Robustness of Machine Learning-Based Intrusion Detection Systems Against Adversarial Attacks in Cyber-Physical Systems

Antonio Carlos C. da Silva Júnior[1], **Silvio E. Quincozes**[2], **Renan G. Cattelan**[1], **Rodrigo S. Miani**[1]

[1]Faculdade de Computação – Universidade Federal de Uberlândia (UFU)

[2]Universidade Federal do Pampa (UNIPAMPA), Campus Alegrete

{antonio.junior,renan,miani}@ufu.br, silvioquincozes@unipampa.edu.br

Abstract. *Cyber-physical systems are integral to various technological ecosystems, enabling interactions between computing systems, communication networks, and physical processes. Machine learning algorithms are employed to train anomaly-based Intrusion Detection Systems (IDS) to secure these ecosystems. However, they remain vulnerable to adversarial attacks, where training and testing data can be manipulated to degrade performance. This paper evaluates machine learning algorithms in anomaly-based IDSs under adversarial attacks using two datasets from the power system context. The results indicate that ensemble algorithms excelled against JSMA attacks, while single classifiers performed better against FGSM attacks, with Decision Tree and Random Forest being the top performers in their respective groups.*

1. Introdução

Sistemas ciber-físicos (CPS, do inglês *Cyber-Physical Systems*) e Internet das Coisas (IoT, do inglês *Internet of Things*) são definições sobrepostas, ambas referindo-se às tendências na integração digital de recursos, incluindo conectividade de rede e capacidade computacional com dispositivos, sistemas e recursos físicos [Greer et al. 2019]. Exemplos desses sistemas podem ser encontrados em diversos setores, como saúde, agricultura, energia, entre outros. Sistemas de Controle Industrial (SCI) são um tipo específico de CPS. Segundo [Anthi et al. 2021], esses sistemas desempenham um papel crucial na Infraestrutura Nacional Crítica, sendo utilizados, por exemplo, em processos de manufatura, redes elétricas/inteligentes (também conhecidas como *Smart Grid* - SG), estações de tratamento de água, refinarias de gás e petróleo, e na assistência médica.

De acordo com [Quincozes et al. 2022], o aumento de dispositivos conectados às SG resultou na criação de novos protocolos de comunicação. Entre eles, destaca-se a norma que define o conjunto de protocolos padrão IEC-61850, a qual estabelece importantes convenções, especialmente nas áreas de automação e comunicação. Dessa forma, a troca de mensagens entre dispositivos de diferentes fabricantes é padronizada por meio de protocolos com estruturas bem definidas. Ainda segundo [Quincozes et al. 2022], o padrão IEC-61850 desempenha um papel crucial na rede elétrica, sendo responsável por dividir, transformar e combinar fluxos de energia.

No âmbito da segurança da informação, proteger sistemas e redes de computadores contra ataques digitais tem sido uma preocupação crescente nos últimos anos

[Martins et al. 2020]. Nesse contexto, os Sistemas de Detecção de Intrusão (IDS, do inglês *Intrusion Detection System*) tornam-se cada vez mais importantes, pois esses mecanismos constituem a primeira linha de defesa, sendo responsáveis por proteger o sistema contra ataques gerados por tráfego malicioso [Alatwi and Aldweesh 2021].

[Ayub et al. 2020] destacam que o uso de algoritmos de aprendizado de máquina (AM) no desenvolvimento de IDSs é amplamente difundido e tem demonstrado desempenho satisfatório, tornando esses sistemas robustos e eficazes. Por outro lado, [Sahani et al. 2023] apontam que pesquisas de IDSs baseados em AM para SG ainda é um campo pouco explorado. [Mohanty et al. 2022] destacam que, apesar do bom desempenho na classificação do tráfego de rede (benigno/maligno), o uso de IDSs baseados em AM mostra-se vulnerável a ataques adversários. Uma das formas de ataque adversário em IDSs é a manipulação de informações. Nesse tipo de ataque, dados que representam um ataque na rede podem ser alterados para parecer tráfego benigno, ou vice-versa, confundindo o classificador. Isso é alcançado por meio da modificação de certos atributos, como o comprimento da carga útil, a taxa de pacotes e a presença de tráfego bidirecional, entre outros [da Silva et al. 2023].

Os estudos de [Anthi et al. 2021] e [da Silva et al. 2023] analisam o impacto dos ataques adversários em IDSs para CPSs. [Anthi et al. 2021] empregam o projeto *CleverHans* para implementar o ataque JSMA, avaliando seu efeito nos classificadores *Random Forest* e J48, utilizando dados do *Power Systems Datasets*, gerados pela *Mississippi State University* e *Oak Ridge National Laboratory*. Já [da Silva et al. 2023] aplicam os ataques JSMA e FGSM em um sistema ciberfísico, implementando-os via CleverHans e TensorFlow para construir uma Perceptron Multi-Camadas (MLP). Os classificadores *Random Forest*, J48 e MLP são avaliados utilizando também o Power Systems Datasets. Trabalhos similares, como [Figueroa et al. 2022], [Gipiškis et al. 2023] e [Khaw et al. 2024], também abordam ataques adversários em IDSs para sistemas ciberfísicos, diferenciando-se pelo uso de diversos conjuntos de dados ou variações nos ataques adversários. Contudo, tais estudos não exploram a comparação de desempenho entre diferentes tipos de algoritmos de aprendizado de máquina, e não investigam o impacto dos ataques adversários em diversos ambientes ciberfísicos.

O objetivo deste trabalho é avaliar algoritmos de AM usados em IDSs baseados em anomalias quando expostos a ataques adversários. Nessa análise, foram utilizados dois tipos de algoritmos: os *single classifier* e os *ensemble classifier*. *Single classifiers* são aqueles que utilizam apenas um classificador em suas predições. Em contraste, *ensemble learning* utiliza vários classificadores em suas predições. A ideia foi verificar se o uso de múltiplos classificadores pode ser uma vantagem na construção de IDSs para CPSs. Um diferencial deste trabalho foi a utlização de dois conjuntos de dados provenientes do contexto de sistemas de energia.

O restante do artigo está organizado da seguinte maneira: a Seção 2 apresenta os principais conceitos associados ao tema da pesquisa, bem como os trabalhos relacionados; a Seção 3 detalha os métodos e materiais da pesquisa; a Seção 4 relata os resultados do estudo, com a comparação dos diferentes algoritmos utilizados; e, por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Fundamentação Teórica

Esta seção detalha os conceitos de IDSs e de ataques adversários, fundamentais para a compreensão da pesquisa, assim como os principais trabalhos relacionados.

2.1. Sistemas de Detecção de Intrusão

De acordo com [Garcia-Teodoro et al. 2009], IDSs são ferramentas de segurança que buscam reforçar os mecanismos de proteção da informação. Os IDSs utilizam duas metodologias de detecção de intrusos. A primeira é a detecção por assinatura, na qual o IDS busca em uma base de dados padrões de possíveis ações maliciosas, conhecidos como assinaturas. Essa metodologia possui uma boa taxa de assertividade para ataques conhecidos, porém é ineficaz contra ataques que não possuem registros em sua base de dados. Um grande desafio desse modelo é manter a base sempre atualizada. A segunda metodologia refere-se à detecção baseada em anomalias. Nesse caso, o IDS é treinado para classificar o tráfego de rede e distinguir padrões normais de padrões maliciosos. O desafio desse modelo é alcançar um alto nível de precisão, evitando falsos positivos. Ressalta-se que, por não se restringir a assinaturas de ataques conhecidos, essa abordagem consegue identificar distúrbios desconhecidos.

2.2. Ataques Adversários

Apesar do sucesso dos modelos de AM na classificação de tráfego de rede, uma consequência típica do uso dessa abordagem é a sua fragilidade em relação a ataques adversários [Huang et al. 2011]. Segundo [Martins et al. 2020], ataques adversários são definidos como mecanismos que tentam burlar algoritmos de classificação, utilizando entradas criadas especificamente para confundir os algoritmos de AM, fazendo com que a predição não ocorra conforme o esperado. Para [Alhajjar et al. 2021], os ataques adversários são eficazes em diferentes cenários, como reconhecimento de imagem, reconhecimento de fala e detecção de spam.

[Mohanty et al. 2022] apontam a existência de três tipos de ataques adversários: envenenamento, evasão e roubo de modelo. O ataque de envenenamento ocorre quando alterações são feitas nas amostras de treinamento. O ataque de evasão acontece quando as amostras de teste são alteradas. Já o ataque baseado em roubo de modelo ocorre quando o atacante possui ou obtém, de alguma forma, informações privilegiadas sobre o modelo.

Existem também três tipos de cenários para ataques adversários: *Gray Box*, *White Box* e *Black Box*. Quando o atacante tem conhecimento do modelo de classificação, como, por exemplo, o algoritmo de AM que será utilizado, a base de dados, além de toda ou parte dos parâmetros do modelo, ele executa um ataque do tipo *White Box*. Quando o invasor possui apenas um conhecimento parcial ou incompleto da arquitetura do alvo, o cenário é denominado *Gray Box*. Já quando o atacante não tem nenhum conhecimento sobre o modelo, o ataque é denominado *Black Box* [Ayub et al. 2020].

Dentre as diversas técnicas de ataque adversário, destacam-se duas: o *Fast Gradient Sign Method* (FGSM) e o *Saliency Map Attack* (JSMA) baseado no *Jacobian*. Ambos os métodos utilizam a estratégia de adicionar pequenas perturbações à amostra original. Além disso, ambos os ataques geralmente fazem uso de uma MLP pré-treinada como modelo no processo de geração das amostras adversárias. O método FGSM busca alterar cada uma das características dos dados de entrada, gerando um valor específico de

modificação nas amostras. Por outro lado, o método JSMA gera suas amostras adversárias por meio dos mapas de saliência (*saliency maps*) [Alatwi and Aldweesh 2021].

A Eq. 1 representa o ataque FGSM. $\mathbf{x}$ indica a entrada original; $\mathbf{x}^*$ caracteriza o exemplo adversário usado; ϵ é o parâmetro que indica o nível de perturbação adicionado à entrada original; $\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, \mathbf{y})$ indica o gradiente da função de perda J em relação à entrada $\mathbf{x}$; $J(\theta, \mathbf{x}, \mathbf{y})$: representa a função de perda; θ denota os parâmetros do modelo de aprendizado de máquina, como pesos e vieses em uma rede neural; $\mathbf{y}$ representa o rotulo correto associado a entrada verdadeira $\mathbf{x}$; ·sign representa a função sinal, que retorna o sinal de cada componente do gradiente.

$$\mathbf{x}^* = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, \mathbf{y})) \tag{1}$$

A Eq. 2 representa o ataque JSMA. $\mathbf{x}$ indica a entrada original; $\mathbf{x}^*$ representa a entrada modificada; ϵ caracteriza o fator de escala; ·sign denota a função sinal; $\left(\frac{\partial F_t(\mathbf{x})}{\partial \mathbf{x}_i}\right)$ é o gradiente da função de saída.

$$\mathbf{x}' = \mathbf{x} + \alpha \cdot \text{sign}\left(\frac{\partial F_t(\mathbf{x})}{\partial \mathbf{x}_i}\right) \tag{2}$$

A Adversarial Robustness Toolbox (ART) [1] é uma biblioteca de segurança para aprendizado de máquina desenvolvida pela *IBM Research* como software livre, utilizando Python. O ART está relacionado a muitos ataques adversários de última geração e a mecanismos de defesa para modelos convencionais de aprendizado de máquina e aprendizado profundo [Woldeyohannes 2021]. A biblioteca pode ser utilizada como uma ferramenta de desenvolvimento e treinamento de modelos de aprendizado de máquina contra ataques adversários já citados, como evasão, envenenamento, extração e ataques de inferência. Além disso, pode ser usada para defender e mensurar a robustez dos modelos. Todos os ataques adversários implementados neste trabalhos usaram a biblioteca ART.

2.3. Trabalhos Relacionados

[Anthi et al. 2021] estudaram o uso de ataques adversários no contexto de SCI, destacando que o seu trabalho é pioneiro na investigação do comportamento de modelos supervisionados contra esses ataques. Além disso, os pesquisadores abordam como os sistemas SCI podem se defender desses mecanismos de ataque. O trabalho mencionado utiliza a biblioteca em Python CleverHans[2] para gerar os ataques maliciosos. O ataque usado foi o JSMA, e os algoritmos testados foram RF e J48. Os resultados mostraram uma queda de desempenho em ambos os algoritmos. No entanto, quando as amostras adversárias foram inseridas no treinamento, os classificadores demonstraram maior robustez em relação a esses ataques.

[Figueroa et al. 2022] abordam que os SCI utilizam algoritmos de AM para executar funções do dia a dia, tornando-se alvos potenciais para ataques adversários. Portanto, pesquisas sobre esses ataques podem melhorar a compreensão do assunto e evitar que usuários maliciosos usem esses recursos de forma indevida. Para esta pesquisa, foram

[1] https://github.com/Trusted-AI/adversarial-robustness-toolbox

[2] https://github.com/cleverhans-lab/cleverhans

usados três tipos de ataques: FGSM, *DeepFool* e JSMA. Os resultados indicam que ataques adversários têm consequências negativas no desempenho de DL e algoritmos de AM em SCI.

O trabalho de [da Silva et al. 2023] também investigou ataques adversários em cenários SCI, utilizando os ataques FGSM e JSMA. Além disso, foi constatado que o conhecimento prévio sobre a estrutura do algoritmo alvo pode resultar em ataques mais intensos. Outro fator que influenciou a intensidade dos ataques foi o volume de dados acessados pelo atacante. Embora o FGSM tenha causado ataques com severidade média maior em comparação ao JSMA, este último apresenta a vantagem de ser menos invasivo e, possivelmente, mais difícil de ser detectado.

De acordo com [Quincozes et al. 2022], apesar de os métodos de detecção de intrusão serem amplamente estudados em redes e sistemas convencionais, existem poucos estudos que consideram os protocolos IEC–61850. Além disso, [Quincozes et al. 2022] revelam que existe um gargalo quando se trata de dados industriais realistas para treinamento, teste e avaliação de IDSs. O trabalho resultou em um framework para gerar conjuntos de dados IEC–61850 com características específicas, extraídas de protocolos de comunicação de subestação em nível de rede e do domínio elétrico, para detectar diferentes tipos de ataques. A geração desse framework é uma alternativa para mitigar o gargalo relacionado aos dados industriais, fornecendo um conjunto de dados que auxilia nesse contexto.

[Hariguna and Hananto 2022] investigaram o uso de algoritmos *single classifier* e *ensemble classifier* em um contexto semelhante (IDS) mas sem focar em ataques adversários. Os autores analisaram qual dos grupos se destaca em termos de desempenho em diferentes modelos de detecção de intrusão. Os resultados indicaram que os *single classifier* alcançaram uma precisão de 77,4%, enquanto os *ensemble classifier* chegaram a uma precisão de 96,8%. Considerando esses aspectos, existem diversos trabalhos que avaliam ataques adversários no contexto de CPSs e outros que buscam comparar *single classifier* e *ensemble classifier*. No entanto, ainda há limitações na literatura, como dificuldades em obter dados que se aproximem da realidade, em definir os atributos mais impactados pelos ataques, em identificar os melhores algoritmos de AM contra ataques adversários dentro de cada grupo, e em estabelecer uma metodologia sólida para lidar com esses ataques.

3. Materiais e Métodos

O método adotado nesta pesquisa compreende 11 etapas: 1) escolha da base de dados associada a um CPS; 2) separação da base de dados em 70% para treino e 30% para teste; 3) realização do pré-processamento; 4) seleção dos algoritmos, com 4.1) a escolha de um conjunto de algoritmos do tipo *single classifier* e 4.2) a escolha de um conjunto de algoritmos do tipo *ensemble classifier*; 5) treinamento do modelo de classificação binário para cada um dos algoritmos; 6) teste do modelo de classificação binário para cada um dos algoritmos; e 7) avaliação o desempenho dos modelos. Em relação aos ataques adversários, as etapas são: 8) roubo do conjunto de treino (70%) e treino da MLP; 9) alteração das amostras de ataque do conjunto de testes usando dois ataques (FGSM e JSMA), juntamente com a MLP pré-treinada, variando os parâmetros dos ataques; e 10) apresentação do conjunto de testes modificado aos modelos de classificação treinados; e 11) reavaliação

do desempenho dos modelos. A Figura 1 ilustra essa metodologia. O cenário pode ser considerado *Gray Box*, uma vez que, o atacante tem apenas informações parciais a respeito dos dados, usando apenas o conjunto de treinamento.

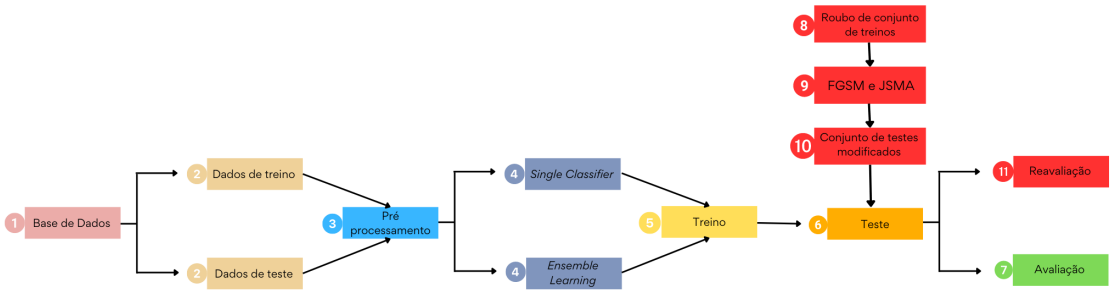

Figura 1. Metodologia de pesquisa adotada

Foram consideradas duas bases de dados: a primeira representa o cenário de SCI, e a segunda, cenários baseados em protocolos IEC–61850. A escolha da base de dados para treinamento e desenvolvimento de testes é crucial para a construção da pesquisa, pois influencia diretamente os resultados obtidos. Foram utilizadas duas fontes de informação: *Power System Smart Grid Monitoring Power*, usada em [Alatwi and Aldweesh 2021], e *Ereno IEC-61850 Intrusion Dataset*, desenvolvida em [Quincozes et al. 2024]. Essas bases foram escolhidas por representarem cenários realistas de sistemas ciber-físicos.

O *Power System Smart Grid Monitoring Power* é um conjunto de dados referente a um sistema de energia, desenvolvido pela Mississippi State University e Oak Ridge National Laboratory em 15 de abril de 2014. Essa base de dados é composta por três conjuntos de dados derivados de um conjunto inicial, que consiste em 15 subconjuntos, cada um com 37 eventos de sistema de energia. Esses eventos são divididos em oito eventos naturais, um evento sem ocorrência, e 28 eventos de ataque. Além disso, a base de dados possui 129 colunas, sendo 116 delas resultantes da medição de 29 atributos, em que cada um é medido por quatro unidades de medição fasorial (UMF), e 12 colunas dedicadas a registros do painel de controle, Snort, e uma última coluna para classificar o evento. O conjunto de dados original consiste de 55.663 amostras maliciosas e 22.714 benignas.

A segunda base de dados selecionada, *ERENO IEC-61850 Intrusion Dataset*, consiste de sete conjuntos de dados, cada um cobrindo os cenários de ataque propostos. Cada conjunto de dados possui 69 características e diferentes quantidades de ataques e amostras normais. No total, o conjunto de dados possui 79.991 amostras maliciosas e 19.998 amostras benignas. Cada exemplo do conjunto de dados corresponde a uma mensagem sobre um evento gerenciado pelo protocolo GOOSE (*Generic Object Oriented Substation Event*) usado para troca de mensagens entre os dispositivos de uma subestação elétrica. Um determinado evento (malicioso ou não) corresponde a múltiplas mensagens. Assim, as duas bases de dados selecionadas auxiliam na construção de uma pesquisa mais próxima da realidade, além de estar em alinhamento com os objetivos deste trabalho, que visam analisar ataques adversários em CPS, considerando dois grupos de algoritmos de AM: os *ensemble classifiers* e os *single classifiers*.

Em seguida, esses dados passam pela fase de pré processamento, onde valores

nulos ou inválidos são removidos, a coluna de classificação é convertida para binária, onde zero representa tráfego de rede normal e 1 tráfego de rede malicioso. Além disso, é utilizado o método Min-Max para normalizar os dados. Posteriormente, uma função de transformação (fit_transform) é aplicada para ajustar os dados de acordo com os valores mínimos e máximos calculados. Para finalizar o pré-processamento, é realizado um balanceamento nas amostras de treino.

Depois da escolha dos conjuntos de dados, o próximo passo foi a divisão do conjunto em treino e teste. O mesmo processo foi adotado em ambas, com uma divisão de 70% para treino e 30% para teste. A decisão para o uso da estratégia *hold-out* se justifica pois os principais trabalhos relacionados [Anthi et al. 2021], [da Silva et al. 2023] e [Figueroa et al. 2022] também a utilizam. Durante o processo de treino, os dados passaram por um processo de balanceamento com o auxílio do método *RandomUnderSampler* do pacote *imbalanced-learn* do Python. O *Power System Smart Grid Monitoring Power* após o balanceamento ficou com 14.441 instâncias benignas e maliciosas enquanto que o *Ereno IEC-61850 Intrusion Dataset* ficou com 13.969 amostras benignas e maliciosas. Em seguida, os algoritmos foram selecionados com base em dois critérios principais: o primeiro foi criar dois grupos de algoritmos, um representando a classe *single classifier* e o outro representando a classe *ensemble classifier*. O segundo critério foi a relevância desses algoritmos na literatura. O resultado foi um conjunto de sete algoritmos, sendo três deles *single classifier* (Support Vector Machine (SVM), K-nearest Neighbors (KNN) e Árvore de Decisão (AD)) e os outros quatro *ensemble classifier* (Random Forest, GBM, AdaBoost e XGBoost). É importante lembrar que os referidos algoritmos foram escolhidos por aparecerem com frequência em trabalhos sobre desenvolvimento de sistemas de detecção de intrusão baseados em aprendizado de máquina. Além disso, foi utilizada a biblioteca scikit-learn [3] do Python para os implementar algoritmos escolhidos. Todos os algoritmos foram executados com os valores padrão da biblioteca.

A seguir, é criada uma linha de base (*baseline*), utilizada como ponto de partida para analisar o desempenho dos classificadores sem a interferência de ataques adversários. Posteriormente, é avaliado o desempenho dos mesmos classificadores, agora com a interferência desses ataques. A *baseline* consiste em treinar os algoritmos pré-selecionados (*single classifier* e *ensemble machine learning*). Após o treinamento, os algoritmos são testados, e suas performances são avaliadas. O desempenho dos algoritmos foi medido com base no F1-Score.

As amostras adversárias são então geradas utilizando a biblioteca ART, desenvolvida em Python ART. Essa ferramenta oferece diversos tipos de ataques adversários, e o estudo selecionou aqueles que melhor se relacionam com os objetivos da pesquisa. Os ataques investigados na presente pesquisa foram o JSMA e o FGSM. Os valores dos parâmetros theta e gama para o ataque JSMA foram definidos como 0,1. Para o FGSM, foi utilizado o valor de 0,08 para o parâmetro eps. Nesse momento, o mesmo conjunto de dados usado no treinamento dos algoritmos é ”roubado”por um indivíduo malicioso e utilizado para treinar uma MLP. Em seguida, as amostras de ataque do conjunto de testes foram alteradas utilizando os dois ataques (FGSM e JSMA) aplicados pela MLP pré-treinada. Os parâmetros dos ataques são variados para permitir comparações.

[3]https://scikit-learn.org/stable/

A última etapa envolve o novo treinamento com as amostras adversárias. Nessa fase, as amostras que sofreram perturbações dos ataques adversários são introduzidas ao processo. Os classificadores treinados agora recebem os dados de teste adulterados pelos ataques adversários e são avaliados novamente, com o objetivo de validar os algoritmos que mais se destacam e o grupo ao qual pertencem. A próxima seção detalha os resultados dos experimentos com os dois conjuntos de dados.

4. Resultados

Inicialmente, sete modelos de aprendizado de máquina foram testados neste estudo. Três desses modelos são *single classifier*: Support Vector Machine (SVM), K-nearest Neighbors (KNN) e Árvore de Decisão (AD). Os outros quatro são *ensemble classifier*: Random Forest, Gradient Boosting Machine (GBM), AdaBoost e XGBoost. Esses modelos foram treinados e testados utilizando duas bases de dados: *Power System Smart Grid Monitoring Power* e *Ereno IEC-61850 Intrusion Dataset*, e posteriormente atacados por dois ataques adversários (JSMA e FGSM). Os resultados alcançados pelos testes iniciais levaram em consideração as configurações padrão dos algoritmos.

A Figura 2 apresenta o resultado dos algoritmos considerando dois cenários: o Cenário 1 não tem ataque adversário, enquanto no Cenário 2 são acrescentados os ataques adversários FGSM e JSMA. Em ambos os ataques, a métrica considerada foi o F1-score e a base de dados usada é a *Power System Smart Grid Monitoring Power*.

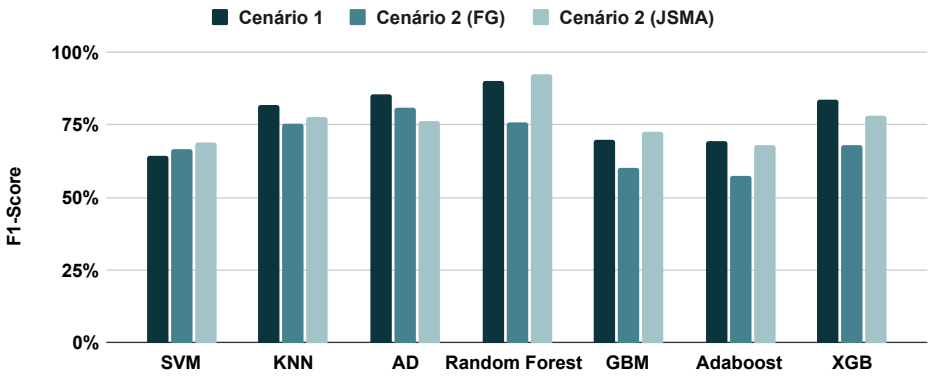

Figura 2. Resultados para o *Power System Smart Grid Monitoring Power*.

A Figura 3 apresenta a mesma metodologia, mudando apenas a base de dados usada, para a *Ereno IEC-61850 Intrusion Dataset*. O F1-Score combina a precisão e a revocação em uma única medida. Quanto mais próximo de um, melhor o desempenho do classificador; quanto mais próximo de zero, pior o desempenho. A fórmula usada para calcular o F1-Score é dada pela Eq. 3. A precisão avalia o número de acertos do modelo, considerando o número total de tentativas, ou seja, das previsões positivas, quantas eram verdadeiras positivas. Já a revocação mede a proporção de acertos do modelo em relação ao total de casos que ele deveria ter acertado.

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{3}$$

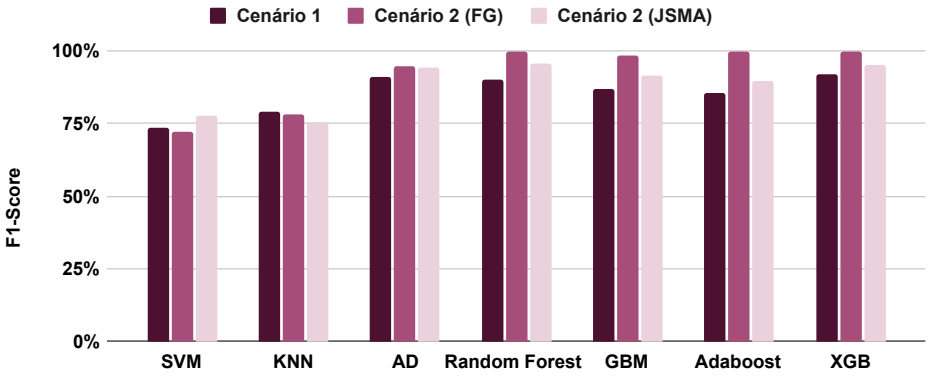

Figura 3. Resultados para o *Ereno IEC-61850 Intrusion Dataset*.

4.1. Power System Smart Grid Monitoring Power

A partir dos resultados obtidos (Figura 2), pode-se perceber que o melhor classificador no Cenário 1 foi o Random Forest, com um desempenho de aproximadamente 90%. A Árvore de Decisão seguiu com uma medida próxima aos 85%, seguida pelos classificadores XGB e KNN, que também se aproximaram dos 85%, com o XGB sendo ligeiramente superior ao KNN. Os demais algoritmos ficaram abaixo dos 70%.

Em relação ao Cenário 1, considerando o ataque FGSM, é possível notar que o grupo *single classifier* apresentou um menor decaimento em seu desempenho em comparação ao grupo *ensemble classifier*, com destaque para o classificador SVM, que inclusive mostrou uma melhora após o ataque. Além disso, a Árvore de Decisão registrou o melhor F1-Score entre todos os classificadores após o ataque pelo FGSM. O padrão observado entre os *ensemble classifiers* foi semelhante, com todos apresentando um declínio considerável, embora o Random Forest ainda tenha registrado o melhor F1-Score dentro do grupo. Com exceção do SVM, os classificadores do grupo *single classifier* também experimentaram um declínio em seus desempenhos.

O classificador do grupo *single classifier* que mais se destacou foi a Árvore de Decisão, com F1-Score superior a 75% em todos os cenários e uma avaliação de aproximadamente 80% quando não exposta a nenhum ataque (Cenário 1). Enquanto isso, o algoritmo que mais se destacou entre os *ensemble learning* foi o Random Forest, com desempenhos superiores a 75% em todos os cenários e uma avaliação de aproximadamente 90% quando não exposto a nenhum ataque (Cenário 1). Curiosamente, o ataque JSMA melhorou o F1-Score do Random Forest.

Ainda considerando o Cenário 2, mas agora com o ataque JSMA, observa-se que os algoritmos do grupo *ensemble learning*, Random Forest e GBM, tiveram um aumento em seu F1-Score, enquanto o AdaBoost apresentou uma leve depreciação, e o XGB teve um decréscimo menor em comparação ao ataque FGSM. O Random Forest alcançou o melhor F1-Score entre todos os classificadores, considerando ambos os grupos, com uma performance próxima de 95%. No grupo *single classifier*, o SVM manteve o mesmo comportamento observado com o ataque FGSM, ou seja, seu F1-Score aumentou, mas com o JSMA o aumento foi ainda maior. Além disso, o KNN apresentou um menor declínio quando exposto ao JSMA em comparação ao FGSM, e a Árvore de Decisão mostrou-se mais vulnerável ao JSMA do que ao FGSM.

4.2. Ereno IEC-61850 Intrusion Dataset

A partir dos resultados obtidos na Figura 3 pode-se perceber que os melhores classificadores de cada grupo, considerando o Cenário 1, foram o XGB e a Árvore de Decisão, ambos com F1-Score superior a 90%. Vale ressaltar que o Random Forest alcançou um desempenho quase similar ao do XGB. Os demais algoritmos do grupo *Ensemble Learning* apresentaram F1-Score de aproximadamente 85%, enquanto os *Single Classifier* tiveram F1-Scores em torno 75%.

No Cenário 2, considerando o ataque FGSM, todos os classificadores do grupo *Ensemble Classifier* tiveram seu F1-Score melhorado, chegando a cerca de 99%, e não foram afetados pelo ataque. Enquanto isso, no grupo *Single Classifier*, o destaque foi novamente a Árvore de Decisão, que não foi afetada e alcançou um F1-Score de aproximadamente 95%, enquanto os demais algoritmos foram impactados pelo ataque e tiveram desempenho próximo de 75%.

Ainda considerando o Cenário 2, mas agora com o ataque JSMA, o Random Forest, o XGB e a Árvore de Decisão se destacaram, com F1-Scores próximos a 95%. O grupo *Ensemble Classifier* novamente não foi afetado e teve seu F1-Score melhorado em todos os classificadores. Em relação ao grupo *Single Classifier*, a Árvore de Decisão e o SVM não apresentaram depreciação no F1-Score, enquanto o KNN teve uma perda maior em comparação ao ataque FGSM.

4.3. Discussão

Os classificadores apresentaram melhor desempenho quando treinados e testados utilizando a base de dados *Ereno IEC-61850 Intrusion Dataset*. Isso se deve ao processo de enriquecimento das características (*features*) ao qual a base de dados foi submetida. Inclusive, em [Quincozes et al. 2024], foi realizado um teste que demonstrou uma melhora significativa nos resultados ao comparar o uso da base de dados enriquecida com a não enriquecida. Apesar disso, o algoritmo *Random Forest* manteve um bom desempenho em ambas as bases de dados, destacando-se entre os *Ensemble Classifiers*. Além disso, a Árvore de Decisão foi o destaque entre os *Single Classifiers*. Por outro lado, o SVM apresentou um F1-Score relativamente menor que os demais classificadores, embora tenha demonstrado certa resistência a ambos os ataques em ambos os cenários.

Em relação à base de dados *Power System Smart Grid Monitoring Power*, o grupo *Single Classifier* apresentou o melhor desempenho geral. No entanto, esse grupo foi o mais afetado quando exposto ao ataque FGSM. Em contraste, no ataque JSMA, o grupo *Ensemble Classifier* foi o mais impactado, com exceção do SVM, que não foi afetado nesse cenário. Na Tabela 1, é possível observar que a média do grupo *Ensemble Classifier* é maior do que a do grupo *Single Classifier* para ambos os ataques adversários em CPS. No entanto, ao considerar o desvio padrão, o grupo *Single Classifier* apresenta um desvio padrão menor em ambas as situações de ataque. Isso sugere que o grupo *Single Classifier* pode ser um pouco mais resistente aos ataques do que o grupo *Ensemble Classifier* em alguns casos.

As principais limitações enfrentadas pela pesquisa foram a consideração apenas de parâmetros fixos para os ataques. Além disso, não foram testadas variações no tamanho das amostras de teste e treino durante a execução do método. Vale ressaltar que,

Tabela 1. Comparativo entre os algoritmos *Emsemble Learning* e *Single Learning*

	Ensemble Learning	Single Learning
Média FGSM	0.817 ± 0.176	0.775 ± 0.086
Média JSMA	0.85 ± 0.1	0.78 ± 0.077

levando em consideração os cenários analisados, os resultados indicam que os algoritmos *Ensemble Classifier* são mais eficazes na construção de IDSs em CPSs.

5. Conclusão

O objetivo deste trabalho foi analisar o impacto de ataques adversários em IDSs baseados em anomalias para CPS, construídos utilizando *single classifiers* e *ensemble classifiers*. Os algoritmos de aprendizado de máquina investigados neste estudo foram Support Vector Machine (SVM), K-nearest Neighbors (KNN) e Árvore de Decisão (AD), representando os *single classifiers*, enquanto Random Forest, Gradient Boosting Machine (GBM), AdaBoost e XGBoost representaram os *ensemble classifiers*. Para testar esses algoritmos, foram utilizados dois ataques: FGSM e JSMA. Com base nos resultados, os algoritmos de *ensemble machine learning* tiveram um desempenho superior aos *single classifiers*. No entanto, em algumas situações específicas, como nos ataques Fast Gradient, os *single classifiers* se sobressaíram. Entre os melhores algoritmos de cada grupo, a Árvore de Decisão apresentou os melhores resultados entre os *single classifiers*, enquanto o Random Forest se destacou no grupo de *ensemble machine learning*.

Como trabalhos futuros, pretende-se analisar a influência dos parâmetros *theta*, *gama* e *eps* nos algoritmos JSMA e FGSM, respectivamente. Além disso, será avaliado o impacto de outros cenários de ataque, como o roubo de uma quantidade menor de amostras de treinamento. Por fim, outra linha de investigação consiste em avaliar a resistência dos classificadores após a inserção de um certo número de amostras adversárias.

Referências

Alatwi, H. A. and Aldweesh, A. (2021). Adversarial black-box attacks against network intrusion detection systems: A survey. In *2021 IEEE World AI IoT Congress (AIIoT)*, pages 0034–0040. IEEE.

Alhajjar, E., Maxwell, P., and Bastian, N. (2021). Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications*, 186:115782.

Anthi, E., Williams, L., Rhode, M., Burnap, P., and Wedgbury, A. (2021). Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *Journal of Information Security and Applications*, 58:102717.

Ayub, M. A., Johnson, W. A., Talbert, D. A., and Siraj, A. (2020). Model evasion attack on intrusion detection systems using adversarial machine learning. In *54th annual conference on information sciences and systems (CISS)*, pages 1–6. IEEE.

da Silva, G. H. E., Miani, R. S., and Zarpelao, B. B. (2023). Investigando o impacto de amostras adversárias na detecção de intrusões em um sistema ciberfísico. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 281–294. SBC.

Figueroa, H., Wang, Y., and Giakos, G. C. (2022). Adversarial attacks in industrial control cyber physical systems. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28.

Gipiškis, R., Chiaro, D., Preziosi, M., Prezioso, E., and Piccialli, F. (2023). The impact of adversarial attacks on interpretable semantic segmentation in cyber–physical systems. *IEEE Systems Journal*, 17(4):5327–5334.

Greer, C., Burns, M., Wollman, D., and Griffor, E. (2019). Cyber-physical systems and internet of things.

Hariguna, T. and Hananto, A. R. (2022). Improved intrusion detection system (ids) performance using machine learning: A comparative study of single classifier and ensemble learning. In *2022 IEEE Creative Communication and Innovative Technology (ICCIT)*, pages 1–7. IEEE.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. D. (2011). Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58.

Khaw, Y. M., Jahromi, A. A., Arani, M. F., and Kundur, D. (2024). Evasive attacks against autoencoder-based cyberattack detection systems in power systems. *Energy and AI*, page 100381.

Martins, N., Cruz, J. M., Cruz, T., and Abreu, P. H. (2020). Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. *IEEE Access*, 8:35403–35419.

Mohanty, H., Roudsari, A. H., and Lashkari, A. H. (2022). Robust stacking ensemble model for darknet traffic classification under adversarial settings. *Computers & Security*, 120:102830.

Quincozes, S. E., Albuquerque, C., Passos, D., and Mossé, D. (2022). Ereno: An extensible tool for generating realistic iec-61850 intrusion detection datasets. In *Anais Estendidos do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 1–8. SBC.

Quincozes, S. E., Albuquerque, C., Passos, D., and Mossé, D. (2024). Ereno: A framework for generating realistic iec–61850 intrusion detection datasets for smart grids. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3851–3865.

Sahani, N., Zhu, R., Cho, J.-H., and Liu, C.-C. (2023). Machine learning-based intrusion detection for smart grid computing: A survey. *ACM Transactions on Cyber-Physical Systems*, 7(2):1–31.

Woldeyohannes, H. D. (2021). Review on "adversarial robustness toolbox (art) v1. 5. x.": Art attacks against supervised learning algorithms case study.