Classificação de Risco de Vulnerabilidades de Segurança via Processos Gaussianos e Aprendizado Ativo

Davyson S. Ribeiro[1], Rafael Lemos[1], Francisco R. P. da Ponte[1], César Lincoln C. Mattos[1], Emanuel B. Rodrigues[1]

[1]Universidade Federal do Ceará (UFC)
Av. da Universidade, 2853 – CEP 60020-181 – Fortaleza – CE – Brasil

{davysonribeiro, rafael.lemos}@alu.ufc.br,
fco.rparente@gmail.com,
{cesarlincoln, emanuel}@dc.ufc.br,

Abstract. *Effective vulnerability management is essential for cybersecurity, but the lack of skilled professionals makes this task challenging. Expert data labeling, in conjunction with machine learning techniques, seeks to obtain models capable of emulating the experience of security professionals. This paper investigates the feasibility of using Gaussian Processes (GPs) with Active Learning to classify security vulnerabilities according to their risk of exploitation. The aim is to reduce the labeled data required for an effective classifier. The proposed methodology combines the uncertainties in predictions provided by GP models with five data selection strategies for labeling available in the literature. The experiments used the recently published CVEjoin data set, which contains information about more than 200,000 vulnerabilities. Three evaluation scenarios are considered, all with the same amount of labeled data but different amounts of Active Learning iterations. The BSB strategy performed best in accuracy and F1 score, especially with more labeling iterations.*

Resumo. *O gerenciamento eficaz de vulnerabilidades é essencial para a segurança cibernética, mas a falta de profissionais especializados torna essa tarefa desafiadora. A rotulação de dados por especialistas em conjunto com técnicas de aprendizado de máquina busca obter modelos capazes de emular a experiência de profissionais da área de segurança. Este trabalho investiga a viabilidade do uso de Processos Gaussianos (GPs) com Aprendizado Ativo para classificar vulnerabilidades de segurança conforme seu risco de exploração. O objetivo é reduzir a quantidade de dados rotulados necessários para obter um classificador eficaz. A metodologia proposta combina as incertezas nas predições fornecidas pelos modelos de GPs com cinco estratégias de seleção de dados para rotulação disponíveis na literatura. Os experimentos realizados utilizam o conjunto de dados CVEjoin, publicado recentemente, que contém informações sobre mais de 200.000 vulnerabilidades. São considerados três cenários de avaliação, todos com a mesma quantidade total de dados rotulados, mas diferentes quantidades de iterações de Aprendizado Ativo. A estratégia Best and Second Best (BSB) apresentou o melhor desempenho em termos de acurácia e F1-score, especialmente no cenário em que há mais iterações de rotulação.*

1. Introdução

O constante aumento de dispositivos conectados e a complexidade dos sistemas de tecnologia da informação pode resultar no aumento da superfície de ataque de ambientes computacionais, os deixando mais vulneráveis a ameaças cibernéticas [Jakkal 2022]. Conforme definido pelo Instituto Nacional de Padrões e Tecnologia (NIST - *National Institute of Standards and Technology*), vulnerabilidades são fraquezas em um sistema de informação ou implementações que podem ser exploradas por uma fonte de ameaça [Ross 2012].

Uma das abordagens para resolver esse problema envolve estratégias que auxiliem analistas de segurança cibernética na correção proativa de vulnerabilidades, ou seja, antes que estas possam ser exploradas por agentes mal-intencionados. Essa disciplina é chamada de Gestão de Vulnerabilidades, e constitui um processo contínuo que visa identificar, categorizar, priorizar e remediar vulnerabilidades presentes em softwares, aplicações e sistemas operacionais [Foreman 2019]. A gestão de vulnerabilidades é crucial para controlar as possíveis falhas em um ambiente computacional. É importante que haja uma gestão eficaz, capaz de priorizá-las conforme sua criticidade e possibilidade de exploração [Sabottke et al. 2015]. Para isso é fundamental que sejam coletadas informações sobre as vulnerabilidades, como por exemplo as contidas nos CVEs (*Common Vulnerabilities and Exposures*) organizados na base de dados NVD (*National Vulnerability Database*).

Dessa forma, torna-se necessária a existência de níveis de classificação de risco das vulnerabilidades do ambiente, que devem ser classificadas quanto ao seu nível de impacto para uma instituição. Devido ao grande fluxo e variedade dos dados coletados em um ambiente computacional, as equipes de segurança enfrentam o desafio de analisar essa grande quantidade de dados de forma minuciosa, para então tomar decisões eficazes com base nas vulnerabilidades encontradas e que representam perigo para as organizações [Tenable 2023]. A quantidade de profissionais de segurança especializados para realizar essa tarefa é pequena quando comparada à grande quantidade de sistemas e vulnerabilidades existentes [Hore et al. 2023]. Rotular adequadamente as fraquezas da rede quanto ao risco é uma tarefa difícil e requer tempo de equipes de segurança especializadas, sendo o processo de rotulação extremamente importante para que haja correção das vulnerabilidades de um ambiente computacional e uma vez que esse processo ocorre, as equipes de segurança têm seu tempo otimizado e o esforço pode ser focado na solução de problemas, e não mais em rotulação [Ponte et al. 2023a].

Dada a necessidade de priorizar vulnerabilidades e os recursos limitados das equipes de segurança da informação, o uso de Aprendizado de Máquina (ML - *Machine Learning*) tem sido incluído nesse contexto para auxiliar na classificação das vulnerabilidades quanto ao risco para diferentes organizações [Alshaya et al. 2023]. Nesse sentido, o Aprendizado Ativo (AL - *Active Learning*), subtópico do ML, é capaz de identificar os exemplos mais informativos para serem encaminhados à etapa de rotulação, reduzindo a quantidade de dados rotulados usados no treinamento e o correspondente esforço de avaliação por um ou mais especialistas. O resultado final é um modelo capaz de simular a experiência de especialistas em segurança da informação na avaliação do risco de gestão de vulnerabilidades [Ponte et al. 2023a].

Modelos de Processos Gaussianos (GP - *Gaussian Processes*) são métodos de aprendizado supervisionado Bayesianos não paramétricos que tendem a apresentar bons

resultados na presença de poucos dados rotulados para treinamento [Williams e Rasmussen 2006]. Por ser uma abordagem Bayesiana, um modelo de GP retorna distribuições de probabilidade em suas predições, ao invés de estimações pontuais. Pela sua capacidade inerente de quantificar a incerteza nas predições fornecidas, GP são bons candidatos a serem usados no contexto de AL.

Neste trabalho, é investigada a viabilidade da aplicação de um modelo de ML utilizando GP em uma metodologia de AL para a classificação de vulnerabilidades de segurança quanto ao seu risco. A proposta visa reduzir o esforço humano na atividade de rotulação enquanto mantém resultados de generalização satisfatórios. Serão exploradas diferentes estratégias de AL para identificar qual delas apresenta melhor desempenho em cenários com poucos dados rotulados disponíveis. A avaliação dessa combinação entre GP e AL será dada ao testar diferentes cenários de interação entre o AL e o especialista, bem como diversas iterações do AL. Será possível observar que o desempenho do modelo proposto tende a melhorar ao longo do tempo, à medida que mais exemplos informativos são rotulados e adicionados ao conjunto de treinamento.

2. Trabalhos Relacionados

Nesta seção é apresentada uma revisão da literatura sobre técnicas de ML e AL cooperando no processo de gestão de vulnerabilidades de segurança (fases de detecção e avaliação), com foco maior na tarefa de classificação de risco das vulnerabilidades.

Kashyap et al. [2022] discutiram a detecção de ataques cibernéticos em sistemas de tráfego automotivo. Os autores elaboraram um modelo com base em GP para identificar veículos maliciosos em ambientes de tráfego misto. Foi discutido a possibilidade de investigar e integrar outros métodos de detecção de anomalias e ML para complementar o modelo baseado em GP, com o objetivo de melhorar a capacidade de identificar e responder a ataques cibernéticos de forma mais ampla e eficaz.

Sun et al. [2023] apresentaram o desenvolvimento de um framework, denominado ASSBert, que combinou AL e aprendizado semi-supervisionado para a detecção de vulnerabilidades em contratos inteligentes. O trabalho destacou que a detecção eficaz de vulnerabilidades em contratos inteligentes enfrenta o desafio da escassez de dados rotulados, por isso o framework utiliza ML para selecionar eficientemente dados valiosos para aumentar o desempenho do modelo de detecção. Em experimentos, a aplicação se mostrou superior aos métodos convencionais, mesmo com uma pequena quantidade de dados rotulados e uma grande quantidade de dados não rotulados.

Kure et al. [2022] propuseram um método integrado para a avaliação de riscos em Sistemas Ciberfísicos (CPS), organizado em três etapas distintas. Na primeira etapa, os ativos são classificados utilizando lógica difusa para determinar sua criticidade. Os analistas respondem a cinco perguntas sobre o potencial impacto na Confidencialidade, Integridade e Disponibilidade (CIA) do ativo, bem como o tempo necessário para recuperação após um ataque. Na segunda etapa, um modelo de ML é empregado para prever a vulnerabilidade dos ativos a dez tipos de ataques cibernéticos específicos. Por fim, na terceira etapa, os analistas avaliam a conformidade dos controles de segurança da empresa com os requisitos estabelecidos pelo ISO 27005 [Firoiu 2015]. Os experimentos realizados demonstraram a eficácia da classificação da criticidade dos ativos e da avaliação dos controles de segurança.

Elbaz et al. [2021] adotaram a técnica de AL para etiquetar um conjunto de

dados contendo informações de CPE (Identificadores Únicos de Software), extraídos das descrições de vulnerabilidades divulgadas pelo NIST. Esses dados rotulados foram utilizados para treinar um modelo de ML capaz de classificar as vulnerabilidades em três níveis distintos: (I) LOG, onde a vulnerabilidade é registrada sem alertas imediatos; (II) TICKET, onde um chamado é gerado para resolução durante o horário comercial; e (III) ALERT, indicando uma vulnerabilidade crítica que requer ação imediata. No entanto, essa solução apresenta limitações, uma vez que se baseia apenas nas informações do CPE, que frequentemente estão ausentes no momento da publicação da vulnerabilidade. Além disso, não é levada em consideração outras informações cruciais sobre as vulnerabilidades, como inteligência de ameaças e contexto específico, o que pode afetar a precisão das decisões de alerta para os analistas de segurança.

Por fim, Ponte et al. [2023a], apresentaram uma metodologia baseada em AL para criar um modelo supervisionado capaz de emular a experiência de especialistas na avaliação de riscos de vulnerabilidades. O estudo destacou a importância de considerar informações de vulnerabilidade, inteligência de ameaças e contexto para uma avaliação eficaz de riscos, em contraste com práticas inadequadas que subestimam a probabilidade e o impacto da exploração de vulnerabilidades. Os experimentos realizados demonstraram que a solução alcançou uma alta precisão na identificação de vulnerabilidades críticas, comparável ao desempenho dos analistas. A abordagem baseada em AL mostrou-se eficaz na classificação de riscos, superando rapidamente a seleção aleatória de instâncias, mesmo em um cenário com um grande número de vulnerabilidades.

O presente trabalho se diferencia significativamente dos demais ao explorar mais estratégias sofisticadas para calcular a incerteza, combinando AL com o modelo GP de aprendizado supervisionado não paramétrico. A escolha de GP em nossa pesquisa se deve à sua capacidade de proporcionar estimativas probabilísticas em suas predições, o que é crucial para quantificar a incerteza nos modelos utilizados no contexto de segurança da informação. Além disso serão considerados diferentes cenários de avaliação para cada uma das estratégias, todas com a mesma quantidade total de dados rotulados, mas diferentes quantidades de iterações de AL. Nesse contexto, a Tabela 1 sumariza as diferenças entre os trabalhos relacionados e o nosso.

Tabela 1: Comparação entre os trabalhos relacionados e o presente artigo.

	Detecção de Vulnerabilidades	**Classificação de Risco**	**Classificadores de Aprendizado de Máquina**	**Medição de Incerteza - Aprendizado Ativo**
Kashyap et al. [2022]	✓		GPR	
Sun et al. [2023]	✓		Bert, Bert-AL Bert-SSL, ASSBert	Entropy
Kure et al. [2022]		✓	KNN, NN, DT, RF LR, NBM, NB	
Elbaz et al. [2021]		✓	CRFs	Least Confident
Ponte et al. [2023a]		✓	RF, GB, RL SVC, MLP	Entropy
Este Trabalho		✓	GP	Entropy, Least Confident, BSB, GPLCB, Random

3. Ciclo de Aprendizado e Estratégias Utilizadas

Os dados analisados constituem um conjunto estruturado para o estudo de vulnerabilidades específicas, cada amostra composta por atributos coletados e organizados, que detalham a probabilidade de exploração de uma vulnerabilidade e as características dos ativos tecnológicos afetados. As amostras são rotuladas de acordo com uma classificação de risco: baixa (*LOW*), moderada (*MODERATE*), importante (*IMPORTANT*) ou crítica (*CRITICAL*). Essas classificações são cruciais para determinar a urgência de resposta e as medidas de mitigação necessárias contra potenciais ataques cibernéticos. A análise desses dados é fundamental para compreender e gerenciar ameaças cibernéticas organizacionais, oferecendo uma base sólida para implementar estratégias defensivas eficazes e aumentar a resiliência das infraestruturas digitais contra vulnerabilidades identificadas.

A metodologia proposta utiliza um modelo de GP para AL, visando melhorar a atividade de classificação das vulnerabilidades. O modelo de ML supervisionado adotado oferece uma abordagem probabilística flexível para previsões, que não só estima valores específicos, mas também quantifica a incerteza associada a essas previsões. Será discutido e analisado o desempenho de cada estratégia de quantificação de incerteza relacionada ao AL, validando suas aplicações no contexto de classificação de vulnerabilidades.

Modelos de aprendizado supervisionado, em geral, são dependentes da quantidade e qualidade dos dados rotulados disponíveis. Nesse cenário, o AL surge como uma abordagem complementar, permitindo que o modelo solicite de forma iterativa e interativa as instâncias de dados mais informativas para rotulagem. As iterações se referem ao fato de ser um procedimento executado em ciclos que se repetem, a interatividade refere-se à troca de informações entre o modelo e o agente rotulador, usualmente um especialista na área de estudo. Esta abordagem pode melhorar a eficiência da etapa de aprendizado, garantindo que o modelo final seja eficaz mesmo na presença de uma pequena quantidade de dados rotulados [Swiler et al. 2020].

3.1. Ciclo Iterativo do Aprendizado Ativo

Na Figura 1 é possível observar a divisão do ciclo iterativo do AL em várias etapas distintas, sendo elas: (1) Inicialização, (2) Treinamento do Modelo de ML, (3) Avaliação da Incerteza, (4) Seleção de Amostras, (5) Rotulação de Amostras e (6) Atualização do Conjunto de Treinamento. Logo após são especificadas cada uma dessas etapas.

Na fase de **Inicialização**, é definido um conjunto de dados inicial pequeno, rotulado, que será utilizado para treinar o modelo de ML. Esse conjunto inicial pode ser selecionado de maneira aleatória ou seguindo critérios específicos. A fase de **Treinamento do Modelo** ocorre utilizando o conjunto de dados rotulados disponível. Durante essa fase, o modelo é ajustado para minimizar a diferença entre as previsões e os rótulos verdadeiros das amostras de treinamento. Logo após, ocorre a **Avaliação da Incerteza**, onde o modelo é aplicado ao conjunto de dados não rotulados (chamado de *pool*, na literatura) para estimar a incerteza associada a cada amostra. A fase de **Seleção de Amostras** acontece com base nas estimativas de incerteza obtidas na etapa anterior, onde são selecionadas uma ou mais amostras mais informativas e incertas para serem rotuladas. Na fase de **Rotulação de Amostras**, ocorre o processo manual de rotulação por um especialista humano. Em ambiente experimental, para a realização de avaliações sistemáticas, o especialista pode ser substituído por um oráculo, ou seja, um sistema simulado que fornece automaticamente os

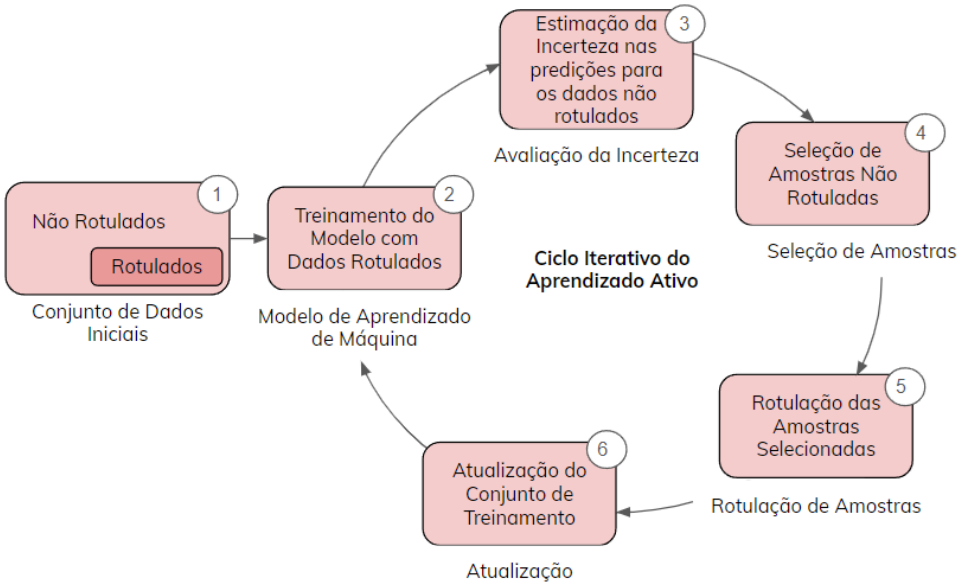

Figura 1: Ciclo iterativo do Aprendizado Ativo.

rótulos necessários para as amostras selecionadas, a partir de uma base de conhecimento não disponibilizada ao modelo. Os novos dados rotulados são então adicionados ao conjunto de treinamento do modelo. Por fim, temos a **Atualização do Conjunto de Treinamento**, com as novas informações oriundas da rotulação das amostras selecionadas. Isso permitirá que o modelo incorpore os novos exemplos rotulados durante o próximo ciclo do processo de AL. Após essa fase, retorna-se ao processo de treinamento e o ciclo se repete.

3.2. Estratégias de Seleção para Aprendizado Ativo

Os critérios de incerteza são métricas usadas para determinar quais amostras de dados são mais úteis para rotular e adicionar ao conjunto de treinamento. A justificativa por trás da seleção dessas amostras é devido ao modelo estar mais incerto sobre suas previsões [Pereira-Santos et al. 2019]. Antes da aplicação do critério de incerteza, é necessário realizar o cálculo da classe predita. Modelos de GP somente possuem inferência analítica em tarefas de regressão com Verossimilhança Gaussiana [Rasmussen e Williams 2006]. Como o problema tratado envolve uma classificação multiclasse, optou-se pelo uso de uma verossimilhança categórica, implementada via função softmax.

Seguindo estratégias de GPs esparsos e inferência variacional [Hensman et al. 2013; Hensman et al. 2015], as predições do modelo são aproximadas por amostras de Monte Carlo, que são passadas para a função softmax para resultar em probabilidades. Em seguida, a média da probabilidade de cada classe pode ser calculada tomando-se a média das amostras geradas. Mais especificamente, considerando-se uma entrada $\boldsymbol{x}_*$, representando uma vulnerabilidade sem rótulo, e a correspondente s-ésima amostra de Monte Carlo $f_c^{(s)}(\boldsymbol{x}_*)$ da posteriori do modelo para a classe de risco c, a probabilidade da

saída (classe predita) y_* será dada por:

$$P(y_* = c \mid \boldsymbol{x}_*) = \frac{1}{S} \sum_s \text{softmax}(f_c^{(s)}(\boldsymbol{x}_*)), \tag{1}$$

$$\text{em que } \text{softmax}(f_c^{(s)}(\boldsymbol{x}_*)) = \frac{\exp(f_c^{(s)}(\boldsymbol{x}_*))}{\sum_c \exp(f_c^{(s)}(\boldsymbol{x}_*))}, \tag{2}$$

Sendo S o total de amostras de Monte Carlo. A classe predita será aquela com maior probabilidade média, i.e., $c_* = \arg\max_c P(y_* = c \mid \boldsymbol{x}_*)$. Ao final desses passos, aplica-se o critério de incerteza desejado. O padrão com maior valor para o critério em questão será o escolhido para ser rotulada. A seguir são detalhados os critérios usados neste trabalho.

Least Confident. Esse critério é dado pelo complemento da maior probabilidade média entre as classes. Trata-se de uma estratégia simples de implementar, entretanto em conjuntos de dados com muitas variações e ruídos pode não ser suficiente para melhorar o modelo. Este critério é calculado por

$$\text{lc}(\boldsymbol{x}_*) = 1 - P(y_* = \ c_* \mid \boldsymbol{x}_*). \tag{3}$$

Nota-se que um maior valor para $\text{lc}(\boldsymbol{x}_*)$ está relacionado a uma menor confiança na predição final.

BSB - *Best and Second Best.* Também chamada de critério de margem, mede a incerteza de um modelo de ML considerando a diferença entre as duas maiores probabilidades preditas para cada amostra, que representa uma vulnerabilidade. É particularmente eficaz em identificar amostras onde o modelo tem dificuldade em distinguir entre duas ou mais classes, especialmente em situações de confiança similar em várias classes. O BSB é menos sensível a classes desbalanceadas em comparação com outros critérios, fornecendo uma medida mais equilibrada de incerteza [Joshi et al. 2009]. Matematicamente, para uma entrada $\boldsymbol{x}_*$, seja $P(y = c_1 \mid \boldsymbol{x}_*)$ a probabilidade predita da classe mais provável c_1, e $P(y = c_2 \mid \boldsymbol{x}_*)$ a probabilidade predita da segunda classe mais provável c_2, a medida de incerteza baseada no BSB é definida por

$$\Delta(\boldsymbol{x}_*) = P(y = c_1 \mid \boldsymbol{x}_*) - P(y = c_2 \mid \boldsymbol{x}_*). \tag{4}$$

Entropy. Quantifica a incerteza ou desordem associada a uma distribuição de probabilidade. As amostras com alta entropia indicam maior incerteza do modelo, pois têm distribuições de probabilidade mais equilibradas entre as classes. Este critério é útil para identificar regiões do espaço de entrada onde o modelo tem menor confiança em suas previsões. A entropia captura a incerteza do modelo de forma mais completa, especialmente em situações com distribuições de probabilidade desbalanceadas, contudo pode ser computacionalmente mais custosa de calcular, especialmente em modelos com muitos rótulos, sendo sensível a desbalanceamento de classes, onde a incerteza pode ser superestimada para classes minoritárias [Joshi et al. 2009]. Para distribuições discretas, como é o caso de tarefas de classificação, a entropia $H(\boldsymbol{x}_*)$ é definida por

$$H(\boldsymbol{x}_*) = -\sum_c P(y = c \mid \boldsymbol{x}_*) \log P(y = c \mid \boldsymbol{x}_*). \tag{5}$$

GPLCB - Gaussian Process Lower Confidence Bound. Estima a incerteza ao considerar a média das probabilidades preditivas e o desvio padrão associado. O desvio é estimado a partir dos valores softmax calculados a partir das amostras de Monte Carlo, sendo dado por $\sigma_c(\boldsymbol{x}_*) = \sqrt{\mathbb{V}[\text{softmax}(f_c(\boldsymbol{x}_*))]}$, em que $\mathbb{V}[\cdot]$ é o estimador de variância amostral, calculada considerando as S amostras $f_c^{(s)}(\boldsymbol{x}_*)$. As amostras com um limite inferior de confiança mais baixo indicam maior incerteza do modelo [Garnett 2023]. O limite inferior de confiança é dado por

$$\text{GPLCB}(\boldsymbol{x}_*) = 1 - (P(y = c_* \mid \boldsymbol{x}_*) - \beta\sigma_c(\boldsymbol{x}_*)), \tag{6}$$

em que β é um parâmetro que regula o tamanho do intervalo de confiança a ser considerado.

Random. Consiste na seleção aleatória de amostras do conjunto de dados para rotulação, sem levar em conta a incerteza do modelo. Sua aplicação geralmente é levada em consideração como linha de base (referência) na avaliação de métodos alternativos de seleção de amostras [Pereira-Santos et al. 2019].

4. Experimentos

4.1. Conjunto de Dados

O presente estudo utilizou o conjunto de dados CVEjoin[1], que foi desenvolvido para auxiliar analistas e pesquisadores na análise de risco, classificação da criticidade de vulnerabilidades e outras atividades relacionadas à segurança da informação. Este conjunto de dados, disponível publicamente, contém informações sobre mais de 200.000 vulnerabilidades coletadas de diversas fontes [Ponte et al. 2023b].

O CVEjoin enfatiza a importância de considerar não apenas o escore CVSS (*Common Vulnerability Scoring System*), mas também a inteligência de ameaças e informações contextuais para avaliar adequadamente o risco de exploração de vulnerabilidades. O conjunto aplicado nos experimentos realizados nesse trabalho contém 208 amostras rotuladas, 29 atributos e 4 classes, classificando as vulnerabilidades de acordo com o risco para o ambiente, definindo rótulos de risco das vulnerabilidades como: baixa, moderada, importante ou crítica. Alguns dos atributos utilizados são características da vulnerabilidade, como o impacto da vulnerabilidade na confidencialidade, integridade e disponibilidade, e o escore CVSS. Além disso, incluem-se atributos de inteligência de ameaças, como a presença de exploração pública, feeds do Twitter e Google Trends.

4.2. Avaliação do modelo de Aprendizado de Máquina

Diversas métricas serão utilizadas para avaliar as estratégias de AL e o modelo de ML. A **acurácia** é uma medida geral da exatidão do modelo, representando a proporção de predições corretas em relação ao total de amostras. A **precisão** mede a proporção de verdadeiros positivos em relação ao total de predições positivas feitas pelo modelo. O **recall** avalia a capacidade do modelo em identificar corretamente todas as amostras positivas, sendo a proporção de verdadeiros positivos em relação ao total de amostras positivas na base de dados. Por fim, o **F1-Score** é uma combinação da precisão e recall em uma única medida, calculada pela média harmônica entre essas duas métricas, proporcionando uma

[1] https://github.com/rodrigoparente/cvejoin-security-dataset

avaliação balanceada do desempenho do modelo em relação à sua capacidade de predição e capacidade de capturar todos os exemplos positivos de forma eficiente [Géron 2019].

Para realização deste estudo foi utilizada a biblioteca GPyTorch que otimiza a utilização do hardware para aplicações de modelos de GPs. Na Tabela 2 são descritas as configurações utilizadas para o AL. **Tamanho Inicial** representa a quantidade de dados rotulados usada para treinar o modelo antes de iniciar o processo de AL, sendo dimensionado proporcionalmente ao número de classes. **Iterações Ativas** representam o número de ciclos de AL executados, onde novas amostras são selecionadas e adicionadas ao conjunto de treinamento. **Seleção Ativa por Iteração** configura o número de amostras escolhidas em cada iteração para serem rotuladas e incluídas no treinamento. **Estratégias de Seleção** representam os diferentes métodos para selecionar as amostras mais informativas (descritas no subseção 3.2). Foram realizados experimentos para avaliar as estratégias de AL repetidas vezes de forma independente, aumentando a robustez estatística e a confiabilidade dos resultados. Com mais repetições, a média e o desvio padrão das métricas de desempenho fornecem uma estimativa mais precisa do desempenho real do modelo, reduzindo o viés aleatório causado por divisões específicas dos dados.

Tabela 2: Configurações do Aprendizado Ativo usadas nos experimentos.

Configurações	**Valor**
Tamanho Inicial	$10\times$ Número de classes
Iterações Ativas	Variação de acordo com o cenário
Seleção Ativa por Iteração	Variação de acordo com o cenário
Estratégias de Seleção	Random, Least Confident, Entropy, BSB, GPLCB
Número de repetições independentes	30
Divisão de dados (Treino/Teste)	90%/10%

Do mesmo modo, a Tabela 3 apresenta as configurações utilizadas pelo GP. **Número de Tarefas** refere-se à quantidade de tarefas simultâneas que o modelo de GP precisa resolver. Neste caso, é o dobro do número de classes para permitir a modelagem multitarefa. **Taxa de Aprendizado** é um parâmetro que controla a velocidade de ajuste do modelo durante o treinamento a fim de garantir a convergência do modelo. **Número de Pontos de Indução** é utilizado para aproximar o GP em grandes conjuntos de dados. **Amostras de Verossimilhança** representam a quantidade de amostras usadas tanto no treinamento quanto no teste para estimar a verossimilhança dos dados, influenciando a precisão e a robustez do modelo. **Número de Épocas** representa o total de passagens completas através do conjunto de dados de treinamento. **Kernel** é usado para medir a similaridade entre dados, sendo o kernel RBF, dado por $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp(-\frac{1}{2}\sum_d w_d^2(x_d - x'_d)^2)$, uma escolha comum na literatura. Os valores σ_f^2 e $w_d^2|_{d=1}^D$, em que D é a dimensão de entrada, são hiperparâmetros ajustados automaticamente durante o próprio treinamento [Hensman et al. 2013; Rasmussen e Williams 2006].

Essas configurações utilizadas em conjunto proporcionam uma avaliação mais justa e equilibrada do desempenho do modelo e das estratégias de seleção ativa, otimizando a avaliação comparativa. Pois com mais dados, podemos comparar melhor as diferentes estratégias de AL, identificando com mais clareza qual estratégia tende a performar melhor em média. Por outro lado sofremos penalidades como o aumento do tempo de execução,

Tabela 3: Configurações do modelo de Processo Gaussiano.

Configurações	**Valor**
Número de tarefas	$2 \times$ número de classes
Taxa de aprendizado	0,05
Número de pontos de indução	$5 \times$ número de classes
Amostras de Monte Carlo (treino)	100
Amostras de Monte Carlo (teste)	1000
Número de épocas	3000
Kernel	RBF

porque cada repetição envolve a realização de um conjunto completo de experimentos de AL, o que inclui o re-treinamento do modelo, a seleção ativa de exemplos e a avaliação do desempenho, além de maior uso de recursos computacionais. O conjunto de dados também passou por uma fase de pré-processamento, uma vez que contém atributos que são contínuos, discretos, booleanos e categóricos. Esse processo envolveu transformar os dados categóricos em valores numéricos usando a técnica de *one-hot-encoding*, pois os elementos não possuem nenhuma hierarquia entre eles e adicionalmente, quando preciso, foi efetuada a normalização dos dados com média zero e desvio padrão unitário.

Para a solução proposta, os experimentos foram divididos em três cenários que emulam diferentes níveis de intervenção de um especialista em segurança da informação para classificar o risco das vulnerabilidades. Os cenários de AL exploram como a frequência de interação com o especialista e o número de iterações do algoritmo influenciam na acurácia da classificação do risco das vulnerabilidades. No Cenário I, são realizadas 100 consultas individuais para rotular cada vulnerabilidade, resultando em 100 iterações do algoritmo de AL. Essa abordagem permite uma análise detalhada e progressiva, porém requer um investimento significativo de tempo e esforço tanto do especialista quanto do sistema computacional, que precisa retreinar o modelo a cada iteração. O Cenário II apresenta um meio-termo, com 20 iterações em que cada uma envolve a rotulação de um conjunto de 5 vulnerabilidades, equilibrando a necessidade de interação especializada com a eficiência do processo de AL. Já o Cenário III busca minimizar as interações, com apenas 10 consultas para rotular 10 vulnerabilidades de uma vez, resultando em 10 iterações do algoritmo. Essa estratégia reduz a demanda sobre o especialista, mas pode afetar a precisão da análise devido à menor frequência de retreinamentos do modelo.

5. Resultados

Nesta seção, são apresentadas tabelas com a média e o desvio padrão das métricas de desempenho para diferentes estratégias de seleção de amostras no final do ciclo de AL. Também são exibidos gráficos comparando as curvas de acurácia média de diferentes estratégias de AL ao longo das iterações de seleção, rotulação e retreinamento do modelo. Por fim, é feito o cálculo da métrica AUC *(Area Under the Curve)*, que representa a área sob a curva de acurácia média, e quantifica melhor o desempenho ao longo das iterações, enquanto as outras métricas só capturam o estado final. Todas as métricas foram calculadas com base em 30 repetições independentes dos experimentos para assegurar robustez estatística.

5.1. Resultados do Cenário I

A Tabela 4 apresenta os resultados da média e desvio padrão das métricas alcançadas pelo modelo de classificação GP ao final do ciclo de diferentes estratégias de AL no cenário I. Nesse cenário ocorrem 100 iterações de AL, onde em cada iteração, uma vulnerabilidade é selecionada por vez pela estratégia e rotulada por um especialista.

Tabela 4: Média e desvio padrão de diferentes métricas de avaliação do modelo de classificação GP considerando várias estratégias de aprendizado ativo no Cenário I.

Estratégia	**Acurácia** $\mu \pm \sigma$	**Precisão** $\mu \pm \sigma$	**Recall** $\mu \pm \sigma$	**F1-score** $\mu \pm \sigma$
BSB	0.78 ± 0.05	0.83 ± 0.06	0.78 ± 0.05	0.78 ± 0.06
Entropy	0.77 ± 0.05	0.86 ± 0.06	0.77 ± 0.05	0.77 ± 0.06
GPLCB	0.76 ± 0.05	0.80 ± 0.02	0.76 ± 0.06	0.76 ± 0.07
Least Confident	0.78 ± 0.04	0.82 ± 0.06	0.77 ± 0.04	0.78 ± 0.05
Random	0.75 ± 0.05	0.78 ± 0.06	0.75 ± 0.05	0.74 ± 0.06

Ao analisar os resultados, observa-se que a estratégia Random (seleção aleatória) apresenta o pior desempenho com os menores valores gerais. Por outro lado, destaca-se que todas as outras estratégias de AL que utilizam algum cálculo de incerteza apresentam métricas melhores que o Random, e próximas entre si. As diferenças de desempenho entre as estratégias podem ser atribuídas à maneira como cada uma delas seleciona os exemplos para rotulação, sendo que a seleção aleatória não proporciona esses benefícios, o que se reflete em valores inferiores nas métricas de avaliação.

Considerando as quatro métricas (acurácia, precisão, recall e f1-score) em conjunto, percebe-se que a estratégia BSB demonstra o melhor desempenho geral. O BSB é eficaz na identificação correta de verdadeiros positivos enquanto minimiza falsos positivos e negativos, resultando em um desempenho equilibrado. Esta estratégia mostra eficácia especial na identificação de amostras em que o modelo enfrenta dificuldades para distinguir entre duas ou mais classes, principalmente em cenários de confiança similar entre diversas classes. Além disso, ele é menos sensível a desequilíbrios de classes em comparação com outros critérios, proporcionando uma medida mais equilibrada de incerteza.

A Figura 2 mostra as curvas de acurácia média da estratégia que apresenta melhor desempenho (BSB) e da estratégia de *benchmark* (Random), em função do número de dados rotulados com suas respectivas AUCs. Com este gráfico, pode-se observar o comportamento das estratégias ao longo de todo o ciclo iterativo do AL.

Como esperado, percebe-se que a acurácia do modelo de classificação do risco de vulnerabilidades aumenta à medida que novas amostras selecionadas pelas estratégias de AL são rotuladas pelo especialista e utilizadas para retreinamento do modelo. Além disso, observa-se que à medida que esse processo avança com mais dados rotulados, a estratégia BSB começa a apresentar um desempenho superior ao Random, culminando com uma acurácia final de 0,78, conforme observado na tabela 4. A acurácia maior do BSB ao longo das iterações do AL é demonstrado quantitativamente com um valor de AUC maior quando comparado à estratégia Random, conforme informado na legenda do gráfico.

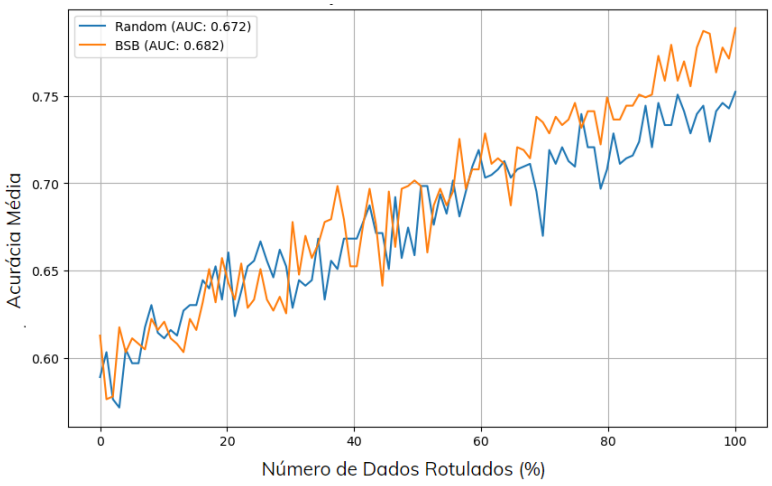

Figura 2: Acurácia média e AUC em função do número de dados rotulados no Cenário I.

5.2. Resultados do Cenário II

A Tabela 5 apresenta os resultados do cenário II, onde 20 iterações do AL são executadas e 5 vulnerabilidades são rotuladas por vez pelo especialista. Conforme observado no cenário anterior, a estratégia de seleção aleatória também é inferior às demais estratégias, e o BSB apresenta as melhores métricas gerais.

Tabela 5: Média e desvio padrão de diferentes métricas de avaliação do modelo de classificação GP considerando várias estratégias de aprendizado ativo no Cenário II.

Estratégia	**Acurácia** $\mu \pm \sigma$	**Precisão** $\mu \pm \sigma$	**Recall** $\mu \pm \sigma$	**F1-score** $\mu \pm \sigma$
BSB	0.77 ± 0.05	0.73 ± 0.06	0.77 ± 0.05	0.76 ± 0.06
Entropy	0.76 ± 0.05	0.72 ± 0.06	0.76 ± 0.05	0.75 ± 0.06
GPLCB	0.75 ± 0.05	0.74 ± 0.02	0.75 ± 0.06	0.75 ± 0.07
Least Confident	0.75 ± 0.04	0.72 ± 0.06	0.75 ± 0.04	0.75 ± 0.05
Random	0.74 ± 0.05	0.72 ± 0.06	0.74 ± 0.05	0.74 ± 0.06

A Figura 3 exibe o gráfico com as curvas de acurácia média e as respectivas AUCs das estratégias BSB e Random em função do número de dados rotulados durante o processo iterativo. Mais uma vez nota-se que o BSB ultrapassa o Random em termos de acurácia média nas iterações finais do AL, atingindo uma acurácia de 0,77 e uma AUC de 0,652.

5.3. Resultados do Cenário III

A Tabela 6 e a Figura 4 exibem os resultados do cenário III, onde menos iterações do AL são executadas (dez) e mais vulnerabilidades são rotuladas por vez pelo especialista (dez). Observa-se um resultado similar aos outros dois cenários: a estratégia BSB obteve os melhores indicadores de desempenho na classificação de risco das vulnerabilidades, enquanto o Random apresentou o pior desempenho. No entanto, nota-se um ganho mais destacado no desempenho do BSB frente ao Random nas curvas de acurácia média e AUC ao longo do processo de AL, conforme pode ser observado na Figura 4.

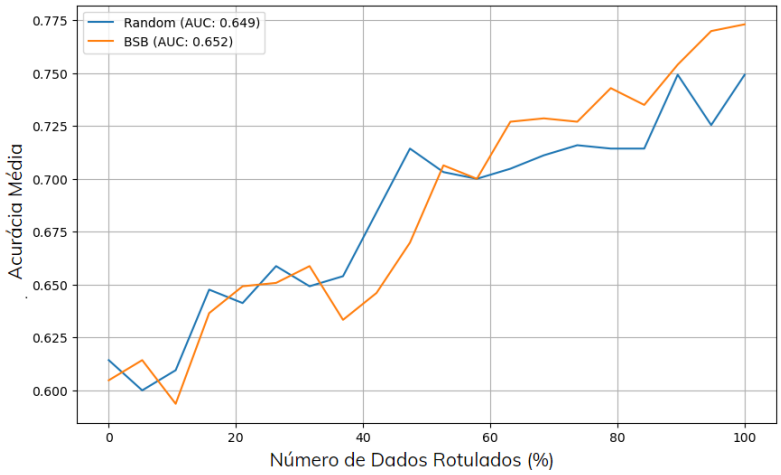

Figura 3: Acurácia média e AUC em função do número de dados rotulados no Cenário II.

Tabela 6: Média e desvio padrão de diferentes métricas de avaliação do modelo de classificação GP considerando várias estratégias de aprendizado ativo no Cenário III.

Estratégia	**Acurácia** $\mu \pm \sigma$	**Precisão** $\mu \pm \sigma$	**Recall** $\mu \pm \sigma$	**F1-score** $\mu \pm \sigma$
BSB	0.76 ± 0.05	0.74 ± 0.07	0.76 ± 0.05	0.76 ± 0.06
Entropy	0.74 ± 0.05	0.74 ± 0.07	0.74 ± 0.05	0.74 ± 0.06
GPLCB	0.74 ± 0.05	0.76 ± 0.07	0.74 ± 0.05	0.73 ± 0.06
Least Confident	0.76 ± 0.05	0.74 ± 0.07	0.76 ± 0.05	0.75 ± 0.04
Random	0.72 ± 0.04	0.71 ± 0.07	0.72 ± 0.04	0.72 ± 0.04

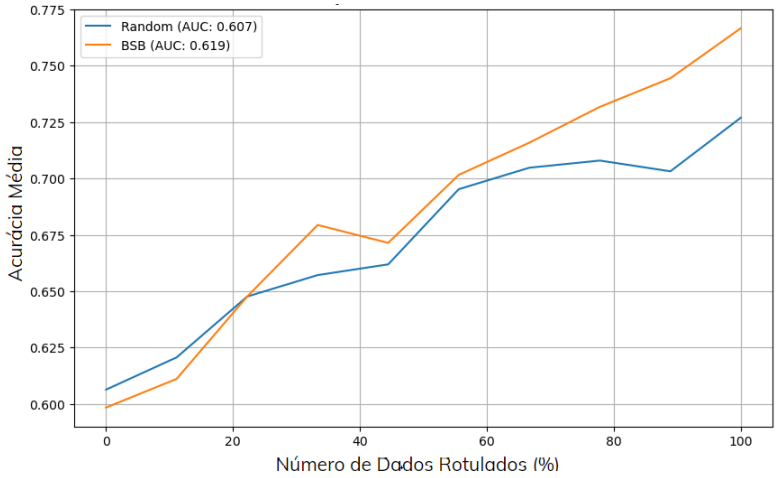

Figura 4: Acurácia média e AUC em função do número de dados rotulados no Cenário III.

5.4. Resultados da Estratégia BSB em Diferentes Cenários

Após a realização dos experimentos nos três cenários, a estratégia BSB emergiu como a mais eficaz na classificação de risco das vulnerabilidades, apresentando resultados superiores na maioria das métricas de desempenho em comparação com as demais técnicas.

Neste sentido, a estratégia com melhor desempenho foi escolhida para analisarmos nesta seção o efeito de fazermos menos ciclos de seleção + rotulação + retreino do modelo no processo de AL, ao agruparmos um subconjunto de vulnerabilidades a serem processadas em cada iteração, conforme os cenários descritos anteriormente.

A Figura 5 apresenta as curvas de acurácia média e AUCs para a estratégia BSB em função do número de dados rotulados. Embora o cenário I, caracterizado por ciclos mais curtos e frequentes, apresente a maior acurácia ao final do processo e uma maior área sob a curva, observa-se que os 3 cenários apresentam resultados semelhantes. Isso indica que pode ser vantajoso fazer menos iterações no processo de AL, com benefício de ganho de tempo oriundo de menos esforço computacional e menos interações com o especialista em segurança da informação.

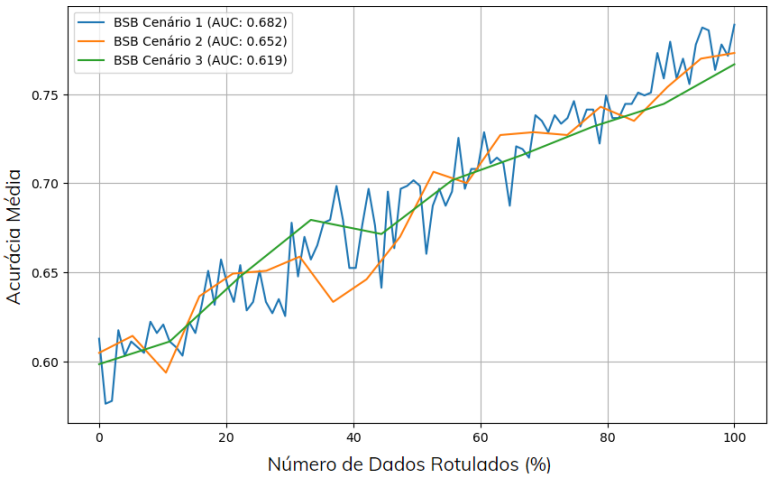

Figura 5: Acurácia média e AUC da estratégia BSB em função do número de dados rotulados para os diferentes cenários analisados.

6. Considerações Finais

Este estudo investigou a aplicabilidade de um modelo de ML baseado em processos gaussianos em conjunto com aprendizado ativo para classificar vulnerabilidades de segurança em sistemas de tecnologia da informação, com base em seu risco de exploração.

A metodologia foi testada em três cenários experimentais, cada um com uma quantidade variável de vulnerabilidades rotuladas por iteração do AL, os resultados evidenciaram que estratégias de AL permitem a redução do esforço humano necessário para a rotulação de vulnerabilidades, sem comprometer a precisão dos modelos de classificação de risco. A comparação entre as estratégias de AL revelou que a seleção de amostras baseada na incerteza do modelo é mais eficiente do que a seleção aleatória, demonstrando que a consideração da incerteza é fundamental para a otimização do processo de aprendizado.

Concluiu-se também que o uso de menos iterações no processo de AL, agrupando o mesmo número de dados a serem rotulados em grupos maiores, tem o benefício de reduzir o esforço computacional para retreinar o modelo de GP e reduzir a interação com o especialista para rotulação dos dados. Para trabalhos futuros, considera-se a investigação do processo de classificação de risco de vulnerabilidades baseado em ML em outros contextos, tais como aplicações web e ambientes de nuvem.

Referências

Alshaya, F. A., S. S. Alqahtani e Y. A. Alsamel (2023). "VrT: A CWE-Based Vulnerability Report Tagger: Machine Learning Driven Cybersecurity Tool for Vulnerability Classification". Em: *2023 IEEE/ACM 1st International Workshop on Software Vulnerability (SVM)*. IEEE, pp. 10–13.

Elbaz, C., L. Rilling e C. Morin (2021). "Automated risk analysis of a vulnerability disclosure using active learning". Em: *C&ESAR 2021-28th Computer & Electronics Security Application Rendezvous*, pp. 1–19.

Firoiu, M. (2015). "General Considerations on Risk Management and Information System Security Assessment According to ISO/IEC 27005: 2011 and ISO 31000: 2009 Standards." Em: *Quality-Access to Success* 16.149.

Foreman, P. (2019). *Vulnerability management*. Auerbach Publications.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.

Géron, A. (2019). *Mãos à obra: aprendizado de máquina com Scikit-Learn & TensorFlow*. Alta Books.

Hensman, J., N. Fusi e N. D. Lawrence (2013). "Gaussian Processes for Big Data". Em: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013*. AUAI Press.

Hensman, J., A. Matthews e Z. Ghahramani (2015). "Scalable variational Gaussian process classification". Em: *Artificial Intelligence and Statistics*. PMLR, pp. 351–360.

Hore, S., A. Shah e N. D. Bastian (2023). "Deep VULMAN: A deep reinforcement learning-enabled cyber vulnerability management framework". Em: *Expert Systems with Applications* 221, p. 119734.

Jakkal, V. (fev. de 2022). *Cybersecurity threats are always changing—staying on top of them is vital, getting ahead of them is paramount*. Microsoft Security Blog. URL: https://www.microsoft.com/en-us/security/blog/2022/02/09/cybersecurity-threats-are-always-changing-staying-on-top-of-them-is-vital-getting-ahead-of-them-is-paramount/.

Joshi, A. J., F. Porikli e N. Papanikolopoulos (2009). "Multi-class active learning for image classification". Em: *2009 ieee conference on computer vision and pattern recognition*. IEEE, pp. 2372–2379.

Kashyap, A., A. Chakravarthy e P. P. Menon (2022). "Detection of Cyber-Attacks in Automotive Traffic Using Macroscopic Models and Gaussian Processes". Em: *IEEE Control Systems Letters* 6, pp. 1688–1693.

Kure, H. I. et al. (2022). "Asset criticality and risk prediction for an effective cybersecurity risk management of cyber-physical system". Em: *Neural Computing and Applications* 34.1, pp. 493–514.

Pereira-Santos, D., R. B. C. Prudêncio e A. C. de Carvalho (2019). "Empirical investigation of active learning strategies". Em: *Neurocomputing* 326, pp. 15–27.

Ponte, F. R. da, E. B. Rodrigues e C. L. Mattos (2023a). "A Vulnerability Risk Assessment Methodology Using Active Learning". Em: *Advanced Information Networking and Applications: Proceedings of the 37th International Conference on Advanced Information Networking and Applications (AINA-2023), Volume 2*. Springer, pp. 171–182.

Ponte, F. R. da, E. B. Rodrigues e C. L. Mattos (2023b). "CVEjoin: An Information Security Vulnerability and Threat Intelligence Dataset". Em: *International Conference on Advanced Information Networking and Applications*. Springer, pp. 380–392.

Rasmussen, C. E. e C. K. I. Williams (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, pp. I–XVIII, 1–248. ISBN: 026218253X.

Ross, R. S. (2012). *Guide for Conducting Risk Assessments*. Special Publication 800-30 Rev. 1. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final. National Institute of Standards e Technology.

Sabottke, C., O. Suciu e T. Dumitraș (2015). "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting {Real-World} exploits". Em: *24th USENIX Security Symposium (USENIX Security 15)*, pp. 1041–1056.

Sun, X. et al. (2023). "ASSBert: Active and semi-supervised bert for smart contract vulnerability detection". Em: *Journal of Information Security and Applications* 73, p. 103423. ISSN: 2214-2126.

Swiler, L. P. et al. (2020). "A survey of constrained Gaussian process regression: Approaches and implementation challenges". Em: *Journal of Machine Learning for Modeling and Computing* 1.2.

Tenable (2023). *Três desafios reais enfrentados pelas organizações de segurança cibernética*. Retrieved from https://www.tenable.com.

Williams, C. K. e C. E. Rasmussen (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA.