Aprimorando a Detecção de APTs com Geração de Dados Sintéticos Baseada em GAN-Transformers

Alfredo Cossetin Neto[1], Rafel C. Pregardier[1], Carlos R. P. dos Santos[1], Vinicius Fulber-Garcia[2], Luis A. L. Silva[1]

[1]Centro de Tecnologia – Universidade Federal De Santa Maria – UFSM
Av. Roraima, 1000, CEP 97.105-900 – Santa Maria, RS – Brasil

[2]Departamento de Informática – Universidade Federal Do Paraná – UFPR
Jardim das Américas, CEP 81530-900 – Curitiba, PR – Brasil

{acneto,rcpregardier,csantos,luisalvaro}@inf.ufsm.br,
vinicius@inf.ufpr.br

Abstract. *This work investigates the generation of synthetic data for Advanced Persistent Threats (APTs) using Generative Adversarial Networks (GANs) adapted to the domain of time series. Given the stealthy and sequential nature of APTs, traditional data generation methods that ignore temporal dynamics are insufficient. To address this limitation, this study explores the Transformer Time-Series Conditional GAN (TTS-CGAN) architecture, originally proposed for biosignals, and proposes specific adaptations for the generation of malicious network traffic flows. The process includes data modeling from the DAPT2020 dataset, architectural adjustments to enhance capacity and diversity, and validation of the synthetic data through qualitative, quantitative metrics and the performance evaluation of machine learning models trained on real, synthetic, and semi-synthetic datasets. Results indicate that the synthetic data generated by the TTS-CGAN can improve APT detection performance, demonstrating the viability and benefits of the proposed approach.*

Resumo. *Este trabalho investiga a geração de dados sintéticos de Advanced Persistent Threats (APTs) utilizando Redes Generativas Adversariais (GANs) adaptadas para o domínio de séries temporais. Considerando a natureza furtiva e sequencial das APTs, abordagens tradicionais de geração de dados que ignoram as dinâmicas temporais tornam-se insuficientes. Para superar essa limitação, este estudo explora a arquitetura Transformer Time-Series Conditional GAN (TTS-CGAN), originalmente proposta para biosinais, e propõe adaptações específicas para a geração de fluxos de rede maliciosos. O processo inclui a modelagem de dados do dataset DAPT2020, ajustes arquiteturais para aumento da capacidade e diversidade, além da validação dos dados sintéticos por meio de métricas qualitativas, quantitativas e do desempenho de modelos de Machine Learning (ML) treinados em conjuntos reais, sintéticos e semi-sintéticos. Os resultados indicam que o uso de dados sintéticos gerados pela TTS-CGAN pode aprimorar a detecção de APTs, demonstrando a viabilidade e os benefícios da abordagem proposta.*

1. Introdução

Com o aumento da digitalização de serviços e a crescente dependência de sistemas online, a cibersegurança tornou-se uma das áreas mais críticas da atualidade. Em resposta a essa tendência, cibercriminosos têm desenvolvido ataques cada vez mais sofisticados e difíceis de detectar. Entre as ameaças modernas, destacam-se especialmente as *Advanced Persistent Threats* (APTs) [Alshamrani et al. 2019]. Diferentemente dos ataques convencionais, as APTs são realizadas por agentes altamente qualificados e com amplos recursos financeiros, utilizando ferramentas personalizadas e estratégias planejadas para se infiltrar em sistemas sensíveis [Alshamrani et al. 2019].

APTs seguem uma estrutura em múltiplos estágios e são frequentemente considerados uma evolução dos ataques multietapa tradicionais [Alshamrani et al. 2019, Ghafir et al. 2018]. A principal característica das APTs é a sua natureza furtiva: são silenciosos e prolongados, capazes de permanecer ocultas por longo período de tempo, avançando discretamente até alcançar seus objetivos [Xiong et al. 2020]. Na literatura, técnicas de *Machine Learning (ML)* têm sido aplicadas na detecção de APTs, incluindo *Support Vector Machines* (SVMs) e *Random Forests* (RFs) [Xin et al. 2018]. Devido à natureza sequencial desses ataques, abordagens que analisam séries temporais e a correlação entre eventos maliciosos são especialmente relevantes [Myneni et al. 2020]. Por exemplo, arquiteturas baseadas em redes neurais profundas para dados sequenciais, como as *Long Short-Term Memory* (LSTM) [Géron 2022], têm sido frequentemente empregadas na análise de ataques multietapa [Liao et al. 2024, Zhou et al. 2021]. Contudo, a eficácia desses modelos depende da qualidade e representatividade dos dados utilizados no treinamento, sendo escassas as fontes que contêm exemplos realistas de APTs [Myneni et al. 2023].

A escassez e complexidade desses dados têm levado a comunidade de cibersegurança a buscar alternativas para sua geração. Trabalhos recentes destacam que *Generative Adversarial Networks* (GANs) [Chakraborty et al. 2024] são promissoras na criação de dados sintéticos realistas para o treinamento de modelos de detecção de ciberataques [Li et al. 2018, Alo et al. 2024]. As GANs têm encontrado aplicação em várias áreas da cibersegurança, incluindo a geração de tráfego de rede [Bianchi et al. 2025], criação de ataques sintéticos, balanceamento de conjuntos de dados e até mesmo diretamente na detecção de ataques [Navidan et al. 2021]. Particularmente na geração de tráfego malicioso, as GANs podem produzir diversos formatos de dados, desde pacotes individuais até fluxos completos de rede. Apesar dessas aplicações serem promissoras, a utilização de GANs para gerar dados sintéticos representativos de APTs permanece uma área aberta, sem soluções amplamente consolidadas na literatura.

O objetivo deste trabalho é investigar a aplicação das GANs para gerar dados sintéticos representativos de APTs, visando aprimorar o desempenho dos modelos de ML usados na detecção dessas ameaças. Para isso, considerando as características sequenciais das APTs, a pesquisa explora a arquitetura *Transformer Time-Series Conditional GAN* (TTS-CGAN) [Li et al. 2022]. Originalmente proposto na área da saúde, o modelo TTS-CGAN emprega mecanismos de auto-atenção baseados em *Transformers* para capturar dependências temporais complexas, possibilitando a geração de dados sintéticos que refletem os padrões sequenciais presentes em séries temporais reais. Neste trabalho, este modelo é explorado na captura de correlações entre diferentes fluxos de rede e a dinâmica

temporal típica desses ataques.

Entre as principais contribuições deste trabalho, destacam-se:

1. A adaptação e aprimoramento da arquitetura TTS-CGAN, originalmente proposta para a área da saúde, ao domínio da cibersegurança. Foram adicionados camadas de normalização de batch e *MiniBatch* para aumentar a diversidade dos *batches* gerados e mitigar a possibilidade de *mode collapse*, além de ajustes de hiperparâmetros, considerando a maior variabilidade e complexidade dos dados de cibersegurança;
2. A validação qualitativa e quantitativa da geração de dados sintéticos com a arquitetura, utilizando PCA, t-SNE e *Dynamic Time Warping* (DTW) para evidenciar a proximidade estatística entre os dados sintéticos e reais;
3. A avaliação do impacto dos dados sintéticos na detecção de APTs, por meio de modelos de *Machine Learning — Random Forest*, SVM, LSTM e *Transformer Encoder* —, demonstrando que todos os modelos treinados com dados semi-sintéticos obtiveram melhor desempenho.

2. Fundamentação Teórica

Atualmente, há poucas fontes de dados disponíveis para o estudo de APTs [Navarro et al. 2018]. Um dos *datasets* mais utilizados em pesquisas sobre APTs é o DARPA 2000 [Lippmann et al. 2000], que inclui dois cenários desse tipo: (i) LLDOS 1.0, um ataque simples e de curta duração; e (ii) LLDOS 2.0.2, um ataque mais longo, furtivo e sofisticado. Apesar de amplamente explorado, esse *dataset* é bastante antigo e não reflete mais as características dos APTs modernos, visto que os cenários de ataque foram construídos em um ambiente simulado, com ferramentas e técnicas da época, como escaneamento com *Nmap* e ataques de negação de serviço do tipo *mstream*, o que limita sua representatividade frente aos vetores de ataque atuais.

Alguns *datasets* populares, como CICIDS2018 [Sharafaldin et al.], ISCX [Shiravi et al. 2012] e UNSW-NB15 [Moustafa e Slay 2015], são empregados em estudos sobre ataques multietapa e APTs. Contudo, a diferença entre tráfego benigno e tráfego de ataque nesses *datasets* é trivial [Liao et al. 2024, Myneni et al. 2020], uma vez que modelos de *Machine Learning* básicos podem alcançar alta precisão em tarefas de classificação simplesmente analisando amostras individuais desses *datasets*, o que não reflete a natureza sequencial e furtiva de uma APT.

Datasets modernos voltados especificamente para APTs são o DAPT2020 [Myneni et al. 2020] e o Unraveled [Myneni et al. 2023]. Como nosso objetivo é gerar dados sintéticos, o Unraveled, que é um *dataset* semi-sintético, torna-se inadequado para o nosso estudo. Dessa forma, neste trabalho, optou-se por utilizar o DAPT2020 como fonte de dados para investigar o emprego da TTS-CGAN na geração de dados sintéticos para APTs.

O DAPT2020 segue a estrutura descrita por [Alshamrani et al. 2019], contemplando quatro estágios de ataque: *Reconnaissance*, *Foothold Establishment*, *Lateral Movement* e *Data Exfiltration*, além de tráfego benigno, que representa atividade não maliciosa. Esses estágios ocorrem na ordem especificada, simulando múltiplos ataques intercalados. Este *dataset* foi utilizado no treinamento e execução de modelos

de classificação de APTs, explorando combinações dos dados originais com os dados sintéticos gerados pelos modelos de *Machine Learning* investigados.

2.1. *Machine Learning* na Detecção de APTs

Técnicas de *Machine Learning* têm sido utilizadas para detectar diferentes APTs. Em [Ghafir et al. 2018], os autores desenvolveram uma framework que utiliza detectores clássicos de eventos suspeitos na rede, os quais eram posteriormente agrupados em diferentes *clusters*, representando possíveis instâncias de APTs. Cada um desses *clusters* serve como entrada para modelos de *Machine Learning*, como *Support Vector Machines* (SVM) e *Random Forests* (RF), que estimam a probabilidade de alertas corresponderem a uma APT completa. No trabalho [Ghafir et al. 2019], os mesmos autores mantiveram o sistema de clusterização, mas introduziram o uso de *Hidden Markov Models* (HMMs) para organizar os alertas em uma possível sequência de estágios de um ataque. Com base nessa sequência, eles aplicaram um algoritmo desenvolvido especificamente para prever o próximo estágio da APT, alcançando uma acurácia de predição de 93,91%.

Em [Liao et al. 2024], os autores propuseram uma *Graph Neural Network* (GNN) com mecanismos de atenção baseados em *transformers*, denominada *Graph Attention Network* (GAT). Essa arquitetura recebe como entrada a topologia da rede, reconstruída a partir de endereços IP e portas, e realiza a correlação entre os nós do grafo (representando combinações de IPs e portas). Paralelamente, o tráfego de rede é processado por uma rede LSTM, com o objetivo de capturar os padrões temporais presentes na comunicação. As saídas dessas duas arquiteturas são, então, concatenadas e passadas por uma camada linear de neurônios (sem função de ativação) para reduzir a dimensão. Por fim, o resultado é inserido em um algoritmo desenvolvido responsável por classificar o estágio do ataque APT. O artigo utiliza o mesmo conjunto de dados adotado no presente trabalho, o DAPT2020, no qual o modelo proposto é comparado com dois algoritmos clássicos — *Support Vector Machines* e Regressão Logística — além de diversos modelos de *Deep Learning*, como LSTM, MLP, CNN e outras variações.

2.2. GANs e Dados Temporais Sintéticos

A geração de dados sequenciais por meio da integração de GANs com LSTMs é um tema de pesquisa recente. Por exemplo, em [Zhu et al. 2019b], é proposto um modelo LSTM-GAN para a detecção de anomalias em séries temporais, que é aplicado em dados de ECG e tráfego de táxis de Nova York. De maneira semelhante, o trabalho em [Harada et al. 2019] emprega modelos de GANs recorrentes baseados em LSTM para gerar biossinais multiclasse, permitindo o controle das características dos dados sintéticos por meio da análise de variáveis latentes.

A proposta em [Hazra e Byun 2020] consiste no modelo SynSigGAN, que combina LSTMs no gerador com redes convolucionais (CNNs) no discriminador para produzir sinais biomédicos sintéticos a partir de um conjunto limitado de dados reais. Seguindo essa abordagem, os autores em [Zhu et al. 2019a] propõem o modelo BiLSTM-CNN GAN, projetado para gerar dados sintéticos de eletrocardiogramas com alta fidelidade, demonstrando a eficácia da combinação entre LSTMs bidirecionais e CNNs na síntese de sinais fisiológicos.

Em [Vaswani et al. 2017], foi proposto o mecanismo de *self-attention*, introduzindo o modelo *transformer*, capaz de capturar relações complexas em sequências

com desempenho superior às arquiteturas anteriores baseadas em LSTM. A TransGAN [Jiang et al. 2021] foi a primeira rede generativa adversarial construída inteiramente com *transformers* para a geração de imagens sintéticas, utilizando o *Vision Transformer (ViT)* [Dosovitskiy et al. 2020] no discriminador. O *ViT* tem um propósito semelhante ao de camadas convolucionais, buscando extrair características relevantes das imagens, porém apresenta, em geral, maior capacidade para capturar padrões globais.

A TTS-CGAN [Li et al. 2022] também é uma arquitetura de GAN baseada exclusivamente em transformers, mas voltada à geração de dados temporais, originalmente projetada para tratar problemas de geração de dados sintéticos na área da saúde. A estrutura da TTS-GAN faz uso de *Vision Transformers* (ViT), mas trata séries temporais como imagens com altura de um único pixel. Tanto o gerador quanto o discriminador utilizam camadas lineares e de convolução (sem função de ativação) para alinhar os dados com a entrada de um único Transformer *multi-head*. Baseado neste modelo, a TTS-CGAN [Li et al. 2022] foi proposta como uma evolução da TTS-GAN, com a capacidade adicional de lidar com dados rotulados. O trabalho explorou técnicas para a incorporação de rótulos nesta arquitetura. Como resultado, uma modificação no discriminador foi adotada, passando a contar com duas cabeças de classificação: uma responsável por distinguir dados reais de sintéticos e outra para a predição das classes dos dados.

2.3. Trabalhos Relacionados

Em [Li et al. 2019, Li et al. 2018], uma GAN foi treinada com dados temporais de um sistema de tratamento de água para detectar ataques por meio da identificação de anomalias. Após o treinamento, o próprio discriminador foi reutilizado para classificar sequências temporais: se uma sequência fosse considerada falsa, ela era interpretada como uma possível anomalia e, portanto, um potencial ataque ao sistema.

Com o mesmo intuito de utilizar o discriminador na detecção de ataques em tráfego de rede, os autores em [Zeeshan e Maasooma 2024] empregaram uma arquitetura de GAN baseada em *transformers* e redes neurais *feed-forward* para detectar ataques em tráfego de rede. O trabalho foi realizado nos *datasets* UNSW-NB15, NSL-KDD e CIC-IDS 2017, modelando o tráfego no formato de fluxos e considerando os rótulos dos ataques e não utilizando o discriminador treinado com outros métodos de detecção de anomalias.

O GANsformer [Hudson e Zitnick 2021] foi uma das primeiras implementações de *transformers* em um contexto de aprendizado adversarial, utilizando atenção bipartida em uma arquitetura híbrida entre *transformers* e redes convolucionais (CNNs) para a geração de imagens. Posteriormente, essa arquitetura foi empregada em [Alzahem et al. 2022] com o objetivo de balancear o *dataset* de malware CLaMP [Kumar et al. 2019], gerando dados sintéticos para aprimorar a performance de modelos de detecção. Como o GANsformer foi originalmente projetado para imagens, os autores converteram os cabeçalhos binários dos malwares em representações bidimensionais, possibilitando seu uso na GAN. Para validar os dados gerados, foram utilizados diferentes modelos baseados em CNNs, e todos apresentaram melhorias na detecção dos malwares ao serem treinados com o *dataset* balanceado. No entanto, o estudo não explora outros métodos clássicos de *Machine Learning*, como *Random Forests*, *Support Vector Machines*

ou quaisquer outros, que, inclusive, são abordados no próprio *dataset* utilizado em [Kumar et al. 2019].

O trabalho em [Alo et al. 2024] investiga a baixa representatividade de certos tipos de ataques nos *datasets* atuais, como APTs e *zero-day*. O estudo propõe o uso de GANs para gerar dados sintéticos e aumentar essa representatividade a fim de aprimorar o desempenho de modelos de *Machine Learning*. A proposta foi validada nos conjuntos de dados UNSW-NB15 e CICIDS2017, utilizando uma arquitetura original de GAN simples, baseada em camadas convolucionais. Embora o estudo não detalhe como os dados sintéticos foram integrados aos dados reais, os dados semi-sintéticos foram utilizados para treinar um modelo de *Deep Learning* composto por camadas convolucionais e recorrentes. Esse modelo foi então comparado com algoritmos clássicos —*Random Forests*, *Suport Vector Machines*, *Naive Bayes* (NB) e *K-Nearest Neighbors* (KNN). O modelo de *Deep Learning* obteve a maior acurácia entre os avaliados, alcançando 95,8%. O artigo, contudo, não testou o desempenho do modelo de *Deep Learning* treinado com os dados originais e nem dos outros modelos com os dados sintéticos.

Observa-se que os trabalhos analisados utilizam conjuntos de dados de cibersegurança com representações simplificadas de APTs, como discutido na Seção **??**. Além disso, as comparações entre modelos treinados com e sem dados sintéticos são, em geral, insuficientes e raramente são avaliadas de forma sistemática em diferentes algoritmos, o que dificulta a mensuração real do impacto dos dados gerados. Em contraste, este trabalho utiliza um *dataset* mais sofisticado e realista, e realiza uma análise comparativa completa, evidenciando de forma quantitativa o efeito da geração de tráfego sintético de APTs no desempenho de múltiplos modelos de *Machine Learning*.

3. Geração de Dados Temporais Sintéticos de APTs Usando a Arquitetura TTS-CGAN

Esta seção descreve a metodologia adotada para a geração de dados sintéticos de APTs baseada na arquitetura TTS-CGAN.

3.1. Pré-processamento e Modelagem de Dados

O DAPT2020 contém aproximadamente 86 mil fluxos de rede coletados ao longo de cinco dias. Cada fluxo representa uma série de pacotes trocados entre cliente e servidor, sendo descrito por 81 colunas, cada uma correspondente a uma característica distinta do tráfego. Dessa forma, é necessário realizar um processamento sobre os dados originais para filtrar as informações mais relevantes dos fluxos de rede e realinhar os dados para o formato de entrada da GAN.

Inicialmente, como o foco deste trabalho é a análise de dados temporais, os endereços *IP* foram removidos do conjunto pois não é um atributo que depende do tempo. A única coluna categórica presente era o rótulo do tráfego. Essa coluna foi padronizada para letras minúsculas e, em seguida, codificada numericamente, atribuindo-se um valor inteiro único para cada tipo de tráfego. Por exemplo, todos os fluxos com rótulo *"benign"* receberam o valor "0", os com rótulo *"exfiltration"* receberam o valor "1", *"establish foothold"* receberam o valor "2", *"lateral movement"* receberam o valor "3"e por fim *"reconnaissance"* receberam o valor "4". Esta conversão é relevante para permitir interpretar cada classe como estágios 1, 2, etc.

Com o objetivo de identificar os atributos mais relevantes para a caracterização do tipo de tráfego (benigno ou ataque), foi calculada a correlação de Pearson entre cada atributo e a coluna de rótulo. Os modelos construídos neste trabalho foram baseados nos atributos que apresentaram correlação positiva com o rótulo, conforme ilustrado na Figura 1. Apenas os atributos com correlação positiva forma considerados neste trabalho para permitir a GAN trabalhar com um número menor de variáveis com influência direta sobre o estágio do ataque. Os demais atributos contidos no conjunto de dados foram removidos.

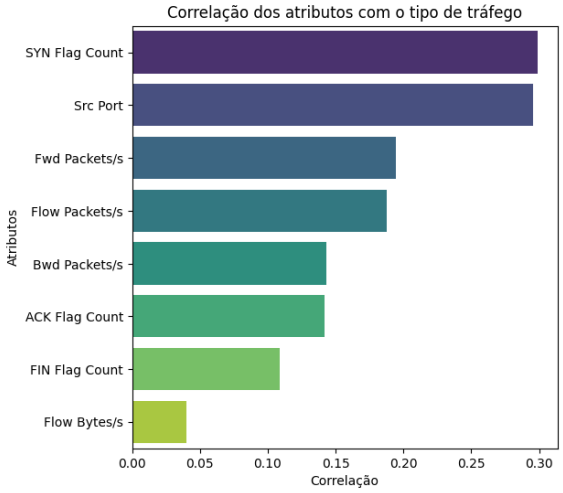


Figura 1. Atributos do DAPT2020 com maior correlação com o tipo de tráfego.

Considerando que o *dataset* representa uma longa sequência de fluxos de pacotes, foi necessário segmentá-lo em subsequências utilizáveis durante o treinamento. Para isso, calcularam-se os comprimentos de todas as sequências contínuas de amostras pertencentes à mesma classe. Após desconsiderar sequências muito curtas (com menos de 10 amostras), observou-se que a maioria continha pouco mais de 30 amostras. Com base nessa análise, definiu-se um comprimento fixo de 30 fluxos por subsequência. O *dataset* resultante foi, então, dividido em pequenas sequências, que serviram como entrada para a GAN, conforme ilustrado na Figura 2, totalizando, ao final, 2889 sequências, cada uma com 8 canais (colunas).

3.2. Arquitetura Adaptada da TTS-CGAN para Dados de APTs

A TTS-CGAN, proposta originalmente em [Li et al. 2022], concatena os rótulos com um vetor de números aleatórios amostrados de uma distribuição Gaussiana, o que é denominado espaço latente em arquiteturas de GANs, e utiliza camadas lineares e convolucionais para alinhar o resultado com a entrada da camada de *transformers*, composta por três blocos *multi-head* em sequência. No artigo original, houve uma forte preocupação com o *model collapse*, resultando em um discriminador significativamente mais complexo do que o gerador e em um alto *dropout* de 0,5.

Em contraste, este trabalho adapta essa arquitetura para a geração de ataques no formato de fluxos de rede, um domínio com muito maior variabilidade entre as amostras. Devido a essa diferença, os hiperparâmetros da arquitetura foram ajustados com o objetivo de aumentar a capacidade da GAN em compreender e gerar dados com alta variabilidade. Para isso, foram testadas diferentes configurações para aumentar a complexidade da

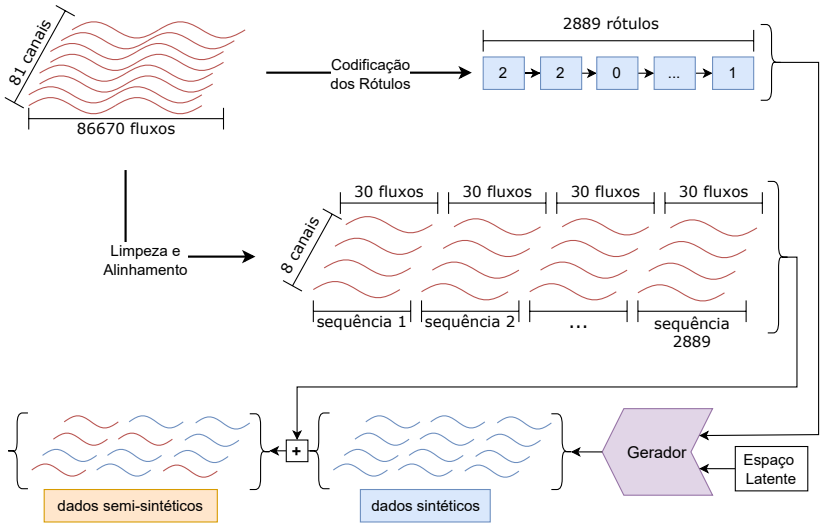
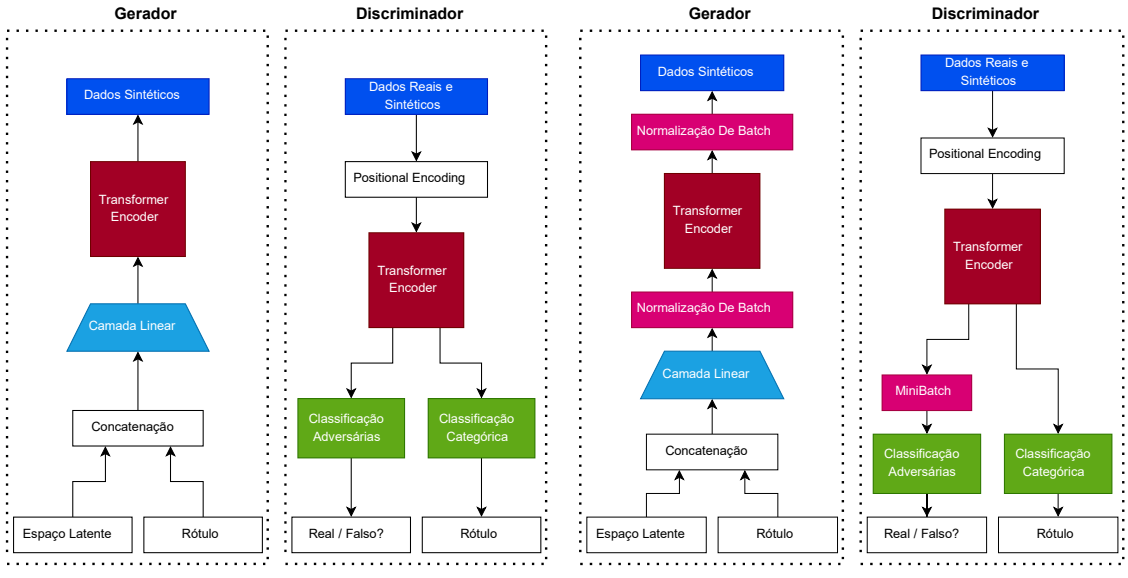


Figura 2. Processamento do *dataset* e geração dos dados sintéticos e semi-sintéticos.

arquitetura de modo a manter a estabilidade do treinamento. Dessa forma, a dimensão de entrada (*embedding*) do gerador foi aumentada de 10 para 32, e a do discriminador, de 50 para 96. A dimensão dos rótulos (*label embedding*) foi expandida de 10 para 32, e o número de cabeças de atenção do gerador foi ampliado para o mesmo número do discriminador: de 5 para 8. Por fim, foi adicionada uma nova camada de *transformer* no discriminador, indo de 3 para 4.



(a) TTS-CGAN original **(b) TTS-CGAN explorada neste trabalho**

Figura 3. Arquiteturas de GANs para tratamento de dados com características temporais.

Com o objetivo de aumentar a diversidade dos dados gerados e mitigar o risco

de *mode collapse*, práticas comuns em ML foram adotadas: adição de duas camadas de normalização em *batch* no gerador, uma antes e outra depois da camada de *multi-head attention* do *transformer*. Além disso, foi inserida uma camada de *MiniBatch Discrimination* na cabeça do discriminador responsável por classificar os dados entre reais ou sintéticos. Essa camada calcula as características estatísticas do lote de entrada, como o desvio padrão, e incorpora essas informações como novos canais nos dados de entrada. O objetivo dessas modificações é permitir que o discriminador identifique lotes com baixa diversidade, facilitando a diferenciação entre dados reais e gerados. Com isso, o gerador é incentivado a produzir amostras mais diversas, reduzindo significativamente a probabilidade de *mode collapse*. Essas alterações podem ser observadas na Figura 3.

Devido às modificações propostas na arquitetura e à alta variabilidade dos dados, o risco de *overfitting* torna-se menos provável. Dessa forma, a taxa de *dropout* no gerador foi reduzida de 0,5 para 0,1, e no discriminador, de 0,5 para 0,0.

O treinamento da arquitetura foi realizado ao longo de 70 épocas, com *batches* de tamanho 30. Para a atualização dos pesos da GAN, foram adotados os mesmos hiperparâmetros descritos em [Li et al. 2022]: o otimizador *Adam*, com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}999$; uma função de perda adversarial baseada na distância de Wasserstein; e o uso de uma média móvel exponencial (EMA) para suavizar as funções de perda ao longo do treinamento. O *dataset* original foi embaralhado e dividido na proporção de 70% para treino e 30% para teste. O treinamento e geração dos dados foi feito em uma GPU NVIDIA RTX 3060, CPU Intel i5-12400F 6 cores e 32 GB RAM, usando PyTorch 2.1.

3.3. Geração dos Dados Sintéticos e Semi-Sintéticos para APTs

Uma das principais vantagens do uso de GANs é a flexibilidade na geração de dados personalizados conforme a necessidade. Com base nisso, foram criados dois conjuntos de dados distintos: um voltado para a avaliação da similaridade com os dados reais e outro com o objetivo de enriquecer o conjunto original, conforme descrito a seguir:

Sintético Puro: Conjunto de dados totalmente sintético, gerado para replicar as amostras utilizadas durante o treinamento da GAN. O gerador recebe como entrada a classe alvo (rótulo) e um vetor amostrado do espaço latente correspondente. Esse conjunto é utilizado principalmente para medir a similaridade entre os dados reais e os dados gerados, já que busca reproduzir o comportamento das amostras originais.

Semi-Sintético: Conjunto baseado nos dados reais utilizados no treinamento, com adição de amostras sintéticas exclusivamente nas classes de ataque. Como o conjunto original é fortemente desbalanceado, a GAN foi empregada para gerar novas instâncias de cada estágio de uma APT, buscando tornar o conjunto mais balanceado. Ao todo, foram geradas 1300 sequências de ataque: 800 de *exfiltration*, 145 de *establish foothold*, 326 de *lateral movement* e 43 de *reconnaissance*.

4. Experimentos e Resultados

Este trabalho avalia a capacidade da TTS-CGAN de gerar ataques APT sintéticos no formato de séries temporais e de melhorar a performance de modelos de *Machine Learning*, respeitando os diferentes rótulos presentes no conjunto de dados DAPT2020. Para alcançar esse propósito, foram definidas as seguintes perguntas a serem respondidas

nos experimentos: a) Qual a similaridade entre os dados sintéticos e os dados originais do DAPT2020? b) As melhorias na arquitetura propostas na Seção 3.2 contribuem para melhorar a qualidade dos dados? e c) Como modelos de *Machine Learning* para a detecção de APTs se comportam ao serem treinados em dados puramente sintéticos e semi-sintéticos?

4.1. Análise Estatística das Distribuições Geradas

As duas primeiras perguntas de pesquisa foram avaliadas por meio da análise das distribuições estatísticas dos dados sintéticos gerados no Grupo 1. Esses dados foram produzidos por duas versões da arquitetura TTS-CGAN, ambas configuradas com os mesmos hiperparâmetros: uma versão original e outra com as modificações propostas neste trabalho.

Para uma análise qualitativa da distribuição dos dados no espaço latente, utilizamos as técnicas *Principal Component Analysis* (PCA) e *t-distributed Stochastic Neighbor Embedding* (t-SNE) [Géron 2022]. O PCA é uma técnica linear de redução de dimensionalidade que preserva a variância dos dados, enquanto o t-SNE é uma técnica não linear que busca preservar as relações locais entre amostras, sendo particularmente eficaz para a visualização de agrupamentos em dados de alta dimensão. Ambas as técnicas foram aplicadas neste trabalho para projetar as séries temporais em duas dimensões, permitindo uma avaliação qualitativa da sobreposição entre os dados reais e os sintéticos.

Figura 4 apresenta os resultados dessas projeções. Em ambas as arquiteturas, os dados sintéticos seguem um padrão de distribuição semelhante ao dos dados reais, ocupando regiões próximas no espaço projetado. Na arquitetura aprimorada neste trabalho, os dados estão mais uniformemente distribuídos e apresentam maior dispersão. Isso é um indicativo de aumento na variabilidade e na capacidade de generalização do modelo gerador.

Para complementar a análise qualitativa, foi calculada a média da métrica *Dynamic Time Warping* (DTW) para cada classe. A DTW é uma métrica consolidada para avaliar a similaridade entre séries temporais, especialmente útil em contextos com alta variabilidade nos dados [Brophy et al. 2023]. Foram comparadas as distâncias entre sequências reais, entre sequências reais e sintéticas geradas pela arquitetura original, e entre sequências reais e sintéticas da versão aprimorada. Quanto mais próximos os dados sintéticos estiverem dos valores dos dados reais, maior a fidelidade dos dados gerados pela arquitetura.

Os resultados, apresentados na Tabela 1, indicam que, com exceção da classe *benign*, a arquitetura aprimorada obteve melhorias significativas, especialmente na classe *establish foothold*, reforçando a eficácia das modificações realizadas. Como o objetivo do trabalho é gerar fluxos de ataques APT para melhorar a representatividade dos dados originais, o desempenho da classe *benign* não impactará a performance dos modelos de *machine learning* testados com o *dataset* semi-sintético.

4.2. Análise de Dados Sintéticos em Modelos de Classificação de Machine Learning

Para responder à última pergunta, sobre como os modelos de *Machine Learning* reagem aos dados sintéticos, e para reforçar o quão semelhantes são os dados sintéticos em relação aos reais, este trabalho avaliou o desempenho de algoritmos de *Machine Learning*

Classe	Real	TTS-CGAN	TTS-CGAN Aprimorada
reconnaissance	14,014	14,666	13,502
benign	10,939	10,454	9,855
establish foothold	15,613	22,651	16,128
lateral movement	13,315	11,910	13,431
exfiltration	8,602	11,677	11,592

Tabela 1. *Dynamic Time Warping* entre os dados reais e sintéticos.

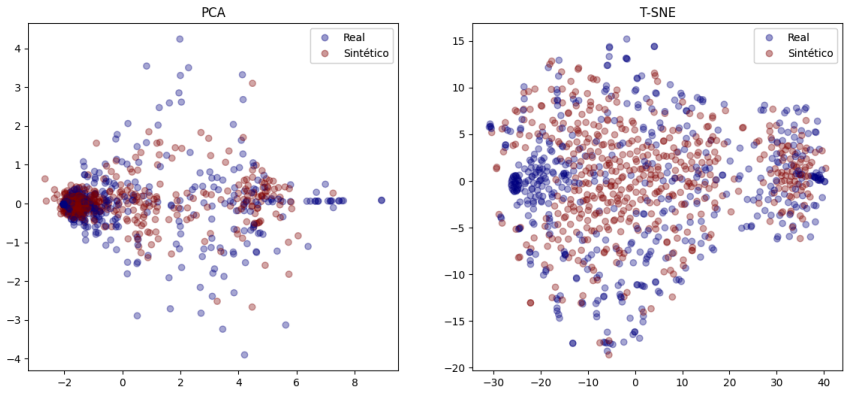
na tarefa de classificação dos diferentes estágios de um ataque APT presentes no *dataset* DAPT2020: *Reconnaissance*, *Foothold Establishment*, *Lateral Movement*, *Data Exfiltration* e tráfego benigno.

Para essa validação, utilizamos três protocolos distintos de avaliação dos modelos de classificação: (i) ***Train on Real, Test on Real*** (TRTR), em que os modelos são treinados e testados exclusivamente com dados reais, servindo como linha de base para comparação; (ii) ***Train on Synthetic, Test on Real*** (TSTR), no qual os modelos são treinados com nossos dados puramente sintéticos e testados com os dados reais, com o objetivo de avaliar o quanto os dados sintéticos conseguem aproximar-se do comportamento real; e (iii) ***Train on Real and Synthetic, Test on Real*** (TRSTR), em que o modelo é treinado em nosso conjunto semi-sintético e testado com dados reais, determinando se a geração de novas amostras de ataques pode melhorar o desempenho dos modelos. Esta validação foi realizada sobre os 30% dos dados reais previamente separados para teste, sendo que o TRTR foi treinado exatamente sobre os mesmos 70% utilizados no treinamento da GAN.

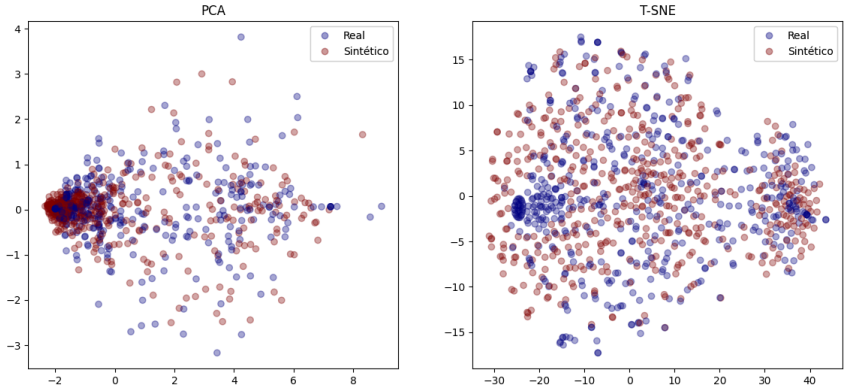
Os testes descritos utilizaram quatro modelos preditivos distintos de *Machine Learning*:

- **Random Forest (RF)**: utiliza a implementação da biblioteca *Scikit-learn*, com 50 estimadores (árvores) e profundidade máxima indefinida;
- **Transformer Encoder**: utiliza um modelo baseado em *transformer encoder*, também implementado em *PyTorch*, com dimensionalidade de entrada de 64, oito cabeças no mecanismo de atenção *multi-head* e *dropout* de 0,3. Os demais hiperparâmetros seguiram os padrões da biblioteca;
- **LSTM**: utiliza uma rede recorrente simples utilizando a biblioteca *PyTorch*, contendo uma camada LSTM com 64 unidades, *dropout* 0.1 e uma camada linear de saída com a mesma dimensionalidade. Os demais hiperparâmetros utilizam os valores padrão da biblioteca;
- **Support Vector Machine (SVM)**: utiliza a implementação padrão da SVM da biblioteca *Scikit-learn*, com os hiperparâmetros padrão e *kernel* com a *Radial Basis Function* (RBF).

As métricas utilizadas para avaliação do desempenho dos modelos foram a **acurácia**, **precisão** e **F1-score**.



(a) TTS-CGAN original



(b) TTS-CGAN com melhorias

Figura 4. Comparação entre os dados reais e sintéticos de APTs: projeções via PCA e t-SNE, para as arquiteturas de GANs analisadas neste trabalho.

Modelos de *Machine Learning*, quando aplicados a *datasets* tradicionais de cibersegurança, podem alcançar altos resultados de precisão quando modelos de classificação binários de detecção de ataques são utilizados. No entanto, esse não é o caso do DAPT2020, especialmente ao considerarmos o desafio adicional de tratar o problema como uma tarefa de classificação multiclasse. Nesse contexto, o objetivo dos testes realizados neste trabalho não é apenas identificar a ocorrência de um ataque, mas também determinar em qual estágio da cadeia de APT ele se encontra.

Em relação aos dados gerados, os resultados apresentados na Tabela 2 indicam que todos os modelos treinados exclusivamente com dados sintéticos puros gerados com as técnicas investigadas neste trabalho obtiveram desempenho inferior ao alcançado com o *dataset* original. Esse comportamento é amplamente esperado, como demonstra a literatura [Esteban et al. 2017, Yoon et al. 2019], uma vez que dados sintéticos, por mais realistas que sejam, dificilmente capturam todas as nuances dos dados reais. Ainda assim, os resultados obtidos evidenciam a similaridade entre os modelos treinados com dados gerados e com dados reais. Por outro lado, os dados semi-sintéticos permitiram melhorar o desempenho de todos os modelos avaliados. Enquanto os modelos SVM e LSTM apresentaram ganhos modestos, o modelo baseado em *transformers* apresentou uma melhoria significativa. A *Random Forest*, por sua vez, destacou-se como o modelo com o maior ganho de desempenho, além de alcançar a melhor pontuação geral.

Modelo	Dados de Treino	Acurácia	Precisão	Recall	F1-score
Random Forest	Real	0.8646	0.8689	0.8280	0.8480
	Sintético	0.7056	0.6938	0.6978	0.6958
	Semi-Sintético	**0.8770**	**0.8812**	**0.8447**	**0.8626**
Transformer (8 heads, 64 dim)	Real	0.8311	0.8258	0.8136	0.8197
	Sintético	0.6994	0.7290	0.6804	0.7039
	Semi-Sintético	**0.8323**	**0.8247**	**0.8287**	**0.8267**
LSTM (64 dim)	Real	**0.8261**	0.7740	**0.8254**	0.7989
	Sintético	0.7267	0.7173	0.7215	0.7194
	Semi-Sintético	0.8124	**0.7989**	0.8057	**0.8023**
SVM (RBF)	Real	**0.8373**	**0.8434**	0.7841	0.8127
	Sintético	0.6745	0.7118	0.6533	0.6813
	Semi-Sintético	**0.8373**	0.8163	0.8107	**0.8135**

Tabela 2. Resultados dos modelos treinados com diferentes conjuntos de dados.

Em [Liao et al. 2024], os autores também utilizaram modelos baseados em SVMs e LSTMs para classificar estágios de APTs dentro do DAPT2020. O artigo apresenta os hiperparâmetros utilizados para a LSTM, que coincidem com os da arquitetura empregada neste trabalho — exceto as dimensões de entrada e saída, que variam em função das diferentes estratégias de pré-processamento adotadas, refletindo objetivos distintos entre os estudos. Apesar dessas divergências, os autores relatam valores de acurácia, precisão e F1-score próximos de 0,80, resultados compatíveis com os obtidos nos nossos experimentos com dados reais. No caso da SVM, os autores em [Liao et al. 2024] reportam métricas superiores às obtidas neste trabalho. Entretanto, como o artigo não detalha os hiperparâmetros utilizados no experimento, torna-se inviável realizar uma comparação direta entre os resultados.

5. Conclusão

Este trabalho explorou o uso de dados sintéticos de APTs, gerados por redes generativas adversariais, para aprimorar o desempenho de modelos de detecção. Para isso, a arquitetura *Transformer Time-Series Conditional* GAN (TTS-CGAN), originalmente voltada à área da saúde, foi adaptada ao domínio de fluxos de rede maliciosos. As adaptações incluíram o aumento da capacidade dos módulos gerador e discriminador, além da adição de camadas de normalização e discriminação por *batch*. A arquitetura resultante foi treinada e avaliada utilizando o confiável *dataset* DAPT2020.

Demonstrando as vantagens de customização da GAN, foram gerados dois conjuntos de dados: sintéticos puros e semi-sintéticos, utilizados tanto em avaliações qualitativas quanto quantitativas. A análise estatística por meio de PCA, t-SNE e *Dynamic Time Warping* evidenciou a proximidade entre os dados gerados e os dados reais. Além disso, os experimentos de detecção mostraram que a utilização de dados semi-sintéticos proporciona ganhos consistentes no desempenho de diferentes modelos de *Machine Learning*, incluindo *Random Forests*, SVMs, LSTMs e *Transformers*.

Como possibilidades para trabalhos futuros, destaca-se a integração de outras

fontes de dados relacionados a ataques APTs reais, bem como a utilização de datasets com características distintas, visando ampliar a generalização do modelo. Além disso, a investigação de arquiteturas generativas mais avançadas, como variações de GANs, modelos de difusão ou *autoencoders* baseados em *transformers*, representa um caminho interessante para aprimorar mais a qualidade e diversidade dos dados sintéticos. O uso de outros tipos de classificadores para a detecção dos estágios de APTs também é um caminho a ser investigado.

Referências

Alo, S. O., Jamil, A. S., Hussein, M. J., Al-Dulaimi, M. K. H., Taha, S. W., e Khlaponina, A. (2024). Automated detection of cybersecurity threats using generative adversarial networks (GANs). In *2024 36th Conference of Open Innovations Association (FRUCT)*, pages 566–577. IEEE.

Alshamrani, A., Myneni, S., Chowdhary, A., e Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, 21(2):1851–1877.

Alzahem, A., Boulila, W., Driss, M., Koubaa, A., e Almomani, I. (2022). Towards optimizing malware detection: An approach based on generative adversarial networks and transformers. In Nguyen, N. T., Manolopoulos, Y., Chbeir, R., Kozierkiewicz, A., e Trawiński, B., editors, *Computational Collective Intelligence*, volume 13501, pages 598–610. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

Bianchi, L., Pregardier, R., Silva, L. A. L., e Santos, C. R. P. (2025). 2pack-gan: Exploring transfer learning to fine-tune generative adversarial networks for network packet generation. In *NOMS 2025-2025 IEEE Network Operations and Management Symposium*, pages 1–9. IEEE.

Brophy, E., Wang, Z., She, Q., e Ward, T. (2023). Generative adversarial networks in time series: A systematic literature review. *ACM Computing Surveys*, 55(10):1–31.

Chakraborty, T., KS, U. R., Naik, S. M., Panja, M., e Manvitha, B. (2024). Ten years of generative adversarial nets (gans): a survey of the state-of-the-art. *Machine Learning: Science and Technology*, 5(1):011001.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Esteban, C., Hyland, S. L., e Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

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

Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., e Aparicio-Navarro, F. J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359.

Ghafir, I., Kyriakopoulos, K. G., Lambotharan, S., Aparicio-Navarro, F. J., Assadhan, B., Binsalleeh, H., e Diab, D. M. (2019). Hidden markov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access*, 7:99508–99520.

Harada, S., Hayashi, H., e Uchida, S. (2019). Biosignal generation and latent variable analysis with recurrent generative adversarial networks. *IEEE Access*, 7:144292–144302.

Hazra, D. e Byun, Y. C. (2020). Synsiggan: Generative adversarial networks for synthetic biomedical signal generation. *Biology (Basel)*, 9(12):441.

Hudson, D. A. e Zitnick, L. (2021). Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR.

Jiang, Y., Chang, S., e Wang, Z. (2021). TransGAN: two pure transformers can make one strong GAN, and that can scale up. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, pages 14745–14758. Curran Associates Inc.

Kumar, A., Kuppusamy, K., e Aghila, G. (2019). A learning model to detect maliciousness of portable executable using integrated feature set. *Journal of King Saud University - Computer and Information Sciences*, 31(2):252–265.

Li, D., Chen, D., Goh, J., e Ng, S.-k. (2018). Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., e Ng, S.-K. (2019). Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*, pages 703–716. Springer.

Li, X., Metsis, V., Wang, H., e Ngu, A. H. H. (2022). Tts-gan: A transformer-based time-series generative adversarial network. In *International conference on artificial intelligence in medicine*, pages 133–143. Springer.

Liao, N., Wang, J., Guan, J., e Fan, H. (2024). A multi-step attack identification and correlation method based on multi-information fusion. *Computers and Electrical Engineering*, 117:109249.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., et al. (2000). Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In *Proceedings DARPA Information survivability conference and exposition. DISCEX'00*, volume 2, pages 12–26. IEEE.

Moustafa, N. e Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6. IEEE.

Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., e Kang, M. (2020). DAPT 2020 - constructing a benchmark dataset for advanced persistent threats. In Wang, G., Ciptadi, A., e Ahmadzadeh, A., editors, *Deployable Machine Learning for Security Defense*, volume 1271, pages 138–163. Springer International Publishing. Series Title: Communications in Computer and Information Science.

Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A., e Huang, D. (2023). Unraveled—a semi-synthetic dataset for advanced persistent threats. *Computer Networks*, 227:109688.

Navarro, J., Deruyver, A., e Parrend, P. (2018). A systematic survey on multi-step attack detection. *Computers & Security*, 76:214–249.

Navidan, H., Moshiri, P. F., Nabati, M., Shahbazian, R., Ghorashi, S. A., Shah-Mansouri, V., e Windridge, D. (2021). Generative adversarial networks (gans) in networking: A comprehensive survey & evaluation. *Computer Networks*, 194:108149.

Sharafaldin, I., Habibi Lashkari, A., e Ghorbani, A. A. A detailed analysis of the CICIDS2017 data set. In Mori, P., Furnell, S., e Camp, O., editors, *Information Systems Security and Privacy*, volume 977, pages 172–188. Springer International Publishing. Series Title: Communications in Computer and Information Science.

Shiravi, A., Shiravi, H., Tavallaee, M., e Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security*, 31(3):357–374.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, U., e Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., e Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6:35365–35381.

Xiong, C., Zhu, T., Dong, W., Ruan, L., Yang, R., Cheng, Y., Chen, Y., Cheng, S., e Chen, X. (2020). Conan: A practical real-time apt detection system with high accuracy and efficiency. *IEEE Transactions on Dependable and Secure Computing*, 19(1):551–565.

Yoon, J., Jarrett, D., e Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Zeeshan, M. e Maasooma (2024). Trans-GAN: A deep learning paradigm for multi-type anomaly detection in network traffic. In *2024 International Conference on Frontiers of Information Technology (FIT)*, pages 1–6. IEEE.

Zhou, P., Zhou, G., Wu, D., e Fei, M. (2021). Detecting multi-stage attacks using sequence-to-sequence model. *Computers & Security*, 105:102203.

Zhu, F., Ye, F., Fu, Y., et al. (2019a). Electrocardiogram generation with a bidirectional lstm-cnn generative adversarial network. *Scientific reports*, 9(1):6734.

Zhu, G., Zhao, H., Liu, H., e Sun, H. (2019b). A novel lstm-gan algorithm for time series anomaly detection. In *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, pages 1–6.