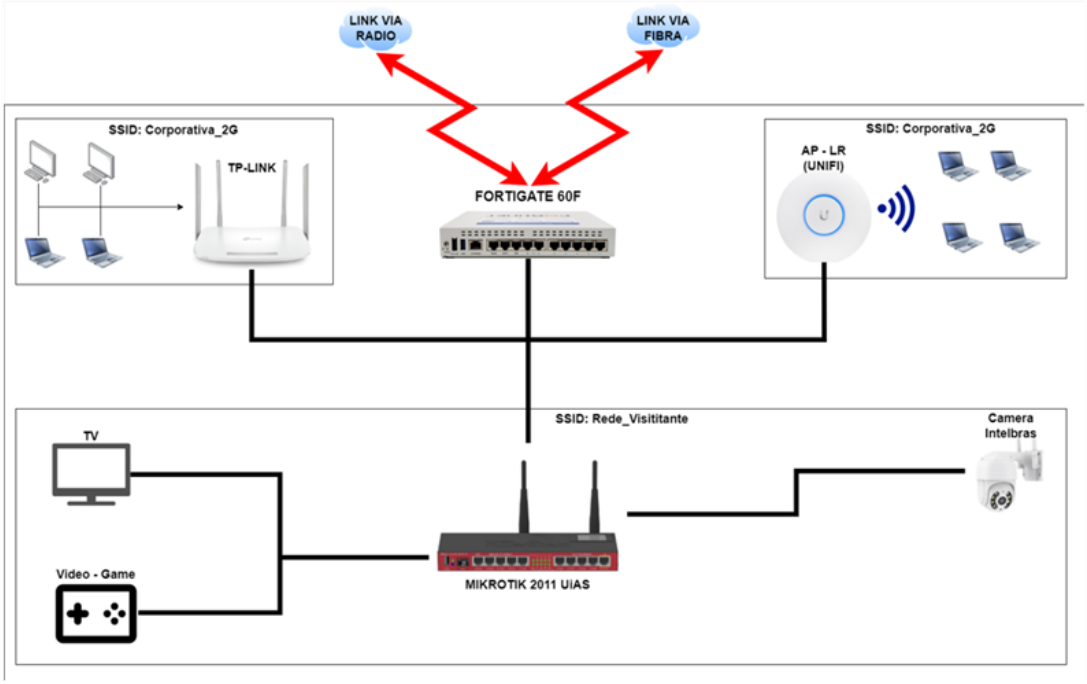
4.1. Montagem do laboratório

A infraestrutura de rede utilizada para este estudo, foi configurada em um ambiente controlado, projetado para simular com precisão um ambiente corporativo. Nesse cenário foram aplicadas técnicas de ML e GAN como métodos para detecção de ataques de *Brute Force* a um *firewall Fortigate 60F* (Figura 5).

A rede é composta por dois *links* de internet, fibra e rádio, conectando-se a um *Firewall. O firewall* utilizado no ambiente foi um *Fortigate 60F que* além de atuar como a primeira linha de defesa contra ameaças cibernéticas, é essencial para implementação da *SD-WAN* (*Software-Defined Wide Area Network*) que atua por meio da *Performance SLA* (*Service Level Agreement*). A *SD-WAN* é uma abordagem de rede que permite o gerenciamento inteligente e o balanceamento dinâmico do tráfego de dados, utilizando os

links de forma segura e eficaz (CISCO, 2024). Já o *Performance SLA* refere-se a um acordo de nível de serviço configurado no *firewall*, no qual são definidas as métricas para avaliação do desempenho dos links e caso seja identificado anormalidades nessas métricas, o *Fortigate* altera automaticamente a navegação para o outro link garantindo a continuidade dos serviços.

Figura 5 – Topologia ambiente controlado.



Fonte: Elaborado pelos autores.

O balanceamento de links proporcionado pelo *firewall* não tem somente a função de otimizar o tráfego conforme as necessidades da rede, mas também proporciona uma prevenção a exploração de vulnerabilidades do ambiente. Sem um equipamento de segurança robusto, o ambiente se torna extremamente vulnerável a ataques, como invasões e comprometimento dos dados, que podem pôr fim resultar em prejuízos financeiro e danos a reputação de uma empresa (GALEGALE *et al*., 2017).

Além disso, o *Fortigate 60F* é o responsável pela distribuição da rede *Dynamic Host Configuration Protocol (DHCP*), que é um protocolo que possuí a função de automatizar a atribuição de *IPs*, facilitando a gestão e o gerenciamento da rede (CISCO, 2017). A rede está dividida em três sub-redes: Rede de gerência, onde se conectaram apenas dispositivos que terão permissões de gerenciar a rede; Rede Corporativa possibilitando a conexão de todos os equipamentos presentes no ambiente desenvolvendo tarefas corporativas; Rede de visitantes, que abriga dispositivos temporários. A funcionalidade do *firewall* é crítica para garantir que cada sub-rede opere de forma segura e isolada. Portanto a certificação da segurança deste dispositivo é fundamental para a integridade e continuidade dos serviços prestados no ambiente corporativo analisado, destacando a importância de manter uma postura proativa em relação a segurança cibernética.

4.2. Ataques de *Brute Force*

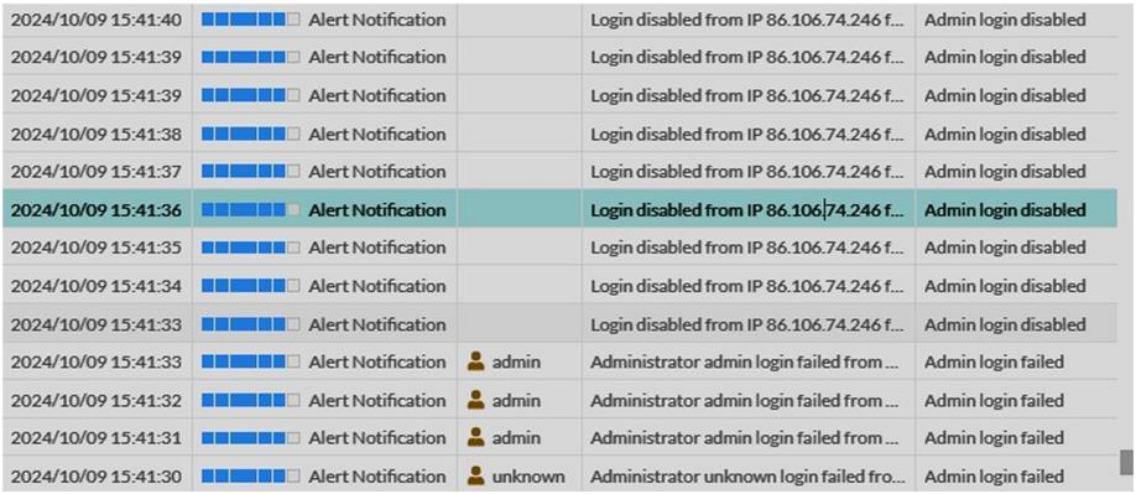
Após montagem e configuração do ambiente controlado, foi realizado escaneamento de portas abertas na rede, com objetivo de encontrar possíveis vulnerabilidades em serviços estabelecidos nesta rede. Ao executá-la foram identificadas portas abertas no *firewall*, caracterizando um possível ponto de vulnerabilidade (Figura 6).

Figura 6 – Registro de escaneamento de portas abertas no *firewall*.

Fonte: Ferramenta *NMAP*.

Durante o período de testes, foram analisados eventos do *firewall* e identificadas diversas tentativas de acesso não autorizado, que utilizaram ataques de *brute force* como método para tentar obter acesso ao sistema (Figura 7).

Figura 7 – Amostra de tentativas de acesso.



2024/10/09 15:41:40	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:39	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:39	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:38	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:37	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:36	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:35	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:34	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:33	Alert Notification		Login disabled from IP 86.106.74.246 f...	Admin login disabled
2024/10/09 15:41:33	Alert Notification	admin	Administrator admin login failed from ...	Admin login failed
2024/10/09 15:41:32	Alert Notification	admin	Administrator admin login failed from ...	Admin login failed
2024/10/09 15:41:31	Alert Notification	admin	Administrator admin login failed from ...	Admin login failed
2024/10/09 15:41:30	Alert Notification	unknown	Administrator unknown login failed fro...	Admin login failed

Fonte: Painel *Fortigate.*

Analisando os *logs* do dispositivo, foi possível identificar que as tentativas de acesso indevido não foram bloqueadas, uma vez que o *firewall* apenas desabilita o remetente por sessenta segundos após três erros consecutivos e após finalizar o tempo de espera, o invasor pode realizar outras três tentativas. Com base nesta análise, foi criado uma *ML* que identifica as tentativas de *logins* e as classifica como *brute force* ou acesso legitimo para mitigar os acessos indesejados.

4.3. *Dataset* de Treinamento

4.3.1. Montagem e Preparação do *Dataset*

O *Dataset* foi elaborado a partir de registros reais de tentativas de acesso, englobando tanto dados de ataques de *brute force* quanto acessos legítimos, que foram normalizados para corresponder a estrutura esperada pelos modelos que irão utilizá-la. A normalização dos dados ocorreu através de um algoritmo simples que efetuou leitura do arquivo de *log* extraído do *syslog* conectado ao *firewall* (Figura 8), a partir do qual foram

coletadas as seguintes informações sobre a tentativas de acesso: Endereço *IP (Internet Protocol),* Método da tentativa, Usuário de *login*, Data e Hora e o *Status* da tentativa (Acesso permitido ou Acesso Negado).

Em seguida foi utilizado uma *API Opensource* (código aberto) para descobrir a geolocalização do *IP* que está realizando a tentativa de acesso, para assim determinar o país de origem, os resultados são armazenados em um cache para otimização. Após isso, o código calcula métricas como número total de tentativas, a porcentagem de sucesso, porcentagem de falha e finalmente, a média de tempo entre as tentativas, organizando esses dados em uma estrutura Tabular.

Figura 8 – Amostra de dados de log do *syslog* conectado ao *firewall* sem tratamento.

Time	IP	Host	Facilit	Priority	Ta	Message
Oct 11 15:39:29	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:18 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:31	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:21 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:33	172.16.0.1	date=2024-10-11	local7	alert		time=15:39:22 devname="FortiGate-60F" devid="FGT60FTK
Oct 11 15:39:36	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:25 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:37	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:26 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:37	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:26 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:37	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:26 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:37	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:26 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:38	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:27 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:39	172.16.0.1	date=2024-10-11	local7	notice		time=15:39:29 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:41	172.16.0.1	date=2024-10-11	local7	info		time=15:39:31 devname="FortiGate-60F" devid="FGT60FTK20074
Oct 11 15:39:43	172.16.0.1	date=2024-10-11	local7	info		time=15:39:33 devname="FortiGate-60F" devid="FGT60FTK20074

Fonte: *Visual Syslog Server*

O resultado desse processamento é um *dataset* gerado em formato *Excel (xlsx)*, que contém as informações normalizadas e agrupadas, permitindo análises posteriores e facilitando a identificação de padrões de comportamento anômalo, como tentativas de *brute force.* Os valores normalizados no arquivo posteriormente foram analisados manualmente para identificação dos acessos legítimos e acessos provenientes de tentativas de ataques de *brute force*, agregando a coluna "*is_bruteforce*" rotuladas com valores binários sendo: *brute force* como 1 e acessos legítimos como 0, posicionado assim como o modelo da Tabela 2 apresenta.

Tabela 2 – Modelo da tabela de saída após normalização e preenchimento.

Endereço IP de origem	**Número de Tentativas**	**Porcentagem de Sucesso (%)**	**Porcentagem de Falha (%)**	**Média de Tempo entre Tentativas (segundos)**	**País**	**is_brute force**
IP de origem do atacante	Tentativas totais da origem no registro	Porcentagem sobre sucesso total de tentativas	Porcentagem sobre falha total de tentativas	Média de tempo entre todas as tentativas executadas	País de origem do IP do atacante	Binário de rotulagem
IP de origem do atacante	Tentativas totais da origem no registro	Porcentagem sobre sucesso total de tentativas	Porcentagem sobre falha total de tentativas	Média de tempo entre todas as tentativas executadas	País de origem do IP do atacante	Binário de rotulagem
IP de origem do atacante	Tentativas totais da origem no registro	Porcentagem sobre sucesso total de tentativas	Porcentagem sobre falha total de tentativas	Média de tempo entre todas as tentativas executadas	País de origem do IP do atacante	Binário de rotulagem

Fonte: Elaborado pelos autores.

Os dados revelam que, de 869 de tentativas de acesso analisadas, 502 (57,7%) foram tentativas de *brute force*, enquanto apenas 367 (42,3%) representaram acessos legítimos, conforme apresentado na Tabela 3. Essa discrepância destaca a necessidade de um sistema mais robusto para detectar e neutralizar ataques de *brute force*.

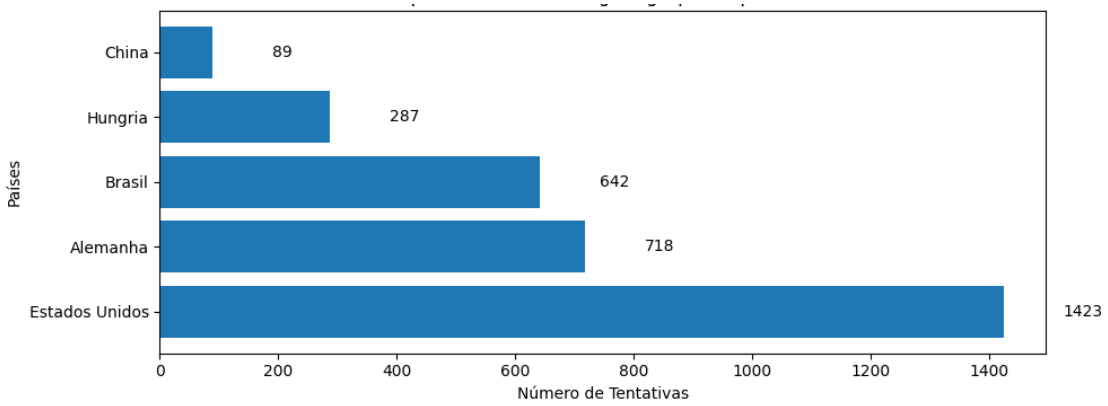
Tabela 3 – Acessos analisados

Classe	**Quantidade**	**Porcentagem (%)**
Acesso Legítimo	367	42,3
Tentativa Brute Force	502	57,7
Total	869	100,0

Fonte: Elaborado pelos autores

Além disso, a normalização dos dados forneceu informações relevantes com base de distribuição geográficas. A Figura 10 demonstra as cinco principais localidades dessas tentativas, destacando que os Estados Unidos registraram o maior número de acessos suspeitos. As informações contribuem com possíveis origens dos ataques, o que pode auxiliar nas tomadas de decisão para implementar medidas de segurança direcionadas.

Figura 10 – Top 5 tentativas de *login* agrupadas por país.



Fonte: Elaborado pelos autores

4.4. Implementação da GAN e ML

4.4.1. Montagem e Treinamento do Modelo de GAN

O modelo GAN foi montado e treinado utilizando a linguagem de programação *Python*, com a biblioteca *PyTorc*h. O treinamento foi realizado em um *hardware* equipado com uma *GPU* NVIDIA *RTX* 3060, utilizando CUDA, tecnologia nativa das placas de vídeo NVIDIA para acelerar o processamento através do uso do processamento gráfico. Durante o processo de treinamento, a *GAN* foi alimentada com dados reais extraídos do *dataset* normalizado. Esses dados foram utilizados para formar grupos que o Discriminador (D) analisou. O D estava responsável por avaliar se os dados são reais ou gerados. Enquanto isso, o Gerador (G) criava dados a partir de um vetor de ruído, adicionando variabilidade aos dados, produzindo uma maior variedade de saídas.

A estrutura do modelo incluiu o uso de camadas densas, compostas por neurônios interconectados. No projeto, a camada oculta do modelo contém 128 neurônios (Código 1), auxiliando o modelo no aprendizado de características dos dados de maneira precisa. A taxa de aprendizado foi definida como 0,0002, frequentemente utilizada em modelos *GAN* para estabilidade do treinamento. Através da aprendizagem contínua tornou-o capaz de gerar dados sintéticos com características dos dados reais.

Código 1 – Estrutura do G e D do algoritmo.

```
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(4, 256),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(256, 256),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(256, 4)
        )


    def forward(self, z):
        output = self.model(z)
        success_percentage = torch.clamp(output[:, 1], 0, 100)
        fail_percentage = 100 - success_percentage
        average_time = torch.clamp(output[:, 3], 0, 1000)
        return torch.stack([output[:, 0], success_percentage,
fail_percentage, average_time], dim=1)
```

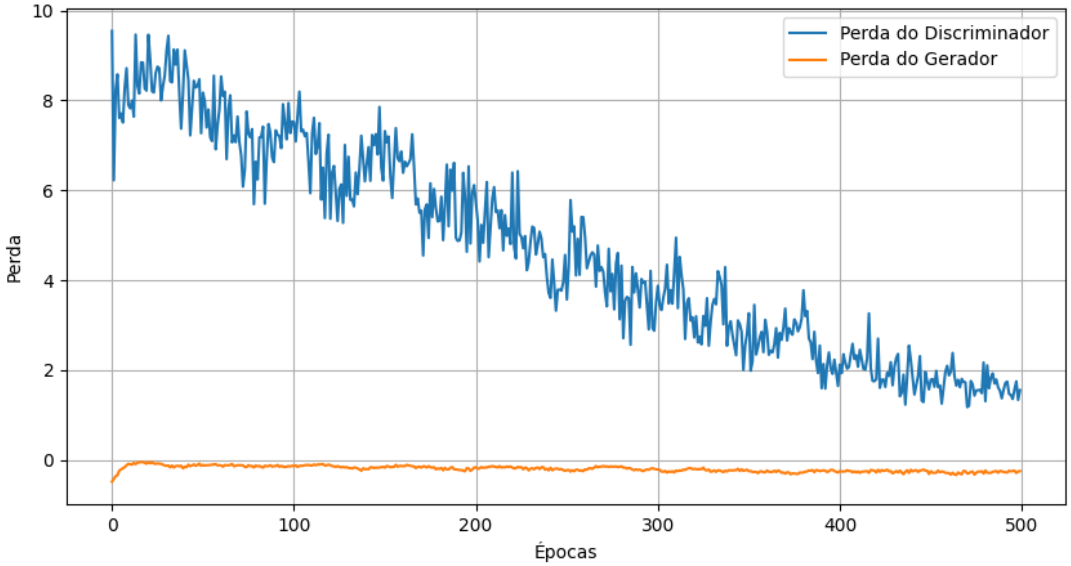
```
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(4, 256),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(256, 128),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Linear(64, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        return self.model(x)
```

Fonte: Elaborado pelos autores.

Os valores de perda do D e do G evoluíram durante o treinamento (Figura 12 e Tabela 4). A perda do D *(d_loss)* incialmente foi de 9,54 na época 0 e, com a sequência do treinamento houve a redução para 1,79 na época 450.

O comportamento indica que o D evoluiu no quesito de distinção entre dados sintéticos e reais. Em paralelo, a perda do G (*g_loss*) apresentou em seu início o valor de -0,48 na época 0, se aproximando a -0,25 ao final na época 450, demonstrando que o G estabeleceu uma linearidade de melhoria no que se refere a qualidade de dados sintéticos produzidos, dificultando o trabalho do D ao distingui-los.

Figura 12 - Gráfico de progressão perdas durante o treinamento da GAN.



Fonte: Elaborado pelos autores.

Por sua vez, é possível observar a variação na precisão do D durante o treinamento. A precisão dos dados reais (*Acc_real*) primariamente flutuava entorno de 81,25%, e finalizou ao final da época 450 com 84,38% com constantes pequenas variações, enquanto a precisão dos dados falsos *(Acc_fake)* iniciou em 59,38% e subiu para 89,06% na mesma época, conforme ilustrado na Tabela 4. A precisão total *(Acc_total)* reflete o progresso geral do modelo, alcançando ao fim do treinamento uma precisão de 86,72% ao final da última época.

O modelo foi treinado por 500 épocas, sendo monitorado a cada 50 épocas para identificação e coleta de estatísticas de treinamento, com objetivo de evitar sobreajuste (*overfitting*) dos dados, que degradam a generalização, fazendo com que o aprendizado refletisse de maneira mais precisa os dados reais.

Tabela 4 – Perda *GAN* por época.

Época	Perda Discriminador (*d_loss*)	Perda Gerador (*g_loss*)	Precisão Real (*Acc_real*)	Precisão Falsa (*Acc_fake*)	Precisão Total (*Acc_total*)
0	9,54	-0,48	81,25%	59,38%	70,31%
50	8,17	-0,11	95,31%	98,44%	96,88%
100	7,52	-0,14	98,44%	100%	99,22%
150	7,32	-0,17	96,88%	100%	98,44%
200	5,72	-0,16	93,75%	95,31%	94,53%
250	3,57	-0,22	92,19%	96,88%	94,53%
300	3,88	-0,21	89,06%	96,88%	92,97%
350	3,27	-0,25	78,12%	93,75%	85,94%
400	2,12	-0,24	85,94%	98,44%	92,19%
450	1,79	-0,25	84,38%	89,06%	86,72%

Fonte: Elaborado pelos autores.

4.5. Montagem e Treinamento do Modelo de ML

O modelo foi desenvolvido na linguagem de programação *Python* utilizando o classificador *Random Forest*, que opera através de um conjunto de múltiplas árvores de decisão, trabalhando isocronicamente para previsões mais desenvolvidas e assertivas.

A implementação iniciou-se através da leitura de dados de treinamento provenientes de amostras reais que coletamos do sistema de evento do *firewall* posteriormente normalizadas. Segundo Kharismadhany (2022), esse aspecto torna essencial a utilização de técnicas de aprendizado para aumentar a eficiência da defesa do sistema, especialmente em relação a padrões de ataques de *brute force* que frequentemente passam despercebidos.

Após a fase de treinamento, o modelo foi testado em um conjunto de dados separados, foi possível avaliar a sua precisão e outras métricas, como *recall e F1-Score*, com a finalidade de determinar o desempenho do classificador após uso do insumo provido pelo algoritmo GAN.

Os resultados demonstram uma acurácia de 98,5% (Figura 13), indicando que o modelo conseguiu identificar corretamente as tentativas de acesso legítimas e os ataques de *brute force* na maioria das vezes. O *recall* de 98,0% sugere que o modelo foi eficaz em capturar a maioria das tentativas de ataque. O *F1-Score*, que é uma média harmônica entre a precisão e o *recall*, ficou em 98,2%, refletindo um bom equilíbrio entre a taxa de verdadeiros positivos e falsos positivos. Os falsos positivos foram apenas 27 e os falsos negativos totalizaram 89, mostrando que há um pequeno número de tentativas legítimas sendo identificadas erroneamente como ataques, conforme demonstrado na Tabela 5.

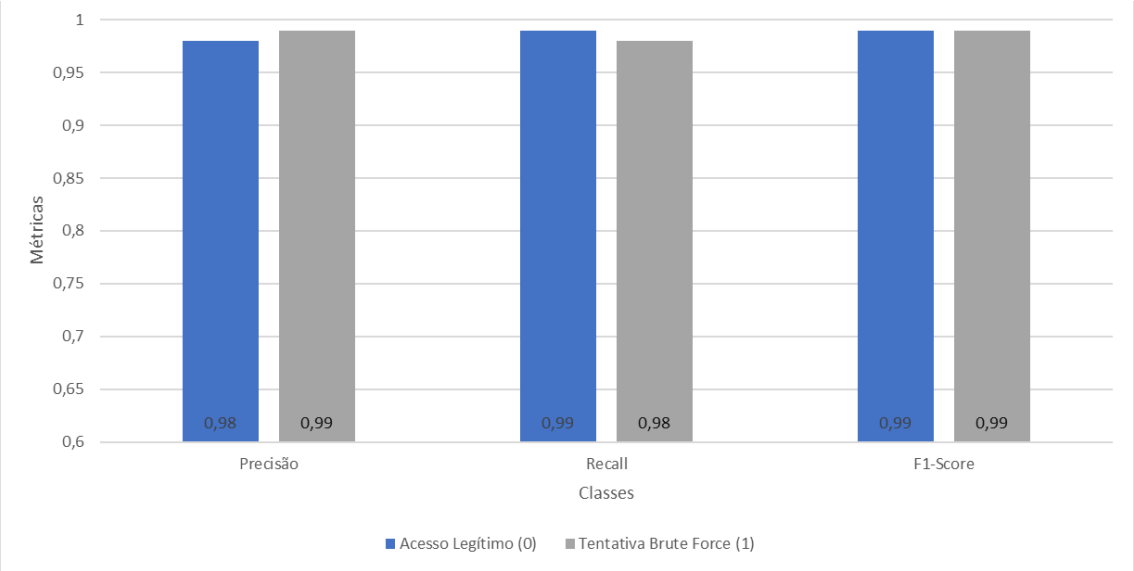
Tabela 5 – Métricas *ML*

Métrica	**Valor**
Precisão	98.5%
Recall	98.0%
F1-Score	98.2%
Acurácia	98.5%
Falsos Positivos (FP)	27
Falsos Negativos (FN)	89

Fonte: Elaborado pelos autores.

1. **Falso Positivo (FP):** O modelo identifica incorretamente uma tentativa legítima como ataque.
2. **Falso Negativo (FN):** O modelo identifica incorretamente uma tentativa de ataque como legítima.

Figura 13 – Métricas de Avaliação da ML por Classe



Fonte: Elaborado pelos autores.

4.6. Integração entre a *GAN* e a *Machine Learning*

Após o treinamento da *GAN*, os dados sintéticos gerados são introduzidos ao *dataset* da *ML* em treinamento. Essa integração expõe o modelo a uma variedade ampliada de dados, incluindo padrões que podem não estar presentes nos dados reais, portanto melhora seu treinamento na identificação de ataques de *brute force*, ajudando a identificar ataques que antes não eram identificados no *firewall* do ambiente controlado, o modelo fortalece a defesa da rede.

Os dados gerados são uteis posteriormente para criação de *blacklists* no *Fortigate* sendo agregado até mesmo a um algoritmo para que seja efetuada a inserção automática baseada na saída, permitindo que o *firewall* reaja proativamente a potenciais ameaças.

5. Conclusão

O projeto demonstrou um desempenho notável na identificação de padrões anômalos associados a ataques de *brute force*. A integração da GAN para geração de dados sintéticos realistas com a *ML* resultou em uma precisão considerável em relação a identificação de *brute force*, aumentando a eficácia do modelo de ML ao enriquecer o conjunto de dados utilizados como treinamento pelo algoritmo. A implementação ampliou a capacidade do modelo em identificar padrões de ataque de *brute force*, que anteriormente poderiam passar despercebido pelos sistemas de *firewall* tradicionais. A precisão alcançada de 98,5% (Figura 13) reforça o potencial da metodologia para fortalecer a segurança de sistemas críticos.

A combinação das técnicas apresentou-se como uma abordagem inovadora para ampliar a segurança de redes de sistemas de informação, possibilitando automatizar a construção de *blacklists*, atribuindo respostas proativas automaticamente a potenciais ameaças com base nas saídas do modelo treinado, tornando os sistemas de defesa mais robustos e precisos. À medida que o cenário de segurança evoluiu, a capacidade de gerar e analisar dados sintéticos realistas será uma ferramenta valiosa para fortalecer as defesas das organizações contra ameaças emergentes.

Por fim, o diferencial do projeto está no uso de dados sintéticos para expansão de modelos baseados em aprendizado de máquina sobre possíveis variações de ataques, um fator que não só aumenta a eficiência da defesa, mas também expande a aplicabilidade da solução a variados ambientes de segurança. Este estudo é um avanço promissor em direção ao desenvolvimento de sistemas autônomos e resilientes, com capacidade para prever e responder a ataques de maneira dinâmica e eficiente.

Embora a abordagem tenha demonstrado benefícios claros, é essencial reconhecer os presentes desafios, como ataques de injeção de dados ou envenenamento de modelos. Pesquisas futuras devem focar no aprimoramento de método de defesas robustos, assegurando que os dados gerados e utilizados mantenham sua integridade e eficiência. Com isso, a aplicação de *GANs* na segurança pode ser ampliada, permitindo que organizações fortaleçam ainda mais seus sistemas para proteção contra ameaças cada vez mais sofisticadas.

Referencial Bibliográfico

ADIBAN, M.; SINISCALCHI, M. S.; SALVI, G. 2023. *A step-by-step training method for multi generator GANs with application to anomaly detection and cybersecurity*. Disponível em: https://www.sciencedirect.com/science/article/pii/S0925231223003065.

ALMEIDA. C.C. 2019. Identificação e classificação de imagens usando rede neural convolucional e "*machine learning*" [recurso eletrônico]: implementação em sistema embarcado. Disponivel em: https://repositorio.unicamp.br/acervo/detalhe/1126679.

ALQAHTANI, H.; KAVAKLI, M.; AHUJA, G. 2019. *Applications of Generative Adversarial Networks (GANs): An Updated Review*. Disponível em: https://www.researchgate.net/publication/338050169_Applications_of_Generative_Adversarial_Networks_GANs_An_Updated_Review.

ARORA, A SHANTANU. 2020. *A Review on Application of GANs in Cybersecurity Domain*. Disponível em: https://www.tandfonline.com/doi/full/10.1080/02564602.2020.1854058?scroll=top&needAccess=true.

AWS. 2023. O que é uma *GAN?* Disponível em: https://aws.amazon.com/pt/what-is/gan/

BROWN, A.; TUOR, A; HUTCHINSON, B.; NICHOLS, N. 2018. *Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection.* Disponível em: https://dl.acm.org/doi/abs/10.1145/3217871.3217872.

CISCO. 2017. *DHCP Overview*. Disponível em: https://www.cisco.com/c/en/us/td/docs/routers/ncs4200/configuration/guide/IP/17-1-1/b-dhcp-17-1-1-ncs4200/b-dhcp-17-1-1-ncs4200_chapter_00.pdf.

CISCO. 2024. *What is SD-WAN?* Cisco. Disponível em: https://www.cisco.com/c/en/us/solutions/enterprise-networks/sd-wan/what-is-sd-wan.html.

DATA SCIENCE ACADEMY. 2022. *Deep Learning Book*. Disponível em: https://www.deeplearningbook.com.br/.

FONTANA, G & CARNEIRO, F. R.P. 2023. Estudo do impacto da variação de parâmetros em uma rede neural artificial aplicado a bases com diferentes características. Disponivel em: https://ri.unipac.br/repositorio/wp-content/uploads/tainacan-items/282/224732/GUILHERME-FONTANA-KILSON-ESTUDO-DO-IMPACTO-DA-VARIACAO-DE-PARAMETROS-EM-UMA-REDE-COMPUTACAO-2023.pdf.

GALEGALE, N.; FONTES, E.; GALEGALE, B. 2017. Uma contribuição para a Segurança da Informação: um estudo de casos múltiplos com organizações brasileiras. Disponível em: https://www.scielo.br/j/pci/a/Srp97XX3Hyb4MfjxRH9gDgd/#.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. 2017. *Deep Learning*. Disponível em: https://link.springer.com/article/10.1007/s10710-017-9314-z.

GOODFELLOW, I. J.; POUGET-ABADIE. J.; MIRZA, M.; XU, B.; WARDE-FARLEY D.; OZAIR, S.; COURVILLE, A; BENGIO, Y. 2014. *Generative Adversarial Nets*. Disponível em: https://arxiv.org/abs/1406.2661.

HAYKIN, S. 1998. *Neural Networks: A Comprehensive Foundation.* Disponível em: https://dl.acm.org/doi/abs/10.5555/521706.

HOSAKI, G. Y. & RIBEIRO, D. F. 2021. *Deep learning*: ensinando a aprender. Disponivel em: https://ric.cps.sp.gov.br/handle/123456789/5060.

JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. 2021. *Machine learning and deep learning*. Disponível em: https://link.springer.com/article/10.1007/s12525-021-00475-2.

KASPERSKY, 2022. Relatório da *Kaspersky* revela como ocorreram os ataques de *ransomware* em 2022. Disponivel em: https://www.kaspersky.com.br/about/press-releases/43-dos-ataques-de-ransomware-em-2022-comecaram-com-a-exploracao-de-aplicativos.

KHARISMADHANY, E.; RUSWIANSARI, M.; HARSONO, T. 2023. *Brute-force Detection Using Ensemble Classification* Disponível em: https://www.researchgate.net/publication/382373555_Brute-force_Detection_Using_Ensemble_Classification.

KOCH, B.; DENTON, E.; HANNA, A.; FOSTER, J. G. 2021. *Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research*. Disponível em: https://par.nsf.gov/biblio/10324721-reduced-reused-recycled-life-dataset-machine-learning-research.

LECUN, Y.; BENGIO, Y.; HINTON, G. 2015. *Deep learning.* Disponível em: https://www.nature.com/articles/nature14539.

LIPPMANN, RICHARD.; HAINES, J. W.; FRIED. J. D.; KORBA, J.; DAS. K. 2000. *The 1999 DARPA off-line intrusion detection evaluation.* Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1389128600001390.

LUCIC. M.; KURACH. K.; MICHALSKI. M.; GELLY. S.; BOUSQUET. O. 2017. *Are GANs Created Equal? A Large-Scale Study*. Disponível em: https://arxiv.org/abs/1711.10337.

MARTINIANO. A.; FERREIRA. R. P.; FERREIRA. A.; FERREIRA. A.; SASSI. R. J. 2016. Utilizando uma rede neural artificial para aproximação da função de evolução do sistema de Lorentz. Disponível em: https://www.researchgate.net/figure/Figura-1-Representacao-do-neuronio-artificial_fig1_329245206/actions#caption.

MIRZA, M. & OSINDERO, S. 2014. *Conditional Generative Adversarial Nets*. Disponível em: https://arxiv.org/abs/1411.1784.

MONARD, M. C & BARANAUSKAS, J. A. 2003. Conceitos sobre Aprendizado de Máquina. Disponível em: https://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf.

NAJAFABADI. M. M.; KHOSHGOFTAAR. T. M.; KEMP. C.; SELIYA. N.; ZUECH. R. 2014. *Machine Learning for Detecting Brute Force Attacks at the Network Level.* Disponível em: https://ieeexplore.ieee.org/document/7033609.

NOOR. S.; BAZAI. S. U.; GHAFOOR. M. I.; MARJAN. S.; AKRAM. S.; ALI. F. 2023. *Generative Adversarial Networks for Anomaly Detection: A Systematic Literature Review*. Disponível em: https://www.researchgate.net/publication/370177350_Generative_Adversarial_Networks_for_Anomaly_Detection_A_Systematic_Literature_Review.

PEREIRA. H. A.; DE SOUZA. A. F.; DE MENEZES. C. S. 2018. *Obtaining evidence of learning in digital games through a deep learning neural network to classify facial expressions of the players. IEEE Transactions on Computational Intelligence and AI in Games, 2019.* Disponível em: https://ieeexplore.ieee.org/abstract/document/8659216

POWERS. D. M. W. 2020. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.* Disponível em: https://arxiv.org/abs/2010.16061.

SALAMA, S.; ALAMOUDI, Y.; ALAMOUDI, G.; ALBESHRI, F. 2023. *Cloud Computing Security Issues and Countermeasure: A Comprehensive Survey.* Disponível em: https://www.ijcaonline.org/archives/volume185/number14/32767-2023922832/.

SCARFONE, K. & MELL, P. 2007. *Guide to Intrusion Detection and Prevention Systems (IDPS)* Disponível em: https://csrc.nist.gov/pubs/sp/800/94/final.

SIEMURI, A.; SELVAN, K.; KUUSNIEMI, H.; VALISUO, P.; ELMUSRATI, M. S. 2022. *A Systematic Review of Machine Learning Techniques for GNSS Use Cases.* Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9937069.

SOMMER. R & PAXSON, V. 2010. *Outside the Closed World:On Using Machine Learning for Network Intrusion Detection*. Disponível em: https://ieeexplore.ieee.org/document/5504793.

WANG, Z.; SHE, Q.; WARD, T. E. 2020. *Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy*. Disponível em: https://arxiv.org/abs/1906.01529.