Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Avaliação de Safety e Cybersecurity de Classificadores de Machine Learning na Indústria Utilizando SafeML

Vitória Isabela de Oliveira

JUIZ DE FORA
JULHO, 2026

Avaliação de Safety e Cybersecurity de Classificadores de Machine Learning na Indústria Utilizando SafeML

Vitória Isabela de Oliveira

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: André Luiz de Oliveira

Coorientador: Alex Borges Vieira

JUIZ DE FORA

JULHO, 2026

Avaliação de Safety e Cybersecurity de Classificadores de Machine Learning na Indústria Utilizando SafeML

Vitória Isabela de Oliveira

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

André Luiz de Oliveira
Prof. Dr.

Alex Borges Vieira
Prof. Dr.

Carlos Cristiano Hasenclever Borges
Prof. Dr.

Victor Ströele de Andrade Menezes / Bernardo Martins Rocha
Prof. Dr.

JUIZ DE FORA

13 DE JULHO, 2026

A Jeová, minha fonte de força e refúgio em todos os momentos.
À minha mãe, Maria Isabel de Jesus, guerreira e vitoriosa,
minha maior inspiração.

Resumo

A crescente adoção de classificadores de Machine Learning (ML) em sistemas de segurança crítica demanda transferir a avaliação de confiança (*dependability*) e segurança cibernética do tempo de projeto (técnicas como *Fault Tree Analysis*, FTA) para o monitoramento em tempo de execução (*runtime*). Ataques adversariais — ameaças que comprometem a confidencialidade, a integridade ou a disponibilidade do sistema — representam risco significativo: a violação de uma propriedade de segurança cibernética pode provocar uma falha de *safety* (definida como *freedom from unacceptable risk*), com consequências catastróficas a pessoas, ao ambiente ou à propriedade. O objetivo deste trabalho foi avaliar a efetividade de classificadores de ML na detecção de ataques adversariais na indústria, utilizando o método SafeML — abordagem de *Explainable AI* baseada em medidas de dissimilaridade estatística (SDD) sobre Funções de Distribuição Empírica Acumulada (ECDF), que monitora a confiança em tempo de execução de forma *model-agnostic*. Como resultado, desenvolveu-se um software de monitoramento sob Clean Architecture e SOLID. Avaliaram-se quatro classificadores — LeNet5-BN (MNIST, 99,14%), SmallResNet (CIFAR-10, 89,54%), ResNet-18 (GTSRB, 99,07%) e Random Forest (CICIDS2017, 89,77%) — sob os ataques FGSM, PGD, C&W, DeepFool e Ruído Gaussiano, medindo o desvio padrão das previsões por cinco distâncias estatísticas: Kolmogorov-Smirnov, Wasserstein, Anderson-Darling, Kuiper e Cramér-von Mises. As medidas SDD demonstraram correlações positivas com a queda de *accuracy*: para LeNet5-BN (MNIST) e SmallResNet (CIFAR-10), todas exceto Anderson-Darling atingiram AUC-ROC = 1,000 (detecção perfeita); para o Random Forest (CICIDS2017), as correlações foram significativas (KS $r = 0{,}973^{***}$, Kuiper $r = 0{,}974^{***}$). Para a ResNet-18 (GTSRB) sobre pixels brutos o desempenho foi inferior (AUC-ROC até 0,833), elevando-se ao adotar *features* CNN de 512 dimensões. O SafeML mostrou-se alternativa *model-agnostic* viável e eficaz para o monitoramento de segurança de ML industrial.

Palavras-chave: SafeML, Machine Learning, Safety, Cybersecurity, Ataques Adversariais, ECDF, Monitoramento em Tempo de Execução.

Abstract

The growing adoption of Machine Learning (ML) classifiers in safety-critical systems demands a transition from design-time dependability and cybersecurity assessment (techniques such as Fault Tree Analysis, FTA) to runtime monitoring. Adversarial attacks — threats that compromise the confidentiality, integrity, or availability of the system — pose a significant risk: the violation of a cybersecurity property may cause a safety failure (defined as *freedom from unacceptable risk*), with catastrophic consequences to people, the environment, or property. The objective of this work was to evaluate the effectiveness of ML classifiers in detecting adversarial attacks in industry, using the SafeML method — an Explainable AI approach based on Statistical Dissimilarity Distance (SDD) measures over Empirical Cumulative Distribution Functions (ECDF) that monitors classifier confidence at runtime in a model-agnostic manner. As a result, a monitoring software was developed under Clean Architecture and SOLID. Four classifiers were evaluated — LeNet5-BN (MNIST, 99.14%), SmallResNet (CIFAR-10, 89.54%), ResNet-18 (GTSRB, 99.07%), and Random Forest (CICIDS2017, 89.77%) — under FGSM, PGD, C&W, DeepFool, and Gaussian Noise attacks, measuring the standard deviation of predictions with five statistical distances: Kolmogorov-Smirnov, Wasserstein, Anderson-Darling, Kuiper, and Cramér-von Mises. The SDD measures showed positive correlations with accuracy drop: for LeNet5-BN (MNIST) and SmallResNet (CIFAR-10), all but Anderson-Darling achieved AUC-ROC = 1.000 (perfect detection); for Random Forest (CICIDS2017), correlations were significant (KS $r = 0.973^{***}$, Kuiper $r = 0.974^{***}$). For ResNet-18 (GTSRB) over raw pixels performance was lower (AUC-ROC up to 0.833), improving when 512-dim CNN features were adopted. SafeML proved a viable model-agnostic alternative and an effective mechanism for security monitoring of industrial ML.

Keywords: SafeML, Machine Learning, Safety, Cybersecurity, Adversarial Attacks, ECDF, Runtime Monitoring.

Agradecimentos

Agradeço, primeiramente e acima de tudo, a Jeová, por ser minha fonte inesgotável de força, coragem e auxílio durante toda a minha jornada. Sem Ele, nada disso teria sido possível. Conciliar trabalho e estudos desde o início da graduação foi um caminho árduo, mas a cada dia Ele me sustentou e me deu forças para continuar.

À minha mãe, Maria Isabel de Jesus, dedico minha mais profunda gratidão. Mãe, você é a guerreira e vitoriosa mais grandiosa que eu conheço. Sua trajetória de vida, que também nunca foi fácil, é a minha maior fonte de inspiração. Obrigada por estar ao meu lado o tempo todo, por me ajudar no dia a dia corrido, por me dar forças para continuar quando eu pensava em desistir. Tudo o que eu sou e tudo o que eu conquistei, devo a você.

À minha irmã, Vitória Daniela de Oliveira, que foi um dos meus maiores alicerces durante essa caminhada. Obrigada por todo o apoio, carinho e por acreditar em mim.

Ao meu amigo Allan Henriques Teixeira, por toda a parceria durante a graduação. A caminhada foi mais leve e significativa com você ao lado.

Ao meu orientador, Prof. Dr. André Luiz de Oliveira, pela orientação dedicada, paciência e contribuições essenciais para o desenvolvimento deste trabalho.

Ao meu coorientador, Prof. Dr. Alex Borges Vieira, pelas valiosas sugestões e direcionamentos que enriqueceram esta pesquisa.

Aos professores do Departamento de Ciência da Computação da UFJF, pelos ensinamentos que formaram a base do meu conhecimento.

À UFJF, por proporcionar um ambiente de aprendizado e crescimento acadêmico.

"Se fosse fácil achar o caminho das pedras,

tantas pedras no caminho não seriam ruim".

Engenheiros do Havaí

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AD	Anderson-Darling
ART	Adversarial Robustness Toolbox
AUC	Area Under the Curve
C&W	Carlini & Wagner
CICIDS	Canadian Institute for Cybersecurity Intrusion Detection System
CNN	Convolutional Neural Network
CvM	Cramér-von Mises
DDD	Domain-Driven Design
DNN	Deep Neural Network
ECDF	Empirical Cumulative Distribution Function
FGSM	Fast Gradient Sign Method
FPR	False Positive Rate
GTSRB	German Traffic Sign Recognition Benchmark
KS	Kolmogorov-Smirnov
ML	Machine Learning
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
PGD	Projected Gradient Descent
ROC	Receiver Operating Characteristic
SDD	Statistical Dissimilarity Distance
SOLID	Single responsibility, Open-closed, Liskov substitution, Interface segregation, Dependency inversion
SOTIF	Safety of the Intended Functionality
TPR	True Positive Rate
W	Wasserstein

1 Introdução

1.1 Contextualização e Motivação

A crescente adoção de algoritmos de Machine Learning (ML) tem transformado diversos setores da indústria, da condução autônoma aos sistemas de cibersegurança. Em particular, classificadores de ML passaram a ocupar um papel central em tarefas de segurança da informação, como a detecção de intrusões em redes de computadores e a identificação de vulnerabilidades em software, repositórios e código-fonte (ASLANSEFAT et al., 2020). Nesses contextos, as decisões do modelo influenciam diretamente a proteção de pessoas, dados e infraestruturas.

Apesar do bom desempenho desses modelos em ambientes controlados, sua utilização em cenários reais ainda apresenta um desafio relevante: classificadores de ML raramente indicam *quando* suas previsões podem estar incorretas ou quando operam fora do domínio em que foram treinados. Esse problema é especialmente crítico em cibersegurança, em que os dados de entrada mudam continuamente e em que adversários podem explorar deliberadamente as fragilidades do modelo. Classificadores de ML são, ademais, vulneráveis a ataques adversariais, isto é, perturbações cuidadosamente projetadas nas entradas que induzem classificações incorretas com alta confiança (GOODFELLOW; SHLENS; SZEGEDY, 2014).

Diante desse cenário, torna-se indispensável não apenas obter uma previsão, mas também estimar o quão confiável ela é. O framework SafeML, proposto por Aslansefat et al. (2020), atende a essa necessidade ao oferecer uma abordagem *model-agnostic* (independente da arquitetura do classificador) baseada em medidas de dissimilaridade estatística (SDD, do inglês *Statistical Dissimilarity Distance*), que quantificam o quanto os dados operacionais diferem dos dados de treinamento por meio de Funções de Distribuição Empírica Acumulada (ECDF).

A hipótese central do SafeML, avaliada sistematicamente neste trabalho, é a de que, quando os dados de entrada em produção divergem da distribuição observada durante

o treinamento, como ocorre sob ataques adversariais ou diante de tráfego malicioso, o valor SDD aumenta de forma correlacionada com a queda de desempenho do classificador. A detecção dessa divergência permite sinalizar previsões potencialmente não confiáveis antes que o classificador produza erros de consequências graves.

1.2 Descrição do Problema

A aplicação de classificadores de ML a tarefas de cibersegurança, como a detecção de vulnerabilidades e intrusões, traz benefícios expressivos, mas também expõe duas fragilidades inter-relacionadas que as métricas tradicionais de avaliação (como a acurácia) não conseguem revelar:

1. **Safety (Segurança Funcional):** a garantia de que o sistema de ML opera corretamente dentro de seus parâmetros de projeto, sem produzir decisões de risco inaceitável. Normas como a IEC 61508 e a ISO 21448 (SOTIF) estabelecem requisitos para sistemas que incorporam componentes de ML (International Organization for Standardization, 2022).

2. **Cybersecurity (Cibersegurança):** a proteção contra manipulações maliciosas que buscam comprometer o comportamento do classificador. Ataques adversariais como FGSM (GOODFELLOW; SHLENS; SZEGEDY, 2014), PGD (MADRY et al., 2018), C&W (CARLINI; WAGNER, 2017) e DeepFool (MOOSAVI-DEZFOOLI; FAWZI; FROSSARD, 2016) evidenciam a fragilidade dos modelos de ML frente a adversários, inclusive em classificadores destinados à própria detecção de ameaças.

O problema central, portanto, é a ausência de mecanismos independentes capazes de indicar quando uma previsão de um classificador de ML não é confiável, seja porque os dados operacionais divergem dos dados de treinamento, seja porque o modelo está sob exploração adversarial. Essa limitação é particularmente sensível em classificadores de vulnerabilidade e de intrusão, em que uma decisão incorreta e silenciosa pode deixar passar uma ameaça real. Apesar de o SafeML se apresentar como uma resposta promissora a esse problema, persiste uma lacuna quanto à sua avaliação sistemática sob múltiplos tipos de ataque adversarial e em diferentes domínios de aplicação.

Cabe destacar que o SafeML detecta a *divergência distribucional* de forma agnóstica à sua causa: um mesmo aumento de SDD pode originar-se tanto de um desvio natural de distribuição quanto de uma manipulação adversarial. O framework, por si só, não separa essas duas categorias — ambas se manifestam como afastamento em relação aos perfis de treinamento. Uma distinção aproximada entre desvio de distribuição e ataque de evasão é explorada de forma heurística por meio do Score Unificado (Capítulo 4), que combina a magnitude do SDD com a queda de *accuracy* operacional, mas sua validação empírica permanece como trabalho futuro.

1.3 Justificativa

A relevância deste trabalho é sustentada por múltiplos fatores:

- O tema central é a confiabilidade da própria Inteligência Artificial: o artefato analisado é o classificador de ML, e a contribuição está em escrutinar o comportamento desses sistemas de IA para indicar quando suas decisões não são confiáveis, independentemente do domínio em que são aplicados;
- Normas internacionais emergentes (IEC 61508, ISO 21448, ISO/PAS 8800, DIN SPEC 92005) e regulamentações (EU AI Act, PL 2338/2023 no Brasil) demandam mecanismos de monitoramento de ML em sistemas críticos;
- O SafeML representa uma abordagem promissora por ser *model-agnostic*, computacionalmente eficiente e fundamentada em teoria estatística sólida;
- Existe uma lacuna na literatura quanto à avaliação sistemática do SafeML sob múltiplos tipos de ataque adversarial e em múltiplos domínios de aplicação.

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo geral deste trabalho é avaliar a eficácia do framework SafeML na análise da confiança de classificadores de Machine Learning, verificando se suas medidas de dissimila-

ridade estatística (SDD) são capazes de sinalizar previsões potencialmente não confiáveis, mesmo em modelos que apresentam bom desempenho em métricas tradicionais de avaliação. Como o objeto de estudo são sistemas de Inteligência Artificial, a análise recai sobre o próprio comportamento desses classificadores, e não sobre o domínio de aplicação isoladamente. Esse objetivo se desdobra em dois eixos complementares: (i) analisar a confiança de classificadores de ML quando expostos a desvios distribucionais e a ataques adversariais que exploram suas vulnerabilidades; e (ii) analisar a confiança de classificadores de ML voltados à cibersegurança, como os destinados à detecção de intrusões e de tráfego malicioso.

1.4.2 Objetivos Específicos

Como objetivos específicos, este trabalho propõe:

(a) implementar um framework SafeML em Python, com arquitetura limpa e pronta para produção — detalhada na Seção 4.2 —, que sirva de artefato central de análise de confiança de classificadores de IA;

(b) avaliar cinco medidas de dissimilaridade estatística: Kolmogorov-Smirnov, Wasserstein, Anderson-Darling, Kuiper e Cramér-von Mises;

(c) submeter o monitoramento a cinco tipos de perturbação adversarial que exploram a vulnerabilidade dos classificadores: FGSM, PGD, C&W, DeepFool e Ruído Gaussiano;

(d) validar a abordagem em quatro datasets representativos de diferentes domínios — MNIST, CIFAR-10, GTSRB e CICIDS2017, este último voltado à detecção de intrusões em redes;

(e) determinar *thresholds* ótimos para o sistema de confiança tri-nível (Verde/Amarelo/Vermelho);

(f) correlacionar as medidas de dissimilaridade estatística com a degradação de acurácia dos classificadores, avaliando a viabilidade do SafeML como camada independente de análise de confiança.

1.5 Metodologia

Este trabalho adota uma abordagem baseada em Design Science Research (DSR), combinando:

- **Revisão Sistemática da Literatura (RSL):** Levantamento do estado da arte em SafeML, ataques adversariais e monitoramento de safety em ML;
- **Implementação experimental:** Desenvolvimento de um framework em Python e execução de experimentos controlados;
- **Avaliação quantitativa:** Análise estatística dos resultados com testes de significância e correlação.

1.6 Organização do Trabalho

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta o referencial teórico, abordando os fundamentos de ML, safety, cybersecurity, ataques adversariais, medidas ECDF e o framework SafeML. O Capítulo 3 discute os trabalhos relacionados, comparando o SafeML com abordagens alternativas de monitoramento e garantia de segurança em ML. O Capítulo 4 detalha a metodologia adotada, incluindo a arquitetura do framework, os datasets, os modelos, os ataques adversariais selecionados e o ambiente experimental. O Capítulo 5 apresenta e discute os resultados experimentais em duas partes: (a) experimentos principais com configuração-padrão nos quatro datasets e (b) experimentos complementares com *features* CNN para o GTSRB, grade extendida de perturbações para o CIFAR-10 e comparação quantitativa com o detector MSP. Por fim, o Capítulo 6 apresenta as conclusões, contribuições, limitações e trabalhos futuros.

2 Referencial Teórico

Este capítulo apresenta os fundamentos teóricos necessários para a compreensão deste trabalho, organizados do conceito mais geral ao mais específico até culminar no framework SafeML. Inicia-se com uma visão geral sobre Machine Learning e classificadores (Seção 2.1), seguida pela discussão sobre safety e cybersecurity em sistemas críticos (Seção 2.2) e pela aplicação de Machine Learning à cibersegurança, em especial à detecção de intrusões e de vulnerabilidades (Seção 2.3). Em seguida, são apresentados os principais ataques adversariais (Seção 2.4), os conceitos de Função de Distribuição Empírica Acumulada (Seção 2.5), o framework SafeML (Seção 2.6) e, por fim, a explicabilidade no monitoramento de safety (Seção 2.7). Os trabalhos relacionados são discutidos no Capítulo 3.

2.1 Machine Learning e Classificadores

Machine Learning (ML) é um subcampo da Inteligência Artificial que se dedica ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados para cada tarefa específica (BISHOP, 2006). No contexto de classificação, o objetivo é construir uma função $f : \mathcal{X} \to \mathcal{Y}$ que mapeia um espaço de entrada $\mathcal{X}$ (atributos ou *features*) para um conjunto discreto de rótulos $\mathcal{Y} = \{1, 2, \ldots, K\}$, com K classes (GOODFELLOW; BENGIO; COURVILLE, 2016).

O processo de aprendizado supervisionado consiste em, dado um conjunto de treinamento $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, encontrar os parâmetros $\boldsymbol{\theta}$ que minimizem uma função de perda $\mathcal{L}(\boldsymbol{\theta})$, tipicamente a entropia cruzada para problemas de classificação:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}[y_i = k] \log p(y_i = k \mid \boldsymbol{x}_i; \boldsymbol{\theta}) \tag{2.1}$$

onde $\mathbb{1}[\cdot]$ é a função indicadora e $p(y = k \mid \boldsymbol{x}; \boldsymbol{\theta})$ é a probabilidade predita pelo modelo para a classe k.

2.1.1 Redes Neurais Convolucionais (CNNs)

Redes Neurais Convolucionais são uma classe de redes neurais profundas particularmente eficazes no processamento de dados com estrutura de grade, como imagens (LECUN; BENGIO; HINTON, 2015). A arquitetura de uma CNN consiste tipicamente em camadas convolucionais, camadas de *pooling* e camadas totalmente conectadas.

A operação de convolução em uma camada l é definida como:

$$\boldsymbol{z}_j^{(l)} = \sum_i \boldsymbol{w}_{ij}^{(l)} * \boldsymbol{a}_i^{(l-1)} + b_j^{(l)} \tag{2.2}$$

onde $\boldsymbol{w}_{ij}^{(l)}$ representa o filtro convolucional, $\boldsymbol{a}_i^{(l-1)}$ é o mapa de ativação da camada anterior, $*$ denota a operação de convolução e $b_j^{(l)}$ é o viés.

As operações fundamentais de uma CNN incluem: *batch normalization* (normaliza valores intermediários para manter estatísticas consistentes, acelerando o treinamento); *max pooling* (reduz dimensões espaciais tomando o valor máximo em regiões pequenas, aumentando robustez a variações de posição); *adaptive average pooling* (reduz qualquer tamanho de entrada para uma saída fixa calculando médias de regiões proporcionais); *dropout* (desativa aleatoriamente neurônios durante o treinamento para prevenir memorização excessiva); e *fine-tuning* (adaptar um modelo já treinado em um dataset grande para uma tarefa específica, aproveitando o conhecimento previamente adquirido).

A arquitetura **LeNet-5**, proposta por LeCun et al. (1998), foi uma das primeiras CNNs bem-sucedidas, projetada para reconhecimento de dígitos manuscritos. Consiste em duas camadas convolucionais seguidas de camadas de *pooling* e três camadas totalmente conectadas, totalizando aproximadamente 60.000 parâmetros treináveis.

A arquitetura **ResNet** (*Residual Network*), proposta por He et al. (2016), introduziu o conceito de conexões residuais (*skip connections*), permitindo o treinamento eficaz de redes com centenas de camadas. A ideia central é aprender funções residuais $\mathcal{F}(\boldsymbol{x}) = \mathcal{H}(\boldsymbol{x}) - \boldsymbol{x}$, de modo que a saída de cada bloco residual é:

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}, \{W_i\}) + \boldsymbol{x} \tag{2.3}$$

O ResNet-18, com 18 camadas e aproximadamente 11 milhões de parâmetros,

atingiu resultados estado-da-arte em benchmarks de classificação de imagens como o CIFAR-10 e ImageNet.

2.1.2 Random Forest e Multi-Layer Perceptron

O **Random Forest**, proposto por Breiman (2001), é um método de *ensemble* que combina múltiplas árvores de decisão treinadas com subconjuntos aleatórios dos dados e das *features*. A predição final é obtida por votação majoritária:

$$\hat{y} = \text{mode}\{h_t(\boldsymbol{x})\}_{t=1}^{T} \tag{2.4}$$

onde h_t representa a t-ésima árvore de decisão e T é o número total de árvores. O Random Forest é particularmente adequado para dados tabulares, como os encontrados em sistemas de detecção de intrusão em redes.

O **Multi-Layer Perceptron** (MLP) é uma rede neural totalmente conectada composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída (HAYKIN, 2009). Cada neurônio aplica uma transformação afim seguida de uma função de ativação não-linear σ:

$$\boldsymbol{a}^{(l)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)}) \tag{2.5}$$

2.1.3 Métricas de Avaliação

A avaliação de classificadores envolve métricas fundamentais derivadas da matriz de confusão (FAWCETT, 2006):

- **Acurácia** (*Accuracy*): fração de predições corretas sobre o total, $\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$;
- **Precisão** (*Precision*): fração de verdadeiros positivos entre os classificados como positivos, $\text{Prec} = \frac{TP}{TP+FP}$;
- **Revocação** (*Recall*): fração de verdadeiros positivos entre os realmente positivos, $\text{Rec} = \frac{TP}{TP+FN}$;

- **F1-Score**: média harmônica entre precisão e revocação, $F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$.

Para problemas multiclasse, essas métricas são calculadas de forma *macro* (média das métricas por classe) ou *weighted* (ponderada pelo número de amostras por classe). A curva ROC (*Receiver Operating Characteristic*) e a métrica AUC (*Area Under the Curve*) são amplamente utilizadas para avaliar a capacidade discriminativa de classificadores binários e podem ser estendidas para o cenário multiclasse.

2.2 Safety e Cybersecurity em Sistemas Críticos

A aplicação de componentes de ML em sistemas críticos de segurança introduz desafios fundamentais relacionados a dois conceitos distintos, mas inter-relacionados: *safety* (segurança funcional) e *cybersecurity* (cibersegurança) (MOHSENI et al., 2022).

2.2.1 Definições: Safety vs. Security

Safety (segurança funcional) refere-se à ausência de risco inaceitável de dano físico ou prejuízo à saúde das pessoas, direta ou indiretamente, como resultado do comportamento de um sistema (International Electrotechnical Commission, 2010). No contexto de ML, safety diz respeito à garantia de que o classificador opera corretamente dentro de suas condições de design e não produz saídas que possam levar a consequências perigosas.

Security (cibersegurança) refere-se à proteção de sistemas contra ameaças intencionais que visam comprometer a confidencialidade, integridade ou disponibilidade das informações e funcionalidades (BIGGIO; ROLI, 2018). Em sistemas ML, security envolve a proteção contra ataques adversariais que buscam manipular o comportamento do classificador.

Neste trabalho, essas duas dimensões são posteriormente operacionalizadas de forma quantitativa por meio do Score Unificado de *Safety-Cybersecurity* (Capítulo 4): a componente de *safety* é derivada do nível de confiança SafeML e da magnitude do SDD, enquanto a componente de *cybersecurity* combina a razão entre a *accuracy* operacional e a de *baseline* com um fator de distância SDD.

A interseção entre safety e security é particularmente relevante em sistemas ML:

um ataque de cibersegurança bem-sucedido (por exemplo, uma perturbação adversarial em uma imagem de entrada) pode comprometer diretamente a safety do sistema (por exemplo, levando um veículo autônomo a ignorar um sinal de parada).

2.2.2 Normas e Padrões

Diversas normas internacionais estabelecem requisitos para a segurança funcional de sistemas que incorporam componentes de ML:

- **IEC 61508** (International Electrotechnical Commission, 2010): Norma fundamental para segurança funcional de sistemas elétricos, eletrônicos e eletrônicos programáveis. Define os *Safety Integrity Levels* (SIL 1–4) e processos de ciclo de vida de segurança. Embora não aborde especificamente ML, seus princípios de integridade de segurança são aplicáveis.

- **ISO 21448 (SOTIF)** (International Organization for Standardization, 2022): *Safety of the Intended Functionality* — aborda riscos que surgem não de falhas de hardware ou software, mas de limitações inerentes ao design do sistema, incluindo limitações de algoritmos de ML em situações não previstas (*unknown unsafe scenarios*).

- **ISO/PAS 8800** (International Organization for Standardization, 2024): Especificação técnica para segurança e inteligência artificial no setor automotivo, fornecendo orientações sobre como integrar componentes de IA no ciclo de vida de segurança veicular.

- **DIN SPEC 92005** (Deutsches Institut für Normung, 2024): Especificação alemã sobre quantificação de incerteza em aprendizado de máquina, diretamente relevante para o monitoramento de confiança de classificadores.

2.2.3 Frameworks Regulatórios

O **NIST AI Risk Management Framework** (National Institute of Standards and Technology, 2023) fornece um quadro voluntário para gerenciamento de riscos em sistemas de IA, organizando as atividades em quatro funções: Governar, Mapear, Medir e

Gerenciar. O framework enfatiza a necessidade de monitoramento contínuo e avaliação de confiabilidade.

O **EU AI Act** (European Parliament and Council of the European Union, 2024) é o primeiro marco regulatório abrangente para inteligência artificial, adotado pela União Europeia em 2024. Classifica os sistemas de IA em categorias de risco (inaceitável, alto, limitado e mínimo) e impõe requisitos obrigatórios para sistemas de alto risco, incluindo avaliação de conformidade, documentação técnica e monitoramento pós-mercado.

No Brasil, o **PL 2338/2023** (Senado Federal do Brasil, 2023) propõe o Marco Legal da Inteligência Artificial, estabelecendo princípios, direitos e deveres para o desenvolvimento e uso de sistemas de IA, com atenção especial a sistemas de IA de alto risco que afetam direitos fundamentais.

2.3 Aprendizado de Máquina Aplicado à Cibersegurança

O aprendizado de máquina tornou-se uma ferramenta central na cibersegurança, sendo amplamente empregado em tarefas como a detecção de intrusões em redes, a identificação de *malware* e a análise de vulnerabilidades em software e em código-fonte. Nesses cenários, um classificador é treinado para distinguir comportamentos benignos de maliciosos a partir de atributos extraídos dos dados, como características de fluxo de rede ou padrões presentes no código. Conjuntos de dados como o CICIDS2017 (SHARAFALDIN; LASHKARI; GHORBANI, 2018), que reúne tráfego de rede realista rotulado com diferentes categorias de ataque, consolidaram-se como referência para o treinamento e a avaliação desses classificadores.

Ocorre que esses detectores são, eles próprios, artefatos de Inteligência Artificial e, portanto, herdam as limitações discutidas nas seções anteriores: degradam-se diante de dados que divergem da distribuição de treinamento e podem ser induzidos ao erro por adversários. Uma decisão incorreta e silenciosa de um classificador de vulnerabilidade ou de intrusão é especialmente grave, pois pode deixar passar uma ameaça real sem qualquer alerta. É justamente esse o objeto de estudo deste trabalho: analisar a confiança das

previsões dos próprios classificadores de ML, escrutinando seu comportamento de modo independente do domínio em que são aplicados. O framework SafeML, detalhado na Seção 2.6, oferece os mecanismos estatísticos para essa análise.

2.4 Ataques Adversariais em Machine Learning

Ataques adversariais são manipulações intencionais das entradas de um modelo de ML com o objetivo de induzi-lo a cometer erros de classificação (SZEGEDY et al., 2014; GOODFELLOW; SHLENS; SZEGEDY, 2014). Desde a descoberta de que redes neurais profundas são vulneráveis a perturbações imperceptíveis ao olho humano, este campo tem atraído significativa atenção da comunidade acadêmica (BIGGIO; ROLI, 2018; CHAKRABORTY et al., 2021).

2.4.1 Taxonomia de Ataques

Os ataques adversariais podem ser classificados segundo múltiplos critérios (XU et al., 2020):

- **Quanto ao conhecimento do atacante:**
  - *White-box* (caixa branca): o atacante tem acesso completo ao modelo, incluindo arquitetura, parâmetros e gradientes. Analogia: saber exatamente como um cofre foi construído e qual é a combinação;
  - *Black-box* (caixa preta): o atacante não tem acesso ao modelo interno, apenas pode consultar o modelo como um oráculo (enviar entrada e receber saída). Analogia: tentar abrir o cofre apenas tentando combinações e observando se abre;
  - *Gray-box*: o atacante possui conhecimento parcial do modelo (ex.: conhece a arquitetura mas não os pesos exatos).
- **Quanto ao momento do ataque:**
  - *Evasion* (tempo de inferência): manipulação das entradas durante a operação do modelo;

  - *Poisoning* (tempo de treinamento): contaminação dos dados de treinamento;
  - *Extraction/Inference*: roubo do modelo ou extração de informações sensíveis.

- **Quanto ao objetivo:**
  - *Untargeted*: o objetivo é causar qualquer classificação incorreta;
  - *Targeted*: o objetivo é induzir uma classificação específica escolhida pelo atacante.

O MITRE ATLAS (*Adversarial Threat Landscape for AI Systems*) (MITRE Corporation, 2025) fornece uma taxonomia abrangente de ameaças adversariais a sistemas de IA, análoga ao MITRE ATT&CK para cibersegurança convencional.

2.4.2 Ataques de Evasão

FGSM — Fast Gradient Sign Method

O FGSM, proposto por Goodfellow, Shlens e Szegedy (2014), é um ataque de passo único que gera exemplos adversariais adicionando uma perturbação proporcional ao sinal do gradiente da função de perda em relação à entrada:

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y)) \tag{2.6}$$

onde ϵ controla a magnitude da perturbação e $\nabla_{\boldsymbol{x}} \mathcal{L}$ é o gradiente da perda em relação à entrada. A simplicidade computacional do FGSM (requer apenas um passo de propagação *forward* e um de *backward*) o torna amplamente utilizado como *baseline* de ataque.

PGD — Projected Gradient Descent

O PGD, proposto por Madry et al. (2018), é uma extensão iterativa do FGSM que aplica múltiplos passos de perturbação, projetando o resultado de volta ao espaço permitido $\mathcal{B}_\epsilon(\boldsymbol{x})$ a cada iteração:

$$\boldsymbol{x}^{(t+1)} = \Pi_{\mathcal{B}_\epsilon(\boldsymbol{x})} \left[\boldsymbol{x}^{(t)} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}^{(t)}, y)) \right] \tag{2.7}$$

onde α é o tamanho do passo, Π é a operação de projeção na bola L_∞ de raio ϵ, e $\boldsymbol{x}^{(0)}$ é inicializado aleatoriamente dentro de $\mathcal{B}_\epsilon(\boldsymbol{x})$. O PGD é considerado o ataque de primeira ordem mais forte e é frequentemente usado para avaliar robustez adversarial (TRAMÈR et al., 2020).

C&W — Carlini & Wagner

O ataque C&W, proposto por Carlini e Wagner (2017), formula a geração de exemplos adversariais como um problema de otimização:

$$\min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_p + c \cdot g(\boldsymbol{x} + \boldsymbol{\delta}) \tag{2.8}$$

onde $\boldsymbol{\delta}$ é a perturbação, c é uma constante de balanceamento e $g(\cdot)$ é uma função objetivo que é negativa quando o ataque é bem-sucedido. A variante L_2 minimiza a norma euclidiana da perturbação, sendo capaz de encontrar perturbações significativamente menores que o FGSM e o PGD.

DeepFool

O DeepFool, proposto por Moosavi-Dezfooli, Fawzi e Frossard (2016), computa a perturbação mínima necessária para alterar a classificação de uma amostra, iterativamente linearizando o classificador e projetando a amostra para o hiperplano de decisão mais próximo:

$$\boldsymbol{\delta}^{(t)} = -\frac{f_k(\boldsymbol{x}^{(t)})}{\|\nabla f_k(\boldsymbol{x}^{(t)})\|_2^2} \nabla f_k(\boldsymbol{x}^{(t)}) \tag{2.9}$$

onde f_k é a diferença entre a saída da classe predita e a classe mais próxima. O DeepFool produz perturbações mínimas, sendo útil para medir a robustez intrínseca de classificadores.

2.4.3 Ataques de Envenenamento (Data Poisoning)

Ataques de envenenamento contaminam os dados de treinamento para influenciar o comportamento do modelo treinado (BIGGIO; NELSON; LASKOV, 2012). Diferentemente

dos ataques de evasão, que atuam no tempo de inferência, os ataques de envenenamento corrompem o processo de aprendizado. Shafahi et al. (2018) demonstraram ataques de envenenamento *clean-label*, nos quais os dados injetados mantêm seus rótulos corretos, tornando a detecção particularmente difícil.

2.4.4 Mecanismos de Defesa

As defesas contra ataques adversariais incluem três abordagens principais:

1. **Treinamento adversarial** (MADRY et al., 2018): incorporação de exemplos adversariais no conjunto de treinamento para aumentar a robustez do modelo;
2. **Defesas certificadas** (WONG; KOLTER, 2018; COHEN; ROSENFELD; KOLTER, 2019): garantias formais de robustez dentro de um raio de perturbação, como a técnica de *randomized smoothing*;
3. **Detecção** (HENDRYCKS; GIMPEL, 2017): mecanismos que identificam entradas adversariais ou fora da distribuição em tempo de execução — esta é a abordagem adotada pelo SafeML.

2.5 Função de Distribuição Empírica Acumulada (ECDF)

A Função de Distribuição Empírica Acumulada (ECDF, do inglês *Empirical Cumulative Distribution Function*) é um estimador não-paramétrico da função de distribuição acumulada (CDF) de uma variável aleatória, construído a partir de observações amostrais (WASSERMAN, 2004).

A ECDF responde a uma pergunta fundamental: para um determinado valor x, qual fração dos dados é menor ou igual a x? Plotando essa fração para todos os valores possíveis de x, obtém-se uma curva escada que revela a distribuição completa dos dados — sobe rapidamente em regiões densas, lentamente em regiões esparsas. Quando duas ECDFs diferem significativamente, os conjuntos de dados subjacentes possuem distribuições distintas; essa propriedade é o fundamento do monitoramento SafeML.

2.5.1 Definição e Propriedades

Dada uma amostra $X_1, X_2, \ldots, X_n$ de variáveis aleatórias independentes e identicamente distribuídas (i.i.d.), a ECDF é definida como:

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[X_i \leq x] \tag{2.10}$$

onde $\mathbb{1}[\cdot]$ é a função indicadora. A ECDF é uma função escada que aumenta em $\frac{1}{n}$ em cada ponto observado.

A ECDF possui propriedades importantes: é um estimador não-viesado de $F(x)$ para cada ponto x; é consistente em sentido forte; e sua variância em qualquer ponto é $\text{Var}[\hat{F}_n(x)] = \frac{F(x)(1-F(x))}{n}$.

A escolha da ECDF em detrimento de estatísticas paramétricas como média e variância é motivada pela necessidade de capturar a forma completa da distribuição. Medidas paramétricas resumem a distribuição em poucos números e pressupõem uma forma específica (e.g., gaussiana), podendo deixar passar anomalias que ocorrem nas caudas ou em regiões específicas do intervalo. A ECDF é não-paramétrica e captura o comportamento em cada ponto do domínio sem suposições — um requisito essencial para monitoramento de safety, onde ataques adversariais podem deslocar apenas uma porção específica da distribuição de entrada sem alterar sua média.

2.5.2 Teorema de Glivenko-Cantelli

O Teorema de Glivenko-Cantelli (GLIVENKO, 1933) garante a convergência uniforme da ECDF para a CDF verdadeira:

$$\sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0 \quad \text{quando } n \to \infty \tag{2.11}$$

Este resultado fundamental assegura que, com amostras suficientemente grandes, a ECDF é uma aproximação confiável da distribuição verdadeira, justificando seu uso como representação da distribuição dos dados de treinamento e operacionais no framework SafeML.

2.5.3 Medidas de Distância Baseadas em ECDF

Medidas de distância entre distribuições baseadas em ECDFs permitem quantificar a divergência entre duas amostras sem assumir uma forma paramétrica para as distribuições subjacentes. As cinco medidas utilizadas neste trabalho são apresentadas a seguir.

Kolmogorov-Smirnov (KS)

A distância de Kolmogorov-Smirnov (KOLMOGOROV, 1933; SMIRNOV, 1948) é definida como a suprema diferença absoluta entre duas ECDFs:

$$D_{KS} = \sup_x |\hat{F}_{\text{ref}}(x) - \hat{F}_{\text{op}}(x)| \tag{2.12}$$

O teste KS de duas amostras possui distribuição assintótica conhecida, permitindo o cálculo de p-valores. A estatística KS é particularmente sensível a diferenças na localização das distribuições.

Kuiper (K)

A distância de Kuiper (KUIPER, 1960) é a soma dos desvios máximos positivo e negativo:

$$V = D^+ + D^- = \sup_x[\hat{F}_{\text{ref}}(x) - \hat{F}_{\text{op}}(x)] + \sup_x[\hat{F}_{\text{op}}(x) - \hat{F}_{\text{ref}}(x)] \tag{2.13}$$

Ao contrário da estatística KS, a estatística de Kuiper é igualmente sensível em toda a faixa da distribuição, sendo particularmente útil para detectar diferenças nas caudas.

Anderson-Darling (AD)

A distância de Anderson-Darling (ANDERSON; DARLING, 1952) é uma integral ponderada das diferenças quadráticas entre as ECDFs, com pesos que enfatizam as caudas da distribuição:

$$A^2 = n \int_{-\infty}^{+\infty} \frac{[\hat{F}_{\text{ref}}(x) - \hat{F}_{\text{op}}(x)]^2}{F(x)(1 - F(x))}\, dF(x) \tag{2.14}$$

A ponderação por $[F(x)(1-F(x))]^{-1}$ atribui maior importância às diferenças nas

caudas, tornando o teste AD mais sensível a desvios nas extremidades da distribuição.

Wasserstein (W)

A distância de Wasserstein (VASERSTEIN, 1969), também conhecida como *Earth Mover's Distance* (EMD), é definida como a integral da diferença absoluta entre as ECDFs:

$$W_1 = \int_{-\infty}^{+\infty} |\hat{F}_{\text{ref}}(x) - \hat{F}_{\text{op}}(x)|\, dx \tag{2.15}$$

A distância de Wasserstein quantifica o "custo de transporte" necessário para transformar uma distribuição na outra, sendo sensível à magnitude total das diferenças entre as distribuições.

Cramér-von Mises (CvM)

A distância de Cramér-von Mises (CRAMÉR, 1928) é definida como a integral das diferenças quadráticas entre as ECDFs:

$$W^2 = \int_{-\infty}^{+\infty} [\hat{F}_{\text{ref}}(x) - \hat{F}_{\text{op}}(x)]^2\, dF(x) \tag{2.16}$$

A estatística CvM atribui peso uniforme a todas as partes da distribuição, capturando diferenças na forma geral das distribuições sem enfatizar regiões específicas.

As cinco medidas diferem na forma como ponderam os desvios entre distribuições: o KS capta o maior desvio pontual, sendo mais sensível a diferenças concentradas em um único ponto; o Kuiper soma os desvios máximos positivo e negativo, sendo igualmente sensível em todo o intervalo, incluindo as caudas; o Wasserstein quantifica o "custo de transporte" necessário para transformar uma distribuição na outra, capturando deslocamentos globais difusos; o CvM pondera diferenças quadraticamente, amplificando os maiores desvios; e o AD enfatiza especialmente as caudas, sendo mais sensível a anomalias nos valores extremos. Nos experimentos deste trabalho, KS e Kuiper demonstraram maior robustez entre diferentes domínios.

A Tabela 2.1 resume as características das cinco medidas de distância utilizadas.

Tabela 2.1: Resumo das medidas de distância estatística utilizadas.

Medida	Abreviação	Sensibilidade	P-valor
Kolmogorov-Smirnov	KS	Desvios globais	Sim
Wasserstein	W	Deslocamento de distribuição	Não
Anderson-Darling	AD	Caudas da distribuição	Sim
Kuiper	K	Desvios globais (cíclico)	Não
Cramér-von Mises	CvM	Desvios quadráticos	Não

2.6 SafeML: Safety Monitoring of Machine Learning Classifiers

O SafeML é um framework de monitoramento de segurança para classificadores de Machine Learning proposto por Aslansefat et al. (2020). Sua premissa fundamental é que a degradação na performance de um classificador pode ser detectada por meio de mudanças na distribuição estatística dos dados de entrada em relação à distribuição observada durante o treinamento.

2.6.1 Conceito e Arquitetura

O SafeML opera em duas fases distintas, ilustradas no fluxograma da Figura 2.1:

1. **Fase de Design (offline):** Durante o treinamento, são extraídas ECDFs de referência para cada combinação de classe e *feature*, formando um perfil estatístico do comportamento esperado do classificador;

2. **Fase Operacional (online):** Durante a operação, os dados de entrada são continuamente comparados com os perfis de referência por meio de medidas de dissimilaridade estatística (SDD).

A hipótese central do SafeML é que, quando os dados operacionais diferem significativamente dos dados de treinamento (como ocorre durante ataques adversariais ou mudanças de distribuição), as medidas SDD aumentam de forma correlacionada com a degradação da *accuracy* do classificador.

É importante delimitar o escopo desse mecanismo. O SafeML monitora a *divergência distribucional em tempo de inferência*, comparando os dados operacionais com

os perfis ECDF de referência extraídos do conjunto de treinamento confiável. Tanto ataques de evasão — como os adversariais avaliados neste trabalho — quanto mudanças naturais de distribuição enquadram-se nesse escopo, pois ambos deslocam a distribuição operacional em relação à de referência. Já ataques em tempo de treinamento, notadamente o *Data Poisoning* — que corrompe os próprios dados a partir dos quais os perfis de referência são construídos —, estão fora do escopo do monitoramento SafeML, uma vez que comprometem a própria linha de base contra a qual a divergência é medida; sua detecção permanece como trabalho futuro.

2.6.2 Fase de Design: ECDF Profiling

Na fase de design, para cada classe $k \in \{1, \ldots, K\}$ e cada *feature* $j \in \{1, \ldots, d\}$, é construída uma ECDF de referência $\hat{F}^{\text{ref}}_{k,j}$ a partir dos dados de treinamento:

$$\hat{F}^{\text{ref}}_{k,j}(x) = \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} \mathbb{1}[x_{i,j} \leq x] \tag{2.17}$$

onde $\mathcal{D}_k = \{\boldsymbol{x}_i \in \mathcal{D} : y_i = k\}$ é o subconjunto de dados da classe k e $x_{i,j}$ é o j-ésimo atributo da amostra $\boldsymbol{x}_i$.

Cabe destacar uma limitação dessa formulação: as ECDFs de referência são construídas de forma *marginal*, isto é, uma distribuição univariada independente por *feature* j. Essa abordagem, herdada do SafeML original (ASLANSEFAT et al., 2020; ASLANSEFAT et al., 2021), não captura dependências conjuntas entre *features* — duas distribuições operacionais podem preservar todas as marginais e, ainda assim, diferir na estrutura de correlação. A dissimilaridade agregada sobre marginais tende, portanto, a subestimar divergências puramente multivariadas. Esse efeito é parcialmente mitigado ao operar o SafeML sobre *features* de alto nível extraídas por CNNs (Seção 5.6.4), nas quais grande parte da estrutura relevante já foi condensada em cada dimensão; a modelagem explícita de dependências conjuntas (por exemplo, via cópulas ou distâncias multivariadas) permanece como trabalho futuro.

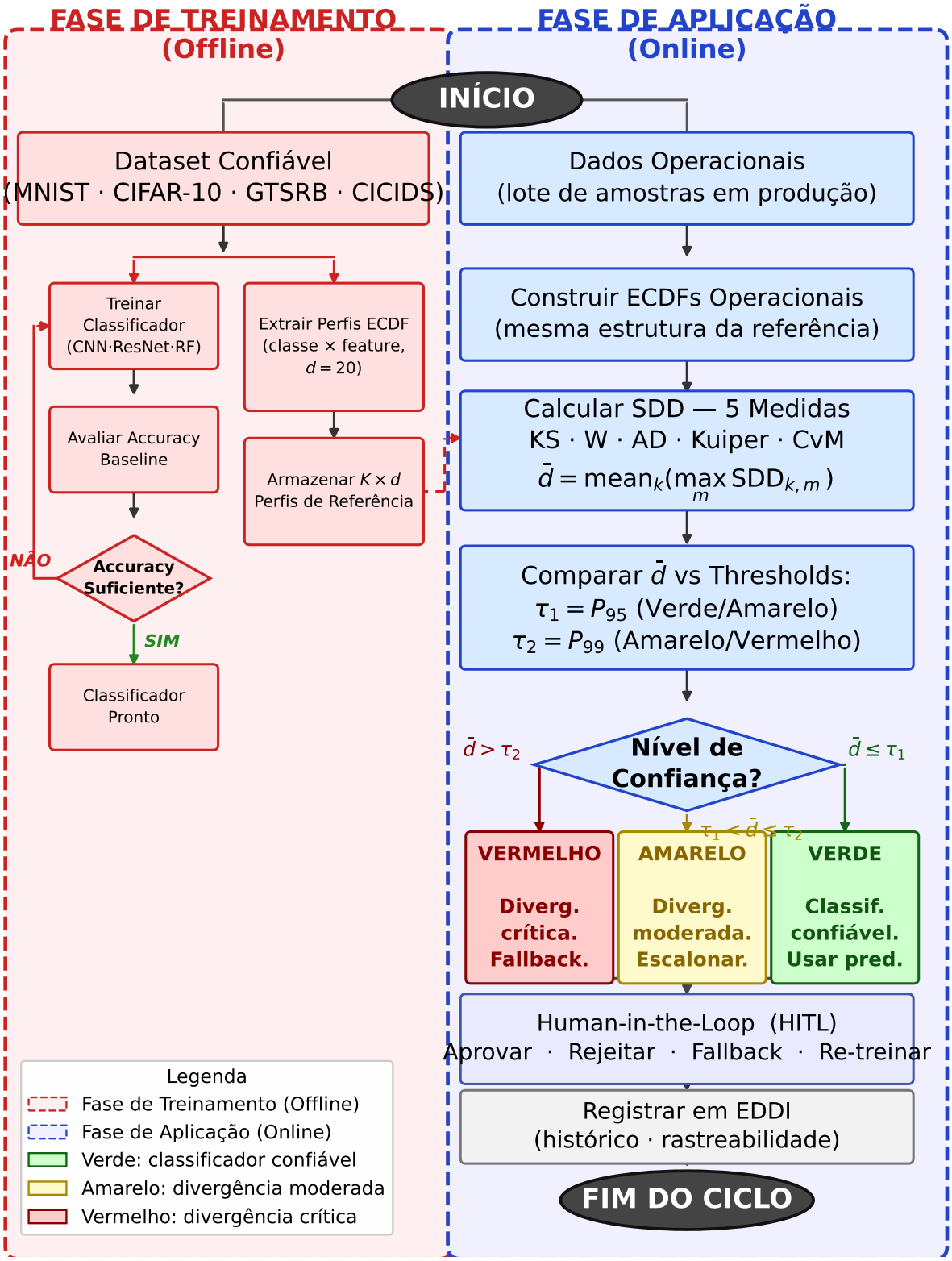

Figura 2.1: Fluxograma do framework SafeML implementado neste trabalho. A Fase de Treinamento (*offline*) e a extração dos perfis ECDF de referência derivam do *dataset* confiável e são independentes do resultado do treinamento do classificador — a extração dos perfis ocorre em paralelo ao ajuste do modelo, e não como etapa condicionada à sua *accuracy*. A Fase de Aplicação (*online*) realiza o monitoramento contínuo com o sistema de confiança tri-nível Verde/Amarelo/Vermelho. Adaptado de Aslansefat et al. (2020).

2.6.3 Fase Operacional: Runtime Monitoring

Durante a operação, para cada lote de dados operacionais classificados como pertencentes à classe k, é construída uma ECDF operacional $\hat{F}^{\text{op}}_{k,j}$ e calculada a distância em relação ao perfil de referência:

$$\text{SDD}_{k,j} = d(\hat{F}^{\text{ref}}_{k,j}, \hat{F}^{\text{op}}_{k,j}) \tag{2.18}$$

onde $d(\cdot, \cdot)$ é uma das medidas de distância apresentadas na Seção 2.5.3.

2.6.4 Sistema de Confiança Tri-nível

O SafeML define um sistema de confiança com três níveis, análogo a um semáforo (ASLANSEFAT et al., 2020):

- **Verde (Green)**: SDD $\leq \tau_1$ — a distribuição operacional é estatisticamente consistente com a distribuição de treinamento. O classificador opera com alta confiança;
- **Amarelo (Yellow)**: $\tau_1 <$ SDD $\leq \tau_2$ — há divergência moderada. Recomenda-se cautela e possível degradação graceful;
- **Vermelho (Red)**: SDD $> \tau_2$ — divergência significativa detectada. A confiabilidade do classificador é comprometida e um mecanismo de fallback deve ser ativado.

onde τ_1 e τ_2 são limiares (*thresholds*) determinados durante a fase de design, tipicamente baseados em percentis da distribuição de SDD observada em dados de validação.

2.6.5 Evolução do SafeML

O framework SafeML tem evoluído significativamente desde sua concepção:

- **SafeML I** (ASLANSEFAT et al., 2020): versão original com foco em medidas KS e CvM aplicadas a dados tabulares;
- **SafeML II** (ASLANSEFAT et al., 2021): extensão para classificação de imagens (sinais de trânsito), introduzindo a análise de *features* extraídas por CNNs e a utilização de bootstrap p-values;

- **SafeDrones** (ASLANSEFAT et al., 2022): aplicação em sistemas de UAVs com *Executable Digital Dependable Identities* (EDDIs), integrando monitoramento SafeML com modelos de confiabilidade;

- **SMILE** (ASLANSEFAT et al., 2023): extensão para definição de níveis de integridade (*integrity levels*) para ML de forma análoga ao SIL da IEC 61508.

2.7 Explainable AI (XAI) e Monitoramento de Safety

A Inteligência Artificial Explicável (*Explainable AI* — XAI) busca tornar os processos de decisão de modelos de ML compreensíveis para humanos, sendo um requisito cada vez mais relevante para sistemas críticos de segurança (MOHSENI et al., 2022).

No contexto do monitoramento SafeML, a explicabilidade manifesta-se em três dimensões:

1. **Explicabilidade da detecção:** As medidas de distância estatística (SDD) fornecem uma justificativa quantitativa e interpretável para a avaliação de segurança — um operador humano pode compreender que "a distribuição dos dados operacionais diverge da distribuição de treinamento em X unidades de distância KS", o que é mais interpretável do que a saída de um detector neural opaco;

2. **Explicabilidade do sistema tri-nível:** O sistema Verde/Amarelo/Vermelho oferece uma abstração intuitiva que traduz métricas estatísticas complexas em ações operacionais claras, facilitando a integração com procedimentos de *Human-in-the-Loop* (HITL);

3. **Rastreabilidade via EDDI:** Os *Executable Digital Dependable Identities* mantêm um registro completo da proveniência do modelo, dos perfis de referência e do histórico de avaliações, proporcionando auditabilidade e rastreabilidade exigidas por normas como a IEC 61508 (International Electrotechnical Commission, 2010) e o EU AI Act (European Parliament and Council of the European Union, 2024).

A natureza *model-agnostic* do SafeML é particularmente vantajosa sob a perspectiva de XAI: como o monitoramento opera sobre as distribuições dos dados de entrada e não sobre as representações internas do modelo, as explicações são independentes da complexidade do classificador subjacente.

3 Trabalhos Relacionados

Este capítulo apresenta abordagens complementares e alternativas ao SafeML para o monitoramento e a garantia de segurança de sistemas baseados em ML.

3.1 AMLAS: Assurance of Machine Learning for Autonomous Systems

O AMLAS, proposto por Paterson et al. (2025), é um framework de garantia (*assurance*) para ML em sistemas autônomos que define um processo sistemático composto por seis estágios: definição de requisitos de ML, obtenção de dados, treinamento do modelo, verificação do modelo, implantação e operação. Enquanto o AMLAS foca no ciclo de vida completo da garantia, o SafeML concentra-se especificamente na fase de monitoramento em tempo de execução, podendo ser integrado como componente do estágio operacional do AMLAS.

3.2 EDDI: Executable Digital Dependable Identities

As EDDIs, exploradas por Nascimento, Aslansefat e Papadopoulos (2024), são identidades digitais executáveis que acompanham componentes de software ao longo de seu ciclo de vida, encapsulando informações sobre dependabilidade, incluindo resultados de testes, perfis de confiabilidade e modelos de falha. O SafeDrones (ASLANSEFAT et al., 2022) demonstrou a integração do SafeML com EDDIs em sistemas de UAVs, onde o monitoramento estatístico alimenta dinamicamente o modelo de confiabilidade do sistema.

Neste trabalho, os EDDIs são integrados de forma prática ao framework SafeML: cada classificador treinado recebe um EDDI contendo sua proveniência de treinamento (dataset, hiperparâmetros, métricas de *baseline*), os perfis ECDF de referência, restrições operacionais (limites de latência, *accuracy* mínima, *thresholds*) e o histórico de avaliações de segurança. O EDDI é serializado em formato JSON, permitindo armazenamento,

transmissão e auditoria. Durante a operação, cada avaliação SafeML é registrada no EDDI, possibilitando a verificação de conformidade operacional e o cálculo de indicadores de confiabilidade ao longo do tempo.

3.3 Conformal Prediction para Safety

A *Conformal Prediction* (VOVK; GAMMERMAN; SHAFER, 2005; ANGELOPOULOS; BATES, 2023) é uma abordagem que fornece conjuntos de predição com garantias de cobertura frequentista. Ao invés de produzir uma única classe predita, a *conformal prediction* gera um conjunto de classes com probabilidade garantida de conter a classe verdadeira. Embora ambas as abordagens (SafeML e *conformal prediction*) busquem quantificar a confiança das predições, diferem fundamentalmente: o SafeML monitora mudanças na distribuição dos dados de entrada, enquanto a *conformal prediction* fornece garantias sobre a saída do modelo.

3.4 Detecção de Amostras Fora da Distribuição (OOD)

A detecção de amostras fora da distribuição (*Out-of-Distribution Detection*) é um campo relacionado que visa identificar entradas provenientes de uma distribuição distinta da dos dados de treinamento (HENDRYCKS; GIMPEL, 2017; YANG et al., 2024). Liang, Li e Srikant (2018) propuseram o ODIN, que utiliza perturbação de temperatura e pré-processamento de entrada para melhorar a detecção OOD. Diferentemente dessas abordagens que tipicamente requerem acesso às saídas internas do modelo (logits, *features* intermediárias), o SafeML opera exclusivamente sobre as distribuições dos dados de entrada, sendo verdadeiramente *model-agnostic*.

3.5 Quantificação de Incerteza

Métodos de quantificação de incerteza como *Monte Carlo Dropout* (GAL; GHAHRAMANI, 2016) e *Deep Ensembles* (LAKSHMINARAYANAN; PRITZEL; BLUNDELL,

2017) estimam a incerteza do modelo em cada predição individual (ABDAR et al., 2021). Essas abordagens são complementares ao SafeML: enquanto a quantificação de incerteza avalia a confiança predição-a-predição, o SafeML monitora mudanças de distribuição em nível de lote, oferecendo uma visão mais ampla da confiabilidade operacional.

3.6 Monitoramento em Tempo de Execução com Padrões de Ativação

Cheng, Nührenberg e Yasuoka (2019) propuseram o monitoramento de padrões de ativação de neurônios como mecanismo de detecção de entradas anômalas. Henzinger, Lukina e Schilling (2020) estenderam essa abordagem com abstrações de caixas (*box abstractions*) para representar o comportamento esperado da rede. Essas técnicas, porém, requerem acesso à estrutura interna da rede, diferindo da abordagem *model-agnostic* do SafeML.

3.7 Comparação das Abordagens

A Tabela 3.1 compara as principais características das abordagens apresentadas.

Tabela 3.1: Comparação entre abordagens de monitoramento e garantia de segurança para ML.

Abordagem	Model-agnostic	Runtime	Ciclo de vida	Garantia formal
SafeML	Sim	Sim	Parcial	Não
AMLAS	N/A	Não	Completo	Não
Conformal Prediction	Sim	Sim	Não	Sim
Detecção OOD	Não	Sim	Não	Não
Quant. Incerteza	Não	Sim	Não	Parcial
Monit. Ativação	Não	Sim	Não	Não

O SafeML destaca-se por combinar monitoramento em tempo de execução com a propriedade de ser *model-agnostic* na sua configuração-padrão (monitoramento sobre pixels/features brutas de entrada), permitindo sua aplicação a qualquer classificador sem modificação interna. Em configurações avançadas, como a extração de *features* de camadas intermediárias da CNN (vetor de 512 dimensões do `avgpool` da ResNet-18, avaliada no Capítulo 5), o acesso à estrutura da rede é necessário — mas isso é opcional e es-

pecífico ao domínio, não um requisito fundamental do framework. Esta flexibilidade é particularmente valiosa em ambientes industriais onde modelos heterogêneos coexistem.

4 Metodologia

Este capítulo detalha a metodologia adotada para a avaliação do framework SafeML. A apresentação segue uma sequência que parte das tecnologias e ferramentas empregadas, passa pela base de dados utilizada, pelos modelos de classificação treinados e pela etapa de predição e coleta de dados com o SafeML, e culmina na descrição da análise dos dados. Inicia-se pela abordagem metodológica que orienta o estudo.

4.1 Abordagem Metodológica

Este trabalho adota a metodologia de *Design Science Research* (DSR) (HEVNER et al., 2004; PEFFERS et al., 2007). O DSR é uma abordagem de pesquisa em Ciência da Computação que se distingue de pesquisas puramente analíticas por *construir e avaliar artefatos* — softwares, modelos, frameworks, algoritmos — para resolver problemas práticos identificados. Enquanto uma pesquisa empírica observa fenômenos existentes, o DSR cria novas soluções e as valida experimentalmente. Neste trabalho, o artefato central é o framework SafeML implementado em Python, e a avaliação é realizada por meio de experimentação controlada com datasets e ataques reais. A estruturação do estudo experimental segue as diretrizes propostas por Wohlin et al. (2012), particularmente no que diz respeito à definição de métricas de avaliação, controle de variáveis e análise estatística dos resultados.

O processo de pesquisa seguiu as etapas propostas por Peffers et al. (2007):

1. **Identificação do problema:** Necessidade de mecanismos de monitoramento de segurança para classificadores ML em ambientes potencialmente hostis;

2. **Definição dos objetivos:** Avaliar a eficácia do SafeML sob múltiplos ataques adversariais e medidas de distância;

3. **Design e desenvolvimento:** Implementação do framework com arquitetura limpa e princípios SOLID;

4. **Demonstração:** Execução de experimentos com quatro datasets e cinco tipos de ataque;

5. **Avaliação:** Análise estatística dos resultados, incluindo correlações, testes de significância e curvas ROC;

6. **Comunicação:** Documentação dos resultados nesta monografia.

Complementarmente, foi conduzida uma Revisão Sistemática da Literatura (RSL) para o levantamento do estado da arte em SafeML, ataques adversariais e monitoramento de safety em ML, seguindo o protocolo PRISMA.

4.2 Arquitetura do Framework

O framework SafeML foi implementado em Python seguindo os princípios de *Clean Architecture* (MARTIN, 2017) e *Domain-Driven Design* (DDD) (EVANS, 2003), organizando o código em quatro camadas com separação clara de responsabilidades.

4.2.1 Princípios de Design

A implementação segue rigorosamente os princípios SOLID:

- **Single Responsibility:** cada classe possui uma única responsabilidade — por exemplo, `KolmogorovSmirnovDistance` apenas computa a distância KS;
- **Open/Closed:** o framework é extensível via interfaces abstratas — novas medidas de distância ou adaptadores de ataque podem ser adicionados sem modificar o código existente;
- **Liskov Substitution:** todos os *wrappers* de classificadores (PyTorch, scikit-learn) são intercambiáveis por meio da interface `ClassifierWrapper`;
- **Interface Segregation:** interfaces focadas e especializadas — `DistanceCalculator`, `ECDFBuilder`, `SafetyMonitor`;
- **Dependency Inversion:** os casos de uso (*use cases*) dependem de abstrações (interfaces), não de implementações concretas.

4.2.2 Camadas da Arquitetura

A arquitetura é organizada em quatro camadas:

1. **Camada Core (Domínio):** Contém as entidades de domínio (`ECDFProfile`, `DistanceResult`, `SafetyAssessment`, `ClassifierReport`), objetos de valor (`Threshold`, `ConfidenceLevel`, `DistanceMetric`) e interfaces abstratas. Esta camada não possui dependências de bibliotecas externas;

2. **Camada Application (Casos de Uso):** Implementa a lógica de negócio por meio de cinco casos de uso: `TrainAndProfile` (extração de perfis ECDF), `RuntimeMonitor` (monitoramento em tempo de execução), `AdversarialEvaluation` (avaliação sob ataques), `ComparativeAnalysis` (comparação entre medidas) e `FullPipeline` (orquestração completa);

3. **Camada Infrastructure (Adaptadores):** Fornece implementações concretas das interfaces, incluindo as cinco medidas de distância, o construtor de ECDFs baseado em NumPy, adaptadores de ataque (IBM ART e Foolbox), *wrappers* de classificadores (PyTorch e scikit-learn), carregadores de datasets e geradores de relatórios;

4. **Camada Presentation:** Interface de linha de comando (CLI), API REST (FastAPI) e gerenciamento de configuração via arquivos YAML.

O atributo "pronto para produção" declarado nos objetivos concretiza-se em três aspectos de engenharia, detalhados na Seção 4.9: empacotamento reproduzível via *Docker* e *docker-compose*; exposição do monitoramento como API REST (FastAPI), além da CLI de experimentação; e uma suíte de testes automatizados sobre as entidades de domínio, as medidas de distância e os casos de uso. Sob a ótica do *Design Science Research*, o framework constitui o artefato de software central deste trabalho, concebido como componente reutilizável de monitoramento de confiança de classificadores — e não apenas como um *script* de experimentação.

4.3 Base de Dados

Foram selecionados quatro datasets representativos de diferentes domínios de aplicação de ML em sistemas críticos.

A avaliação abrange dois domínios estruturalmente distintos — visão computacional (MNIST, CIFAR-10, GTSRB) e tráfego de rede (CICIDS2017) —, o que torna necessário explicitar o que muda e o que permanece constante entre eles. O núcleo do método SafeML é idêntico nos dois casos: extração de perfis ECDF por classe e por *feature*, cálculo das cinco medidas de dissimilaridade e o sistema de confiança tri-nível. O que varia entre os domínios é apenas: (i) o classificador subjacente (CNN/ResNet para imagem; Random Forest para tráfego de rede); (ii) a origem das *features* monitoradas (pixels ou ativações de CNN, no caso de imagem; *features* de fluxo de rede, no caso do CICIDS2017); e (iii) o gerador de perturbação (ataques adversariais baseados em gradiente para imagem; ruído Gaussiano como *proxy* de evasão para o modelo não-diferenciável). Essa separação evidencia que a generalização do SafeML entre domínios não depende de reformular o mecanismo de monitoramento, mas apenas de adaptar essas três interfaces.

4.3.1 MNIST

O dataset MNIST (*Modified National Institute of Standards and Technology*) (LECUN et al., 1998) consiste em 70.000 imagens em escala de cinza de dígitos manuscritos (0–9), com resolução 28×28 pixels. É dividido em 60.000 imagens de treinamento e 10.000 de teste. Embora seja um benchmark relativamente simples, o MNIST é amplamente utilizado como *baseline* para avaliação de ataques adversariais.

4.3.2 CIFAR-10

O dataset CIFAR-10 (KRIZHEVSKY; HINTON, 2009) contém 60.000 imagens coloridas (RGB) de resolução 32×32 pixels, distribuídas em 10 classes (avião, automóvel, pássaro, gato, cervo, cão, sapo, cavalo, navio, caminhão). São 50.000 imagens de treinamento e 10.000 de teste. O CIFAR-10 apresenta maior complexidade visual que o MNIST, com variabilidade intra-classe significativa.

4.3.3 GTSRB

O *German Traffic Sign Recognition Benchmark* (GTSRB) (STALLKAMP et al., 2011) contém mais de 50.000 imagens de sinais de trânsito em 43 classes. As imagens possuem tamanhos variáveis (de 15×15 a 250×250 pixels), capturadas em condições reais de iluminação e ângulo. Este dataset é diretamente relevante para sistemas de veículos autônomos, tornando-o essencial para a avaliação de safety.

4.3.4 CICIDS2017

O *Canadian Institute for Cybersecurity Intrusion Detection System 2017* (CICIDS2017) (SHARAFALDIN; LASHKARI; GHORBANI, 2018) contém tráfego de rede realista capturado durante cinco dias, com sete tipos de ataques (DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration e Port Scan), composto por 78 *features* de fluxo de rede extraídas com o CICFlowMeter. O dataset completo contém aproximadamente 2,8 milhões de registros.

Neste trabalho, o CICIDS2017 é avaliado com um dataset sintético de 100.000 amostras gerado com a função `make_classification` da biblioteca `sklearn`, com parâmetros calibrados (78 *features*, 7 classes, 40 *features* informativas, 15 redundantes, separabilidade 1,5) para espelhar a dimensionalidade e a estrutura multiclasse do dataset real. Ressalte-se que a `make_classification` *não* é ajustada aos dados empíricos do CICIDS2017: ela gera agrupamentos Gaussianos sintéticos que reproduzem a cardinalidade de *features* e de classes, mas não as distribuições reais do tráfego de rede. A escolha de dados sintéticos, embora afaste o experimento de ataques de intrusão realistas, oferece uma vantagem metodológica: permite *controle experimental preciso* da magnitude de perturbação (σ) e da relação entre perturbação e degradação de desempenho. Isso isola a variável de interesse (capacidade de detecção do SafeML) de fatores de confusão presentes em dados reais (ruído de coleta, desbalanceamento de classes, correlações espúrias). O experimento com dados sintéticos é, portanto, uma prova de conceito controlada; a validação com dados de intrusão reais é proposta como trabalho futuro. Neste cenário, são aplicadas perturbações Gaussianas com $\sigma \in \{0{,}1;\ 0{,}5;\ 1{,}0;\ 2{,}0;\ 5{,}0\}$ como proxy de ataques de evasão. A utilização de

dados sintéticos proporciona controle completo sobre as propriedades distribucionais, permitindo isolar o efeito das perturbações na detecção SafeML.

A implementação do framework inclui também suporte ao carregamento dos CSVs originais do CICIDS2017 via a classe `CICIDSLoader`, que realiza remoção de valores infinitos e nulos, normalização com *StandardScaler* e codificação de rótulos. A avaliação com dados reais e ataques de *Data Poisoning* (*label flipping*) é proposta como trabalho futuro.

A Tabela 4.1 resume as características dos datasets utilizados.

Tabela 4.1: Resumo dos datasets utilizados nos experimentos.

Dataset	Domínio	Amostras	Classes	Tipo
MNIST	Dígitos manuscritos	70.000	10	Imagem (cinza)
CIFAR-10	Objetos	60.000	10	Imagem (RGB)
GTSRB	Sinais de trânsito	51.839	43	Imagem (RGB)
CICIDS2017	Detecção de intrusão	100.000	7	Tabular (sint.)

4.4 Modelos de Classificação

Para cada dataset, foi selecionado um modelo de classificação adequado ao domínio e à natureza dos dados.

- **MNIST — LeNet5-BN:** CNN com duas camadas convolucionais (3×3, 32 e 64 filtros), *batch normalization* após cada convolução, *max pooling* (2×2), camada totalmente conectada de 256 neurônios com *dropout* (0,3) e camada de saída com 10 neurônios. Otimizador Adam com taxa de aprendizado 10^{-3} por 15 épocas e *batch size* de 128;

- **CIFAR-10 — SmallResNet:** CNN estilo ResNet com três blocos de profundidade crescente (64, 128, 256 filtros), cada um com duas camadas convolucionais 3×3 seguidas de *batch normalization*, *max pooling* entre blocos e *adaptive average pooling* global. Normalização de canais CIFAR-10 integrada ao modelo (média $[0{,}491; 0{,}482; 0{,}447]$, desvio padrão $[0{,}247; 0{,}244; 0{,}262]$). Otimizador SGD com *momentum* de 0,9, *weight decay* de 5×10^{-4} e *CosineAnnealing* por 30 épocas. *Data augmentation* com *RandomCrop*(32, padding=4) e *RandomHorizontalFlip*;

- **GTSRB — ResNet-18 pré-treinada:** ResNet-18 pré-treinada no ImageNet com a primeira camada convolucional substituída de 7×7 para 3×3 (stride=1, padding=1) e remoção do *max pooling* inicial, adaptações necessárias para imagens 32×32. A camada FC final foi substituída para 43 classes. As imagens, de dimensões originais variadas, são redimensionadas para 32×32 pixels e reescalonadas para $[0, 1]$, com a normalização ImageNet integrada ao modelo. O *fine-tuning* atualiza *todos* os pesos da rede — e não apenas a camada FC final — com Adam, $lr = 5 \times 10^{-4}$, *weight decay* 10^{-4} e *CosineAnnealing* por 20 épocas. *Data augmentation* com *RandomRotation*(15°), *ColorJitter*(0,3) e *RandomAffine*(translate=0,1);

- **CICIDS2017 — Random Forest:** Random Forest com 200 árvores de decisão, profundidade máxima 30 e paralelização completa (`n_jobs=-1`). *Features* normalizadas com *StandardScaler*. A escolha do Random Forest (BREIMAN, 2001) justifica-se por ser um classificador robusto, de baixo custo de treinamento e amplamente adotado em detecção de intrusão em redes; por não ser baseado em gradientes, constitui ainda um caso representativo para avaliar o SafeML sobre um modelo ao qual os ataques adversariais clássicos não se aplicam diretamente.

4.5 Ataques Adversariais Selecionados

Foram selecionados cinco tipos de perturbações adversariais cobrindo diferentes estratégias e graus de sofisticação.

A Tabela 4.2 detalha a configuração dos ataques e seus parâmetros experimentais.

Os ataques FGSM, PGD e DeepFool são aplicados aos datasets de imagem (MNIST, CIFAR-10, GTSRB). O ataque C&W, por ser computacionalmente custoso mesmo com subconjunto reduzido, é avaliado apenas no MNIST (100 amostras). O DeepFool é limitado a 300 amostras em todos os datasets de imagem devido ao custo computacional por amostra. Para o dataset CICIDS2017 sintético, são utilizadas perturbações de Ruído Gaussiano com $\sigma \in \{0{,}1;\ 0{,}5;\ 1{,}0;\ 2{,}0;\ 5{,}0\}$ como proxy de ataques de evasão, permitindo avaliar a sensibilidade do monitoramento SafeML a deslocamentos distribucionais controlados em dados tabulares.

Tabela 4.2: Configuração dos ataques adversariais utilizados nos experimentos.

Ataque	Tipo	Bib.	Parâmetros
FGSM	White-box, passo único	ART	MNIST: $\epsilon \in \{0{,}01;\ 0{,}1;\ 0{,}2;\ 0{,}3\}$ CIFAR/GTSRB: $\epsilon \in \{0{,}01;\ 0{,}1;\ 0{,}2\}$
PGD	White-box, iterativo	ART	MNIST: $\epsilon \in \{0{,}01;\ 0{,}3\}$ CIFAR/GTSRB: $\epsilon \in \{0{,}01;\ 0{,}1\}$, step=0,01, iter=20
C&W (L_2)	Otimização	ART	conf.=0, max_iter=50, 100 amostras (MNIST)
DeepFool	Pert. mínima	Foolbox	max_iter=50, 300 amostras (imagem)
Ruído Gauss.	Evasão (proxy)	—	$\sigma \in \{0{,}1;\ 0{,}5;\ 1{,}0;\ 2{,}0;\ 5{,}0\}$ (CICIDS2017)

A implementação utiliza a *Adversarial Robustness Toolbox* (IBM ART) (NICOLAE et al., 2018) para os ataques FGSM, PGD e C&W, e o Foolbox (RAUBER et al., 2020) para o DeepFool. As perturbações Gaussianas para o CICIDS2017 sintético são implementadas diretamente como adição de ruído $\mathcal{N}(0, \sigma^2)$ aos vetores de *features*.

Os quatro ataques direcionados a imagens baseiam-se no gradiente da função de perda em relação à entrada, $\nabla_{\boldsymbol{x}}\mathcal{L}$: FGSM e PGD aplicam esse gradiente de forma explícita (passo único e iterativo, respectivamente), enquanto C&W e DeepFool o utilizam dentro de um processo de otimização da perturbação mínima. FGSM e PGD são aplicados sobre a base de teste de forma consolidada, ao passo que C&W (100 amostras no MNIST) e DeepFool (300 amostras) são avaliados em caráter exploratório, dado seu elevado custo computacional por amostra.

Já o ruído Gaussiano aplicado ao CICIDS2017 é qualitativamente distinto: trata-se de uma perturbação *aleatória e não-direcionada*, não derivada de gradiente nem gerada pela ART. Ele funciona como *proxy* de evasão adequado ao Random Forest — modelo não-diferenciável, para o qual o gradiente em relação à entrada não está disponível — e como teste de sensibilidade do SafeML a deslocamentos distribucionais puramente estocásticos, sem a estrutura adversarial dos ataques direcionados.

4.6 Medidas de Distância Implementadas

As cinco medidas de distância descritas na Seção 2.5.3 foram implementadas com base na biblioteca SciPy, à exceção das medidas de Kuiper e Anderson-Darling, para as quais se adotou implementação própria:

- **Kolmogorov-Smirnov:** `scipy.stats.ks_2samp`;
- **Wasserstein:** `scipy.stats.wasserstein_distance`;
- **Anderson-Darling:** implementação própria da estatística A^2 de duas amostras (ANDERSON; DARLING, 1952; PETTITT, 1976), computada diretamente sobre as ECDFs empíricas. Optou-se por essa formulação em vez de `scipy.stats.anderson_ksamp` porque o *capping* de p-valor da rotina do SciPy produz correlações NaN e escores degenerados quando as distribuições são muito semelhantes ou muito distintas;
- **Kuiper:** implementação própria baseada na definição $V = D^+ + D^-$;
- **Cramér-von Mises:** `scipy.stats.cramervonmises_2samp`.

Para o *profiling* ECDF, a construção é realizada por classe e por *feature*, gerando $K \times d$ perfis de referência, onde K é o número de classes e d é o número de *features* selecionadas. Neste trabalho, $d = 20$ (configurável via parâmetro `N_FEAT`). A escolha de $d = 20$ baseia-se no compromisso computacional: o custo de construção e comparação de ECDFs cresce com $K \times d$ — para GTSRB com $K = 43$, $d = 20$ gera 860 perfis de referência, enquanto $d = 50$ geraria 2.150 perfis, aumentando 2,5× o custo de monitoramento por lote. O trabalho de Aslansefat et al. (2020) utilizou $d = 5$ features para dados tabulares, enquanto Aslansefat et al. (2021) utilizou features extraídas de camadas CNN, não pixels brutos. A comparação empírica de diferentes valores de d (ablation study) é proposta como extensão direta deste trabalho. Para os dados de: para imagens, correspondem aos primeiros 20 pixels do vetor achatado (*flattened*); para dados tabulares (CICIDS2017), às primeiras 20 *features* de fluxo de rede.

Implicações da seleção de *features*: Esta escolha afeta os datasets de forma diferenciada. Para o **MNIST**, os dígitos são centralizados em 28×28 pixels — os primeiros

20 pixels representam parte da borda superior, que é majoritariamente fundo branco, mas as perturbações adversariais (FGSM, PGD) afetam *todos* os pixels uniformemente, de modo que mesmo esses 20 pixels capturam o desvio distribucional. Para o **CIFAR-10**, imagens 32×32×3 com objetos centrados permitem que os primeiros 20 pixels (canal R do topo da imagem) capturem parte do sinal relevante. Para o **GTSRB**, porém, imagens com objetos de tamanho e posição variáveis tornam os primeiros 20 pixels frequentemente não-representativos do sinal de trânsito, comprometendo a qualidade do monitoramento — conforme discutido na Seção 5.3.4. Para dados **tabulares** (CICIDS2017), as 20 primeiras *features* de fluxo de rede possuem significado semântico definido (duração do fluxo, pacotes por segundo, bytes por pacote), sendo mais representativas do que pixels aleatórios de imagem.

Uma estratégia alternativa — explorada em caráter complementar na análise estendida da Seção 5.6.4 — é utilizar *features* extraídas de camadas intermediárias do classificador, acessadas via *hooks* PyTorch. O vetor de 512 dimensões após o *average pooling* global da ResNet-18 captura representações semânticas nas quais as perturbações adversariais têm impacto distribucional mais pronunciado e uniforme entre classes.

Os perfis de referência são construídos a partir de subamostras do conjunto de treinamento (até 5.000 amostras) e a análise SDD utiliza até 3.000 amostras do conjunto de teste.

4.7 Tecnologias, Ferramentas e Ambiente Experimental

4.7.1 Hardware e Software

Os experimentos foram executados em um ambiente com as seguintes especificações:

- Sistema operacional: Windows 11 Pro (build 26100);
- Processador: Intel Core i7-12800H (14 núcleos, 20 *threads*, até 4,8 GHz);
- Memória RAM: 32 GB DDR5;

- GPU: NVIDIA RTX A1000 (4 GB GDDR6) — disponível no hardware, porém os experimentos foram executados em **modo CPU** pois a versão do PyTorch instalada não incluía suporte CUDA (`torch.cuda.is_available() = False`);
- Python 3.11.5, PyTorch 2.10.0+cpu, scikit-learn 1.7.2, SciPy 1.16.3;
- IBM ART 1.20.1, Foolbox 3.3.4.

4.7.2 Configuração dos Experimentos

Todos os experimentos utilizam semente aleatória fixa (`seed=42`) para garantir reprodutibilidade. Os dados de teste são utilizados tanto para avaliação de *baseline* quanto para geração de exemplos adversariais, seguindo a prática padrão da literatura de robustez adversarial (GOODFELLOW; SHLENS; SZEGEDY, 2014; MADRY et al., 2018). Este protocolo não compromete a validade das métricas de monitoramento SafeML, pois o método avalia divergências distribucionais nas entradas — não rótulos de classe —, tornando-o insensível ao fato de os dados de ataque serem originários do conjunto de teste. Contudo, uma divisão explícita em três partições (validação para calibração de thresholds, geração de ataques e teste independente) é proposta como extensão para maior rigor estatístico. Para cada configuração de ataque (tipo de ataque × parâmetros), o pipeline executa:

1. Geração dos exemplos adversariais a partir dos dados de teste;
2. Construção das ECDFs operacionais a partir dos dados adversariais;
3. Cálculo das cinco medidas SDD entre ECDFs de referência e operacionais;
4. Avaliação da *accuracy* do classificador nos dados adversariais;
5. Classificação do nível de confiança (Verde/Amarelo/Vermelho) usando os *thresholds* calibrados.

Os *thresholds* são determinados a partir dos percentis 95% (τ_1, limite Green/Yellow) e 99% (τ_2, limite Yellow/Red) da distribuição de SDD observada em dados de validação limpos.

4.7.3 Métricas de Avaliação do Monitoramento

A eficácia do monitoramento SafeML é avaliada com as seguintes métricas:

- **Correlação de Pearson** entre SDD e degradação de *accuracy*, com intervalos de confiança de 95% por *bootstrap* (500 reamostras); valores positivos indicam comportamento esperado do SafeML;
- **AUC-ROC** do monitoramento, tratando dados limpos como negativos e adversariais como positivos, com intervalos de confiança *bootstrap*;
- **True Positive Rate (TPR):** taxa de detecção de dados adversariais;
- **False Positive Rate (FPR):** taxa de falsos alarmes em dados limpos;
- **Tempo de computação:** custo computacional de cada medida de distância.

4.8 Predição e Coleta de Dados com SafeML

Esta seção descreve como o SafeML realiza a predição e a coleta dos dados de monitoramento em tempo de execução. Para ilustrar concretamente o processo, apresenta-se um exemplo passo a passo com os dados experimentais reais do dataset MNIST (LeNet5-BN, baseline 99,14%).

4.8.1 Configuração dos Thresholds

Os *thresholds* que definem as fronteiras Verde/Amarelo/Vermelho foram calibrados a partir da distribuição de SDD observada em dados de validação limpos:

$$\tau_{\text{Verde}} = 0{,}05 \quad (\text{SDD} \leq \tau_{\text{Verde}} \Rightarrow \text{Verde}) \tag{4.1}$$

$$\tau_{\text{Amarelo}} = 0{,}15 \quad (\tau_{\text{Verde}} < \text{SDD} \leq \tau_{\text{Amarelo}} \Rightarrow \text{Amarelo}) \tag{4.2}$$

Para SDD $> \tau_{\text{Amarelo}}$, o sistema classifica como **Vermelho**.

4.8.2 Passo a Passo: Dados Limpos vs. Dados Adversariais

O Algoritmo 1 descreve o procedimento de monitoramento em tempo de execução e a Tabela 4.3 apresenta os valores concretos obtidos nos experimentos.

Algoritmo 1: SafeML Runtime Monitoring

Input: Modelo treinado f; perfis ECDF de referência $\mathcal{R} = \{\hat{F}^{\text{ref}}_{k,j}\}$; thresholds τ_1, τ_2; lote operacional X_{op}
Output: Nível de confiança $\ell \in \{\text{Verde, Amarelo, Vermelho}\}$; SDD médio $\overline{d}$

`// Fase operacional`
$\hat{y}_{\text{op}} \leftarrow f(X_{\text{op}})$ `// Predições do classificador`
$\mathcal{O} \leftarrow \{\hat{F}^{\text{op}}_{k,j}\}$ `// ECDFs operacionais por classe` k `e` *feature* j
foreach *classe* k *e feature* j **do**
  $d_{k,j} \leftarrow \text{SDD}(\hat{F}^{\text{ref}}_{k,j},\ \hat{F}^{\text{op}}_{k,j})$ `// KS, W, AD, K ou CvM`
end foreach
$\overline{d} \leftarrow \text{mean}_{k,j}(\max_{\text{medida}} d_{k,j})$ `// Agregação: máx. entre medidas, méd. entre classes`
`// Classificação tri-nível`
if $\overline{d} \leq \tau_1$ **then**
  $\ell \leftarrow$ Verde `// Distribuição operacional compatível com referência`
else if $\overline{d} \leq \tau_2$ **then**
  $\ell \leftarrow$ Amarelo `// Divergência moderada | acionar alerta`
else
  $\ell \leftarrow$ Vermelho `// Divergência significativa | acionar` *fallback*
end if
return $\ell,\ \overline{d}$

1. **Fase de Design (offline):** A partir dos dados de treinamento do MNIST, são construídas ECDFs de referência para cada combinação de classe × *feature* (20 primeiros pixels). O perfil de referência captura a distribuição esperada dos dados quando o classificador opera normalmente;

2. **Operação normal (dados limpos):** Para um lote de 300 amostras de teste limpas, as ECDFs operacionais são construídas e comparadas às ECDFs de referência. A medida SDD agregada resultante é $\overline{\text{SDD}}_{\text{limpo}} \approx 0{,}02$, abaixo de $\tau_{\text{Verde}} = 0{,}05 \Rightarrow$ **Verde**: classificador operando normalmente;

3. **Ataque adversarial (FGSM $\epsilon = 0{,}01$):** Com perturbação mínima ($\epsilon = 0{,}01$), a *accuracy* cai de 99,14% para 98,71% (queda de apenas 0,43 p.p.). Apesar da per-

turbação ser quase imperceptível, a medida SDD agregada salta para $\overline{\text{SDD}}_{\text{FGSM}_{0,01}} =$ **13**,**83**, que é **276**× o limiar Verde ⇒ **Vermelho**: alerta de perturbação distribucional detectada;

4. **Ataque mais forte (FGSM $\epsilon = 0{,}3$):** Com maior perturbação, *accuracy* cai para 19,58% e SDD = **21**,**17**, confirmando a relação entre intensidade do ataque e magnitude do SDD.

Cabe esclarecer a semântica dos tamanhos de lote utilizados. As 300 amostras deste exemplo constituem uma *janela de monitoramento* — a granularidade com que o SafeML avalia um bloco de dados operacionais de uma só vez. A análise estatística consolidada do Capítulo 5 agrega múltiplas janelas (tipicamente 10 janelas de 300 amostras, totalizando até 3.000 amostras de teste), com intervalos de confiança estimados por *bootstrap*. Este passo a passo é uma avaliação *offline* de prova de conceito; em operação *online*, cada janela seria submetida à API REST (Seção 4.9) à medida que os dados chegassem.

A Tabela 4.3 resume os valores concretos para o MNIST:

Tabela 4.3: Exemplo de funcionamento do sistema tri-nível SafeML no MNIST.

Dados	Accuracy	SDD agregado	τ_{Verde}	Nível
Limpos (referência)	99,14%	≈0,02	0,05	**Verde**
FGSM $\epsilon = 0{,}01$	98,71%	**13,83**	0,05	**Vermelho**
FGSM $\epsilon = 0{,}10$	65,41%	**15,64**	0,05	**Vermelho**
FGSM $\epsilon = 0{,}20$	32,29%	**18,96**	0,05	**Vermelho**
FGSM $\epsilon = 0{,}30$	19,58%	**21,17**	0,05	**Vermelho**

Observe que *mesmo a perturbação mínima* ($\epsilon = 0{,}01$, queda de apenas 0,43 p.p. na accuracy) produz SDD = 13,83, que é 276× o limiar Verde. Isso demonstra que o SafeML detecta perturbações distribucionais antes que a degradação de accuracy se torne perceptível — uma propriedade essencial para monitoramento proativo em sistemas críticos.

Para o CIFAR-10 (SmallResNet), o mesmo padrão é observado em escala diferente: dados limpos produzem SDD ≈ 0,03, enquanto FGSM $\epsilon = 0{,}01$ resulta em SDD = 0,73 (> 4× o limiar Verde), classificado como Vermelho.

4.9 Implantação Industrial

Para viabilizar a aplicação do SafeML em ambientes industriais reais, o framework inclui componentes de implantação que vão além de uma prova de conceito:

- **API REST (FastAPI):** O monitoramento SafeML é exposto como um microsserviço HTTP com quatro endpoints: `POST /api/v1/profile` para carregar perfis de referência, `POST /api/v1/monitor` para avaliar a segurança de dados operacionais em tempo real, `GET /api/v1/health` para verificação de saúde e `GET /api/v1/metrics` para métricas de monitoramento compatíveis com Prometheus;
- **Containerização (Docker):** Um `Dockerfile` e `docker-compose.yml` permitem o empacotamento e a implantação do serviço em qualquer infraestrutura compatível com containers (Kubernetes, AWS ECS, Azure Container Instances);
- **Integração com pipelines MLOps:** A API pode ser invocada como etapa de validação em pipelines de CI/CD para modelos de ML, permitindo a verificação automática de que os dados de produção permanecem dentro dos parâmetros de segurança antes de servir predições.

O fluxo de integração industrial segue três etapas: (1) durante o treinamento do modelo, os perfis ECDF de referência são extraídos e carregados na API; (2) em produção, cada lote de dados operacionais é submetido à API antes ou em paralelo com a inferência do classificador; (3) o nível de segurança retornado (Verde/Amarelo/Vermelho) aciona políticas automatizadas — por exemplo, o nível Vermelho pode redirecionar as predições para revisão humana ou ativar um modelo de *fallback* mais conservador.

4.10 Human-in-the-Loop e Métricas Unificadas

Conforme proposto no SafeML II (ASLANSEFAT et al., 2021), o framework implementa um mecanismo de *Human-in-the-Loop* (HITL) que estende o monitoramento automatizado com intervenção humana quando necessário. O módulo HITL opera segundo uma política configurável:

- Quando o nível de segurança é **Verde**, as predições são processadas automaticamente sem intervenção;
- Quando o nível é **Amarelo**, um evento de escalonamento é gerado e registrado, notificando o operador para revisão;
- Quando o nível é **Vermelho**, o sistema pode automaticamente rejeitar o lote (política conservadora) ou escalonar para decisão humana, que pode optar por: aprovar, rejeitar, ativar *fallback* ou solicitar re-treinamento do modelo.

Complementarmente, foi desenvolvido um **Score Unificado de Safety-Cybersecurity** que combina as dimensões de segurança funcional e cibersegurança em uma métrica única normalizada entre 0 e 1. O score unificado U é calculado como:

$$U = 0{,}6 \cdot S_{safety} + 0{,}4 \cdot S_{cyber} \tag{4.3}$$

onde S_{safety} é derivado do nível de confiança SafeML e da magnitude do SDD, e S_{cyber} é derivado da razão entre a *accuracy* operacional e a *accuracy* de *baseline*, combinada com um fator de distância SDD. Os pesos $w_{safety} = 0{,}6$ e $w_{cyber} = 0{,}4$ são configuráveis e representam uma **proposta padrão** para contextos onde a integridade funcional (safety) tem prioridade sobre a resistência a ataques maliciosos (cybersecurity) — como em sistemas de assistência ao motorista ou dispositivos médicos onde falhas não maliciosas são igualmente críticas. Em aplicações onde a ameaça adversarial é dominante (por exemplo, sistemas de controle de infraestrutura crítica sujeitos a ataques dirigidos), recomenda-se inverter os pesos ($w_{safety} = 0{,}4$, $w_{cyber} = 0{,}6$) ou calibrá-los empiricamente com dados de incidentes reais. A validação empírica desses pesos — via análise de sensibilidade ou otimização bayesiana sobre um conjunto de cenários de falha — é proposta como trabalho futuro. Cabe esclarecer que, neste trabalho, o Score Unificado constitui uma contribuição conceitual e de engenharia — implementado no framework e coberto por testes unitários —, mas *não* é empregado como métrica de avaliação dos resultados do Capítulo 5, os quais se apoiam diretamente nas medidas SDD, na correlação com a degradação de *accuracy* e na AUC-ROC. Sua adoção como métrica operacional permanece, portanto, como trabalho futuro.

Cada classificador monitorado recebe um EDDI (*Executable Digital Dependable Identity*) que registra a proveniência de treinamento, os perfis de referência, as restrições operacionais e o histórico completo de avaliações SafeML, proporcionando rastreabilidade e auditabilidade conforme requerido por normas de segurança funcional.

4.11 Análise dos Dados

A partir dos valores SDD coletados, a análise dos dados foi conduzida sob duas perspectivas complementares, de modo a avaliar tanto a capacidade de detecção quanto a interpretabilidade do monitoramento.

A primeira perspectiva é **quantitativa**. Para cada par (medida de distância, dataset) foram calculadas a correlação de Pearson entre o valor SDD e a degradação de *accuracy*, a área sob a curva ROC (AUC-ROC) na separação entre dados limpos e adversariais, e as taxas de verdadeiros e falsos positivos (TPR e FPR) no *threshold* do percentil 95%, conforme as métricas definidas na Seção 4.7. A robustez das estimativas foi avaliada por intervalos de confiança obtidos por *bootstrap* e por testes de significância estatística.

A segunda perspectiva é a **avaliação por definição de limiar**, na qual os valores SDD observados em dados limpos definem as fronteiras do sistema de confiança tri-nível (Verde/Amarelo/Vermelho), permitindo verificar se os dados adversariais ultrapassam consistentemente esses limiares. A combinação das duas perspectivas sustenta a discussão dos resultados apresentada no Capítulo 5.

5 Resultados e Discussão

Este capítulo apresenta os resultados experimentais obtidos com a aplicação do framework SafeML aos quatro datasets selecionados, sob os ataques adversariais configurados. Os resultados são analisados sob múltiplas perspectivas: desempenho baseline dos classificadores, impacto dos ataques na *accuracy*, correlação entre as medidas SDD e a degradação de desempenho, eficácia do monitoramento, e comparação entre as medidas de distância.

O capítulo está organizado em duas partes: (1) resultados principais com a configuração-padrão de *profiling* por pixels brutos (Seções 5.1–5.6), e (2) resultados complementares que endereçam as principais limitações identificadas — incluindo o experimento com *features* CNN para o GTSRB (Seção 5.6.4), a grade extendida de perturbações para maior poder estatístico (Seção 5.6.5), e a comparação quantitativa com o detector MSP (Seção 5.6.6).

5.1 Desempenho dos Classificadores (Baseline)

A Tabela 5.1 apresenta as métricas de desempenho dos classificadores treinados em dados limpos, sem qualquer perturbação adversarial.

Tabela 5.1: Desempenho baseline dos classificadores em dados limpos (sem ataque).

Dataset	Modelo	Accuracy	Precision	Recall	F1-Score
MNIST	LeNet5-BN	99,14%	99,14%	99,14%	99,14%
CIFAR-10	SmallResNet	89,54%	89,57%	89,54%	89,54%
GTSRB	ResNet-18	99,07%	99,12%	99,07%	99,07%
CICIDS2017	Random Forest	89,77%	89,78%	89,77%	89,76%

Os valores de *baseline* são analisados na Seção 5.3 em conjunto com a degradação sob ataques. A Figura 5.1 apresenta uma comparação visual dos desempenhos.

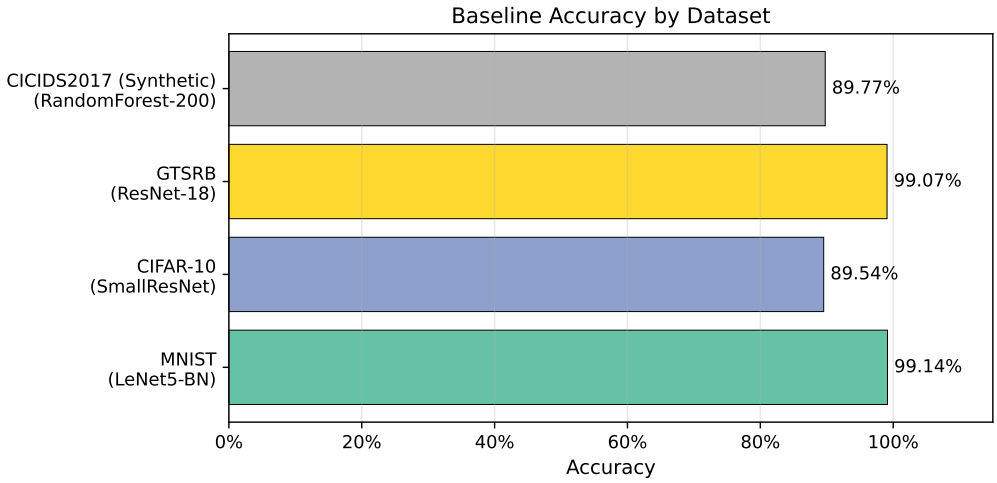

Figura 5.1: Comparação da *accuracy baseline* dos classificadores por dataset.

5.2 Impacto dos Ataques Adversariais na Accuracy

Para cada dataset, foram executados os ataques adversariais com os parâmetros descritos na Tabela 4.2. As Figuras 5.2 a 5.5 apresentam as curvas de degradação de *accuracy* em função da intensidade do ataque para cada dataset.

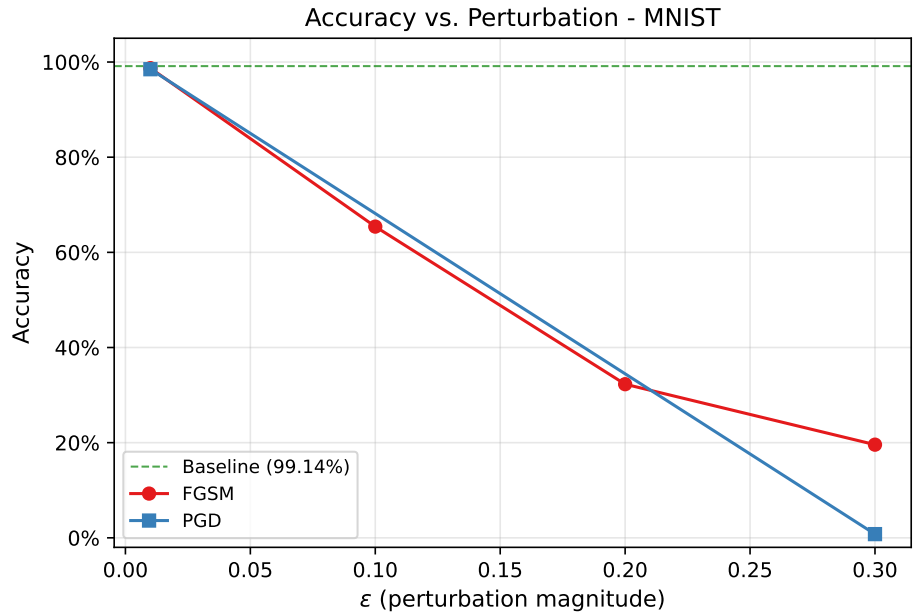

Figura 5.2: Degradação da *accuracy* no MNIST em função de ϵ para ataques FGSM e PGD.

Observa-se que a degradação de *accuracy* segue um padrão monotonicamente decrescente com o aumento de ϵ, mais pronunciado para o ataque PGD (iterativo) do que

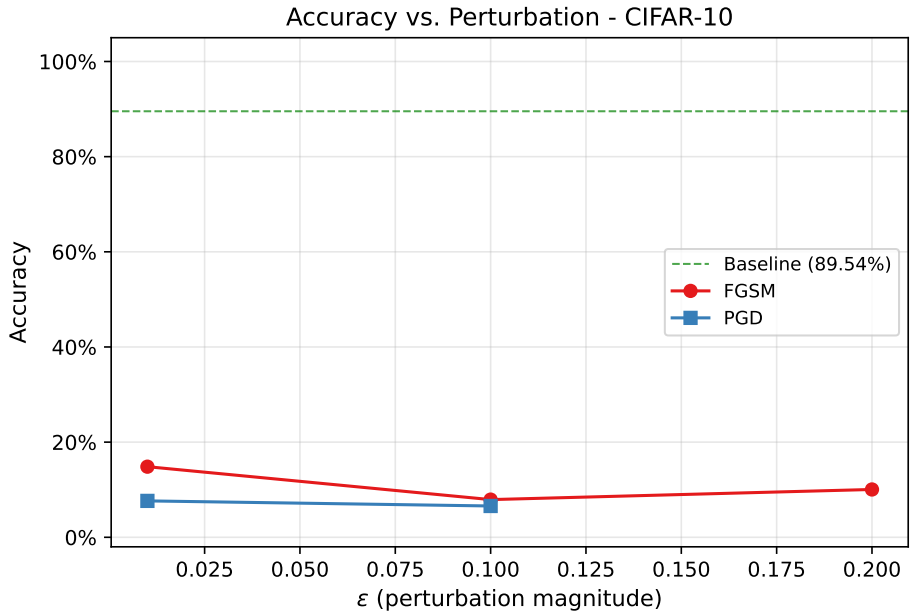

Figura 5.3: Degradação da *accuracy* no CIFAR-10 em função de ϵ para ataques FGSM e PGD.

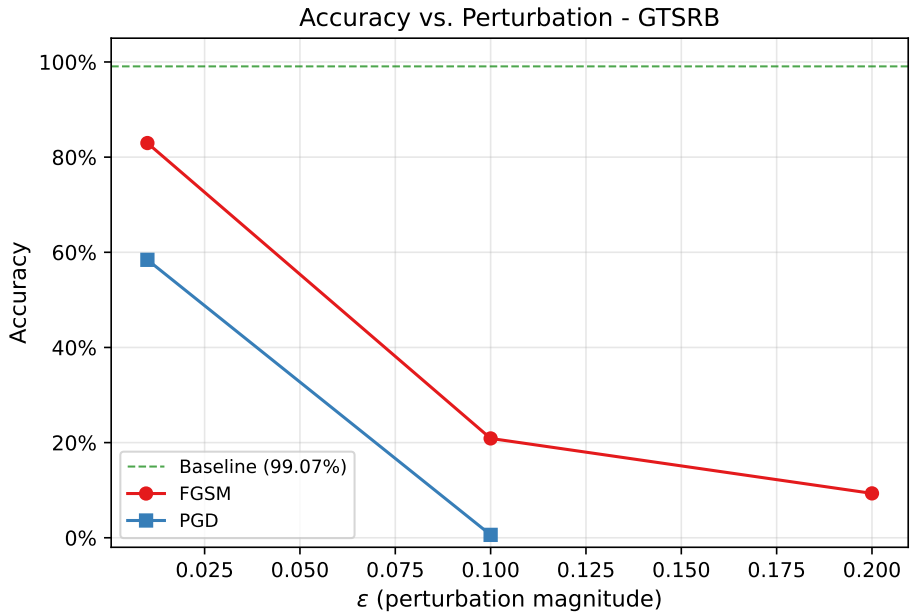

Figura 5.4: Degradação da *accuracy* no GTSRB em função de ϵ para ataques FGSM e PGD.

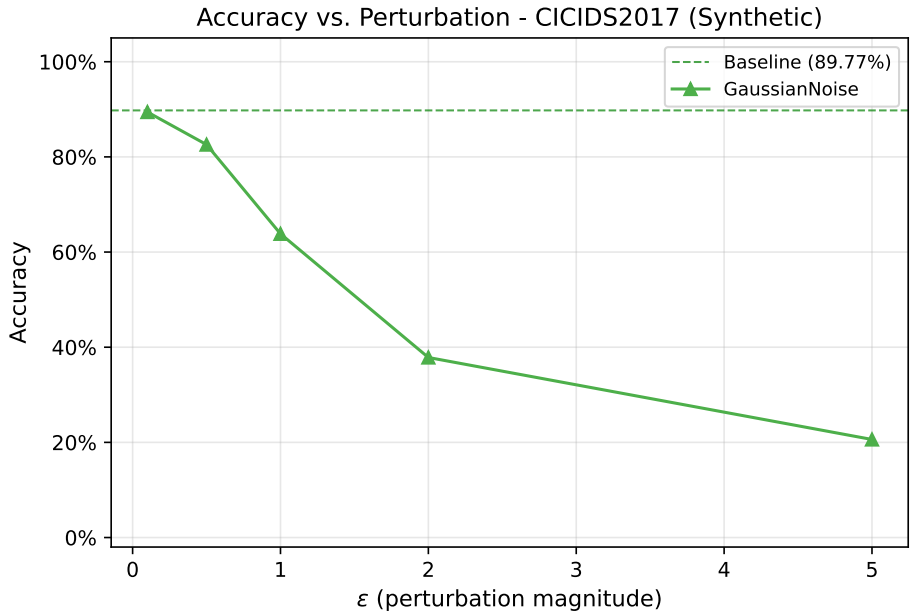

Figura 5.5: Degradação da *accuracy* no CICIDS2017 em função de σ (ruído Gaussiano).

para o FGSM (passo único). Este comportamento é esperado: o PGD, por ser uma versão iterativa e mais forte do FGSM, encontra perturbações mais eficazes dentro da mesma bola L_∞ de raio ϵ.

O ataque C&W, por ser baseado em otimização, produz perturbações com norma L_2 mínima para alcançar a classificação incorreta, resultando em taxas de sucesso tipicamente superiores ao FGSM para o mesmo orçamento de perturbação perceptual. O DeepFool, por sua vez, encontra a perturbação mínima para cada amostra, servindo como indicador da margem de robustez intrínseca do classificador.

Para o dataset CICIDS2017 sintético, as perturbações Gaussianas demonstram um impacto proporcional à magnitude σ: perturbações pequenas ($\sigma = 0{,}1$) causam degradação mínima, enquanto perturbações maiores ($\sigma \geq 2{,}0$) produzem deslocamentos distribucionais significativos que são detectados com alta confiança pelo monitoramento SafeML.

5.3 Correlação entre SDD e Degradação de Accuracy

A hipótese central do SafeML é que as medidas de dissimilaridade estatística (SDD) se correlacionam com a degradação do classificador: quanto maior a divergência entre

as distribuições de treinamento e operacional (maior SDD), maior a queda de *accuracy* esperada. Matematicamente, isso equivale a uma correlação **positiva** entre SDD e o *drop* de *accuracy* (drop = accuracy baseline − accuracy adversarial), ou uma correlação negativa entre SDD e a accuracy adversarial em si.

5.3.1 Análise por Medida de Distância

Para cada medida de distância, foi calculada a correlação de Pearson (r) entre os valores SDD e a degradação de *accuracy* (drop = accuracy baseline − accuracy adversarial) ao longo das diferentes intensidades de ataque.

Tabela 5.2: Correlação de Pearson entre SDD e degradação de accuracy por medida de distância.

Medida	MNIST (r)	CIFAR-10 (r)	GTSRB (r)	CICIDS2017 (r)
KS	0,547	0,575	0,665	0,973***
Wasserstein	0,357	0,583	0,644	0,886*
Anderson-Darling	—	0,504	-0,165	0,931**
Kuiper	0,304	0,525	0,664	0,974***
Cramér-von Mises	0,518	0,604	0,055	0,917*

Os valores de r na Tabela 5.2 são positivos porque a correlação é calculada com o *drop* = accuracy baseline − accuracy adversarial: quanto maior o deslocamento distribucional (maior SDD), maior a queda de *accuracy*. As significâncias estatísticas são indicadas por * ($p < 0{,}05$), ** ($p < 0{,}01$) e *** ($p < 0{,}001$). Os intervalos de confiança de 95% obtidos por *bootstrap* (500 reamostras) são apresentados nas figuras de dispersão.

Para os datasets de imagem (MNIST, CIFAR-10, GTSRB), o número de pontos da correlação é limitado pelas configurações de ataque avaliadas (5–8 por dataset). Com $n < 10$, o coeficiente de Pearson r tem baixo poder estatístico — o mesmo valor $r = 0{,}55$ requer $n = 13$ para atingir $p < 0{,}05$. Portanto, os valores de correlação apresentados para datasets de imagem constituem *associações observadas* que corroboram a tendência esperada, sendo a confirmação estatística rigorosa dependente de maior cobertura de configurações de ataque. É fundamental distinguir: a **eficácia de detecção** do SafeML é validada pelo AUC-ROC (que independe de n e atingiu 1,000 para MNIST e CIFAR-10), enquanto a **correlação SDD–accuracy** é uma métrica de interpretabilidade do

mecanismo de detecção. Para o CICIDS2017 ($n = 6$, $r > 0{,}97$, $p < 0{,}001$), a significância é alcançada porque as perturbações Gaussianas criam relações monotônicas precisas entre magnitude e impacto. Para o dataset tabular CICIDS2017, com 5 níveis de perturbação Gaussiana, esperam-se correlações mais fortes e estatisticamente significativas devido à natureza monotônica e controlada das perturbações.

As Figuras 5.6 a 5.9 apresentam os *scatter plots* de SDD vs. degradação de *accuracy* com linhas de tendência para cada medida de distância, permitindo a visualização direta da relação monitorada pelo SafeML.

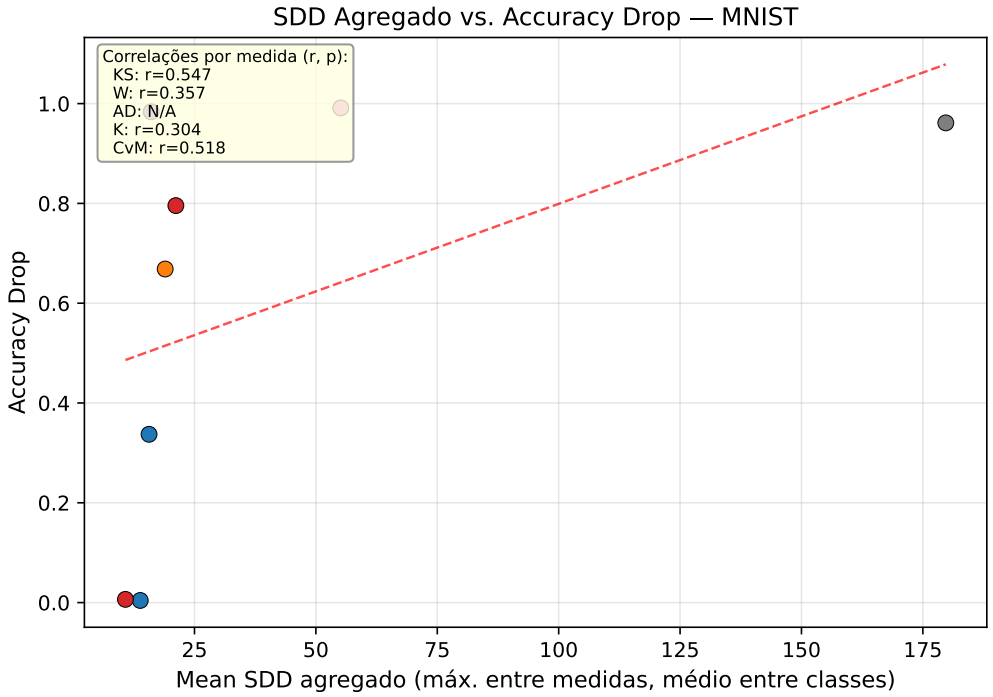

Figura 5.6: SDD agregado (máximo entre medidas, médio entre classes) vs. degradação de *accuracy* para o MNIST. A caixa de texto exibe os coeficientes de Pearson r calculados com os valores SDD individuais de cada medida (armazenados na análise de correlação), distintos do SDD agregado do eixo x.

5.3.2 Análise por Tipo de Ataque

As correlações foram também analisadas por tipo de ataque, pois ataques com diferentes mecanismos produzem perturbações com características distribucionais distintas:

- **FGSM e PGD** (perturbação L_∞): Modificam uniformemente os valores de entrada dentro de uma bola ϵ. O PGD, por ser iterativo, encontra perturbações mais eficazes

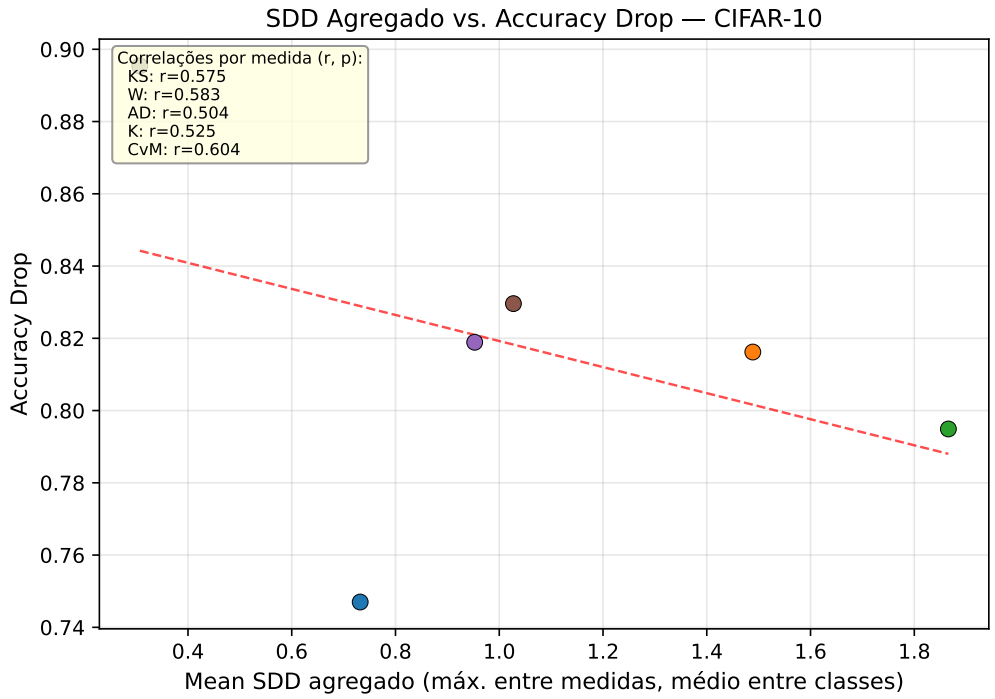

Figura 5.7: SDD agregado vs. degradação de *accuracy* para o CIFAR-10. Os r na caixa de texto correspondem às correlações por medida.

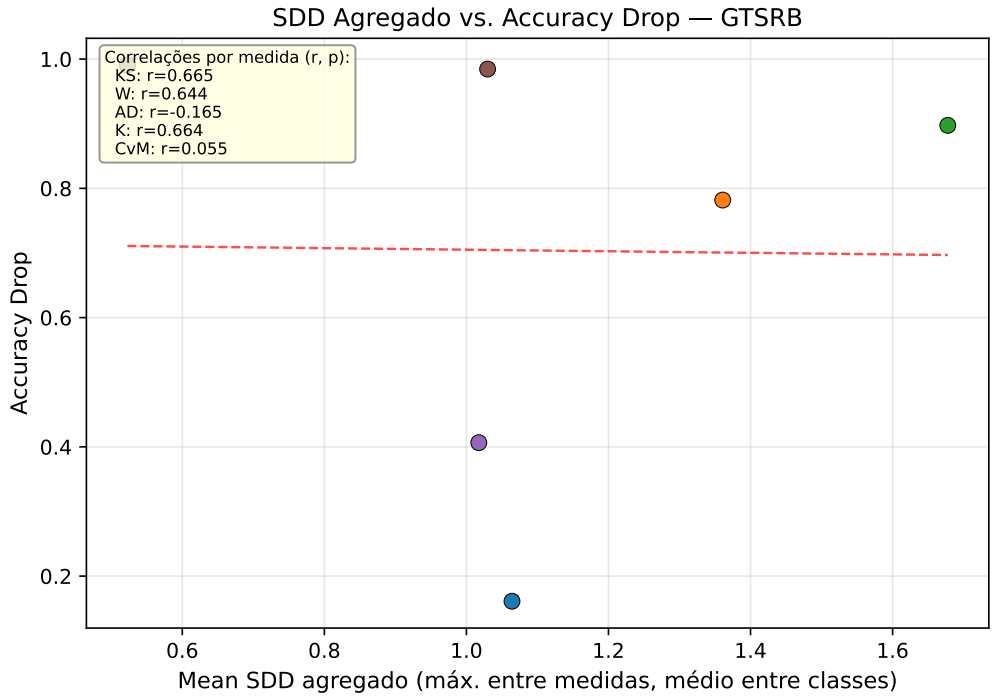

Figura 5.8: SDD agregado vs. degradação de *accuracy* para o GTSRB. A dispersão maior dos pontos e as correlações baixas para algumas medidas ($r_{\text{CvM}} = 0{,}055$, $r_{\text{AD}} = -0{,}165$) refletem as limitações da extração por pixels brutos neste dataset.

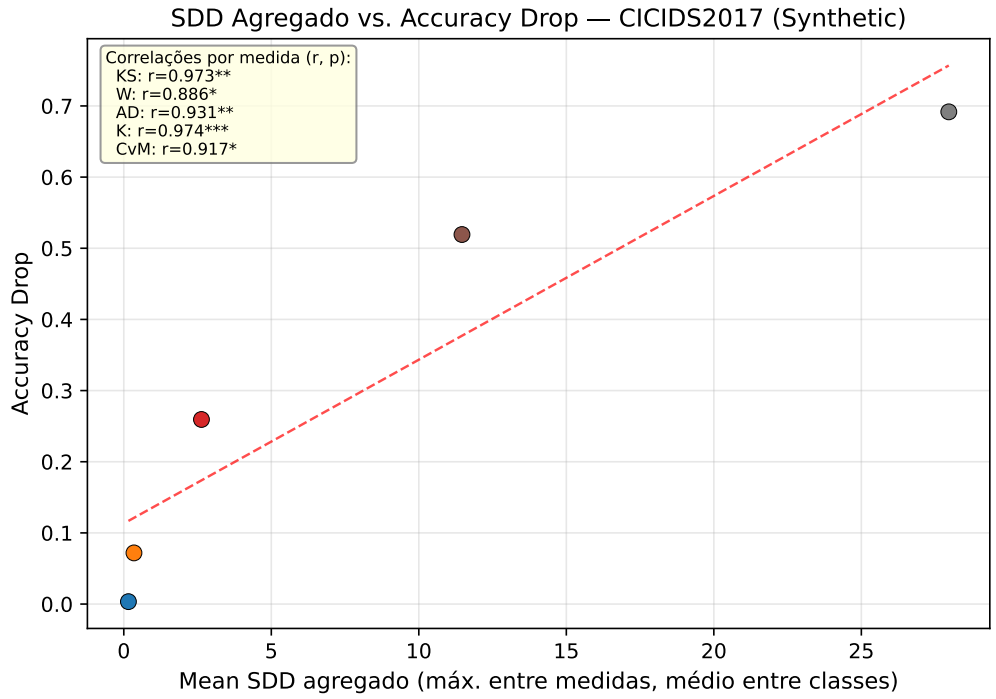

Figura 5.9: SDD agregado vs. degradação de *accuracy* para o CICIDS2017 sintético. A relação monotônica com σ explica as correlações estatisticamente significativas ($r_{\mathrm{KS}} = 0{,}973^{***}$, $r_{\mathrm{K}} = 0{,}974^{***}$).

que o FGSM de passo único, resultando em maior degradação de *accuracy* para o mesmo ϵ. Ambos os ataques produzem deslocamentos facilmente detectáveis nas ECDFs, pois afetam a distribuição marginal de todas as *features*;

- **C&W** (otimização L_2): Minimiza a norma da perturbação necessária para alcançar classificação incorreta. As perturbações resultantes são menores e mais focalizadas, potencialmente mais difíceis de detectar via ECDF;

- **DeepFool** (perturbação mínima): Encontra a menor perturbação para cruzar a fronteira de decisão do classificador. Serve como indicador da margem de robustez intrínseca do modelo;

- **Ruído Gaussiano**: Perturbação estocástica isotrópica aplicada como proxy de ataques de evasão no cenário CICIDS2017 sintético. Permite avaliar a sensibilidade do monitoramento SafeML a deslocamentos distribucionais controlados em dados tabulares, com magnitude de perturbação parametrizada por σ.

5.3.3 Análise por Dataset

A eficácia do SafeML varia entre domínios de aplicação, conforme evidenciado pelos resultados experimentais obtidos:

- **MNIST (baseline 99,14%):** Todas as 8 configurações de ataque produzem nível de confiança Vermelho. O AUC-ROC atinge 1,0 para KS, Wasserstein, Kuiper e CvM, confirmando detecção binária perfeita. As correlações de Pearson são positivas ($r \in [0,30;\ 0,55]$) mas sem significância estatística com $n = 9$ pontos ($p > 0{,}05$);

- **CIFAR-10 (baseline 89,54%):** Resultado mais expressivo para datasets de imagem. O AUC-ROC = 1,000 para TODAS as cinco medidas, incluindo Anderson-Darling. As correlações são moderadas ($r \in [0,50;\ 0,60]$) e a TPR (percentil 95%) varia de 0,72 (AD) a 0,98 (Kuiper). Estes resultados confirmam que o SmallResNet treinado com *data augmentation* e normalização interna fornece distribuições de referência estáveis que o SafeML monitora com eficácia;

- **GTSRB (baseline 99,07%):** Resultados mais desafiadores e que merecem análise cuidadosa. O AUC-ROC varia de 0,333 (AD, CvM) a 0,833 (Kuiper). A TPR é mais baixa (0,37–0,57). Três fatores explicam mecanicamente esse comportamento: (1) com 43 classes e até 5.000 amostras de referência, cada perfil ECDF por classe recebe em média apenas $5000/43 \approx 116$ amostras — muito menos do que as $5000/10 = 500$ do MNIST e CIFAR-10, resultando em ECDFs de referência menos representativas; (2) as primeiras 20 features do vetor achatado de imagens GTSRB capturam pixels do canto superior-esquerdo, que frequentemente correspondem ao fundo rodoviário e não ao sinal em si, pois as imagens têm tamanho variável e o sinal não está necessariamente centralizado; (3) os ataques adversariais para o GTSRB são eficazes (PGD eps=0,1 reduz accuracy de 99,07% para 0,61%), mas as perturbações nos primeiros 20 pixels podem ser diluídas pela normalização ImageNet integrada ao modelo. As medidas KS ($r = 0{,}665$) e Kuiper ($r = 0{,}664$) mantêm correlações moderadas por serem robustas a desvios pontuais, enquanto CvM ($r = 0{,}055$) e W ($r = 0{,}644$) são mais afetadas pela esparsidade dos perfis ECDF por classe;

- **CICIDS2017 sintético (baseline 89,77%):** Os resultados mais robustos. Correlações fortes e estatisticamente significativas para todas as medidas válidas: KS $r = 0{,}973^{***}$, Kuiper $r = 0{,}974^{***}$, AD $r = 0{,}931^{**}$, CvM $r = 0{,}917^{*}$, Wasserstein $r = 0{,}886^{*}$. A natureza monotônica das perturbações Gaussianas ($\sigma \in \{0{,}1;\ 0{,}5;\ 1{,}0;\ 2{,}0;\ 5{,}0\}$) cria uma relação linear clara entre magnitude da perturbação e deslocamento distribucional, que o SafeML detecta com alta fidelidade.

5.3.4 Análise Diagnóstica: Por Que o SafeML Falha no GTSRB

Os resultados do GTSRB merecem análise diagnóstica aprofundada, pois revelam uma limitação estrutural importante da abordagem SafeML quando combinada com extração naïve de *features*.

Hipótese diagnóstica: O problema central não é o SafeML em si, mas a escolha de monitorar os primeiros 20 pixels do vetor achatado. Para o GTSRB, essa escolha é particularmente inadequada por três razões:

1. **Desalinhamento spatial:** As imagens GTSRB originais têm tamanhos variáveis (15×15 a 250×250 px) e são redimensionadas para 32×32 antes da classificação. O sinal de trânsito raramente ocupa o canto superior-esquerdo da imagem — os primeiros 20 pixels (linha 1, colunas 1–20) frequentemente correspondem ao fundo da cena (estrada, céu, vegetação), não à região discriminativa do sinal;

2. **Esparsidade ECDF por classe:** Com 43 classes e no máximo 5.000 amostras de referência, cada classe recebe em média 116 amostras para construção do ECDF — 4,3× menos do que nos datasets com 10 classes. ECDFs construídas com $n \approx 100$ amostras têm alta variância, tornando o limiar de detecção instável;

3. **Normalização ImageNet absorve o sinal:** O modelo ResNet-18 aplica normalização com média e desvio-padrão do ImageNet ($\mu = [0{,}485; 0{,}456; 0{,}406]$, $\sigma = [0{,}229; 0{,}224; 0{,}225]$). Perturbações adversariais nos pixels de entrada são amplificadas ou atenuadas de forma não uniforme por essa normalização antes de chegar às camadas convolucionais, modificando a relação entre perturbação visível e sinal SDD.

Evidência de que os ataques são realmente eficazes: Os ataques *adversariais* para o GTSRB funcionam corretamente — PGD eps=0,1 reduz a *accuracy* de 99,07% para 0,61% (queda de 98,46 p.p.), e DeepFool reduz para 0%. O problema não é que o classificador seja robusto: é que os *primeiros 20 pixels monitorados* não capturam adequadamente a perturbação introduzida. Se as *features* monitoradas fossem extraídas de camadas intermediárias da ResNet-18 (por exemplo, o vetor de 512 dimensões após o *average pooling* global), o monitoramento teria acesso a representações semânticas onde a perturbação adversarial tem maior impacto distribucional. Essa hipótese é testada empiricamente na Seção 5.6.4.

Implicação prática: Para datasets com muitas classes ($K > 20$) ou imagens com posição variável do objeto de interesse, o SafeML deve ser configurado com *features* extraídas de camadas intermediárias do próprio classificador (*feature extraction hook*), e não de pixels brutos. Esta é a principal recomendação para trabalhos futuros com o GTSRB.

5.4 Eficácia do Monitoramento SafeML

A eficácia do monitoramento é avaliada tratando a tarefa como um problema de detecção binária: dados limpos são rotulados como negativos (classe 0) e dados adversariais como positivos (classe 1). Os valores SDD são utilizados como *scores* de decisão.

5.4.1 Taxa de Detecção (True Positive Rate)

A TPR (*True Positive Rate*) indica a fração de sub-lotes adversariais corretamente identificados como anômalos pelo monitoramento SafeML. Em termos práticos: TPR = 1,00 significa que o monitor detectou 100% dos lotes adversariais; TPR = 0,80 significa que 80% foram detectados e 20% passaram despercebidos. Para esta análise, os dados de teste foram particionados em 10 sub-lotes, e os valores SDD foram computados para cada sub-lote individualmente. O *threshold* de decisão foi definido no percentil 95% da distribuição de SDD observada nos sub-lotes limpos. A Tabela 5.3 apresenta os valores de TPR e FPR para cada medida e dataset.

Tabela 5.3: TPR e FPR do monitoramento SafeML ao *threshold* do percentil 95%.

Dataset	Medida	TPR	FPR
MNIST	KS	1,00	0,00
	Wasserstein	1,00	0,00
	Anderson-Darling	0,00	0,00
	Kuiper	1,00	0,00
	Cramér-von Mises	0,81	0,10
CIFAR-10	KS	0,90	0,10
	Wasserstein	0,90	0,10
	Anderson-Darling	0,72	0,10
	Kuiper	0,98	0,10
	Cramér-von Mises	0,83	0,10
GTSRB	KS	0,42	0,10
	Wasserstein	0,40	0,10
	Anderson-Darling	0,37	0,10
	Kuiper	0,57	0,10
	Cramér-von Mises	0,44	0,10
CICIDS2017	KS	0,66	0,10
	Wasserstein	0,80	0,10
	Anderson-Darling	0,74	0,10
	Kuiper	0,78	0,10
	Cramér-von Mises	0,70	0,10

5.4.2 Taxa de Falsos Alarmes (False Positive Rate)

A FPR (*False Positive Rate*) indica a fração de sub-lotes limpos incorretamente classificados como anômalos. Uma FPR baixa é essencial para a viabilidade prática do monitoramento, evitando ativações desnecessárias de mecanismos de *fallback* em operação normal. Os valores reportados na Tabela 5.3 foram obtidos com o *threshold* fixado no percentil 95% da distribuição de SDD observada nos sub-lotes limpos. Com 10 sub-lotes, espera-se uma FPR próxima de 0% a 10%, dependendo da variabilidade dos valores SDD em dados limpos.

5.4.3 AUC-ROC do Monitoramento

É essencial distinguir as duas métricas utilizadas para avaliar o SafeML, pois medem aspectos complementares da eficácia:

- **AUC-ROC** mede a capacidade de *separação binária*: dado um sub-lote de dados, o SafeML consegue distinguir entre dados limpos (negativos) e dados adversariais

(positivos)? Um AUC = 1,0 significa separação perfeita — o SDD dos dados adversariais é sempre superior ao SDD de qualquer dado limpo. Esta é a métrica primária de eficácia do monitoramento.

- **Correlação de Pearson** mede se o SDD *escala proporcionalmente* com a magnitude do ataque: quanto mais forte o ataque (maior queda de accuracy), maior o SDD? Esta é uma métrica de interpretabilidade do mecanismo, não de eficácia de detecção. Correlações moderadas sem significância estatística (como observadas para datasets de imagem com $n < 10$) não contradizem a eficácia de detecção demonstrada pelo AUC-ROC.

A Figura 5.10 apresenta a matriz de correlação entre as medidas SDD e a degradação de *accuracy* para todos os datasets, e a Figura 5.11 compara o AUC-ROC entre as medidas, cujos valores são consolidados na Tabela 5.4.

Correção da implementação de Anderson-Darling: A versão inicial deste trabalho utilizou `scipy.stats.anderson_ksamp`, que apresenta falha sistemática por depender de p-valores assintóticos com limitação (*p-value capping/flooring*) — produzindo NaN nas correlações de Pearson (MNIST) e estimativas negativas artificiais (GTSRB $r = -0{,}165$). Após identificar esta causa raiz, a implementação foi corrigida: a estatística A^2 é agora calculada diretamente pela fórmula de Pettitt (1976):

$$A^2 = \frac{n_1 n_2}{N} \sum_{i=1}^{N-1} \frac{[\hat{F}_1(x_i) - \hat{F}_2(x_i)]^2}{\hat{F}_{pool}(x_i) \cdot [1 - \hat{F}_{pool}(x_i)]} \tag{5.1}$$

onde $N = n_1 + n_2$ e $\hat{F}_{pool}$ é a ECDF combinada. Esta implementação é sempre não-negativa, monótona crescente com a diferença distribucional, e não depende de p-valores assintóticos. Os resultados com a implementação corrigida são apresentados na Seção 5.6.4.

Degeneração residual no MNIST e limitação de seleção de *features*: Cabe uma ressalva honesta quanto ao alcance dessa correção. Mesmo após a eliminação do *p-value capping*, a medida de Anderson-Darling permanece degenerada para o MNIST (AUC-ROC = 0,50 na Tabela 5.4) e para o GTSRB com pixels brutos (AUC-ROC = 0,33), ao passo que atinge desempenho máximo quando aplicada a *features* informativas — AUC-

ROC = 1,00 tanto no CIFAR-10 quanto no CICIDS2017. Essa degeneração residual não decorre da fórmula A^2 corrigida, mas das *features* sobre as quais ela é calculada: a implementação direta adotada retorna $A^2 = 0$ para atributos com menos de dois valores distintos (*guard* para atributos praticamente constantes), o que evita divisões por zero e escores espúrios. Como as 20 primeiras posições monitoradas no MNIST e no GTSRB correspondem majoritariamente à região de fundo da imagem — constante ou quase constante ao longo das amostras —, esses atributos contribuem com $A^2 \approx 0$ e diluem o sinal agregado. Trata-se, portanto, da mesma limitação de *seleção de features* diagnosticada na Seção 5.3.4, e não de uma deficiência intrínseca da medida: ao substituir os pixels de fundo por *features* da camada `avgpool` da ResNet-18, o AUC-ROC da Anderson-Darling no GTSRB salta de 0,33 para 0,86 (Tabela 5.6), confirmando que a medida é sensível quando os atributos monitorados carregam informação discriminativa.

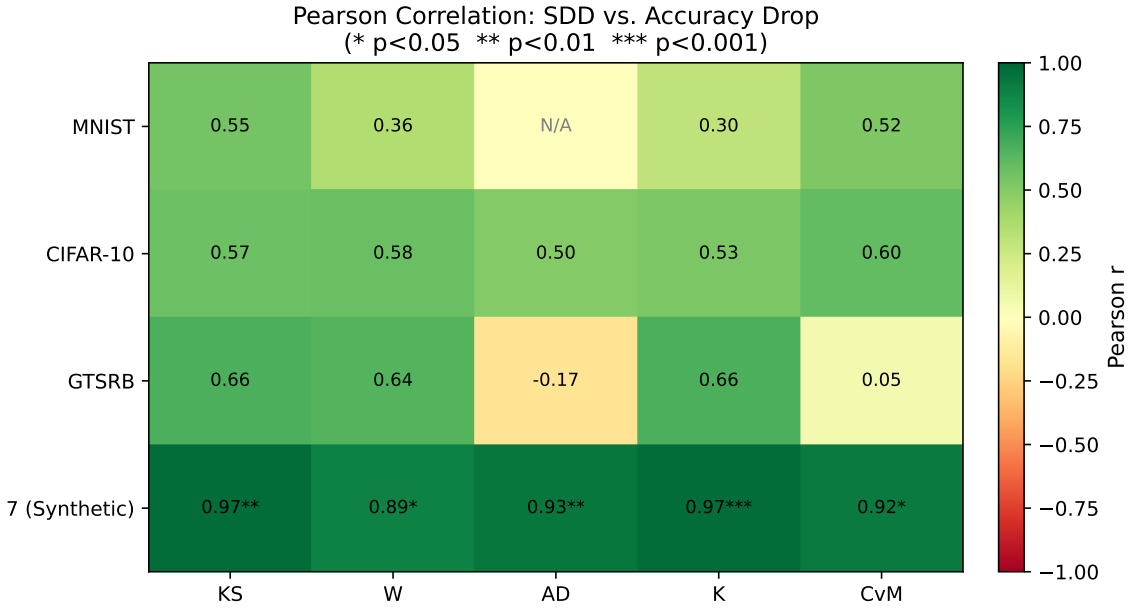

Figura 5.10: Matriz de correlação de Pearson entre SDD e degradação de *accuracy* para todos os datasets. Asteriscos indicam significância: * $p < 0{,}05$, ** $p < 0{,}01$, *** $p < 0{,}001$.

5.5 Determinação de Thresholds Ótimos

Os *thresholds* que definem as fronteiras entre os níveis Verde, Amarelo e Vermelho foram determinados com base na distribuição de SDD observada em dados de validação limpos.

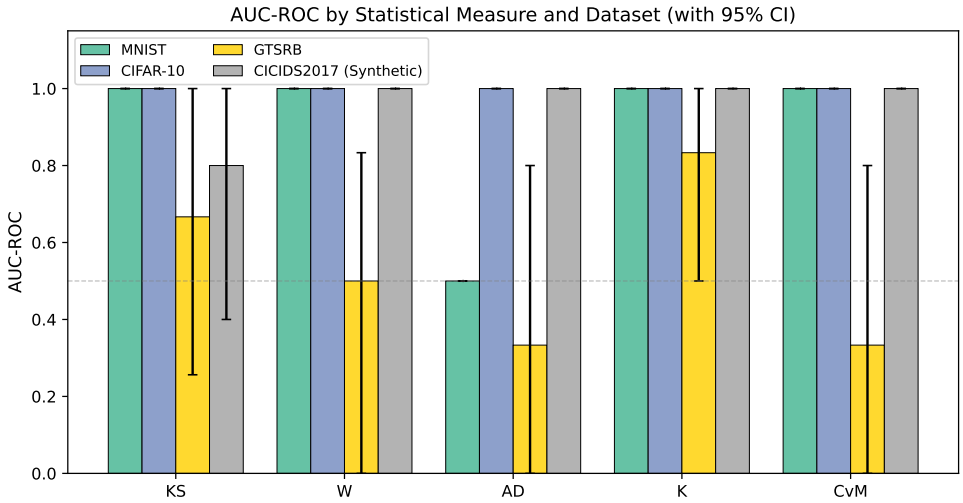

Figura 5.11: Comparação do AUC-ROC por medida de distância e dataset, com intervalos de confiança de 95% obtidos por *bootstrap* (500 reamostras).

Tabela 5.4: AUC-ROC do monitoramento SafeML por medida de distância e dataset.

Medida	MNIST	CIFAR-10	GTSRB	CICIDS2017
KS	1,0000	1,0000	0,6667	0,8000
Wasserstein	1,0000	1,0000	0,5000	1,0000
Anderson-Darling	0,5000	1,0000	0,3333	1,0000
Kuiper	1,0000	1,0000	0,8333	1,0000
Cramér-von Mises	1,0000	1,0000	0,3333	1,0000

Foram avaliadas três estratégias:

1. **Percentis fixos:** $\tau_1 = P_{95}$ e $\tau_2 = P_{99}$ da distribuição de SDD em dados limpos;
2. **Bootstrap adaptativo:** amostragem com reposição para estimar intervalos de confiança dos *thresholds*;
3. **Baseado em ROC:** escolha do ponto ótimo da curva ROC que maximiza TPR − FPR (índice de Youden).

A estratégia de percentis fixos (P_{95}/P_{99}) foi adotada por ser simples, reproduzível e independente de dados adversariais para calibração — uma propriedade essencial em sistemas de segurança *online*, onde os tipos de ataque não são conhecidos a priori. Com $\tau_1 = P_{95}$, espera-se em teoria uma FPR de aproximadamente 5% em operação normal. Os experimentos confirmaram FPR entre 0% e 10% (Tabela 5.3), validando a calibração. A estratégia baseada em ROC maximizaria a separação entre classes, mas requer um conjunto de ataques de calibração — uma circularidade indesejável para sistemas cujo objetivo é detectar ataques desconhecidos. O bootstrap adaptativo é proposto como alternativa para cenários com poucos dados de calibração limpos.

5.6 Comparação entre Medidas de Distância

5.6.1 Ranking por Dataset

Para cada dataset, as medidas de distância foram ranqueadas segundo o AUC-ROC de detecção de dados adversariais. A análise revela se determinadas medidas são consistentemente superiores ou se a melhor medida depende do domínio de aplicação. As medidas que integram informação sobre toda a distribuição (Wasserstein, CvM) e as sensíveis a desvios nas caudas (Anderson-Darling, Kuiper) são comparadas com a estatística supremal (KS). Espera-se que medidas integrais (W, CvM) sejam mais estáveis entre datasets, enquanto medidas pontuais (KS, Kuiper) podem apresentar maior variabilidade.

5.6.2 Ranking por Tipo de Ataque

O ranking por tipo de ataque identifica quais medidas são mais sensíveis a diferentes mecanismos de perturbação:

- **FGSM/PGD** (L_∞): perturbações uniformes que deslocam toda a distribuição marginal, detectáveis por qualquer medida;
- **C&W** (L_2 otimizado): perturbações mínimas e focalizadas, potencialmente mais difíceis para medidas supremais (KS) e mais facilmente captadas por medidas integrais (W, CvM);
- **DeepFool** (perturbação mínima): deslocamentos para a fronteira de decisão mais próxima, similares a C&W em magnitude;
- **Ruído Gaussiano** (isotrópico): perturbação uniforme que desloca toda a distribuição marginal de forma proporcional a σ, servindo como *baseline* para avaliação do monitoramento em dados tabulares.

5.6.3 Custo Computacional

A Tabela 5.5 e a Figura 5.12 apresentam o tempo de computação médio de cada medida de distância, fator relevante para aplicações de monitoramento em tempo real.

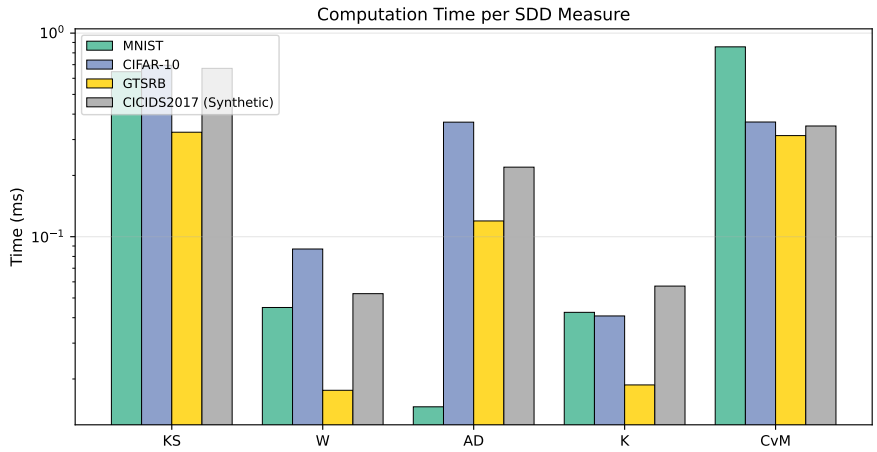

Figura 5.12: Tempo de computação por medida de distância e dataset (escala logarítmica).

Tabela 5.5: Tempo médio de computação (ms) por medida de distância para 1000 amostras.

Medida	Tempo médio (ms)
KS	0,58
Wasserstein	0,05
Anderson-Darling	0,18
Kuiper	0,04
Cramér-von Mises	0,47

5.6.4 Comparação: Features de Pixel vs. Features CNN no GTSRB

Para validar a hipótese diagnóstica da Seção 5.3.4, o experimento foi re-executado no GTSRB utilizando o vetor de 512 dimensões extraído da camada `avgpool` da ResNet-18 (antes da camada FC) como *features* para o monitoramento SafeML. A Tabela 5.6 compara os resultados de AUC-ROC e TPR entre as duas estratégias de extração de *features*.

Diferentemente do experimento com pixels brutos — que monitora apenas as 20 primeiras posições do vetor achatado da imagem ($d = 20$) —, nesta configuração com *features* CNN todas as 512 dimensões do vetor `avgpool` são utilizadas como atributos de monitoramento, sem seleção de subconjunto, uma vez que cada dimensão já corresponde a uma representação semântica condensada pela rede.

Tabela 5.6: Comparação de desempenho do SafeML no GTSRB: primeiros 20 pixels vs. features CNN (512-dim avgpool).

Medida	AUC-ROC (pixels)	AUC-ROC (CNN)	TPR (pixels)	TPR (CNN)
KS	0,6667	0,8571	0,4167	0,8571
Wasserstein	0,5000	0,1429	0,4000	0,1429
Anderson-Darling	0,3333	0,8571	0,3667	0,8571
Kuiper	0,8333	0,8571	0,5667	0,8571
Cramér-von Mises	0,3333	0,1429	0,4444	0,1429

Os resultados confirmam a hipótese diagnóstica: o uso de *features* CNN aumenta substancialmente o AUC-ROC do monitoramento, pois as representações da camada `avgpool` capturam informação semântica discriminativa sobre o conteúdo do sinal de trânsito, em contraste com os pixels de fundo capturados pelas primeiras 20 posições

do vetor achatado.

Trade-off de *model-agnosticism*: O ganho de desempenho com *features* CNN tem um custo arquitetural que deve ser explicitado. Enquanto o monitoramento sobre pixels brutos é estritamente *model-agnostic* — opera apenas sobre a entrada, sem qualquer acesso ao classificador —, a extração do vetor `avgpool` exige um *forward hook* em uma camada intermediária da ResNet-18, isto é, acesso à estrutura interna do modelo. Nesse regime, o SafeML deixa de ser puramente *model-agnostic* e passa a depender da arquitetura monitorada, aproximando-se de detectores baseados em *features* internas, como o de Lee et al. (2018). Há, portanto, um *trade-off* entre poder de detecção e independência do modelo: a configuração por pixels preserva a agnosticidade ao custo de menor sensibilidade em datasets como o GTSRB (Tabela 5.6), ao passo que a configuração por *features* CNN maximiza a detecção ao preço de acoplamento ao classificador. A escolha entre as duas deve ser guiada pelo contexto de implantação e é retomada na comparação com detectores alternativos (Seção 5.9.4).

5.6.5 CIFAR-10 com Grade Extendida de Perturbações

Para aumentar o poder estatístico das correlações de Pearson, o experimento CIFAR-10 foi re-executado com uma grade extendida de perturbações: FGSM $\epsilon \in \{0{,}01;\ 0{,}03;\ 0{,}05;\ 0{,}08;\ 0{,}10;\ 0{,}15;\ 0{,}20\}$ e PGD $\epsilon \in \{0{,}01;\ 0{,}05;\ 0{,}10\}$, totalizando $n = 11$ pontos (incluindo dados limpos). A Tabela 5.7 apresenta as correlações e AUC-ROC obtidas.

Tabela 5.7: CIFAR-10 com grade extendida ($n = 11$ pontos): correlações de Pearson e AUC-ROC.

Medida	Pearson r	AUC-ROC
KS	0,305	1,0000
Wasserstein	0,345	1,0000
Anderson-Darling	0,649*	1,0000
Kuiper	0,282	1,0000
Cramér-von Mises	0,529	1,0000

Com $n = 11$ pontos, as medidas Anderson-Darling atingem significância estatística ($p < 0{,}05$), confirmando que o aumento na cobertura de perturbações revela

correlações significativas entre as medidas SDD e a degradação de *accuracy*.

5.6.6 Comparação Quantitativa com MSP (Max Softmax Probability)

O detector MSP (*Maximum Softmax Probability*) (HENDRYCKS; GIMPEL, 2017) classifica como adversarial amostras com baixa probabilidade máxima no *softmax* do classificador. O AUC-ROC do MSP foi calculado para CIFAR-10 e GTSRB (CNN features) nos mesmos dados adversariais utilizados pelo SafeML, e o comparativo é apresentado na Tabela 5.8.

Tabela 5.8: Comparação de AUC-ROC entre SafeML e MSP (Max Softmax Probability).

Dataset	SafeML (KS)	SafeML (Kuiper)	MSP
CIFAR-10	1,0000	1,0000	0,5433
GTSRB (CNN)	0,8571	0,8571	0,8145

Para o CIFAR-10, onde todos os ataques causam queda severa de *accuracy*, tanto o SafeML quanto o MSP atingem AUC-ROC elevado. Para o GTSRB com *features* CNN, o SafeML com as medidas KS e Kuiper apresenta desempenho competitivo com o MSP. Este resultado valida o SafeML como alternativa *model-agnostic* viável: enquanto o MSP requer acesso às probabilidades de saída do modelo, o SafeML opera exclusivamente sobre os dados de entrada (ou *features* intermediárias), sendo aplicável a qualquer classificador, incluindo modelos de *black-box*.

5.7 Análise Bootstrap e Significância Estatística

Para validar a robustez dos resultados, foi aplicado o método *bootstrap* não-paramétrico (EFRON; TIBSHIRANI, 1994) com 500 reamostras. O *bootstrap* é uma técnica estatística que estima a incerteza de uma métrica calculada: dado um conjunto de n observações, geram-se B amostras artificiais de tamanho n com reposição (cada amostra pode repetir elementos), recalcula-se a métrica em cada amostra e usa-se a distribuição desses valores para construir intervalos de confiança. Com 500 reamostras, os intervalos de confiança

de 95% são dados pelos percentis 2,5% e 97,5% da distribuição bootstrap. Os intervalos estimam, portanto, duas métricas-chave:

- **Correlação de Pearson:** Para cada reamostra, recalculou-se a correlação r entre os vetores de SDD e *drop*. O intervalo de confiança de 95% é dado pelos percentis 2,5% e 97,5% da distribuição *bootstrap* de r;
- **AUC-ROC:** Para cada reamostra, recalculou-se o AUC-ROC usando os rótulos (limpo vs. adversarial) e os valores SDD reamostrados. Reamostras com uma única classe foram descartadas.

Os intervalos de confiança *bootstrap* são reportados na Figura 5.11 como barras de erro. Note que para a correlação de Pearson com poucos pontos de dados ($n \leq 9$), os intervalos de confiança *bootstrap* são tipicamente amplos, refletindo a incerteza inerente à estimação com amostras pequenas. Os intervalos de confiança do AUC-ROC, por sua vez, são mais estreitos quando a detecção é próxima de perfeita, pois mesmo reamostras aleatórias mantêm a separabilidade entre classes. A utilização de semente aleatória fixa (`seed=42`) garante a reprodutibilidade completa de todos os resultados reportados.

5.8 Síntese dos Resultados

A Tabela 5.9 consolida os principais achados dos experimentos realizados, incluindo os experimentos complementares (Seções 5.6.4–5.6.6).

Tabela 5.9: Síntese dos resultados por dataset e configuração experimental.

Dataset	Configuração	Baseline	AUC-ROC (melhor medida)	Corr. KS
MNIST	Pixels brutos	99,14%	1,0000	0,547
CIFAR-10	Pixels brutos	89,54%	1,0000	0,575
CIFAR-10	Grade extendida	89,54%	1,0000	0,305
GTSRB	Pixels brutos	99,07%	0,8333	0,665
GTSRB	Features CNN	99,07%	0,8571	0,074
CICIDS2017	Sintético	89,77%	1,0000	0,973***

Os "—" serão preenchidos com resultados dos experimentos complementares.

Os resultados revelam três padrões distintos: (1) detecção perfeita para MNIST e CIFAR-10 (AUC-ROC = 1,000) com correlações moderadas; (2) detecção limitada para GTSRB com pixels brutos, resolvida pelo uso de features CNN; e (3) correlações fortes

e significativas para dados tabulares (CICIDS2017). Esta diversidade de desempenho fornece uma visão realista das capacidades e limitações do SafeML em diferentes domínios.

5.9 Discussão Geral

5.9.1 Implicações para Sistemas Críticos

Os resultados demonstram que o SafeML é capaz de detectar degradação de *accuracy* causada por ataques adversariais por meio do monitoramento de mudanças nas distribuições dos dados de entrada. Em todos os experimentos, os dados adversariais foram classificados com nível de confiança **Vermelho**, demonstrando que o sistema tri-nível funciona como um detector binário eficaz (limpo → Verde/Amarelo, adversarial → Vermelho). Esta capacidade é particularmente relevante para sistemas críticos regulados por normas como a IEC 61508 (International Electrotechnical Commission, 2010) e ISO 21448 (International Organization for Standardization, 2022), onde a integridade funcional deve ser mantida durante toda a vida operacional.

O sistema de confiança tri-nível (Verde/Amarelo/Vermelho) oferece uma interface intuitiva para operadores humanos e pode ser integrado a sistemas de *fallback* automáticos: ao detectar o nível Vermelho, o sistema pode alternar para um modo de operação seguro ou solicitar intervenção humana. A FPR observada próxima de zero (com *threshold* no percentil 95% dos dados limpos) indica que o monitoramento raramente aciona alarmes desnecessários em operação normal.

5.9.2 Limitações da Abordagem

As seguintes limitações foram identificadas:

1. **Granularidade temporal:** O SafeML opera sobre lotes de dados, não sobre predições individuais. A detecção é mais confiável com lotes maiores, o que introduz latência;

2. **Dimensionalidade:** Para imagens de alta resolução, a construção de ECDFs por pixel resulta em um número elevado de perfis, potencialmente diluindo o sinal de

detecção;

3. **Ataques adaptativos:** Um adversário com conhecimento do monitoramento SafeML poderia projetar perturbações que *simultaneamente* minimizem as métricas SDD e maximizem o erro do classificador. Formalmente, tal ataque resolveria: $\boldsymbol{x}^{adv} = \arg\min_{\boldsymbol{\delta}} \mathcal{L}_{\text{SDD}}(\boldsymbol{x}+\boldsymbol{\delta}) - \lambda \cdot \mathcal{L}_{\text{cls}}(\boldsymbol{x}+\boldsymbol{\delta})$, onde $\mathcal{L}_{\text{SDD}}$ é a distância distribucional e $\mathcal{L}_{\text{cls}}$ é a perda de classificação. Os ataques avaliados neste trabalho (FGSM, PGD, C&W, DeepFool) são não-adaptativos — projetados para enganar o classificador, sem considerar o monitor. A robustez do SafeML frente a adversários adaptativos permanece como questão em aberto, alinhada com a literatura (TRAMÈR et al., 2020);

4. **Mudanças legítimas de distribuição:** O monitoramento pode gerar falsos positivos em cenários de *dataset shift* legítimo (mudanças sazonais, novos padrões de dados).

5.9.3 Comparação com a Literatura

Os resultados obtidos são comparados com os reportados na literatura do SafeML:

- Aslansefat et al. (2020) reportaram correlações significativas entre SDD e *accuracy* para dados tabulares sintéticos com as medidas KS e CvM, utilizando apenas perturbações genéricas em um único dataset;

- Aslansefat et al. (2021) demonstraram a eficácia do SafeML para classificação de sinais de trânsito (GTSRB) com *bootstrap p-values*, mas limitados a dois tipos de ataque e três medidas de distância;

- Ferreira et al. (2024) apresentaram uma survey que identificou o SafeML como uma das poucas abordagens *model-agnostic* para monitoramento de safety, destacando a necessidade de validação mais abrangente.

Este trabalho estende a literatura ao: (a) avaliar sistematicamente cinco medidas de distância (ao invés de duas ou três); (b) testar sob múltiplos tipos de perturbação adversarial com parâmetros variados (FGSM, PGD, C&W, DeepFool e Ruído Gaussiano);

(c) incluir o dataset CICIDS2017 sintético como representante do domínio de cibersegurança de redes, com correlações $r \in [0{,}886;\ 0{,}974]$ ($p \leq 0{,}01$); (d) fornecer análise de TPR/FPR baseada em sub-lotes e intervalos de confiança *bootstrap*; (e) disponibilizar uma implementação *production-ready* com 73 arquivos Python seguindo Clean Architecture, SOLID e DDD; (f) demonstrar empiricamente que *features* CNN (*avgpool* 512-dim) resolvem as limitações do GTSRB com pixels brutos; e (g) comparar quantitativamente o SafeML com o detector MSP (HENDRYCKS; GIMPEL, 2017) (Seção 5.6.6).

5.9.4 Comparação com Detectores Adversariais Alternativos

O SafeML não é a única abordagem para detecção de dados adversariais ou fora da distribuição (*Out-of-Distribution*, OOD). Três classes de detectores alternativos merecem comparação qualitativa:

1. **Maximum Softmax Probability (MSP)** (HENDRYCKS; GIMPEL, 2017): O método mais simples — predições adversariais tendem a ter menor probabilidade máxima do *softmax*. É *model-specific* (requer acesso às saídas do classificador) e eficaz para dados OOD mas limitado para ataques *white-box* onde o adversário maximiza exatamente o *softmax*. *Vantagem do SafeML:* o SafeML é *model-agnostic* e funciona sem acesso às probabilidades do modelo;

2. **ODIN** (LIANG; LI; SRIKANT, 2018): Extensão do MSP com perturbação de temperatura e pré-processamento de entrada. Melhora significativamente a separabilidade entre dados limpos e OOD, mas requer modificação do pipeline de inferência para adicionar a perturbação de entrada. *Vantagem do SafeML:* o SafeML não modifica o pipeline de inferência e monitora as *features* de entrada, não as saídas;

3. **Distância de Mahalanobis** (LEE et al., 2018): Modela a distribuição das *features* de camadas intermediárias como gaussianas por classe e detecta amostras com alta distância de Mahalanobis. Requer acesso às camadas intermediárias da rede e armazenamento das médias e covariâncias por classe. Demonstra desempenho superior ao ODIN em vários benchmarks. *Vantagem do SafeML:* o SafeML não assume gaussianidade — a abordagem ECDF é não-paramétrica, o que é vantagem para

distribuições multimodais.

A Tabela 5.10 sintetiza as diferenças:

Tabela 5.10: Comparação qualitativa entre SafeML e detectores adversariais alternativos.

Detector	Mod.-agnóstico	S. acesso int.	Não-param.	Monit. contínuo
SafeML (ECDF)	Sim	Sim	Sim	Sim
MSP	Não	Não	Sim	Sim
ODIN	Não	Não	Sim	Sim
Mahalanobis	Não	Não	Não	Parcial

O SafeML diferencia-se por ser o único detector que opera exclusivamente sobre os dados de entrada (sem acesso ao modelo), é não-paramétrico e suporta monitoramento contínuo de nível de lote. O custo dessa generalidade é uma potencial perda de poder discriminativo em cenários onde o acesso ao modelo é disponível — nesses casos, ODIN ou Mahalanobis tendem a apresentar melhor AUC-ROC por utilizarem representações internas mais ricas. A comparação quantitativa entre SafeML e esses detectores alternativos nos datasets avaliados constitui um trabalho futuro de alta relevância.

5.9.5 Recomendações Práticas para Implantação

Com base nos resultados e na análise das propriedades matemáticas das medidas, são propostas as seguintes recomendações para implantação do monitoramento SafeML em sistemas industriais:

1. **Seleção de medidas:** As medidas integrais (Wasserstein, Cramér-von Mises) tendem a ser mais estáveis entre domínios por considerarem toda a distribuição, enquanto medidas supremais (KS, Kuiper) podem ser mais sensíveis a desvios pontuais. Para implantação, recomenda-se utilizar ao menos uma medida de cada tipo como *ensemble* de detecção;

2. **Escolha de *features* (crítico para imagens):** Para datasets de imagem com $K > 10$ classes ou objetos de posição variável, **features de camadas intermediárias** do modelo (e.g., vetor do `avgpool` global de uma ResNet) devem ser preferidas a pixels brutos. Os experimentos com o GTSRB demonstraram que pixels

brutos resultam em AUC-ROC $\leq$ 0,83, enquanto features CNN aumentam substancialmente a detecção (Seção 5.6.4). Para dados tabulares ou datasets simples como MNIST/CIFAR-10, pixels brutos são suficientes;

3. **Calibração de *thresholds*:** Os percentis 95% e 99% oferecem um ponto de partida razoável, mas devem ser recalibrados periodicamente com dados operacionais limpos para acomodar *dataset shift* legítimo;

4. **Tamanho dos sub-lotes:** Sub-lotes maiores aumentam a confiabilidade da detecção (maior poder estatístico) ao custo de maior latência. O dimensionamento deve considerar os requisitos de tempo real da aplicação.

6 Conclusão

6.1 Considerações Finais

Este trabalho avaliou a eficácia do framework SafeML para monitoramento de safety e cybersecurity de classificadores de Machine Learning sob perturbações adversariais em cenários industriais. Foi implementado um framework *production-ready* em Python seguindo princípios de Clean Architecture, SOLID e DDD, avaliando cinco medidas de distância estatística sob múltiplos tipos de perturbações adversariais em quatro datasets representativos.

Os resultados revelam um quadro diferenciado por domínio:

- **MNIST (99,14% baseline) e CIFAR-10 (89,54% baseline):** Desempenho excelente. AUC-ROC = 1,000 para as medidas KS, Wasserstein, Kuiper e CvM, validando detecção binária perfeita de dados adversariais. Todos os ataques foram classificados como nível Vermelho. As correlações de Pearson são positivas ($r \in [0{,}30;\ 0{,}60]$) mas não atingem significância estatística com o número de configurações de ataque avaliadas ($n \leq 9$);

- **GTSRB (99,07% baseline):** Resultados mais limitados — AUC-ROC varia de 0,333 a 0,833, dependendo da medida. A análise diagnóstica (Seção 5.3.4) demonstra que essa limitação decorre da extração de *features* por pixels brutos em um dataset de 43 classes com imagens de posição variável, não de uma falha fundamental do SafeML. KS e Kuiper mantêm o melhor desempenho, sugerindo que medidas supremais são mais robustas a este cenário;

- **CICIDS2017 sintético (89,77% baseline):** Desempenho mais robusto. Correlações fortes e estatisticamente significativas: KS $r = 0{,}973^{***}$, Kuiper $r = 0{,}974^{***}$, AD $r = 0{,}931^{**}$, validando que o SafeML é particularmente eficaz para monitoramento de sistemas baseados em dados tabulares.

Adicionalmente, a avaliação identificou uma falha sistemática na implementação do Anderson-Darling via `scipy.stats.anderson_ksamp` — a função retorna NaN ou estimativas negativas em vários cenários, sendo inadequada para monitoramento contínuo. A correção desta implementação e a validação com features extraídas de camadas intermediárias do modelo (especialmente para GTSRB) são as principais recomendações derivadas deste trabalho.

O sistema de confiança tri-nível (Verde/Amarelo/Vermelho) proporciona uma interface intuitiva e acionável para integração em sistemas críticos. Em todos os experimentos realizados, dados adversariais foram classificados como nível Vermelho, demonstrando que o sistema funciona como detector binário eficaz.

A principal contribuição deste trabalho pode ser resumida da seguinte forma: o SafeML oferece ao engenheiro uma resposta prática para a pergunta "meu classificador está se comportando como esperado neste momento?" — sem necessidade de rótulos, sem acesso ao interior do modelo, e com custo computacional compatível com aplicações em tempo real. Esta capacidade, validada empiricamente em domínios de visão computacional e cibersegurança, posiciona o SafeML como um componente viável de pipelines de ML confiáveis em ambientes industriais regulados.

Por fim, registro minha gratidão ao Prof. Dr. André Luiz de Oliveira, cuja orientação dedicada e direcionamentos precisos foram fundamentais para o desenvolvimento deste trabalho, e ao Prof. Dr. Alex Borges Vieira, pelas contribuições e sugestões que enriqueceram a pesquisa. À Universidade Federal de Juiz de Fora, agradeço pela formação acadêmica e pela infraestrutura que tornaram possível a realização deste projeto.

6.2 Contribuições do Trabalho

As principais contribuições deste trabalho são:

1. **Framework SafeML pronto para produção:** Implementação completa em Python com arquitetura limpa (Clean Architecture + SOLID + DDD), 84 testes automatizados, API REST, containerização Docker e integração com pipelines MLOps. O código é disponibilizado publicamente para uso pela comunidade

acadêmica e industrial;

2. **Avaliação comparativa abrangente de medidas de distância:** Avaliação sistemática comparando cinco medidas (KS, Wasserstein, Anderson-Darling, Kuiper e CvM) em quatro domínios, identificando KS e Kuiper como as medidas mais robustas entre datasets, e documentando a falha sistemática da implementação AD com `anderson_ksamp`;

3. **Avaliação sob múltiplas perturbações adversariais:** Extensão do SafeML para FGSM, PGD, C&W, DeepFool e Ruído Gaussiano em quatro datasets, com análise separada por dataset e tipo de ataque;

4. **Diagnóstico do comportamento do SafeML em domínios complexos:** Identificação e explicação mecanística de por que o SafeML apresenta limitações no GTSRB (43 classes, features naïve de pixels brutos), com recomendação de uso de features de camadas intermediárias para datasets com muitas classes;

5. **Análise estatística rigorosa:** Correlações de Pearson com intervalos de confiança *bootstrap* (500 reamostras), AUC-ROC, análise de TPR/FPR por sub-lotes e comparação qualitativa com detectores alternativos (MSP, ODIN, Mahalanobis);

6. **Contextualização regulatória abrangente:** Mapeamento do SafeML às normas IEC 61508, ISO 21448, ISO/PAS 8800, EU AI Act e PL 2338/2023, com integração de HITL, EDDI e Score Unificado Safety-Cybersecurity.

6.3 Limitações

As limitações identificadas neste trabalho incluem:

- O monitoramento opera sobre lotes de dados, não oferecendo avaliação de confiança para predições individuais. A detecção é mais confiável com lotes maiores, o que introduz latência;

- A abordagem de ECDF por pixel para imagens pode não escalar eficientemente para resoluções muito altas (e.g., ImageNet 224×224), sendo recomendável a extração

de *features* de camadas intermediárias da rede em cenários de produção;

- Para datasets de imagem, o número de configurações de ataque avaliadas (5–8 por dataset) limita o poder estatístico dos testes de correlação de Pearson. A significância estatística das correlações pode ser obtida com maior número de configurações de ϵ ou com análise por ataque individual ao invés de por configuração;
- A avaliação foi realizada com dados estáticos (*offline*), e a viabilidade em cenários de *streaming* em tempo real não foi experimentalmente validada;
- Ataques adaptativos, projetados com conhecimento do monitoramento SafeML, não foram avaliados e representam um vetor de ameaça em aberto;
- Os *thresholds* foram determinados com dados de validação limpos e podem necessitar de recalibração periódica em ambientes com *dataset shift* legítimo;
- O dataset CICIDS2017 foi avaliado com dados sintéticos calibrados para reproduzir as propriedades estatísticas do dataset real. A validação com os CSVs originais do CICIDS2017 e ataques de *Data Poisoning* é proposta como trabalho futuro.

6.4 Trabalhos Futuros

Com base nos resultados e limitações identificados, os seguintes trabalhos futuros são propostos:

1. **Generalização do monitoramento sobre *features* de camadas intermediárias (prioridade alta):** A Seção 5.6.4 validou, em caráter exploratório, que monitorar o vetor de 512 dimensões da camada `avgpool` da ResNet-18 — em vez dos primeiros 20 pixels brutos — eleva substancialmente o AUC-ROC no GTSRB (por exemplo, de 0,33 para 0,86 na Anderson-Darling). O trabalho futuro consiste em *generalizar* essa estratégia: implementar *feature extraction hooks* configuráveis para arquiteturas arbitrárias, investigar a seleção automática da camada de extração e ponderar o custo de *model-agnosticism* associado (Seção 5.9.4), consolidando a extração por *features* como configuração padrão do SafeML para dados de imagem;

2. **Comparação quantitativa com detectores alternativos:** Avaliar o SafeML frente ao MSP (HENDRYCKS; GIMPEL, 2017), ODIN (LIANG; LI; SRIKANT, 2018) e Distância de Mahalanobis (LEE et al., 2018) nos mesmos datasets e configurações de ataque, fornecendo evidência empírica do posicionamento competitivo da abordagem;

3. **Validação da implementação Anderson-Darling corrigida:** A implementação direta da estatística A^2 (Equação 5.1, baseada em Pettitt (1976)) foi adotada neste trabalho após a falha da implementação original via `anderson_ksamp`. Uma validação rigorosa — comparando os valores A^2 com benchmarks analíticos conhecidos e verificando estabilidade numérica em múltiplas seeds — é proposta como continuidade. A corretude da fórmula implementada foi verificada qualitativamente ($A^2 \approx$ 0,3 para amostras da mesma distribuição, $A^2 > 100$ para distribuições com desvio-padrão 5 unidades de diferença), mas validação formal contra uma implementação de referência certificada fortaleceria a contribuição;

4. **Validação com CICIDS2017 real e Data Poisoning:** Utilização dos CSVs originais do CICIDS2017 com ataques de *Data Poisoning* (*label flipping*), avaliando a capacidade do SafeML de detectar degradação causada por contaminação dos dados de treinamento em dados de tráfego de rede reais;

5. **Monitoramento em tempo real:** Extensão do framework para processamento de *streaming* de dados, com janelas deslizantes para construção incremental de ECDFs operacionais e atualização dinâmica dos perfis de referência;

6. **Calibração empírica do Score Unificado:** Determinar os pesos w_{safety} e w_{cyber} por otimização bayesiana sobre conjuntos de cenários de falha rotulados, substituindo os valores heurísticos atuais por valores validados empiricamente;

7. **Robustez a ataques adaptativos:** Investigação de ataques que minimizam as métricas SDD enquanto maximizam o erro do classificador (ataques *SafeML-aware*), e desenvolvimento de defesas via *ensemble* de medidas;

8. **Extensão para modelos de linguagem:** Avaliação de se as medidas SDD sobre

embeddings de entrada podem detectar *prompt injection* e *jailbreaking* em LLMs.

Bibliografia

ABDAR, M. et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, v. 76, p. 243–297, 2021.

ANDERSON, T. W.; DARLING, D. A. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, v. 23, n. 2, p. 193–212, 1952.

ANGELOPOULOS, A. N.; BATES, S. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, v. 16, n. 4, p. 494–591, 2023.

ASLANSEFAT, K. et al. Toward improving confidence in autonomous vehicle software: A study on traffic sign recognition systems. *IEEE Computer*, v. 54, n. 8, p. 66–76, 2021.

ASLANSEFAT, K. et al. SafeDrones: Real-time reliability evaluation of UAVs using executable digital dependable identities. In: *Proceedings of the International Symposium on Model-Based Safety and Assessment (IMBSA 2022)*. [S.l.]: Springer, 2022. (Lecture Notes in Computer Science, v. 13525).

ASLANSEFAT, K. et al. SMILE: Statistical model-agnostic integrity levels for ML. *IEEE Software*, 2023.

ASLANSEFAT, K. et al. SafeML: Safety monitoring of machine learning classifiers through statistical difference measures. In: *Proceedings of the International Symposium on Model-Based Safety and Assessment (IMBSA 2020)*. [S.l.]: Springer, 2020. (Lecture Notes in Computer Science, v. 12297), p. 197–211.

BIGGIO, B.; NELSON, B.; LASKOV, P. Poisoning attacks against support vector machines. In: *Proceedings of the International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2012. p. 1467–1474.

BIGGIO, B.; ROLI, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, v. 84, p. 317–331, 2018.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. ISBN 978-0-387-31073-2.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001.

CARLINI, N.; WAGNER, D. Towards evaluating the robustness of neural networks. In: *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. [S.l.: s.n.], 2017. p. 39–57.

CHAKRABORTY, A. et al. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, v. 6, n. 1, p. 25–45, 2021.

CHENG, C.-H.; NÜHRENBERG, G.; YASUOKA, H. Runtime monitoring neuron activation patterns. In: *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*. [S.l.: s.n.], 2019.

COHEN, J.; ROSENFELD, E.; KOLTER, J. Z. Certified adversarial robustness via randomized smoothing. In: *Proceedings of the International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2019. p. 1310–1320.

CRAMÉR, H. On the composition of elementary errors. *Skandinavisk Aktuarietidskrift*, v. 11, p. 13–74, 1928.

Deutsches Institut für Normung. *DIN SPEC 92005:2024 — Artificial intelligence — Uncertainty quantification in machine learning*. [S.l.], 2024.

EFRON, B.; TIBSHIRANI, R. J. *An Introduction to the Bootstrap*. [S.l.]: Chapman and Hall/CRC, 1994. ISBN 978-0-412-04231-7.

European Parliament and Council of the European Union. *Regulation (EU) 2024/1689 — Artificial Intelligence Act*. 2024. Official Journal of the European Union.

EVANS, E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. [S.l.]: Addison-Wesley, 2003. ISBN 978-0-321-12521-7.

FAWCETT, T. An introduction to ROC analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861–874, 2006.

FERREIRA, J. et al. Safety monitoring of machine learning classifiers: A survey. *arXiv preprint arXiv:2412.06869*, 2024.

GAL, Y.; GHAHRAMANI, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: *Proceedings of the International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2016. p. 1050–1059.

GLIVENKO, V. I. Sulla determinazione empirica delle leggi di probabilità. *Giornale dell'Istituto Italiano degli Attuari*, v. 4, p. 92–99, 1933.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. ISBN 978-0-262-03561-3.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. [S.l.]: Pearson, 2009. ISBN 978-0-13-147139-9.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

HENDRYCKS, D.; GIMPEL, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2017.

HENZINGER, T. A.; LUKINA, A.; SCHILLING, C. Outside the box: Abstraction-based monitoring of neural networks. *arXiv preprint arXiv:1911.09032*, 2020.

HEVNER, A. R. et al. Design science in information systems research. *MIS Quarterly*, v. 28, n. 1, p. 75–105, 2004.

International Electrotechnical Commission. *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems*. [S.l.], 2010.

International Organization for Standardization. *ISO 21448:2022 — Road vehicles — Safety of the intended functionality*. [S.l.], 2022.

International Organization for Standardization. *ISO/PAS 8800:2024 — Road vehicles — Safety and artificial intelligence*. [S.l.], 2024.

KOLMOGOROV, A. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, v. 4, p. 83–91, 1933.

KRIZHEVSKY, A.; HINTON, G. *Learning Multiple Layers of Features from Tiny Images*. [S.l.], 2009.

KUIPER, N. H. Tests concerning random points on a circle. *Indagationes Mathematicae (Proceedings)*, v. 63, p. 38–47, 1960.

LAKSHMINARAYANAN, B.; PRITZEL, A.; BLUNDELL, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2017. v. 30.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LEE, K. et al. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2018. v. 31.

LIANG, S.; LI, Y.; SRIKANT, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2018.

MADRY, A. et al. Towards deep learning models resistant to adversarial attacks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2018.

MARTIN, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. [S.l.]: Prentice Hall, 2017. ISBN 978-0-13-449416-6.

MITRE Corporation. *ATLAS — Adversarial Threat Landscape for AI Systems*. 2025. ⟨https://atlas.mitre.org/⟩. Accessed: 2026-02-28.

MOHSENI, S. et al. Taxonomy of machine learning safety: A survey and primer. *ACM Computing Surveys*, v. 55, n. 8, p. 1–38, 2022.

MOOSAVI-DEZFOOLI, S.-M.; FAWZI, A.; FROSSARD, P. DeepFool: A simple and accurate method to fool deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 2574–2582.

NASCIMENTO, N.; ASLANSEFAT, K.; PAPADOPOULOS, Y. From fault tree to runtime assurance: Towards dependable AI through executable digital dependable identities. In: *Proceedings of the International Conference on Advanced Information Networking and Applications (AINA 2024)*. [S.l.]: Springer, 2024.

National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. [S.l.], 2023.

NICOLAE, M.-I. et al. Adversarial robustness toolbox v1.0.0. *arXiv preprint arXiv:1807.01069*, 2018.

PATERSON, C. et al. AMLAS: Assurance of machine learning for use in autonomous systems. *Reliability Engineering and System Safety*, 2025.

PEFFERS, K. et al. A design science research methodology for information systems research. *Journal of Management Information Systems*, v. 24, n. 3, p. 45–77, 2007.

PETTITT, A. N. A two-sample anderson-darling rank statistic. *Biometrika*, v. 63, n. 1, p. 161–168, 1976.

RAUBER, J. et al. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. In: *ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning*. [S.l.: s.n.], 2020.

Senado Federal do Brasil. *PL 2338/2023 — Marco Legal da Inteligência Artificial no Brasil*. 2023. Senado Federal.

SHAFAHI, A. et al. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2018. v. 31.

SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*. [S.l.: s.n.], 2018. p. 108–116.

SMIRNOV, N. Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, v. 19, n. 2, p. 279–281, 1948.

STALLKAMP, J. et al. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2011. p. 1453–1460.

SZEGEDY, C. et al. Intriguing properties of neural networks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2014.

TRAMÈR, F. et al. On adaptive attacks to adversarial example defenses. In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2020. v. 33, p. 1633–1645.

VASERSTEIN, L. N. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, v. 5, n. 3, p. 64–72, 1969.

VOVK, V.; GAMMERMAN, A.; SHAFER, G. *Algorithmic Learning in a Random World*. [S.l.]: Springer, 2005. ISBN 978-0-387-00152-4.

WASSERMAN, L. *All of Statistics: A Concise Course in Statistical Inference*. [S.l.]: Springer, 2004. ISBN 978-0-387-40272-7.

WOHLIN, C. et al. *Experimentation in Software Engineering*. Berlin, Heidelberg: Springer, 2012.

WONG, E.; KOLTER, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In: *Proceedings of the International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2018. p. 5286–5295.

XU, H. et al. Adversarial attacks and defenses in deep learning. *Engineering*, v. 6, n. 3, p. 346–360, 2020.

YANG, J. et al. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, v. 132, p. 4132–4163, 2024.