Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Programa de Pós-Graduação em Tecnologia da Informação

Nível Mestrado Profissional

NILTON DA TRINDADE HERTHEL JUNIOR

PROTEÇÃO ADAPTATIVA DE APLICAÇÕES QUANTO A ATAQUES DE *WEB SCRAPING* UTILIZANDO META-HEURÍSTICAS

DISSERTAÇÃO DE MESTRADO

JOÃO PESSOA – PB

2024

Nilton da Trindade Herthel Junior

Proteção Adaptativa de Aplicações Quanto a Ataques de *Web Scraping* Utilizando Meta-Heurísticas

Dissertação de Mestrado apresentada como requisito final para obtenção do título de Mestre em Tecnologia da Informação pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB.

Orientador: Prof. Dr. Thiago Gouveia da Silva

João Pessoa – PB

2024

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha - *Campus* João Pessoa, PB.

H574p Herthel Junior, Nilton da Trindade.

Proteção adaptativa de aplicações quanto a ataques de *Web Scraping* utilizando meta-heurísticas / Nilton da Trindade Herthel Junior. – 2024.

78 f. : il.

Dissertação (Mestrado em Tecnologia da Informação, Nível Profissional) Instituto Federal de Educação da Paraíba / Programa de Pós-Graduação em Tecnologia da Informação (PPGTI), 2024.

Orientação: Prof. D.r Thiago Gouveia da Silva

1. Segurança da informação. 2. Meta-heurística. 3. Raspagem de dados. I. Título.

CDU 004.056(043)

Elaboração: Lucrecia Camilo de Lima, Bibliotecária – CRB 15/132

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU*

MESTRADO PROFISSIONAL EM TECNOLOGIA DA INFORMAÇÃO

NILTON DA TRINDADE HERTHEL JUNIOR

PROTEÇÃO ADAPTATIVA DE APLICAÇÕES QUANTO A ATAQUES DE WEB SCRAPING UTILIZANDO META-HEURÍSTICAS

Dissertação apresentada como requisito para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB – Campus João Pessoa.

Aprovado em 31 de maio de 2024

Membros da Banca Examinadora:

Dr. Thiago Gouveia da Silva

IFPB – PPGTI

Dr. Paulo Ditarso Maciel Junior

IFPB – PPGTI

Dr. Diego Ernesto Rosa Pessoa

IFPB – PPGTI

Dr. Teobaldo Leite Bulhões Junior

UFPB

João Pessoa, 2024

Documento assinado eletronicamente por:

- **Thiago Gouveia da Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 03/06/2024 15:32:24.
- **Paulo Ditarso Maciel Junior**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 03/06/2024 15:33:16.
- **Diego Ernesto Rosa Pessoa**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 03/06/2024 16:51:41.
- **Teobaldo Leite Bulhões Junior**, **PROFESSOR DE ENSINO SUPERIOR NA ÁREA DE ORIENTAÇÃO EDUCACIONAL**, em 05/06/2024 11:33:04.

Este documento foi emitido pelo SUAP em 24/05/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 566614

Código de Autenticação: ac0b42c99f

Av. Primeiro de Maio, 720, Jaguaribe, JOAO PESSOA / PB, CEP 58015-435
http://ifpb.edu.br - (83) 3612-1200

Este trabalho é dedicado a todos que buscam, através do trabalho voltado para o serviço público, a eficiência e qualidade dos serviços oferecidos à população.

AGRADECIMENTOS

Agradeço primeiramente ao meu orientador e amigo, Prof. Dr. Thiago Gouveia, pelo incentivo e paciência diante dos inúmeros contratempos trazidos à execução deste trabalho.

Também à minha família, pela compreensão com as ausências no dia a dia enquanto assistia aulas, pesquisava, desenvolvia, coletava resultados da pesquisa e escrevia esta dissertação. Minha mãe Margarethe, que sempre se dispôs a ajudar quando preciso, e meu filho Lucas, fonte de inspiração para todos os esforços em criar um mundo melhor.

Além disso, todos os amigos, colegas de curso e de trabalho que contribuíram direta e indiretamente com a pesquisa. Destaco as contribuições de Yuri, amigo de longa data, debatedor de diversas ideias deste trabalho e colaborador em diversas frentes profissionais e acadêmicas.

Agradeço aos membros da banca, Prof. Dr. Teobaldo Leite Bulhões Júnior (UFPB), Prof. Dr. Paulo Ditarso Maciel Júnior (IFPB) e Prof. Dr. Diego Ernesto Rosa Pessoa (IFPB) pelas valiosas contribuições para a versão final do trabalho.

E, por último, mas não menos especial, minha namorada e companheira de pesquisa Ms. Elba Quirino, que enriqueceu este trabalho em curso com seu conhecimento e habilidade em análise de Dados, além do carinho e paciência com as demandas acadêmicas.

RESUMO

Na era atual, caracterizada pela crescente digitalização dos serviços públicos, o conceito de Governo Digital tem alcançado significativa relevância. Este conceito implica no fornecimento de acesso amplo e democrático dos cidadãos aos serviços públicos através da Internet, uma iniciativa que desafia a tradicional dependência de métodos de autenticação. Em um cenário onde a informação detém um papel central e é considerada um ativo de inestimável valor em termos financeiros, políticos e estratégicos, torna-se imperativo abordar e mitigar os riscos associados à raspagem de dados. Esta técnica, que envolve a extração automatizada de dados de serviços online, representa um obstáculo significativo ao acesso seguro e confiável à informação. O presente trabalho busca investigar e desenvolver uma solução para contrapor essa ameaça. A abordagem proposta baseia-se em um estudo aprofundado das ferramentas de proteção já existentes e no monitoramento contínuo das atividades mal-intencionadas. A intenção é estabelecer um conjunto de medidas de controle, integradas ao funcionamento da aplicação *web*, que seja capaz de identificar e responder de maneira eficiente e adaptável às tentativas de exploração automatizada, bem como demais tentativas de ataque, com foco na utilização de meta-heurísticas para seleção das características relevantes nesta análise. Após a implementação e avaliação da eficácia da solução proposta, pôde-se constatar uma eficácia média de 86% sobre os apontamentos realizados com base na classificação, o que valida a proposta como ferramenta adicional para a proteção de sistemas. O estudo sugere a extensão da pesquisa, baseada na aplicação da solução desenvolvida em diversos de cenários, avaliando sua adaptabilidade e eficiência em diferentes contextos e modelos de uso. Por fim, conclui-se que a pesquisa pode fornecer informações valiosas para a evolução contínua das práticas de segurança em aplicações Web voltadas a contextos específicos de Governo Digital, contribuindo significativamente para a proteção de dados e a promoção de um acesso mais seguro e inclusivo aos serviços públicos.

Palavras-chaves: Segurança, Meta-Heurísticas, Raspagem de Dados

ABSTRACT

In the current era, marked by the increasing digitalization of public services, the concept of Digital Government has gained significant relevance. This concept entails providing citizens with broad and democratic access to public services via the Internet, an initiative that challenges the traditional reliance on authentication methods. In a scenario where information plays a central role and is considered an invaluable asset in financial, political, and strategic terms, it becomes imperative to address and mitigate the risks associated with data scraping. This technique, involving the automated extraction of data from online services, poses a significant obstacle to secure and reliable information access. The present work seeks to investigate and develop an innovative and adaptable solution to counter this threat. The proposed approach is based on an in-depth study of existing protection tools and continuous monitoring of malicious activities. The aim is to establish a set of control measures, integrated into the functioning of the web application, capable of efficiently and adaptively identifying and responding to automated exploitation attempts, as well as other attack attempts, focusing on the use of metaheuristics for selecting relevant characteristics in this analysis. After the implementation and evaluation of the effectiveness of the proposed solution, an average effectiveness of 86% was observed based on the classifications made, validating the proposal as an additional tool for system protection. The study suggests extending the research by applying the developed solution in various scenarios, evaluating its adaptability and efficiency in different contexts and usage models. Finally, it is concluded that the research can provide valuable information for the continuous evolution of security practices in web applications focused on specific contexts of Digital Government, significantly contributing to data protection and promoting more secure and inclusive access to public services.

Keywords: Security, Meta-heuristics, Web Scraping

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	*Algoritmos Genéticos*
CERT	*Computer Emergency Response Team*
CNN	*Convolutional Neural Network*
C&C	*Command and Control*
DDoS	*Distributed Denial of Service*
ED	*Evolução Diferencial*
GAN	*Generative Adversarial Network*
GDPR	*General Data Protection Regulation*
GRASP	*Greedy Randomized Adaptive Search Procedure*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
kNN	*k-Nearest Neighbor*
LGPD	Lei Geral de Proteção de Dados Pessoais
PSO	*Particle Swarm Optimization*
SVM	*Support-vector Machine*
TCP	*Transmission Control Protocol*
TTL	*Time to Live*
WAF	*Web Application Firewall*

SUMÁRIO

1. INTRODUÇÃO

No mundo atual, a proliferação de dispositivos eletrônicos produz uma quantidade bastante expressiva de dados. Estima-se que 328,77 milhões de terabytes[1], sejam produzidos diariamente a partir de tais quais computadores, *smartphones*, sensores, *wearables* e demais equipamentos eletrônicos. O aumento nas capacidades computacionais e nas tecnologias de interconexão que sustentam as redes de computadores também é notável, e permite que estas informações sejam coletadas e armazenadas para usos diversos. Como referência, calcula-se que 90% dos dados disponíveis no planeta tenham sido criados nos últimos dois anos[2] e segue evoluindo, dado o aumento do número de dispositivos conectados e suas capacidades.

Este trabalho apresenta uma nova abordagem para a construção de uma aplicação visando a proteção de sistemas contra acessos automatizados, comumente conhecidos como *bots*, e diversas ameaças cibernéticas. O objetivo central é desenvolver um mecanismo eficaz para identificar e inibir tais acessos automatizados e indevidos, que frequentemente visam explorar recursos computacionais primariamente destinados a usuários humanos. Esta pesquisa se destaca pela busca de soluções que não apenas detectem e bloqueiem tais atividades mal-intencionadas, mas também pela capacidade de adaptar-se a diferentes cenários arquiteturais e tipos de ameaças.

Ao considerar o crescente volume e sofisticação dos ataques automatizados, a importância de proteger sistemas contra essas invasões torna-se evidente. Estes ataques representam não apenas um risco à segurança e privacidade dos dados, mas também implicam em custos operacionais significativos para a manutenção dos serviços. Além disso, o acesso indevido por soluções automatizadas pode levar ao vazamento de informações estratégicas, comprometendo a integridade e confiabilidade dos sistemas afetados.

A pesquisa neste trabalho foca em desenvolver uma ferramenta versátil e eficiente, capaz de se adaptar a diferentes ambientes digitais e arquiteturas de sistemas, explorando técnicas avançadas de detecção e mitigação de acessos automatizados, e integrando-as em um sistema robusto que possa ser adaptável às necessidades específicas de cada aplicação. Através desta abordagem, busca-se não apenas reforçar a segurança dos sistemas, mas também reduzir o impacto econômico associado à gestão de acessos indevidos, assegurando a continuidade e a eficiência dos serviços prestados.

---

[1] https://whatsthebigdata.com/data-generated-every-day/, acessado em maio de 2024

[2] https://www.oracle.com/br/data-science/what-is-data-science/, acessado em março de 2024

Este estudo, portanto, visa contribuir para o campo da Segurança Cibernética, oferecendo uma solução abrangente para o desafio crescente dos ataques automatizados, beneficiando tanto organizações quanto usuários finais ao garantir um ambiente digital mais seguro e confiável.

1.1. Motivação e Definição do Problema

A evolução da Internet vem simplificando, ao longo dos anos, a criação de conteúdo por parte de seus usuários. A facilidade para publicar e consumir informações resulta em grandes quantidades de dados disponíveis nesta rede, produzindo uma dinâmica de oferta e procura que se retroalimenta na criação de produtos e serviços voltados para a qualidade do acesso a estes dados.

Entre os diversos atores do processo de disponibilização de dados na Internet, destaca-se o trabalho de órgãos de governo na publicidade e na simplificação de processos burocráticos através do que é conhecido como Governo Digital, como descrito por Robertson et al. (2010). O conceito ganhou ainda mais notoriedade nos tempos atuais devido à pandemia do COVID-19, onde diversos serviços passaram a ser oferecidos de forma *online*, visando diminuir a circulação de pessoas em órgãos públicos. A capacidade de entregar informações e serviços com agilidade e qualidade é um dos pilares que sustentaram estratégias governamentais de contenção da propagação da pandemia, permitindo que os cidadãos permanecessem em casa enquanto a tecnologia provê o acesso a serviços fornecidos pelo governo.

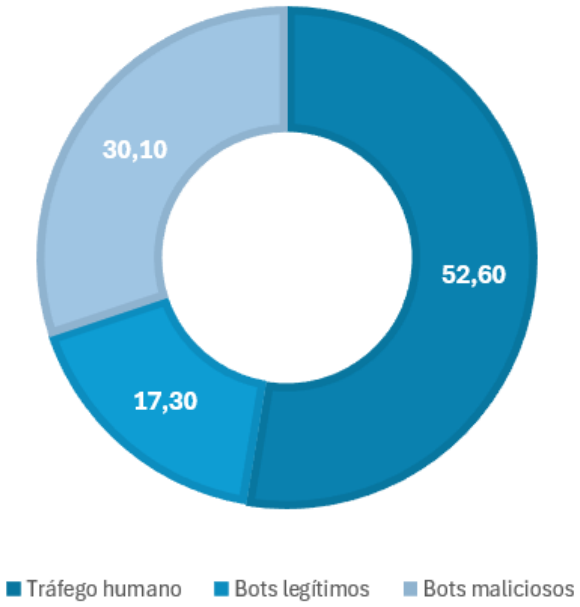
No entanto, tal plataforma de divulgação de informações e serviços resulta em considerável aumento da superfície de ataque, e vem se tornando alvo de estratégias para obtenção de lucro a partir da extração indevida de informações. A exploração de falhas de segurança, além da automação de acessos através de robôs, técnica conhecida como *Web Scraping*, como visto em Zhao (2007), aliadas ao crescente poder computacional dos atacantes disponível em serviços como nuvens computacionais públicas, aumenta o risco à segurança das informações disponibilizadas na Internet. Os dados extraídos por estes métodos são agrupados e comercializados em ambientes da *deep web*, fornecendo informações valiosas sobre populações inteiras, capazes de influenciar a tomada de decisões estratégicas em processos tão críticos quanto a alocação de recursos em campanhas eleitorais, por exemplo.

O tráfego gerado por requisições automatizadas representa 47.4%, ou seja, quase a metade de todo o tráfego realizado através da Internet, conforme dados[3] de 2022 representados na Figura 1. Estes dados representam um aumento de 5.1% com relação ao ano anterior. Destes, apenas 17.3%

[3] https://www.imperva.com/resources/resource-library/reports/2023-imperva-bad-bot-report/. Acessado em 30 de março de 2024.

são considerados “bons” robôs, como indexadores de páginas de pesquisa e ações legítimas como monitoramento de desempenho de aplicações *web*, enquanto 30.1% consistem em serviços responsáveis pela exploração da lógica de negócio de aplicações para obtenção de informações em larga escala, ataques de negação de serviço, entre outros.

Figura 1 - Percentuais de Tráfego Gerado por Humanos e *Bots* na *Internet* em 2022



O assunto adquire especial gravidade quando, diante do aumento nos episódios de mau uso de dados pessoais da população em geral, organismos ao redor do mundo pressionam governos para a adoção de legislações mais duras quanto ao vazamento de dados pessoais, através de instrumentos de regulamentação como o General Data Protection Regulation[4] (GDPR), da União Europeia, e a Lei Geral de Proteção de Dados Pessoais[5] (LGPD), do governo brasileiro. Estes normativos regulam o papel do dono da informação (a pessoa natural) e dos organismos controladores e operadores do dado, definindo direitos e deveres de cada um deles, e estabelecendo sanções financeiras em caso de transgressão.

Neste contexto dinâmico, onde a necessidade de proteção robusta de informações se torna cada vez mais premente, torna-se urgente a implementação de mecanismos de controle de informações eficazes. Esses mecanismos devem garantir a preservação da tríade crítica de segurança: Integridade, Confidencialidade e Disponibilidade dos dados, particularmente no âmbito dos órgãos governamentais. Uma abordagem essencial para alcançar esse objetivo, especialmente ao considerar

[4] https://gdpr-info.eu/

[5] http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709compilado.htm

a proteção em tempo real de acessos realizados via Internet, é o uso de um *Web Application Firewall* (WAF). Este ativo de segurança, operando na camada de aplicação, desempenha um papel fundamental ao interceptar e analisar o tráfego de entrada e saída, com o propósito de identificar e bloquear potenciais ameaças e acessos não autorizados. Ao filtrar o tráfego específico para aplicações web, o WAF oferece uma camada adicional de segurança que é crucial para a proteção de dados sensíveis e para a manutenção da operacionalidade contínua dos serviços governamentais digitais. Suas capacidades, assim como as dos atacantes, vêm aumentando na proporção da evolução tecnológica, e o bom uso destas capacidades está frequentemente associado à interpretação dos dados de bloqueio, visando direcionar os esforços sobre as tentativas de ataque mais frequentes.

Esta interpretação nem sempre é um trabalho simples. A capacidade de traduzir grandes volumes de dados gerados a partir das interceptações provocadas pelo WAF em ações práticas é um desafio que perpassa diversas áreas do conhecimento em Ciência da Computação, do entendimento sobre protocolos de rede aos conceitos de segurança de aplicações, bem como a aplicação de ciência de dados para obtenção de *insights* sobre os tipos de ataque. Uma boa técnica de classificação da informação é capaz de definir perfis de ataque, para os quais as equipes de desenvolvimento podem se preparar com antecedência, observando suas características principais e definindo heurísticas de bloqueio mais eficientes, se adaptando aos ataques para a utilização dos recursos com maior qualidade.

O uso de WAFs não deve ser considerado a única forma de proteção contra acessos maliciosos e/ou automatizados. Este serviço, quando contratado na modalidade SaaS (*Software as a Service*), possui um custo operacional elevado, enquanto a instalação deste na rede física da instituição (*on premises*) pode esbarrar na necessidade de escalabilidade para atender a um número crescente de aplicações, processo que também traz impacto financeiro, relacionado ao custo computacional para atender à demanda. Na prática, o uso pleno de suas funcionalidades ocorre quando há uma vulnerabilidade a ser corrigida em alguma aplicação, de modo a inviabilizar o ataque enquanto não ocorre a mitigação através da atualização desta.

Outras formas de proteção ocorrem de maneira integrada às aplicações. A utilização de mecanismos como CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*), que visam identificar se o usuário é legítimo a partir da análise comportamental ou da resolução de um desafio, oferecem uma proteção compatível com o que é conhecido como o princípio da defesa em camadas, um dos padrões que norteia a Segurança da Informação e que se baseia na necessidade de aplicar controles complementares para a garantia da segurança. Vale ressaltar que o dimensionamento e utilização destes controles é mais um desafio, pois sua aplicação também pode ocasionar um aumento de custos.

Observa-se que, para compreender os mecanismos de defesa aplicados em um contexto de proteção de aplicações *web*, é necessário aprofundar os conhecimentos sobre os meios de exploração das falhas. As técnicas de *Web Scraping*, por exemplo, podem variar de acordo com a necessidade do utilizador, bem como dos mecanismos aplicados na defesa. O trabalho de Manjushree et al. (2020) oferece uma perspectiva geral sobre as pesquisas no tema, bem como menciona os utilitários aplicados na extração automatizada de dados.

Uma das formas de obter o perfilamento destes ataques, quando tratamos de grandes volumes de dados, é através de algoritmos de agrupamento automático. A heurística mais conhecida para este tipo de ação é o *k-means*, como constatado por Bock et al. (2007). Por um lado, trata-se de um mecanismo de fácil implementação, capaz de produzir bons resultados quando submetido a conjuntos de dados adequados para seu uso. Por outro lado, a utilização de métodos exatos para agrupamento comumente obtém melhores resultados, como abordou Ikotun et al. (2022).

Entendendo as formas de ataque utilizadas e identificáveis através do processo de agrupamento de dados, é importante também compreender conceitos relacionados à proteção oferecida por mecanismos ativos de segurança. Trabalhos acadêmicos como o realizado por Azad et al. (2020) buscam comparar ferramentas de mercado e seus mecanismos de identificação de padrões aplicáveis a requisições automatizadas, técnica conhecida como *fingerprinting*. Este processo é ponto de partida para estabelecer o entendimento sobre acessos automatizados, e o perfilamento de usuários pode auxiliar a identificar potenciais fontes de *fingerprint* confiáveis.

Este trabalho utiliza técnicas de agrupamento para classificar e destacar características dominantes derivadas dos apontamentos do WAF e avaliar sua relevância na efetividade de um mecanismo adaptável de bloqueio. Esta abordagem serve como base para o desenvolvimento de uma aplicação baseada em uma heurística de proteção contra ataques, especialmente aqueles relacionados a *web scraping*. Uma vez que a grande massa de dados gerada a partir da observação de ativos de proteção exige equilíbrio entre a capacidade de obter informações e os recursos computacionais aplicados nesta atividade, o método proposto busca um equilíbrio entre a eficiência na extração de informações significativas e ao bom uso dos recursos computacionais alocados nesta tarefa. Para isso, propõe-se uma via intermediária, com base na utilização de meta-heurísticas, visando a identificação e a compreensão das características principais das informações coletadas. A partir da análise dos atributos predominantes em cada agrupamento definido pelo algoritmo baseado na meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*), é proposto um mecanismo de proteção robusto, baseado nas características mais proeminentes identificadas, visando proteger efetivamente os ativos computacionais que estão expostos ao tráfego da *web*.

Por fim, a aplicação proposta concentra-se em maximizar os efeitos dos bloqueios através da correta aplicação dos recursos computacionais na identificação e bloqueio de acessos indevidos

realizados por *bots*, com o intuito de minimizar os custos associados à aplicação de controles de segurança e, em consequência de sua efetividade, dos custos operacionais da própria aplicação, através da utilização dos conceitos de aprendizagem de máquina nas etapas de perfilamento e de tomada de decisão para o bloqueio de usuários maliciosos. Esta efetividade resultará na limitação do impacto prejudicial do vazamento de informações em larga escala, extraídas por usuários mal-intencionados utilizando mecanismos de ataque, resultando em uma infraestrutura digital mais resiliente e segura.

1.2. Objetivos

O objetivo geral da pesquisa consiste em limitar o impacto de requisições automatizadas maliciosas através do desenvolvimento de uma aplicação que seja capaz de ser treinada para conter estes acessos em um contexto de aplicações *web*, considerando dois fatores principais: a proteção a informações sensíveis e a redução do custo em comparação a soluções estabelecidas como padrão para as atividades de detecção e contenção de acessos indevidos e automatizados.

Como passos necessários para o alcance do objetivo supracitado, são considerados os seguintes objetivos específicos:

- Compreender e apresentar os trabalhos acadêmicos que sirvam como base conceitual para a pesquisa e as propostas de solução em cenários análogos de proteção, como *botnets*, *malware*, biometria;
- Elaborar uma heurística de classificação capaz de identificar características predominantes em ataques e acessos automatizados a partir da observação de ativos de proteção;
- Desenvolver um mecanismo capaz de reduzir a eficácia dos acessos automatizados, considerando como fatores principais a adaptabilidade a diferentes cenários, a capacidade de identificação de uma ameaça, e a capacidade de adaptar-se ao longo do tempo por mecanismos de aprendizagem;
- Comparar a eficácia da classificação baseada nas heurísticas de agrupamento com a classificação baseada na coleta simples de tráfego quanto ao índice de sucesso nas classificações.

1.3. Estrutura do Documento

Após esta introdução, os capítulos subsequentes deste trabalho estão organizados da seguinte maneira:

Os conceitos relacionados à Aprendizagem de Máquina e Segurança da Informação são

apresentados no Capítulo 2, onde podemos detalhar os temas nas Seções 2.1 (Segurança da Informação), onde são considerados os conceitos de *web scraping*, assim como as ameaças à Segurança da Informação derivadas do uso desta técnica, assim como na 2.2 (Ciência de Dados e Aprendizagem de Máquina), através da exploração das definições e nuances relacionadas a Aprendizagem Supervisionada e Não-Supervisionada e de Heurísticas de Agrupamento.

O Capítulo 3 da pesquisa, por sua vez, dedica-se integralmente ao exame de literatura relevante relacionada aos temas de Aprendizagem de Máquina e Segurança da Informação. Este capítulo não apenas aborda um conjunto relevante de trabalhos acadêmicos pertinentes a estes dois campos, mas também detalha os resultados de uma Revisão Sistemática, concentrada especificamente na análise da intersecção entre Aprendizagem de Máquina e Segurança da Informação e sua aplicação na detecção e prevenção de ameaças cibernéticas. Os resultados são categorizados sob tópicos principais, como Detecção de Ataques, Biometria Computacional, *Botnets*, entre outros assuntos relevantes.

No Capítulo 4, a arquitetura proposta é detalhada, partindo da sua aplicabilidade (Seção 4.1), e do detalhamento da metodologia utilizada (Seção 4.2), onde está descrita a fase inicial de captura e análise de informações, a construção da solução, contendo a fase da heurística de agrupamento para observação das características do tráfego monitorado e da solução de proteção. Por fim, a Seção 4.3 (Desenvolvimento da Solução Proposta) detalha a implementação das técnicas de agrupamento (Subseção 4.3.1 –Heurística de Agrupamento Proposta), bem como a construção da solução de proteção (4.3.2 – Arquitetura da Solução) a ser implantada. Por fim, são fornecidos detalhes (4.3.3 – Desenvolvimento da Solução Proposta) sobre a arquitetura adotada, tecnologias utilizadas no desenvolvimento e integrações com mecanismos e bases de dados que sustentam o funcionamento desta solução.

O Capítulo 5 apresenta os resultados obtidos na pesquisa, através da aplicação prática da solução em contextos de tráfego real capturado pelas ferramentas de proteção, utilizando-se dos mecanismos definidos no Capítulo 4 para a detecção e prevenção de possíveis ataques. A Seção 5.1 representa os resultados da aplicação da heurística de agrupamento e classificação dos dados em agrupamentos. A Seção 5.2 define os cenários utilizados no teste comparativo entre a coleta de dados a partir dos critérios utilizados para o agrupamento e mecanismos que simulam a captura de toda a requisição, de modo a validar a solução, enquanto a Seção 5.3 apresenta a eficácia da solução com base nos resultados da sua utilização. Por fim, a Seção 5.4 discute a validação dos objetivos da pesquisa e a análise dos resultados obtidos.

As considerações finais e as propostas de trabalhos futuros são descritas no Capítulo 6, com uma perspectiva das contribuições acadêmicas e profissionais sobre os temas abordados na construção desta solução, bem como as formas em que os resultados podem influenciar o desenvolvimento futuro nos campos da Segurança da Informação e da Aprendizagem de Máquina aplicados ao problema em questão.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda, de forma suscinta, os assuntos mais importantes para o bom entendimento deste trabalho. Inicialmente, serão apresentados os conceitos básicos da Segurança da Informação, assim como as definições de raspagem de dados (*web scraping*) e suas contramedidas. Em sequência, são introduzidos alguns conceitos de Ciência de Dados e Aprendizagem de Máquina, com foco em heurísticas de agrupamento de dados.

2.1. Segurança da Informação

A Segurança da Informação (SI) é a disciplina que abrange a proteção da Informação e dos Sistemas de Informação quanto ao acesso não-autorizado, uso, divulgação, perturbação, modificação ou destruição, de modo a garantir Confidencialidade, Integridade e Disponibilidade aos sistemas[6]. É aplicável em um amplo contexto, através da utilização de recursos humanos, processos e tecnologia na construção de um arcabouço de proteção às aplicações, suas informações, e mecanismos de integração (canais de comunicação e ativos de rede, por exemplo).

De acordo com a definição do tema por Stallings et al. (2008), o termo *Confidencialidade* aborda dois conceitos: *Confidencialidade de Dados*, que assegura que informações privadas e confidenciais não estejam disponíveis nem sejam reveladas para indivíduos não autorizados e *Privacidade*, que assegura que os indivíduos controlem ou influenciem quais de suas informações podem ser obtidas e armazenadas. Por sua vez, a *Integridade* também abrange a *Integridade de Dados*, que assegura que as informações e os programas sejam modificados somente de uma maneira especificada e autorizada; e a *Integridade do Sistema*, que assegura que um sistema execute as suas funcionalidades de forma ilesa, livre de manipulações deliberadas ou inadvertidas. Por fim, a *Disponibilidade* assegura que os sistemas operem prontamente e seus serviços se mantenham disponíveis para usuários autorizados.

2.1.1. Raspagem de Dados

Dentro do contexto da Segurança da Informação, este trabalho tem como foco principal a mitigação da extração automatizada de informações de sistemas de informação, um desafio cada vez mais presente na era digital. A extração automatizada, conhecida como *web scraping*, é um processo no qual robôs são desenvolvidos para extrair dados de sistemas. Embora essa prática possa ter aplicações legítimas, como a coleta de dados realizada por autoridades policiais para investigações, seu uso indiscriminado pode ter consequências severas para o desempenho e a segurança de aplicações *web*.

---

[6] https://csrc.nist.gov/glossary/term/infosec

Quando robôs executam tarefas de *web scraping* em um sistema, eles consomem recursos computacionais significativos, competindo com usuários legítimos pelo acesso e pela resposta da aplicação. Essa disputa pode resultar em desaceleração ou até mesmo em indisponibilidade, afetando negativamente a experiência do usuário. Do ponto de vista da SI, mais preocupante é o risco associado ao vazamento de dados. Um robô que acessa funcionalidades de um sistema de forma irrestrita (como uma consulta processual destinada a usuários através de um número de protocolo, por exemplo) pode extrair uma quantidade substancial de informações, comprometendo a integridade e a confidencialidade dos dados, não apenas ameaçando a privacidade dos usuários, mas também expondo a organização a riscos legais e de reputação.

As técnicas de *web scraping* são frequentemente aprimoradas para simular o comportamento humano, tornando mais desafiadora a detecção desses *bots*. Ferramentas como o Selenium[7], um popular utilitário para automação de comandos, são utilizadas para registrar e reproduzir ações em navegadores *web*. Com scripts bem elaborados, o Selenium pode navegar por páginas, preencher formulários, clicar em botões e realizar outras operações, tornando os robôs quase indistinguíveis de usuários legítimos.

Além dos impactos diretos sobre a segurança e o desempenho, a raspagem de dados também levanta questões éticas e legais. Por exemplo, a coleta de dados sem consentimento pode violar regulamentos de privacidade de dados, como o GDPR na União Europeia e a LGPD no Brasil. Além disso, a raspagem pode infringir os termos de uso de muitos sites, resultando em consequências jurídicas.

Nesse cenário, a abordagem deste trabalho se concentra na criação de uma aplicação capaz de detectar e impedir atividades de raspagem de dados. Isso envolve o desenvolvimento de soluções que possam distinguir entre o tráfego gerado por humanos e robôs, e implementar estratégias eficazes para bloquear ou limitar o acesso automatizado sem comprometer a acessibilidade para usuários legítimos. O objetivo é proteger os sistemas de informação contra o uso abusivo de seus recursos e salvaguardar os dados contra acessos não autorizados, mantendo assim a integridade, a confidencialidade e a disponibilidade das informações.

2.1.2. Contramedidas de Proteção

A proteção contra ameaças como *web scraping* geralmente ocorre durante a execução do sistema, utilizando-se de regras e heurísticas pré-definidas para identificar comportamentos suspeitos. Essas regras podem incluir critérios como a origem do acesso, o tamanho da requisição, padrões de comportamento do usuário, entre outros.

Há duas abordagens principais para implementar essas proteções. A primeira abordagem considera incorporar mecanismos de segurança diretamente no código-fonte da aplicação. Isso é

[7] https://www.selenium.dev/

feito por meio de utilitários conhecidos como RASP (*Runtime Application Self-Protection*)[8]. O RASP atua como um agente de proteção integrado que monitora o comportamento da aplicação em tempo real e responde imediatamente a atividades suspeitas ou maliciosas. Isso é realizado através de um conjunto de verificações e proteções que são executadas junto com a aplicação, permitindo uma resposta rápida e eficiente a ameaças potenciais.

A segunda abordagem envolve o monitoramento e a filtragem do tráfego na entrada da rede, especialmente na camada de aplicação. Isso é feito por meio de um modelo conhecido como WAF (*Web Application Firewall*)[9]. O WAF funciona como um filtro ou uma barreira entre a aplicação web e o tráfego de Internet, analisando as requisições de entrada para identificar e bloquear tráfego malicioso com base em um conjunto predefinido de regras.

Ambos os modelos, RASP e WAF, estabelecem um conjunto de regras para a identificação de ameaças, baseadas em padrões conhecidos de ataques e comportamentos mal-intencionados. No entanto, para aumentar a eficácia na detecção de ameaças e adaptar-se a novos tipos de ataques, esses sistemas podem ser aprimorados com a aplicação de técnicas de aprendizagem de máquina. Integrando aprendizagem de máquina, os sistemas podem aprender com os dados de tráfego observados, aprimorando continuamente a precisão na identificação de padrões suspeitos e adaptando-se dinamicamente a novas táticas utilizadas por atacantes. Isso resulta em uma defesa mais robusta e adaptável, capaz de entregar resultados consistentes e eficientes na proteção contra ameaças emergentes e em evolução.

Além das abordagens RASP e WAF para a segurança de aplicações *web*, outra técnica amplamente utilizada na proteção contra acessos automatizados como *web scraping* é conhecida como CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*). O CAPTCHA é um tipo de desafio-resposta utilizado em ambientes computacionais para determinar se o usuário é ou não um humano. Essa técnica é especialmente eficaz para prevenir abusos automatizados, como preenchimentos de formulários por *bots* ou tentativas de login em massa.

A implementação de CAPTCHA em aplicações *web* é uma maneira de adicionar uma camada adicional de defesa contra a raspagem de dados automatizada. Ao exigir interações que são desafiadoras para os *bots*, mas relativamente fáceis para os humanos, o CAPTCHA ajuda a garantir que apenas usuários legítimos possam acessar certas funcionalidades da aplicação. Isso é particularmente útil em pontos críticos como formulários de inscrição, páginas de login e durante transações online.

---

[8] https://www.gartner.com/en/information-technology/glossary/runtime-application-self-protection-rasp. Acessado em maio de 2024.

[9] https://www.cloudflare.com/pt-br/learning/ddos/glossary/web-application-firewall-waf/. Acessado em maio de 2024.

Existem diversas formas de CAPTCHA, que variam desde a digitação de textos distorcidos apresentados em imagens até a identificação de objetos específicos em uma grade de imagens. Versões mais recentes, como o reCAPTCHA da Google, têm evoluído para testes menos intrusivos, utilizando análises de comportamento do usuário para determinar a probabilidade de ser um humano sem a necessidade de desafios explícitos.

No entanto, o CAPTCHA não está isento de desafios. Por um lado, *bots* sofisticados estão se tornando cada vez melhores em resolver alguns tipos de CAPTCHAs, diminuindo sua eficácia. Sua utilização depende de mudanças no código-fonte da aplicação *web* para promover a alteração do fluxo de usuário de acordo com o resultado da análise. Além disso, CAPTCHAs podem criar barreiras de usabilidade para usuários legítimos, especialmente se forem muito complexos ou difíceis de interpretar, representando uma barreira de acessibilidade para pessoas com deficiências visuais ou outras limitações.

Portanto, enquanto o CAPTCHA é uma ferramenta valiosa na prevenção de acesso automatizado, ele é frequentemente mais eficaz quando usado em conjunto com outras técnicas de segurança, como RASP e WAF, formando uma abordagem de defesa em camadas que equilibra segurança e usabilidade.

Compreender a forma como estas tecnologias interagem é fundamental para desenvolver um mecanismo robusto e eficiente na detecção e bloqueio de usuários maliciosos. O conceito de segurança em camadas propõe que uma única proteção é menos eficiente do que a combinação entre vários destes elementos sobrepostos, de modo a maximizar a área de atuação das contramedidas de segurança.

No entanto, a utilização de diversas contramedidas esbarra no desafio de compor um ambiente seguro sem prejudicar a capacidade computacional dos ambientes. Há a necessidade de manter um equilíbrio entre a entrega da funcionalidade e o custo associado, seja ele computacional (como no caso de soluções associadas ao código-fonte e integradas ao ambiente), ou mesmo financeiro, já que muitos destes produtos são oferecidos na modalidade SaaS – Software as a Service – e apresentam valores que podem se tornar proibitivos em um cenário corporativo de grande volume de requisições, especialmente considerando serviços oferecidos a grande parte da população.

A necessidade de reagir a estes cenários diversos faz com que uma única solução adequada para todos os tipos de aplicação seja um desafio, e muitas vezes faz-se necessária a verificação prática da eficácia das soluções diante de um cenário específico para validar o seu funcionamento.

2.2. Ciência de Dados e Aprendizagem de Máquina

Diz-se que a Ciência de Dados, ou *Data Science*, é o estudo dos dados para extrair informações significativas para os negócios.[10] A Ciência de Dados consiste na aplicação de técnicas estatísticas e de análise de dados para a criação de valor a partir de conjuntos de informação, gerando conhecimento estratégico sobre o comportamento de usuários, clientes e cidadãos que podem ser aplicados no âmbito governamental, comercial, entre outros.

Durante o trabalho de desenvolvimento de uma solução para identificar e bloquear automaticamente requisições de um atacante, há a necessidade de adaptar as capacidades de proteção deste *software* de maneira dinâmica, uma vez que os usuários maliciosos estabelecem um processo contínuo de refinamento de técnicas. Por isso, é fundamental compreender e aplicar o conceito de Aprendizagem de Máquina (*Machine Learning*, em inglês), visando oferecer uma solução em constante evolução quanto às proteções. Este conceito baseia-se no autoaprendizado por parte do *software*, permitindo que este assimile novas capacidades à medida em que é utilizado. Assim sendo, à medida em que há possibilidade de utilizar grandes volumes de dados (gerados em tempo real na Internet e/ou armazenados em volumes físicos e virtuais) para oferecer uma ampla gama de usos nos mais diversos contextos, a criação de sistemas de proteção à informação é um segmento que vem se beneficiando constantemente desta evolução.

Os dois conceitos (ciência de dados e aprendizagem de máquina) se complementam em muitos aspectos, à medida em que os cientistas de dados coletam cada vez mais informações através de sistemas que realizam aprendizagem de máquina, e a partir da ciência de dados, definem modelos e heurísticas ainda mais eficazes na extração de informação a partir dos dados coletados.

2.2.1. Aprendizagem Supervisionada e Não-Supervisionada

A Aprendizagem de Máquina pode ser classificada de diversas formas, como a Aprendizagem Supervisionada, Não-Supervisionada, Semisupervisionada ou por Reforço, entre outros. Para compreensão dos conceitos aplicáveis à pesquisa, consideraremos os conceitos dos dois primeiros tipos (Supervisionada e Não-Supervisionada). A aprendizagem supervisionada ocorre quando um usuário ou cientista de dados é capaz de inserir informações manualmente em um sistema contendo rótulos ou saídas pré-estabelecidos, ao passo em que a aplicação constrói, dentro de seus modelos, a capacidade de elaborar respostas de maneira autônoma. Este tipo de aprendizagem é adotado em contextos em que o sistema auxilia na tomada de decisões a partir de entradas e saídas bem conhecidas, simulando a capacidade humana de interpretar informações para solucionar problemas.

[10] https://aws.amazon.com/pt/what-is/data-science/. Acessado em maio de 2024.

Na segunda categoria, aprendizagem não-supervisionada, encontram-se desafios onde o próprio modelo estabelece os rótulos a serem utilizados. Em heurísticas de agrupamento, um sistema computacional recebe um conjunto de dados e os agrupa por similaridade com base em parâmetros obtidos destes dados, de modo a manter os grupos tão coesos quanto possível. A partir dos grupos similares, é possível utilizar o próprio aprendizado para classificar informações futuras, bem como identificar dentre os grupos aquelas informações mais ou menos relevantes para a diferenciação dos registros. Este tipo de conhecimento normalmente permite inferências invisíveis aos olhos humanos, como correlações entre informações aparentemente dissociadas, uma vez que não aplica os rótulos conhecidos pelos cientistas de dados que treinaram o sistema.

2.2.2. Heurísticas de Agrupamento de Dados

Um dos fundamentos da Ciência de Dados diz respeito à capacidade de análise e interpretação de uma vasta quantidade de informações. Diante deste cenário, torna-se fundamental a utilização de um método de tratamento das informações para a identificação de padrões e relações em conjuntos complexos de dados. As heurísticas de agrupamento são ferramentas associadas a este objetivo, permitindo a associação e organização dos dados de modo a facilitar a compreensão das relações entre eles. Por se tratar de uma técnica de aprendizado não-supervisionado, é recomendada quando os dados não são anotados ou rotulados previamente, o que permite que seja utilizada em uma ampla gama de cenários, como a segmentação de clientes baseando-se em padrões de compra para sites de *web commerce*, gerando assim recomendações mais eficazes.

Além disso, o uso das heurísticas de agrupamento pode ser observado para detecção de anomalias, ou *outliers*, em um grande volume de dados. Esta aplicação desempenha papel crucial no tratamento de fraudes, por exemplo, em um modelo que pode ser associado ao de tráfego artificialmente produzido para subverter soluções de proteção à informação, como o CAPTCHA. Na prática, as anomalias são pontos que não se aproximam de nenhum dos grupos destacados pelo algoritmo de agrupamento, o que pode ser percebido com clareza após a execução da heurística. Neste caso, é possível utilizar a detecção de anomalias visando a melhoria do tratamento dos dados em análises subsequentes, por permitir que dados muito distintos dos grupos identificados sejam desconsiderados, reduzindo o ruído destes sobre as inferências realizadas na análise dos grupos.

Por fim, a visualização das características dos grupos é facilitada pelo agrupamento dos mesmos em características similares, o que representa uma vantagem que acompanha o volume de dados utilizados. Quanto maior o número de informações coletadas e analisadas, maior é a importância de uma boa heurística de agrupamento para a identificação de suas características, ampliando a possibilidade da obtenção de informações relevantes.

Uma das mais conhecidas técnicas de agrupamento é o *k-means*, que visa agrupar os dados em torno de centroides, ou seja, dados que representem a média dos atributos coletados a partir de

cada um dos registros. A heurística *k-means*, como visto por Bock et al. (2007), é, provavelmente, o método mais conhecido para o processo de segmentação de dados, que pode ser descrito como o problema de agrupar *n* pontos do espaço $\mathbb{R}^d$ em *k* grupos, de modo a maximizar a semelhança entre os membros de cada grupo. Dados *k* pontos em $\mathbb{R}^d$, denominados centroides, o *k-means* alterna entre dois procedimentos: (a) associa cada ponto ao centroide mais próximo; e (b) atualiza as coordenadas de cada centroide para a média de todos os pontos associados a este. Cada iteração do algoritmo permite que os centroides sejam ajustados entre os registros (comumente representados por pontos n-dimensionais, onde *n* representa a quantidade de atributos de cada registro), minimizando a distância entre os integrantes do seu grupo. Sendo o centroide a média daquele conjunto de dados, estima-se que o grupo apresente características similares que permitam classificar o agrupamento, mesmo na inexistência de um rótulo ou conceito definido para este.

2.2.3. Agrupamento Automático

Ao utilizar as heurísticas de agrupamento para encontrar características comuns entre dados e categorizá-los como parte do processo de classificação de risco aplicada ao objeto da pesquisa, é fundamental definir, além do algoritmo, a quantidade ideal de agrupamentos a ser adotada. Para tal, uma técnica comumente adotada é conhecida como método do cotovelo.

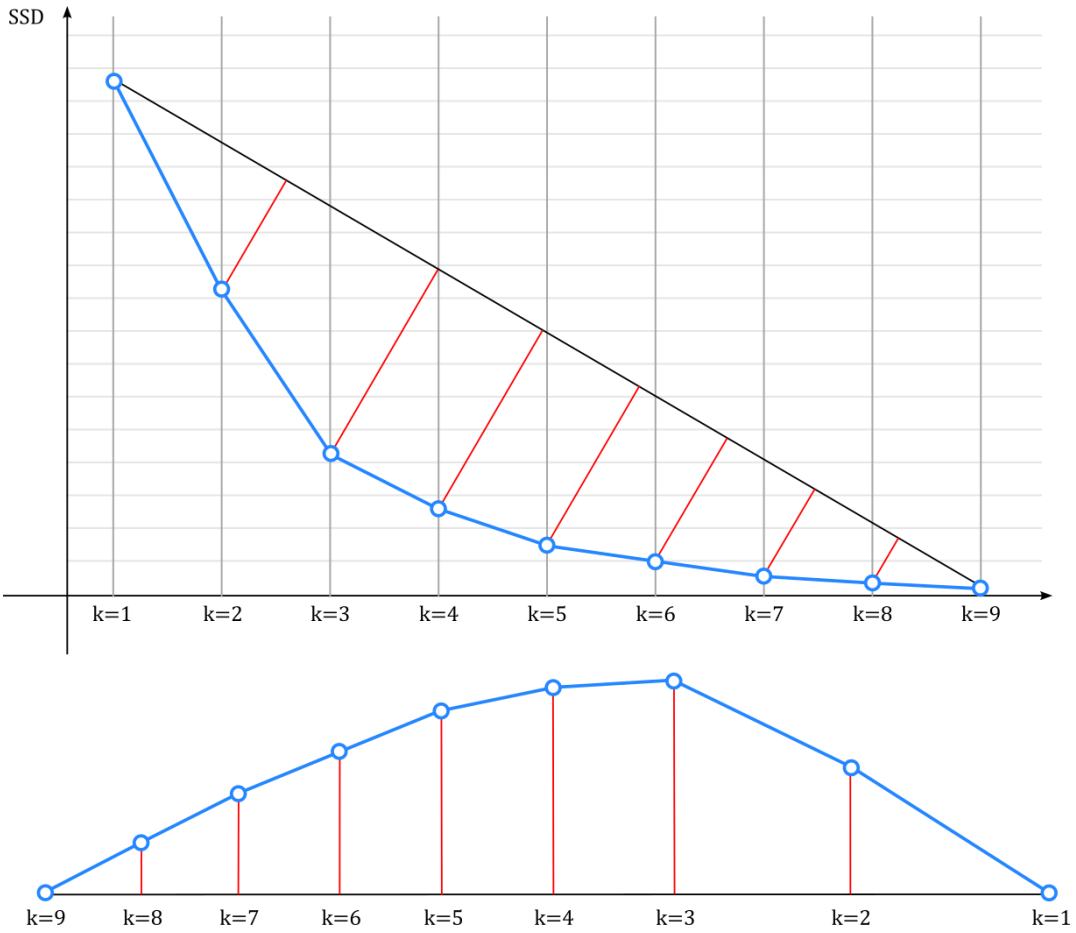
O método baseia-se em um mecanismo visual para observação do número ideal de agrupamentos em um gráfico entre a variação intra-*cluster* e o número (*k*) de grupos. Esta variação é obtida através da soma das distâncias quadráticas entre os elementos de cada grupo, resultando em um valor mais alto quanto mais distantes do centroide estes forem.

A soma das distâncias quadráticas entre os pontos (SSD, do inglês S*um of Squared Distances*), quando refletida em gráfico relativo a *k*, resulta em uma curva onde ocorre rápida variação nos primeiros valores de *k* e gradual diminuição, resultando em um gráfico com aspecto de "cotovelo", de onde deriva o nome do método. Portanto, o ponto de inflexão da curva (o cotovelo) é considerado o ponto em que o número de agrupamentos apresenta a melhor relação entre quantidade e diminuição da distância dos pontos pertencentes a ele, auxiliando a tomada de decisão sobre o valor de *k* a ser escolhido. No gráfico de exemplo na Figura 2, é possível perceber o ponto de inflexão ("cotovelo") em *k=3*, a partir do qual a diminuição entre as distâncias quadráticas dos valores de *k* torna-se cada vez menor. Tal percepção torna-se mais clara quando é traçada uma reta entre o menor e o maior valor de *k* e são medidas as distâncias perpendiculares entre os pontos da curva e esta reta. Ao rotacionarmos a curva, torna-se fácil perceber que o ponto *k=3* é aquele que está mais distante desta reta.

É importante destacar que o método do cotovelo não consiste em uma abordagem rígida para a definição do número de agrupamentos a ser utilizado, e sim em uma heurística que favorece a identificação visual deste número. O domínio do problema pode ser uma métrica igualmente eficaz

para a definição, dado que em determinados gráficos, a diferenciação pode tornar-se difícil. Nestas situações, a diferenciação das características predominantes de cada agrupamento pode ser fundamental para a escolha de um *k* específico para o problema alvo da pesquisa.

Figura 2 – Exemplo de Aplicação da Técnica do Cotovelo



3. TRABALHOS RELACIONADOS

Neste capítulo, apresentam-se uma variedade de trabalhos acadêmicos que possuem relação direta e relevante com o foco central desta pesquisa. Busca-se, através de uma revisão sistemática, conduzida sobre as interações entre os temas de Análise de Dados, Aprendizagem de Máquina e Segurança da Informação, observar de que forma os trabalhos acadêmicos avaliam as contribuições das capacidades oferecidas pela Inteligência Artificial para a proteção das propriedades de Segurança da Informação e a correlação destes artigos com os objetivos da pesquisa.

Ao longo deste capítulo, busca-se estabelecer as fundações teóricas e práticas sobre as quais a pesquisa atual se apoia, destacando o estado da arte em técnicas de segurança, agrupamento de dados e análise de padrões. A revisão destes trabalhos não apenas enriquece a compreensão do campo em estudo, mas também fornece um contexto crítico para as contribuições que este trabalho se propõe a fazer. Ao examinar as abordagens existentes e seus resultados, é possível identificar lacunas, oportunidades e potenciais direções inovadoras para avançar no tratamento a ameaças de segurança cibernética.

3.1. Metodologia da Revisão Sistemática

Visando aprofundar o referencial teórico da pesquisa, foram avaliados os contextos nos quais há aplicabilidade das técnicas de agrupamento e Aprendizagem de Máquina na resolução de problemas do campo da Segurança da Informação, especificamente quanto ao seu uso na detecção de robôs. Utilizando a técnica de revisão sistemática, avaliamos os resultados obtidos a partir da seguinte questão: Em quais contextos de ataque de robôs é possível utilizar técnicas de agrupamento para melhoria da proteção dos sistemas?

Como critérios de inclusão, foram utilizadas as palavras *cybersecurity* e *clustering* (direcionando a pesquisa para os temas de cibersegurança e agrupamento) na pesquisa realizada utilizando as ferramentas de busca ACM Digital Library, IEEE Xplore, Scopus e SpringerLink, e para critérios de exclusão, foram adotadas regras que excluem trabalhos publicados antes de 2020. Este período de corte objetiva a criação de uma análise baseada no estado da arte dos temas pesquisados, de modo a avaliar o direcionamento das pesquisas recentes sobre o tema. Como a palavra *clustering* também pode estar relacionada a contextos de *cluster* de servidores presentes em discussões sobre computação em nuvem e *data centers*, optou-se pela exclusão dos resultados relacionados à computação em nuvem com o afixo "-cloud", bem como reforçar as referências relacionadas a aprendizado (*learning*), dados (*data*) e classificação (*classification*).

O uso de mais um critério de inclusão (palavra-chave *logs*) e outro de exclusão (*IoT*, que representa um extenso domínio cujo escopo poderia interferir nos resultados da busca) foi capaz de refinar os resultados em mais um nível, enquanto, por fim, utilizando a possibilidade de análise

comportamental de usuários (*behavior* OR *behaviour*, a depender do país de publicação) e foco em ataques (*attack*) e robôs (*bot*), chegou-se ao número de 42 artigos. A Tabela 1 resume os critérios de inclusão e exclusão utilizados.

Tabela 1. Critérios de Inclusão e Exclusão da Revisão Sistemática

Critérios de Inclusão	***Critérios de Exclusão***
Artigos em português, inglês ou espanhol	Artigos anteriores a 2019
Cybersecurity	*IoT*
Clustering	*Cloud*
Learning	
Data	
Classification	
Logs	
Behavior OR behaviour	
Attack	
BOT	

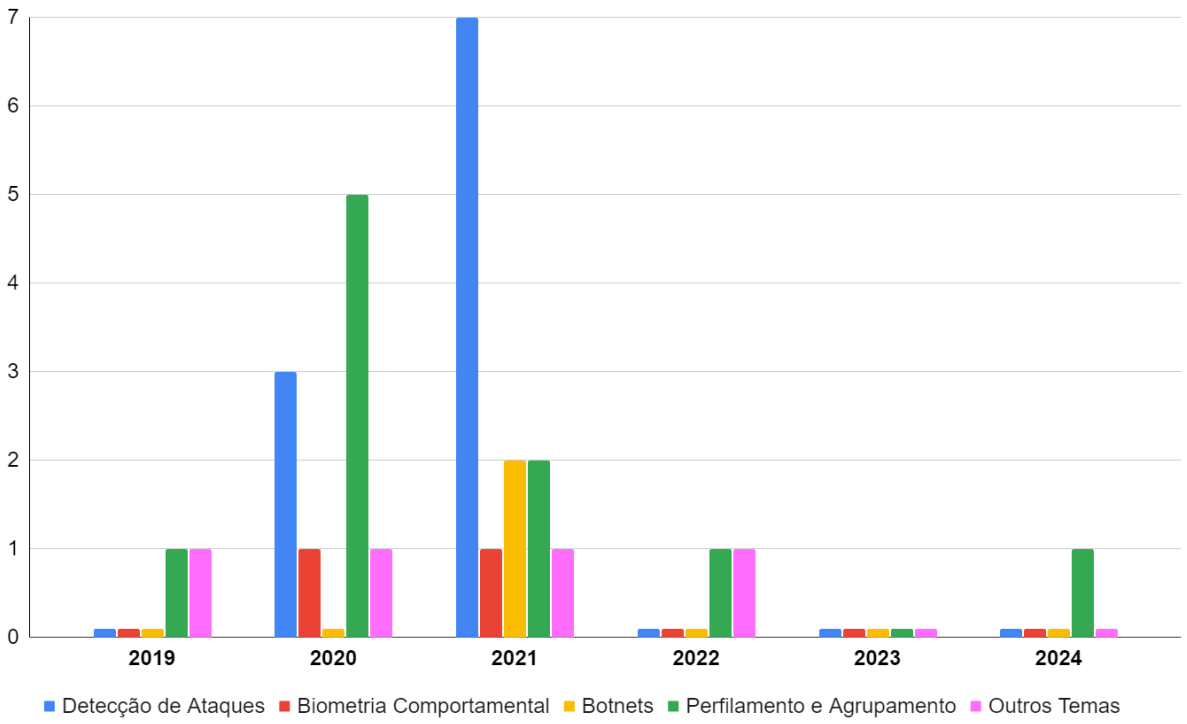
Em uma segunda etapa, foram observados as palavras-chave e o resumo de cada um dos artigos e inferido o tema e sua relação com o contexto da segurança cibernética. Nesta análise, foi possível descartar mais 14 artigos, grande parte deles relacionados ao uso de algoritmos para detecção de robôs que realizam interações fraudulentas em redes sociais, finalizando o escopo de análise em 28 artigos voltados para o uso de Aprendizagem de Máquina e Agrupamento, com foco principal nos contextos de segurança.

Entre os temas de segurança identificados na análise, é possível avaliar que grande parte dos artigos possui direcionamento ao desafio de detecção de ataques, *malware* e ameaças de intrusão (IDS, *Intrusion Detection Systems*). Dez artigos selecionados abordam técnicas sobre o assunto. Outros dez artigos abordam temas relacionados a Perfilamento e Agrupamento. Entre outros temas relacionados com a pesquisa, há predominância de análises de origem maliciosa de tráfego (*botnets*), com 2 artigos pesquisados, além de Biometria Comportamental (identificação de características humanas na interação com sistemas), com 2 trabalhos. A Tabela 2 traz uma visão geral sobre os trabalhos selecionados para investigação adicional. Por sua vez, a Figura 3 apresenta um gráfico relacionando a quantidade de trabalhos selecionados por tema e por ano, oferecendo uma visão mais ampla da distribuição temporal dos artigos relacionados a cada tema.

Tabela 2. Artigos Selecionados por Tema

Citação	Tema	Citação	Tema
Jin et al. (2020)	Detecção de Ataques	Liu et al. (2019)	Perfilamento e Agrupamento
Pawlicki et al. (2020)	Detecção de Ataques	Harifi et al. (2020)	Perfilamento e Agrupamento
Prasad et al. (2020)	Detecção de Ataques	Quezada et al. (2020)	Perfilamento e Agrupamento
McGahagan et al. (2021)	Detecção de Ataques	Maculan et al. (2020)	Perfilamento e Agrupamento
Liu et al. (2021)	Detecção de Ataques	Gelli et al. (2020)	Perfilamento e Agrupamento
Yuan et al. (2021)	Detecção de Ataques	Goßen et al. (2020)	Perfilamento e Agrupamento
Zoppi et al. (2021)	Detecção de Ataques	Guerreiro et al. (2021)	Perfilamento e Agrupamento
Atefinia et al. (2021)	Detecção de Ataques	Rezaei et al. (2021)	Perfilamento e Agrupamento
Shams et al. (2021)	Detecção de Ataques	Liu (2022)	Perfilamento e Agrupamento
Yousefnezhad et al. (2021)	Detecção de Ataques	Guo et al. (2024)	Perfilamento e Agrupamento
Hazan et al. (2020)	Biometria Comportamental	Easttom (2019)	Outros Temas
Hazan et al. (2021)	Biometria Comportamental	Samtani et al. (2020)	Outros Temas
Alahmadi et al. (2020)	*Botnets*	Samtani et al. (2021)	Outros Temas
Rahal et al. (2020)	*Botnets*	Wu et al. (2022)	Outros Temas

Figura 3 – Número de Artigos Selecionados por Tema e por Ano



3.2. Detalhamento das Categorias

As seções a seguir descrevem os artigos selecionados em cada uma das áreas predominantes, a saber: Detecção de Ataques, Biometria Comportamental, *Botnets*, assim como Perfilamento e Agrupamento de Dados. Por fim, serão apresentados outros trabalhos considerados relevantes ao tema, e será feita uma pequena discussão sobre os resultados obtidos neste capítulo.

3.2.1. Detecção de Ataques

O trabalho de McGahagan et al. (2021) é o que possui características mais próximas ao que se pretende implementar nesta pesquisa, ao ponto em que analisa respostas a requisições *web* para identificar características que permitam identificar sites maliciosos a partir de heurísticas de *Machine Learning.* O processo analítico de alto nível utilizado consiste em seleção de características, amostragem e técnicas de aprendizagem supervisionada e não-supervisionada, considerando características como URL do site, IP e porta conhecidos, total de caracteres aplicados nas URLs, extensões e TLDs, entre outros, que permitem caracterizar cada um dos sites e classificá-los com base em uma amostragem de 40000 *websites* válidos e 7039 maliciosos.

Jin et al. (2020) apresenta uma perspectiva interessante de uso de categorização para aplicação de IDS, resultando em um algoritmo de *Machine Learning* chamado SwiftIDS que reduz o tempo necessário para identificar uma intrusão, e consequentemente seu impacto em um ambiente monitorado. Já Liu et al. (2021) aplica um modelo adaptativo de geração de dados sintéticos utilizando as intrusões identificadas para amplificação das amostras de dados maliciosos que podem ser ignorados em um processo de aprendizagem de máquina. O pré-processamento aplicado neste contexto envolve a limpeza e normalização dos dados para facilitar o aumento das amostras a serem avaliadas pelo algoritmo de classificação LightGBM, apresentando resultados qualitativamente superiores.

Pawlicki et al. (2020) também avalia a necessidade de balanceamento entre os dados de ataque e de acessos legítimos, considerando que, na maioria dos casos, o tráfego predominante em uma amostragem é de usuários reais do sistema. Uma das abordagens utilizada neste caso é a sub-amostragem aleatória do dado predominante (acessos reais) para torná-lo mais próximo do número de dados que se deseja pesquisar (ataques). Quando comparado a técnicas de amplificação das amostras de ataque, o resultado obtido a partir da sub-amostragem aleatória encontra sucesso similar a um custo computacional inferior.

Ainda na linha de detecção de intrusão, o trabalho realizado por Prasad et al. (2020) tem como foco a identificação não-supervisionada a partir da seleção e agrupamento de características aplicáveis ao monitoramento de intrusos a partir de *datasets* conhecidos, selecionando 58 *features* aplicáveis ao algoritmo e oferecendo resultados interessantes para grandes volumes de dados.

Em outro aspecto da identificação de ameaças, o trabalho descrito por Yuan et al. (2021) oferece uma revisão sistemática sobre identificação de *insiders*, ou seja, ameaças internas aos sistemas (funcionários insatisfeitos ou maliciosos, que possuem permissões acima de um usuário normal) através do uso de *deep learning*. O trabalho apresentou dificuldades na escolha de um *dataset* real para identificação de *insiders*, mas utilizou o trabalho do CERT da Carnegie Mellon University como uma fonte simulada capaz de fornecer as informações necessárias nos estudos do tipo. Entre os principais desafios encontrados no trabalho de identificação de ameaças internas, ressalta a necessidade de detecção rápida para evitar perdas a partir dos atacantes identificados, assim como a heterogeneidade de suas ações maliciosas como impacto para a criação de um modelo confiável, traço perceptível em outros trabalhos da área de segurança.

O artigo descrito por Zoppi et al. (2021) prevê o uso de algoritmos de detecção de anomalias para identificar intrusões e ataques através de *datasets* que representam o comportamento de atacantes. Através do monitoramento de desempenho e outras características da aplicação e da rede, o método proposto se baseia menos em assinaturas de ataque, e mais em uma tentativa de detectar acessos fora do padrão, o que pode representar um número maior de falsos positivos. Com base em 11 *datasets* de ataque, o trabalho realiza uma comparação entre 17 métodos de identificação global, contextual e coletivo de anomalias onde se destacaram os algoritmos One-Class *Support Vector Machine* (SVM) e Isolation Forest, seguidos de perto pelos algoritmos de agrupamento. O trabalho também destaca que alguns ataques fazem uso dos mecanismos de proteção baseados em *Machine Learning* para burlar o mecanismo de falsos positivos, sendo este um ponto de atenção a ser considerado à medida em que os ataques se tornam mais sofisticados e direcionados a este tipo de proteção.

Em um complemento do trabalho anterior, o artigo escrito por Atefinia et al. (2021) apresenta o uso de uma rede neural para limitar o número de falsos positivos no uso de um algoritmo de detecção de anomalias para detecção de intrusão em uma arquitetura descentralizada que separa as verificações em subproblemas. Desta forma, o artigo obtém resultados positivos expressivos a partir de *datasets* conhecidos de maneira superior a redes neurais monolíticas.

Shams et al. (2021) aborda o problema de classificação de dados de intrusão a partir de uma nova perspectiva, utilizando um tipo de rede neural, *Convolutional Neural Network* (CNN), frequentemente associado a heurísticas de detecção de imagens utilizando *Machine Learning*. O método é abordado através de um pré-processamento que codifica características obtidas a partir dos *datasets* em valores categóricos de 0 a 255, representando um *pixel* na entrada das CNN, e realizando o processamento destas informações na rede neural. Os resultados obtidos pelo método são comparados a diversos outros modelos de *Machine Learning* para detecção de intrusão e atingem uma precisão superior aos demais avaliados.

Por fim, entre os artigos selecionados, Yousefnezhad et al. (2021) apresenta a abordagem combinada (*ensemble methods*) de diversos métodos de classificação para extração de características a serem adotadas em um modelo de *Deep Learning* em dados de intrusão em comparação com as técnicas kNN (*K-Nearest Neighbors*) e SVM isoladas.

3.2.2. Biometria Comportamental

Hazan et al. (2021) oferece uma visão sobre o tema de biometria comportamental ao avaliar o problema de compartilhamento de credenciais por vários usuários. Utilizando algoritmos de identificação de dinâmicas de digitação que permitem identificar unicamente o padrão de escrita de um usuário, o trabalho oferece uma alternativa para identificar múltiplos humanos utilizando uma mesma conta de usuário apenas a partir de suas características de interação com o teclado.

Hazan et al. (2020) também apresentam um trabalho na área de identificação de padrões de digitação que reduz o risco de interceptação dos metadados de digitação, que podem conter informações confidenciais, como senhas. Este tipo de tratamento tem grande importância na utilização de dados sensíveis para pesquisa uma vez que, mesmo a prática sendo contemplada na LGPD como um uso aceitável para tratamento de dados de terceiros, sua segurança deverá ser garantida.

3.2.3. Botnets

No campo das *botnets*, o trabalho de Alahmadi et al. (2020) oferece uma proposta de uso de Cadeias de Markov para classificação do fluxo de atuação dos acessos automatizados providos por estas redes. Com a seleção de características a partir dos fluxos mapeados, é possível identificar padrões de comunicação destes robôs quanto a ações como *Command and Control* (C&C), *Distributed Denial of Service* (DDoS) e *scanning* de portas e classificá-los entre 12 *botnets* existentes com taxa de acerto superior a 99%. O trabalho ressalta, porém, que a inserção de um atraso aleatório nas comunicações a partir da *botnet* pode comprometer a capacidade de identificação do modelo proposto, sendo um ponto a mais de atenção no tratamento de requisições automatizadas.

Rahal et al. (2020) aplicam o conhecimento sobre detecção de robôs para tentar prevenir ataques de DDoS em uma arquitetura distribuída, aplicando a *datasets* de ataques a extração de características de rede como protocolos utilizados, *time-to-live* (TTL), tamanho da janela TCP, entre outros. Com base neste modelo, a proposta utiliza 4 módulos (Coleta de dados e extração de características, Processamento, Análise e Notificação) capazes de identificar uma tentativa de ataque de DDoS e notificar os administradores dos sistemas para a adoção de medidas de contenção. Dentro dos *datasets* utilizados, a arquitetura proposta obteve resultados positivos em mais de 99% dos casos.

3.2.4. Perfilamento e Agrupamento

No aspecto da pesquisa voltado para os temas de perfilamento e agrupamento, Harifi et al. (2020) apresentam um estudo comparativo com dez meta-heurísticas populares, realizando experimentos em conjuntos de dados sintéticos e reais. Após a análise estatística, os autores concluem que todos os métodos possuem desempenho melhor que o *k-means* e que, dentre estes, o *Particle Swarm Optimization* (PSO) alcança o melhor desempenho, seguido pelos Algoritmos Genéticos (AG) e pelas técnicas de Evolução Diferencial (ED). Tal conclusão é corroborada pelos trabalhos de Guerreiro et al. (2021), que obtiveram melhor desempenho utilizando PSO e AG para agrupar e analisar componentes da indústria automobilística.

Quando o conjunto de dados de entrada é pequeno, é possível utilizar métodos exatos para encontrar o melhor agrupamento possível. Quezada et al. (2020) utilizam duas formulações matemáticas, uma não linear e outra que utiliza o conceito de representantes para designar o centro de cada grupo. Por sua vez, Maculan et al. (2020) apresentam duas novas formulações para segmentação de dados, aplicando novas estratégias para obter um modelo quadrático convexo que pode ser resolvido de forma eficiente por algoritmos de pontos interiores, possibilitando a resolução de problemas com até 20 pontos.

Em relação ao uso de técnicas de agrupamento para análise de dados, Quezada et al. (2020) utilizaram um modelo baseado em Programação Matemática Inteira para identificar padrões em um banco de dados de animais bovinos, com objetivo de melhorar as vendas daqueles com as melhores características e encontrar novas estratégias para melhorar o desenvolvimento daqueles com as piores. Por sua vez, Gelli et al. (2020) utilizaram o *k-means* para descrever o modo como times de futebol se comportam em relação à defesa, progressões, lateralidade e eficiência. Os resultados apontaram para a importância da eficiência nos fundamentos deste esporte.

No contexto das técnicas de perfilamento e agrupamento associadas à Segurança da Informação, Rezaei et al. (2021) aplicou métodos de agrupamento em conjunto com técnicas de aprendizado profundo para definir novos meios para diferenciar aplicativos maliciosos de não maliciosos. Para tal, a saída da rede neural foi usada como entrada para o *k-means*, responsável por segmentar estes dados entre dois grupos: aplicativos maliciosos e benignos. É importante destacar, no entanto, que uma solução baseada apenas na identificação de *fingerprints* pode não ser eficaz no tratamento de todos os ataques. O trabalho de Goßen et al. (2020), por exemplo, oferece uma perspectiva de remoção de informações de identificação de um agente de *web scraping*.

Mais ocorrências que constatam a relevância das heurísticas de agrupamento para o campo da Segurança da Informação podem ser encontradas em trabalhos como o descrito por Guo et al. (2024), que visam utilizar a técnica de análise através de heurísticas de agrupamento para identificar riscos de envenenamento em redes neurais, a partir da análise dos grupos em busca de injeções de dados maliciosos no *dataset* utilizado para treinamento. A análise de *clusters* também é parte

integrante do trabalho realizado por Liu (2022), que aplica o *k-means* sobre dados de teste de maneira iterativa, visando obter o número de grupos com menor quantidade de falsos positivos.

O uso de técnicas de inteligência artificial encontra sucesso em abordagens relacionadas ao comportamento de usuários. O trabalho de Liu et al. (2019) demonstra a capacidade do uso de bases centralizadas de logs na detecção de incidentes de segurança através de *deep learning*. Embora focado em ameaças internas (*insiders*), o aprendizado provido por registros de uso de aplicações e elementos de infraestrutura prova-se fundamental na construção de uma base de conhecimento capaz de identificar e prevenir novos ataques.

3.2.5. Outros Temas Relacionados

O trabalho realizado por Easttom (2019) possui como característica principal a utilização de *Machine Learning* em um processo ofensivo, ao invés de defensivo. O artigo permite visualizar aplicações de algoritmos probabilísticos, árvores de decisão e algoritmos de agrupamento para amplificar as capacidades de um *malware*, permitindo que se adapte dinamicamente ao contexto que está sendo atacado, gerando uma ameaça ainda mais difícil de ser devidamente identificada e tratada por técnicas defensivas.

Samtani et al. (2020) se propõe a oferecer um *roadmap* de aquisição de conhecimento para aplicação de uma IA para ações de cibersegurança, ou seja, auxiliar a tomada de decisão (ou até mesmo tomá-las de maneira autônoma) no tratamento de incidentes de segurança, detecção de ameaças, priorização de recursos e gerenciamento de vulnerabilidades. Através de uma extensa lista de fontes de informação relacionadas à disciplina de Segurança da Informação, partindo desde recursos internos como armazenamento de dados, logs, dados biométricos e tráfego de rede e se expandindo até os externos, como repositórios públicos de código, fóruns, redes sociais e sites de notícias, o artigo apresenta uma série de ações necessárias para a construção deste conhecimento em segurança a ser aplicado por uma IA especializada. Entre as ações propostas, podemos ressaltar a possibilidade do uso de *Generative Adversarial Networks* (GANs) para produção de simulações próximas a dados reais (com propósitos ofensivos ou defensivos) e a aplicação da IA na construção de soluções autônomas, capazes de responder dinamicamente a uma ameaça com maior agilidade.

Em mais uma contribuição, Samtani et al. (2021) propõe a utilização de *data sources* da Deep Web na tentativa de prever movimentos de grupos de *hackers* em um processo de monitoramento contínuo das atividades maliciosas, visando prever cenários de ataque. A solução, denominada AZSecure HAP, utiliza uma ampla gama de estratégias para burlar os controles de proteção dos sites da Deep Web, alguns desses baseados em técnicas comuns aos próprios atacantes, como o ajuste fino da frequência de requisições para evitar disparos de controles contra DDoS e alteração constante de IP para evitar *blocklists*. A iniciativa conta com o apoio de mais de 200 órgãos de pesquisa, indústria e segurança.

Wu et al. (2022) utiliza uma premissa similar, mas observa os conceitos disponíveis em plataformas de informação sobre segurança para identificar as tendências da abordagem do tema cibersegurança desde o ano 2000 e como estas se comportam na ocorrência de ataques. O trabalho resultou em 16 classificações que foram avaliadas ao longo do tempo e tiveram sua popularidade analisada dentro deste período, de modo a exibir as tendências entre os temas avaliados e sua correlação com formas de ataque.

3.3. Discussão

Dentre os artigos selecionados, puderam ser observadas diversas aplicações do conhecimento de Aprendizagem de Máquina na área de Segurança da Informação. A possibilidade de explorar características dos protocolos de rede como fontes de informação capazes de identificar tráfego suspeito é um tema relevante em diversas pesquisas, e deve ser uma das informações importantes na construção de um sistema de tomada de decisão quanto ao acesso a partir de robôs. O mesmo pode ser aplicado a partir dos estudos sobre *botnets*, que apresentam formas de identificar origens maliciosas através do rastreamento de origens e comportamentos comuns a atacantes.

Ressalta-se, no entanto, a atenção dada a diversos estudos no aspecto do tratamento de falsos-positivos oriundos desta análise. Determinados padrões de tráfego podem ser confundidos com ataques reais e bloqueados por sistemas de proteção, quando na verdade tratam de cenários válidos de acesso (por exemplo, uma das regras de proteção pode se estabelecer a partir da grande quantidade de requisições originadas do mesmo IP, quando na verdade este acesso é uma implementação legítima de saída de uma rede interna a partir de um NAT). Por isso, qualquer solução de proteção deve ser capaz de tratar falsos positivos de maneira ágil e adaptativa.

O grande volume de implementações propostas que utilizam tráfego simulado também pode ser indicativo de um grande diferencial do presente trabalho, uma vez que este se propõe a aplicar a solução a ser desenvolvida em um ambiente real, onde há demanda substancial da ordem de milhões de requisições. Este contexto real proporciona uma oportunidade valiosa para avaliar o desempenho da solução em condições dinâmicas e autênticas, diferenciando-se significativamente das pesquisas que se baseiam em dados de ameaças previamente definidos e estáticos, e não foi observado nos artigos da revisão. Mesmo aqueles que utilizaram tráfego real o fizeram com base em massas de dados conhecidas de ataques, limitando o seu alcance a situações já conhecidas e observadas pelo campo acadêmico.

Além deste caráter inovador, outra vantagem encontrada ao aplicar a solução em cenários reais diz respeito à sua aplicabilidade em diferentes arquiteturas de sistemas. Essa característica é especialmente relevante em ambientes corporativos diversificados, onde os produtos e serviços oferecidos variam amplamente. Ao aplicar esta abordagem, é possível avaliar o comportamento da solução em uma gama de contextos operacionais, desde sistemas mais simples a infraestruturas

complexas, fornecendo uma compreensão abrangente sobre como a solução pode ser adaptada e implementada de maneira eficaz em diferentes cenários, e as formas como este comportamento pode variar.

4. ARQUITETURA PROPOSTA

Neste segmento da pesquisa, o objetivo é desenvolver uma solução baseada em Aprendizagem de Máquina (Machine Learning), especificamente projetada para identificar e neutralizar requisições mal-intencionadas em sistemas *web*. Esta solução visa abordar duas categorias principais de ameaças: ataques direcionados a sistemas *web* e o uso de robôs para tarefas de *web scraping* em ambientes de aplicativos reais. O foco da solução está em proteger os recursos computacionais e os dados sensíveis sob duas óticas cruciais: prevenindo o uso indevido e intensivo dos recursos do sistema e salvaguardando informações confidenciais contra acessos não autorizados.

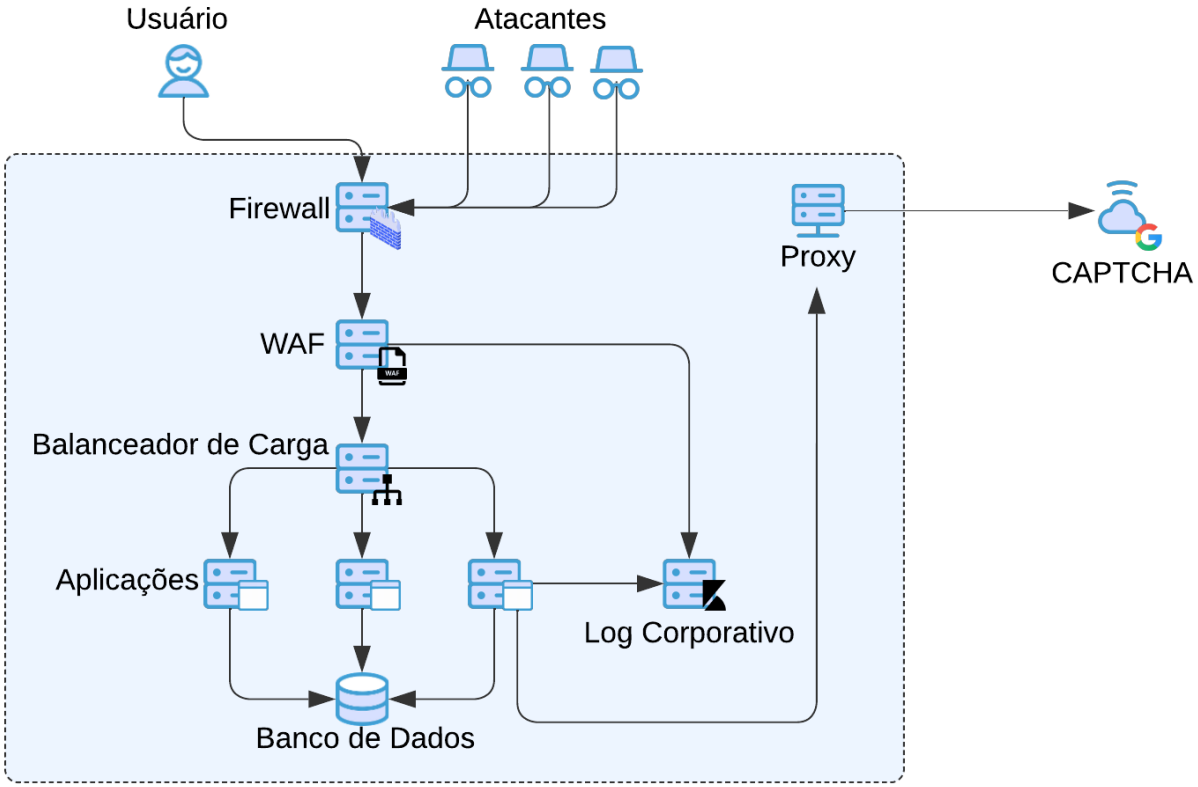
A eficiência da solução proposta será avaliada em relação à sua capacidade de detecção de ataques. Esta avaliação incluirá uma análise aprofundada da eficácia da solução em identificar e responder a tentativas de exploração, especialmente em situações de ataques contínuos ou repetidos. A ênfase será dada à precisão, rapidez e confiabilidade da solução em um cenário dinâmico e realista, refletindo a complexidade e a variedade dos ataques que os sistemas *web* modernos enfrentam regularmente.

4.1. Aplicabilidade

A proposta de solução apresentada tem um papel crucial no contexto empresarial, especialmente no que diz respeito ao fornecimento de informações em aplicações *web* de grande escala, que são acessadas por um vasto público. Observa-se uma tendência crescente nas aplicações *web* em priorizar a amplitude de acesso aos serviços, muitas vezes em detrimento da implementação de controles de segurança mais rigorosos. Esta abordagem é motivada tanto por questões de usabilidade quanto pela intenção de alcançar um público mais amplo. Essa dinâmica tornou-se particularmente evidente durante a pandemia do COVID-19, quando foi essencial que serviços governamentais atendessem a populações vulneráveis. Frequentemente, esses serviços incluem a oferta de informações sensíveis, aumentando a necessidade de equilibrar a acessibilidade com medidas de segurança adequadas para prevenir abusos.

No cenário atual, o fornecedor destes serviços dispõe de uma série de controles de segurança capazes de realizar esta proteção de maneira eficaz. A correta aplicação destes recursos é parte fundamental da construção de uma arquitetura robusta e eficiente, uma vez que grande parte das proteções representa um consumo substancial de recursos financeiros e computacionais. O tênue equilíbrio entre um controle efetivo de segurança e a disponibilidade da informação deve ser perseguido por uma arquitetura da solução *web* dentro dos moldes descritos através da Figura 4:

Figura 4. Arquitetura Padronizada



A Figura 4 representa um fluxo tradicional de proteção de aplicações *web*, com o *firewall* de rede posicionado na entrada da área gerenciada, em azul claro. Como primeira linha de defesa, o *firewall* de rede (ou apenas *firewall*, como é convencionalmente chamado) apresenta a proteção no nível de rede, bloqueando e autorizando IPs com base em políticas previamente estabelecidas. O WAF, por sua vez, realiza este tipo de proteção na camada de aplicação, através da interceptação do tráfego. Ressalta-se neste momento que os pacotes necessitam ser decriptados para que ocorra a inspeção através do WAF, e este possa atuar através de políticas da camada de aplicação. Então o tráfego HTTPS deve ser então decriptado através do compartilhamento da mesma chave privada utilizada no balanceador. Caso alguma das políticas aplicadas retorne à ação de bloqueio, o WAF atua para derrubar esta conexão.

Por outro lado, caso o *request* seja validado, passa então ao balanceador de carga, que direciona a requisição às múltiplas instâncias da aplicação. Neste momento, pode ocorrer uma verificação adicional baseada no serviço de CAPTCHA, acessível através de um servidor *proxy* situado na rede gerenciada. Este servidor realiza a conexão com o serviço externo, validando (ou não) o acesso e servindo como uma verificação adicional ao WAF e ao *firewall* de rede. Só após as verificações dos múltiplos pontos de acesso, os dados podem ser consumidos pelo usuário final.

Todos os acessos são registrados através de um serviço de log corporativo, que realiza a persistência dos *logs* em um servidor centralizado. Este tipo de serviço também permite a aplicação de políticas de segurança sobre seus registros, trabalhando como SIEM (*Security Information and Event Management*), com alertas para eventos de segurança especificados.

Este modelo arquitetural simplificado permite a visualização dos diversos elementos de segurança necessários à proteção de uma aplicação quanto a ataques e acessos automatizados oriundos do ambiente externo. A atuação dos diversos recursos de segurança em conjunto oferece uma série de camadas de proteção, elevando a resiliência da arquitetura.

Porém, o custo da manutenção destes recursos tende a aumentar em um ambiente corporativo de maior porte. Instanciar um WAF pode ter custo elevado, muitas vezes tratando-se de um *appliance* instalado no *data center*. A aplicação de políticas em excesso também representa um risco, pois o WAF dispõe de uma quantidade limitada de memória, o que representa um desafio para políticas de bloqueio baseadas em número de requisições por tempo (por exemplo, para a construção de uma política de bloqueio acima de 100 acessos por hora, é necessário que todas as sessões de usuário sejam monitoradas no *appliance* durante este período, com a contabilização das tentativas de acesso). Soluções de WAF em nuvem representam uma alternativa a estas questões, mas trazem também um questionamento: por fazer necessária a introspecção do tráfego *web*, incorre em risco à soberania das comunicações, especialmente caso a nuvem esteja situada fora do território nacional.

O CAPTCHA, por sua vez, possui modelo de cobrança baseado em número de requisições, e deve ter sua utilização restrita a cenários onde é mais eficaz, por exemplo em telas de autenticação e consultas não-autenticadas de grande visibilidade. Enquanto seu funcionamento aplica diversos recursos de monitoramento das ações de usuário e parâmetros de navegação, a utilização dos serviços do CAPTCHA para todas as operações registradas pelo usuário pode ser proibitiva do ponto de vista econômico. Sua eficácia também é reduzida em contextos em que ocorre a automação "humana" do processo, através de serviços que simulam ações legítimas a partir de um grande grupo de usuários.

Em face desses desafios, a este trabalho consiste em apresentar uma solução adaptável e eficaz, com o objetivo de identificar tráfego *web* indevido com alta precisão e, consequentemente, intervir para impedir o consumo excessivo de recursos por agentes mal-intencionados. Este equilíbrio entre acessibilidade e segurança é vital para garantir que os recursos computacionais não sejam sobrecarregados, permitindo que todos os usuários legítimos tenham acesso ininterrupto e eficiente aos serviços oferecidos. Além disso, a solução proposta é projetada para ser flexível o suficiente para se adaptar a diferentes cenários arquiteturais de aplicações *web*, tornando-a uma ferramenta valiosa para uma ampla gama de contextos empresariais e governamentais, onde a segurança da informação é tão crítica quanto a sua disponibilidade.

4.2. Metodologia

Dentro do contexto proposto, para a definição da metodologia aplicável, deve-se iniciar pela identificação do tráfego malicioso gerado pelos atacantes. O primeiro passo envolve a minuciosa classificação das informações capturadas a partir do monitoramento de acessos. Estas informações são processadas e avaliadas quanto à sua natureza, utilizando-se heurísticas de agrupamento capazes de alocar acessos de características similares em grupos específicos, diferenciando-as e auxiliando na observação de suas distinções, bem como a eficácia dessa diferenciação a partir da análise de quais destas características tornam mais evidente a diferenciação entre acessos legítimos e fraudulentos. A partir da identificação realizada, é possível elaborar um mecanismo prático para a identificação de tráfego gerado maliciosamente, e adotar estratégias de mitigação do impacto destes agentes.

4.2.1. Captura e Análise das Informações

O processo se inicia com aplicação de recursos computacionais para a obtenção de arquivos de *log* reais dos ativos de rede, com foco nas soluções de proteção. Estes registros, em formato JSON padronizados para consumo a partir dos agregadores de *log*, passam por um estágio de pré-processamento para auxiliar a subsequente Análise, a ser realizada na próxima etapa do trabalho.

Após a verificação das informações relevantes para a pesquisa, será aplicada a Análise de Dados sobre um grande volume de informações coletadas. Esta análise, realizada utilizando ferramentas *open source*, pretende confrontar as informações recebidas com dados de bloqueio de ativos de proteção, de modo a identificar quais são as informações mais relevantes na tomada de decisão de sistemas de proteção. Este conhecimento será fundamental na próxima etapa do trabalho, de construção da solução.

4.2.2. Construção do Classificador

A solução proposta contempla a criação de um arcabouço de identificação de ameaças capaz de apontar de maneira eficaz a probabilidade de um *request* representar um ataque ou um acesso legítimo. Este processo se dá em três etapas: captura dos dados coletados pelos ativos de rede, classificação dos dados obtidos com base no aprendizado existente, e disparo da solução de bloqueio baseada na interpretação da classificação.

Ao receber um *request*, a primeira etapa do tratamento, denominada identificação de ameaças, terá como objetivo classificar o risco de uma requisição a partir de suas características, com base no conhecimento estabelecido pela análise de dados constante da fase anterior. Este conhecimento inicial poderá ser aplicado na alocação de recursos de proteção, além de ser fundamental para a tomada de decisão do módulo de contenção na próxima fase do tratamento.

Após a identificação do risco, a requisição será tratada por um módulo de contenção associado à aplicação *web*. Durante seu desenvolvimento, serão considerados os elementos necessários para a proteção em diversos contextos, considerando-se a possibilidade de a solução integrar-se a *firewalls* corporativos (através de APIs) e do próprio sistema operacional (implementando a execução de comandos), permitindo que o controle ocorra em ambientes gerenciados e não-gerenciados.

Por fim, o monitoramento constante da eficácia dos controles implantados irá retroalimentar a solução de Aprendizagem de Máquina a partir dos resultados obtidos. A partir das informações obtidas sobre variações nos padrões de ataque, a solução poderá se adaptar a mudanças realizadas pelos atacantes na tentativa de subverter a proteção implementada, além de promover a capacidade de observação sobre o funcionamento da solução para eventuais ajustes.

4.2.3. Experimentação e Resultados

A solução desenvolvida será submetida à avaliação de sua eficácia em suas três etapas: incialmente, avaliar-se-á o mecanismo de coleta das informações quanto à otimização do tratamento dos dados obtidos a partir dos registros de ferramentas de proteção, de modo a torná-lo eficaz e não atuar como gargalo do processo; no mecanismo de classificação, será avaliada a eficácia no tratamento a partir da comparação entre os resultados obtidos com o seu treinamento e os rótulos oferecidos pela solução de proteção; e, por fim, o mecanismo de bloqueio adotado será avaliado quanto à estratégia adotada na definição das regras de bloqueio das origens identificadas como maliciosas, de modo a minimizar o risco de indisponibilização para usuários reais ao mesmo tempo em que reduz a possibilidade de exploração para usuários que apresentem comportamentos indevidos de acordo com a classificação da etapa anterior.

4.3. Desenvolvimento da Solução Proposta

Uma vez que um dos principais objetivos da pesquisa é analisar o impacto da seleção de características relevantes na classificação de informações, ao iniciar o desenvolvimento da solução optou-se pela captura e análise de dados, de modo a obter conhecimento sobre os protocolos e métodos utilizados pelos atacantes e, através da compreensão das características de destaque do tráfego analisado, gerar o conhecimento necessário para embasar a tomada de decisão quanto à adoção de proteções do sistema, na fase seguinte.

4.3.1. Heurística de Agrupamento Proposta

Partindo do pressuposto de que diferentes métodos apresentam diferente desempenhos a depender do conjunto de dados de entrada, optou-se pela proposição de um novo método de agrupamento, cujo objetivo é a melhor performance para o conjunto de dados utilizado neste trabalho. Esta seção descreve este novo método de agrupamento proposto, baseado na meta-

heurística GRASP. O Algoritmo 1 traz uma visão geral do procedimento, que recebe um conjunto de pontos *P*, e os inteiros *k* e t_{lim}, representando o número de grupos e o tempo limite, respectivamente. Na linha 1, é inicializada a melhor solução com um valor trivial, enquanto o laço *MultiStart* das linhas 2 a 5 busca por uma solução melhor. Na linha 3 é construída uma solução inicial com 50% de chance de usar o algoritmo *k-means* e 50% de chance de usar sua variante *k-medoids*. Ambos são detalhados na sequência. Na linha 4 é executada a busca local, enquanto a linha 5 mantém a melhor solução encontrada.

Algoritmo 1. Implementação do GRASP

Procedimento GRASP(P, k, t_{lim})

```
1  Seja K* o melhor particionamento encontrado
2  Enquanto t < t_lim Faça
3      K ← BuildKMeans(P, k, α) OU BuildKMedoids(P, k, α), equiprovavelmente
4      K ← BuscaLocal(P, k, K)
5      K* ← Melhor( K*, K )
6  Retorna K*
```

O Algoritmo 2 detalha a heurística construtiva proposta no BuildKMeans, que recebe um parâmetro alfa para controlar a sua aleatoriedade. Na linha 1, são escolhidos *k* centroides em $\mathbb{R}^d$ e o laço das linhas 2 a 10 é executado até que o método convirja. O método trabalha exatamente como o *k-means*, exceto que, na primeira iteração há uma probabilidade alfa de cada ponto se conectar com o segundo centroide mais próximo, como detalhado nas linhas 6 e 7. Tal alteração possibilita ao método a exploração de novas áreas do espaço de soluções. Por sua vez, o procedimento BuildKMedoids altera a linha 1 de modo que sejam escolhidos aleatoriamente *k* pontos de *P* como centroides.

Algoritmo 2. Implementação do BuildKMeans

Procedimento BuildKMeans(*P, k,* α)

```
1  Seja C_o um conjunto de pontos posicionados aleatoriamente em ℝ^d
2  Para Cada i = 1, 2, 3, ... Faça
3      C_i ← C_i-1
4      Para Cada p ∈ P Faça
5          Associe p ao ponto mais próximo em C_i
6          Se i=1 E rand(0,1) < α Então
7              Associe p ao segundo ponto mais próximo em C_i
8      Para Cada c ∈ C_i Faça
9          Atualize c para a média dos pontos associados a este
```

10 **Se** $C_i=C_{i-1}$ **Retorna** $K = (C_i, P)$

Finalmente, dado um agrupamento K, composto pelo conjunto de centroides C e pelo conjunto de pontos P com suas associações, o Algoritmo 3 detalha a heurística de busca local proposta. São executadas até 50 tentativas de melhora da solução, apresentadas no laço das linhas 2 a 6, enquanto as linhas 1 e 6 mantêm a melhor solução encontrada. Para cada iteração, um ponto é escolhido aleatoriamente (linha 3) e é associado a outro grupo, também escolhido de forma aleatória (linha 4). Então, o método *k-means* é utilizado para levar esta nova solução a um mínimo local.

Algoritmo 3. Implementação da Busca Local

Procedimento BuscaLocal($P, k, K = (C, P)$)

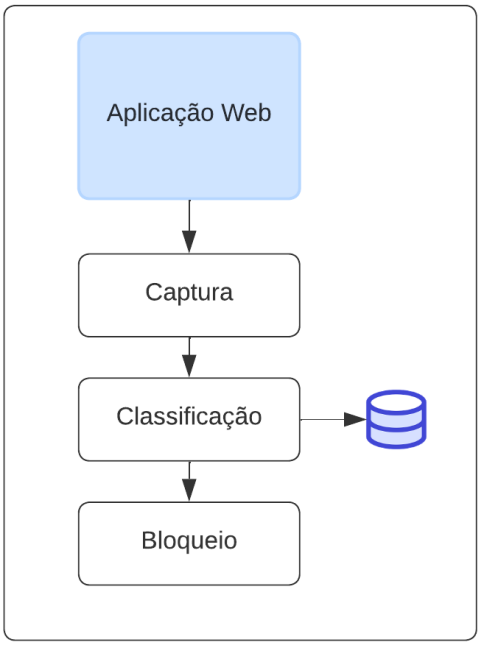
1 $K^* \leftarrow K$
2 **Para Cada** $i = 1, 2, 3, ..., min(|P|,50)$ **Faça**
3 Escolha um $p \in P$ aleatoriamente
4 Associe p a um ponto $c \in C$ escolhido aleatoriamente
5 $K' \leftarrow$ KMeans($K^* = (P, C)$)
6 $K^* \leftarrow$ Melhor(K^*, K')
7 **Retorna** K^*

4.3.2. Arquitetura da Solução

Após a pesquisa identificar os campos predominantes na aplicação de heurísticas de agrupamento, definiu-se a estratégia para desenvolvimento de uma solução que pudesse assimilar, a partir destes campos, a classificação das requisições recebidas em categorias de risco similares àquelas utilizadas pelo WAF. Esta abordagem pretende alcançar dois objetivos: permitir a descentralização (e consequente independência) do modelo quanto ao contrato com fornecedores de soluções de proteção, além de diminuir o custo (computacional e financeiro) envolvido na operação destes utilitários.

Para alcançar estes dois objetivos, a Figura 5 detalha uma solução consistindo em uma API (do inglês, A*pplication Programming Interface*) para classificação de tráfego baseado no conhecimento adquirido a partir da categorização, através de um modelo de aprendizagem de máquina baseado nos campos da requisição considerados mais relevantes. Estes campos alimentam uma base de conhecimento que é capaz de inferir, a partir de uma nova requisição, se os campos extraídos indicam uma tendência pelo bloqueio ou não de um dado IP.

Figura 5. Arquitetura Autocontida para a Solução

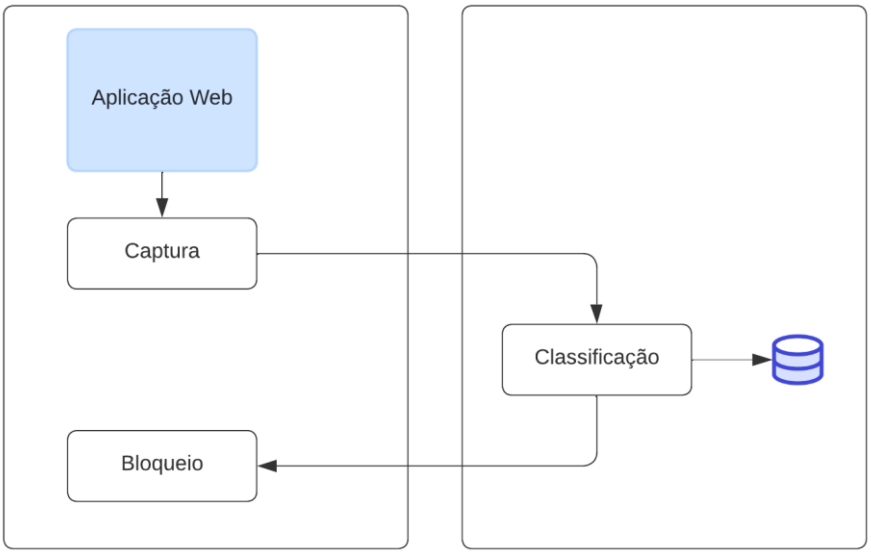


O modelo da Figura 5 representa três módulos com propósitos distintos: um de captura e tratamento de metadados, que funciona como um filtro associado à aplicação e irá apenas rotear os metadados coletados a partir da interceptação da requisição para a API de classificação. Desta forma, a solução pretende onerar o mínimo possível o funcionamento do sistema, uma vez que os métodos de classificação e bloqueio serão independentes e utilizarão ferramentas da própria infraestrutura para o tratamento do risco.

O módulo de classificação, por sua vez, representa a lógica de tratamento da requisição. A API disponibiliza um método que recebe os metadados e os confronta com a base de conhecimento alimentada inicialmente, e que pode ser expandida através de entradas externas. A partir do resultado da classificação, a solução dispara o terceiro módulo, responsável pelo bloqueio. Este foi desenvolvido considerando a possibilidade de uso em sistemas operacionais Unix e derivados, aplicando o conceito de bloqueio através da ferramenta **iptables** e o serviço de agendamento do próprio sistema para definir os tempos de bloqueio e liberação do usuário.

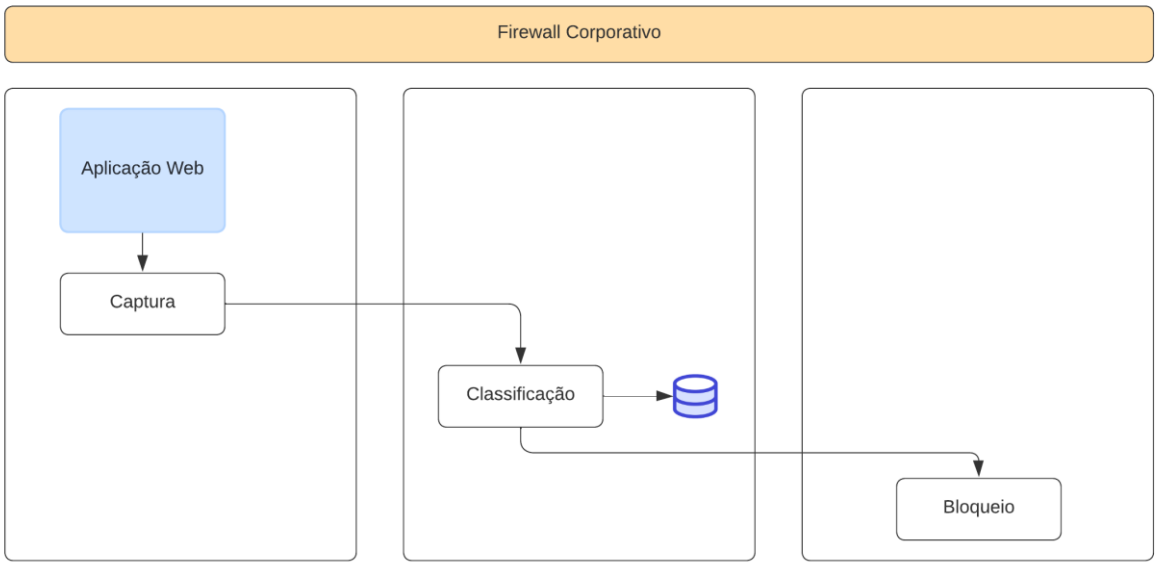
A separação dos módulos visa diminuir a interdependência entre eles, bem como escalá-los de forma independente. O conceito se assemelha a uma arquitetura de microsserviços, e pode ser bastante eficaz em um contexto diverso quanto a tecnologias de desenvolvimento, formas de coleta dos metadados e *firewalls*. Pode-se operacionalizar desta forma em dois modelos distintos, representados nas Figuras 6 e 7, em uma estratégia distribuída entre vários servidores ou através de uma arquitetura de *container*, de modo a prover a escalabilidade entre os módulos proposta.

Figura 6. Proposta de Arquitetura Distribuída Apenas para o Módulo de Classificação



Enquanto a opção retratada na Figura 6 fornece o mesmo padrão da aplicação autocontida, baseada em *firewall* do sistema operacional, o segundo modelo, representado na Figura 7, propõe uma integração com o *firewall* corporativo, integrando-se assim na estratégia global de segurança da organização, através da API de gerenciamento deste dispositivo. Esse layout reflete uma abordagem modular e escalável, permitindo que os módulos sejam atualizados ou substituídos independentemente, conforme necessário, para responder a ameaças emergentes e evoluir com as necessidades do negócio.

Figura 7. Proposta de Arquitetura Distribuída dos Módulos Entre Containers



Como dito, a modularidade deste modelo promove uma flexibilidade operacional significativa. Cada contêiner funciona como uma unidade autônoma, permitindo atualizações, manutenção e escalabilidade independentes. Isso significa que melhorias em algoritmos de classificação ou mecanismos de bloqueio podem ser implementadas em contêineres específicos sem a necessidade de paralisar toda a infraestrutura de segurança.

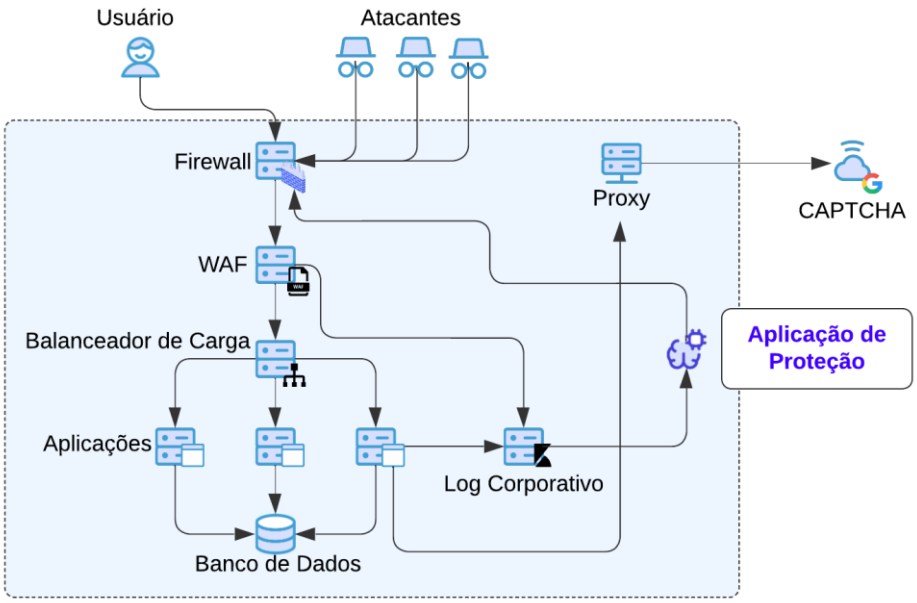
Aliado a este aspecto, a descentralização melhora a resiliência do sistema. A distribuição de carga entre múltiplos contêineres pode reduzir o risco de pontos únicos de falha, aumentando assim a robustez da arquitetura de segurança. Adicionalmente, a utilização de contêineres para tarefas específicas como captura, classificação e bloqueio permite a especialização funcional. Essa especialização possibilita uma eficiência operacional superior e uma resposta mais rápida a ameaças cibernéticas emergentes.

Por último, o modelo descentralizado favorece a inovação contínua. Como cada contêiner pode ser desenvolvido e gerenciado por equipes diferentes, há uma oportunidade para a adoção de novas tecnologias e abordagens de segurança assim que se tornam disponíveis. Isso estimula a colaboração interdisciplinar e o desenvolvimento ágil, alinhando a infraestrutura de segurança cibernética com as melhores práticas e pesquisas de ponta na área.

Em suma, a abordagem descentralizada reflete um paradigma estratégico que enfatiza a adaptabilidade, resiliência, especialização e inovação contínua, elementos essenciais para a segurança cibernética.

No contexto da pesquisa, foi adotada uma arquitetura híbrida entre a versão autocontida e a distribuída, considerando as restrições de tempo para implantação das versões mais refinadas. A sua estratégia de implantação utilizada adotou os três módulos em uma única aplicação. No entanto, o modelo de bloqueio utilizou a integração via API com o *firewall* corporativo. Resgatando o contexto da Arquitetura Padronizada (Figura 4), uma versão atualizada com a ferramenta de proteção ativa pode ser vista na Figura 8.

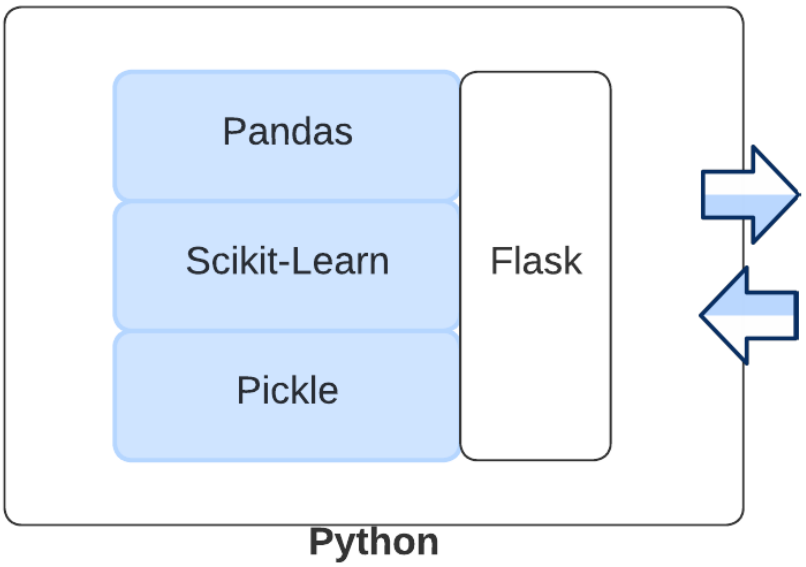
Figura 8. Arquitetura da Solução Adotada



4.3.3. Desenvolvimento da Solução Proposta

O funcionamento da solução se baseia em um filtro de requisições HTTP submetendo os metadados do *request* para a API, que procederá à classificação do risco. Em caso afirmativo para bloqueio, a solução interage com a configuração do *firewall* através de uma classe adaptadora, procedendo ao bloqueio daquele usuário por um tempo parametrizável. Em uma instância local, o bloqueio pode ser realizado utilizando o próprio *firewall* do sistema operacional, enquanto em uma rede corporativa a regra pode ser implantada através de um acesso à API do recurso. Como o objetivo principal da pesquisa é limitar o número de requisições automatizadas a partir de um mesmo usuário, a proposta é definir uma regra que aumente o tempo de bloqueio de forma exponencial: a partir do momento em que o identificador de usuário é reincidente, os períodos da regra de bloqueio tornam-se maiores. A Figura 9 representa a pilha tecnológica utilizada no desenvolvimento da solução:

Figura 9. Tecnologias Utilizadas no Desenvolvimento



Para o desenvolvimento das APIs, optou-se pela linguagem Python devido ao grande quantitativo de bibliotecas disponíveis para análise e tratamento dos dados. Foi utilizado o Flask versão 3.0.0, um *framework* extensivamente aplicado na construção de APIs *web* devido à sua simplicidade e flexibilidade. O Flask permite a criação de *endpoints* de API de maneira rápida e com poucas linhas de código, facilitando o desenvolvimento e a manutenção da solução.

No que diz respeito ao núcleo de aprendizagem de máquina, a escolha recaiu sobre o uso conjunto das bibliotecas Pandas 2.1.3 e Scikit-Learn 1.3.2. O Pandas proporciona estruturas de dados de alta performance e ferramentas de análise de dados fáceis de usar. Com ele, é possível manipular grandes conjuntos de dados, realizar a sanitização e a preparação do conjunto de dados de forma eficiente. Por sua vez, o Scikit-Learn é amplamente reconhecido por dispor de uma vasta gama de algoritmos de aprendizagem de máquina, desde regressão até algoritmos de classificação, o que representa uma aplicabilidade bastante significativa na pesquisa. Esta biblioteca oferece uma ampla gama de algoritmos de aprendizado supervisionado e não supervisionado, além de ferramentas para seleção de modelo, avaliação e pré-processamento de dados. Assim, foi possível desenvolver um modelo preciso e eficiente para as necessidades específicas da aplicação, testando diferentes algoritmos e ajustando parâmetros de forma iterativa.

Finalmente, uma vez concluído o desenvolvimento do modelo de aprendizagem de máquina, sua persistência torna-se necessária para que possa ser reutilizado sem a necessidade de retreino a cada execução. Aqui, o módulo Pickle foi utilizado para serialização do modelo treinado e, em seguida, gravação no disco. A serialização com Pickle é um método eficaz para salvar objetos Python complexos, permitindo que o estado exato do modelo seja salvo e recuperado

posteriormente. Isso facilita não apenas a distribuição e o uso do modelo em diferentes ambientes, mas também a realização de inferências de maneira rápida e eficiente.

Na solução desenvolvida, faz-se necessária a categorização de informações quanto à Lei Geral de Proteção de Dados. Assim como na etapa de Agrupamento, são revisados os dados utilizados na heurística de classificação quanto à sua sensibilidade, de modo a auxiliar a compreensão de como o mecanismo de captura dos *logs* poderá atuar na construção dos atributos. A Tabela 3 fornece uma visão dos campos de requisição utilizados na classificação do modelo, bem como os tipos de dados armazenados e a decisão por anonimizar ou não o dado para a ferramenta.

Tabela 3. Classificação e Tratamento dos Metadados Utilizados no Algoritmo

Dado	Tipo	Categorização	Anonimização
User ID	**Numérico (ou UUID)**	**Dado pessoal**	**Sim**
Timestamp	Numérico	Dado de sistema	Não
Source_IP	Numérico	Dado de sistema	Não
Source_Port	Numérico	Dado de sistema	Não
Destination_IP	Numérico	Dado de sistema	Não
Destination_Port	Numérico	Dado de sistema	Não
Server Group	**Categoria**	**Dado de sistema**	**Sim**
Data Center	**Categoria**	**Dado de sistema**	**Sim**
Action	Categoria	Dado de sistema	Não
UserAgent_Client	Categoria	Dado de sistema	Não
UserAgent_Platform	Categoria	Dado de sistema	Não
UserAgent_OS	Categoria	Dado de sistema	Não
Geo_Latitude	Numérico	Dado de sistema	Não
Geo_Longitude	Numérico	Dado de sistema	Não
Geo_Distance	Numérico	Dado de sistema	Não
CAPTCHA_Score	Numérico	Dado de sistema	Não
Severity	Categoria	Dado de sistema	Não

As justificativas para anonimização dos campos Server Group e Data Center são referentes à proteção de informações arquiteturais e de infraestrutura que possam ser sensíveis para a corporação. Como o dado em si não está sendo avaliado, e sim o fato de pertencer a um ou outro ambiente corporativo, esta categorização é suficiente para a análise do algoritmo.

Quanto ao identificador de usuário, a ferramenta adota por padrão o IP de origem da requisição, mas pode ser adaptada para extrair outra forma de identificação, como o nome de usuário ou UUID

para sistemas autenticados, ou o *fingerprint* do cliente para uma requisição não autenticada. A anonimização deste campo se dá pela necessidade de garantir a proteção das informações pessoais em caso de uso do nome de usuário, e para este fim é adotada a técnica de supressão do dado original aplicando-se o *hash* na geração do metadado. Desta forma, a solução é conforme com a legislação, ao utilizar o dado pessoal apenas para o fim informado nos termos de uso, e uma versão anonimizada do mesmo para a identificação de ameaças.

Por fim, os dados de geolocalização não foram anonimizados por se tratar de uma simplificação que não permite a identificação geográfica exata do usuário. A tabela de geolocalização utilizada no cálculo dos valores indica apenas uma aproximação baseada na localidade, e esse dado, isoladamente, não oferece risco de identificação. No entanto, em casos específicos, pode ser considerada a necessidade de aplicar um método de generalização sobre o campo, de modo a diminuir a área identificável pelos dados de latitude e longitude, e este elemento deve ser parametrizável na solução. Deve ser considerado, no entanto, o impacto desta mudança na reclassificação das informações já incorporadas ao modelo, bem como a potencial diminuição da efetividade da classificação oferecida.

5. EXPERIMENTOS E RESULTADOS

Com a arquitetura da solução estabelecida, a etapa subsequente consistiu na implementação prática do modelo de classificação, integrando-o efetivamente aos sistemas existentes de captura e bloqueio de requisições. Para avaliar o desempenho do modelo, decidiu-se por mensurá-lo em termos de precisão – a proporção de identificações corretas de requisições maliciosas em relação ao total de requisições classificadas como tal pelo modelo. Este indicador é crucial para avaliar a capacidade do modelo de minimizar os falsos positivos, ou seja, situações nas quais as requisições legítimas são incorretamente sinalizadas como ameaças, o que poderia levar a interrupções desnecessárias no serviço e consequente desgaste com o cliente. Além da precisão, faz-se necessário avaliar a métrica de recall para quantificar a proporção de requisições maliciosas efetivamente capturadas pelo modelo. Um alto valor de recall é indicativo de que poucas ameaças passam despercebidas pelo sistema, um aspecto vital para a segurança cibernética.

No entanto, um dos desafios de um bom modelo de aprendizagem consiste em balancear precisão e recall, de modo a otimizar ambos sem comprometer significativamente um ou outro. Para abordar essa dualidade, o F1-Score foi calculado como uma medida harmônica entre precisão e recall, oferecendo uma perspectiva unificada sobre o desempenho do modelo. Um alto F1-Score sugere um equilíbrio adequado entre capturar a maioria das ameaças (recall alto) e manter um baixo número de alertas falsos (alta precisão), um parâmetro essencial para sistemas de detecção de intrusão eficientes em ambientes corporativos.

5.1. Perfilamento e Agrupamento

A fim de selecionar o melhor método para o agrupamento dos dados, foram realizados experimentos comparando o método GRASP (alfa=0.2) proposto com a heurística *k-means* e com as meta-heurísticas PSO e AG propostas por Guerreiro, et al. (2019). Os métodos estudados neste trabalho foram desenvolvidos em linguagem C++, utilizando o compilador G++ 9.3.0-17 com opção de compilação -O3. Os experimentos computacionais foram executados em um computador com processador Intel Core i7-3612QM, com 8 núcleos de 2,1 GHz. Adicionalmente, a máquina utilizada dispõe de 8 GB de memória RAM e utiliza sistema operacional Ubuntu 20.04.

Para os experimentos, foram efetuadas 150 amostragens aleatórias do conjunto completo de dados, divididas em 15 grupos, cada um com 10 amostragens com o mesmo número de linhas e colunas, i.e., pontos e dimensões. Foram amostradas cada combinação de tamanhos de pontos P com valores entre 200, 400, 600, 800 e 1000, e dimensões d entre os valores 8, 10, e 12, e cada método teve um tempo limite de 10 segundos. As Tabelas 4 e 5 detalham os resultados obtidos, com destaque negrito para o melhor resultado em cada grupo. Cada coluna detalha um grupo de amostragens com dimensões iguais e cada linha representa um método estudado. A medida de

qualidade do agrupamento obtido foi dada pala média da soma das distâncias de cada ponto para o seu centroide, e o valor reportado é a média dos resultados obtidos em cada grupo.

Tabela 4. Comparativo Entre os Métodos de Agrupamento

	d=8				d=10				d=12			
N	**PSO**	**GRASP**	**GEN**	**KMO**	**PSO**	**GRASP**	**GEN**	**KMO**	**PSO**	**GRASP**	**GEN**	**KMO**
200	**83.83**	84.54	86.38	103.59	107.20	**105.96**	109.93	119.76	135.54	**133.79**	140.45	143.67
400	**182.00**	182.48	191.87	210.73	231.18	**229.53**	239.07	248.06	268.41	**266.42**	282.51	286.31
600	**301.11**	301.23	316.46	330.47	330.15	**325.13**	340.66	364.91	402.40	**399.37**	422.46	460.15
800	338.04	**338.00**	350.11	431.13	481.92	**477.79**	514.23	517.91	520.74	**518.98**	548.57	564.42
1000	445.48	**440.97**	464.73	521.30	558.63	**555.52**	583.87	626.07	679.45	**664.03**	717.90	743.74
Méd	270.09	**269.44**	281.91	319.44	341.81	**338.79**	357.55	375.34	401.31	**396.52**	422.38	439.66

Tabela 5. Comparativo Entre os Métodos de Agrupamento (GRASP x PSO)

	d=8				d=10				d=12			
N	**PSO**	**vit.**	**GRASP**	**vit.**	**PSO**	**vit.**	**GRASP**	**vit.**	**PSO**	**vit.**	**GRASP**	**vit.**
200	**83.83**	**8**	84.54	2	107.20	4	**105.96**	**6**	135.54	3	**133.79**	**7**
400	**182.00**	**7**	182.48	3	231.18	**5**	**229.53**	**5**	268.41	**6**	**266.42**	4
600	**301.11**	**5**	301.23	**5**	330.15	3	**325.13**	**7**	402.40	3	**399.37**	**7**
800	338.04	**5**	**338.00**	**5**	481.92	4	**477.79**	**6**	520.74	4	**518.98**	**6**
1000	445.48	**5**	**440.97**	**5**	558.63	4	**555.52**	**6**	679.45	3	**664.03**	**7**
Méd	270.09	**6.0**	**269.44**	4.0	341.81	4.0	**338.79**	**6.0**	401.31	3.8	**396.52**	**6.2**

Inicialmente, podemos observar que os resultados corroboram que o *k-means* não apresenta um bom desempenho e que o PSO desempenha muito bem para todos os grupos dados. Contudo, quando o número de pontos aumenta, o GRASP consegue superar levemente o PSO. Uma vez que o conjunto de dados que iremos avaliar possui grande número de pontos, decidimos utilizar o GRASP como método de agrupamento. Vale salientar que este resultado não pode ser generalizado, indicando, apenas, que o GRASP obteve melhores resultados especificamente para este conjunto de dados.

5.1.1. Execução do Perfilamento

Esta subseção descreve o procedimento realizado para o perfilamento dos dados. Diante dos resultados descritos anteriormente, optamos pela utilização do método baseado na meta-heurística GRASP, visto que este alcançou os melhores resultados para o maior conjunto de dados experimentado.

Os dados utilizados no experimento são reais e foram coletados a partir de uma ferramenta proprietária de WAF ao longo de 24 horas de operação, gerados e armazenados em repositório corporativo em formato JSON. A janela de tempo foi selecionada com base na utilização de um dia de operação, visando considerar entre as métricas de utilização a variação da natureza dos bloqueios

de acordo com o horário. Ao longo do período, foram identificados registros relacionados a 306.190 intervenções do WAF.

A ferramenta proprietária realiza a persistência dos dados informativos do bloqueio, relativos à heurística aplicada pela ferramenta, além de informações da requisição e da resposta da aplicação, como cabeçalhos e cookies utilizados. O extenso volume de dados (cerca de 1.4 GB, em formato JSON) necessita de tratamento para a extração das informações fundamentais para a identificação dos perfis de bloqueio.

Para cada registro, foram verificados um total de 21 características, listadas na Tabela 6. Este conjunto de informações é a referência para a análise de um classificador de tráfego padronizado, sem a classificação baseada nos resultados da meta-heurística de agrupamento.

Tabela 6. Lista das Características Extraídas Diretamente dos Registros de *Log*

Campo	Descrição
create-time	*Timestamp* de criação do registro
gateway-name	Nome da instância que registrou a ocorrência
datacenter	Local físico onde está instalada a instância que registrou a ocorrência
server-group-name	Nome do grupo de servidores onde está situada a aplicação
server-group-simulation-mode	Modo de atuação do WAF no grupo de servidores (ativo ou em simulação)
violation-type	Tipo de regra apontada pelo WAF como responsável pelo bloqueio
service-name	Nome do serviço (subgrupo) onde a regra foi aplicada
application-name	Nome da aplicação protegida
sourceip	IP de origem da requisição
clientip	IP de origem do usuário da requisição (pode diferir do anterior, em caso de uso de NAT)
sourceport	Porta de origem da requisição
protocol	Protocolo (TCP/UDP)
destinationip	IP de destino da requisição
destinationport	Porta de destino da requisição
violation-id	Identificador único da violação
violation-attributes	Atributos da violação, caso existam
policy-name	Nome da política de segurança aplicada na ocorrência
action	Ação tomada pelo WAF
alert.severity	Severidade do aponte

alert.description	Descrição do alerta no qual a aplicação se baseou
http	O conteúdo da requisição, com campos como identificador de sessão, cabeçalhos, método HTTP, user-agent e URL acessada, entre outros

Vale destacar que os dados acima representam a extração crua dos registros de *log* da navegação via WAF. Vários destes elementos podem ser explodidos em outras características, como em particular o campo HTTP, que contém informações de identificação de sessão, método HTTP utilizado e *user agent* da requisição. Muitos destes campos, embora possam ser manipulados pelo atacante, representam um indicativo geral da forma como a aplicação cliente se comporta, e podem ser agregados às demais informações obtidas para a construção de inferências fundamentais ao classificador.

Objetivando a extração adequada das informações relevantes, foi realizada uma etapa inicial de limpeza nos dados JSON obtidos. Devido a uma limitação do tamanho de campo para persistência da resposta do servidor, foram eliminados 14.259 registros incompletos, além de 859 apontes que, embora possuíssem tamanho compatível com o registro de auditoria, não apresentavam estrutura válida do JSON, resultando em 291.072 registros válidos para o escopo do trabalho.

Atendendo à necessidade descrita na LGPD sobre anonimização de dados de pesquisa, foi realizado o processo de descaracterização de informações das aplicações protegidas através de substituição simples, além da generalização dos dados de IP de origem com base em geolocalização. Para a geolocalização dos IPs obtidos pela ferramenta, foi utilizada a API gratuita FreeGeoIP[11], através da qual foi possível obter registros sobre os 76.172 IPs únicos identificados dentre os bloqueios. Também visando conformidade com a legislação vigente no que tange à proteção dos dados pessoais, foi limitado o uso dos dados internos às requisições e respostas, evitando expor campos relativos a consultas e operações baseadas em dados pessoais legítimos que possam ter, inadvertidamente, ativado alguma das políticas de bloqueio ativas no WAF.

Após a anonimização dos dados, foram definidas características numéricas para a submissão aos mecanismos de agrupamento. Alguns campos textuais escolhidos, por apresentarem dificuldades na categorização, impactaram as execuções iniciais das meta-heurísticas de perfilamento, e, por isso, foram desconsiderados entre as características relevantes avaliadas pelos mecanismos de classificação.

[11] https://freegeoip.app/

Para categorização dos dados, foi adotado o uso de ferramenta de Notebook [12], utilizando a linguagem Python e a biblioteca Pandas para análise de dados. O trabalho iterativo com as ferramentas permitiu que, a cada execução, os grupos formados fossem observados quanto ao grau de separação fornecida pelo resultado da aplicação da heurística e sua relevância para a análise de risco predominante nos grupos. A cada execução, foram avaliados os aspectos mais significativos e predominantes nos grupos baseando-se nos dados pré-selecionados, considerando-se como base essencial a média do critério Severidade, que serviu como rótulo para identificar quais requisições apresentavam maior risco. Após cada execução, um novo conjunto de dados foi submetido à heurística de agrupamento e seu resultado avaliado quanto à separação baseada na média de Severidade entre os membros de cada grupo.

Através do método adotado, foi possível definir as 10 categorias onde houve maior relevância na associação dos resultados, minimizando o número de anomalias: Comunicação via HTTPS, Ação de Bloqueio, Plataforma (desktop ou *mobile*), *Bot* explícito (Googlebot e outros), Plataforma Indefinida (integrações de API e indicativo de código de terceiros), Origem de IP (Nacional, Internacional ou Intranet), Distância Geográfica (com relação às coordenadas médias do Brasil), Porta de Origem, *Timestamp* e Severidade, sendo os quatro últimos campos normalizados para evitar distorções no uso dos algoritmos.

5.1.2. Agrupamento

Para calcular o número ideal de *clusters*, foi adotado o método do cotovelo. Após a execução da meta-heurística GRASP com *k* variando entre os valores 2 e 12 sobre os dados selecionados, foi possível identificar, através do método do cotovelo, a melhor eficácia do agrupamento quando *k=5.* O gráfico com o cotovelo e o valor do SSD dos pontos para cada valor de *k* pode ser observado na Figura 10.

Dentre os 5 clusters observados após a execução da meta-heurística escolhida (nomeados pelas letras A até E), foi possível identificar algumas características dominantes em cada um. Importante ressaltar que, dentre os bloqueios identificados, há a aplicação de muitas heurísticas customizadas pela equipe de operação do WAF que, identificando riscos de segurança presentes no uso das aplicações, desenvolve padrões de bloqueio que devem ser considerados dentro do contexto das aplicações protegidas pelo *appliance.*

O Cluster A se destaca pelo grande número de bloqueios a partir de requisições sem *User-Agent* definido, como pode ser observado na Figura 11.

[12] https://colab.google/

Figura 10. Obtenção do Número Ótimo de Clusters com Base no Método do Cotovelo

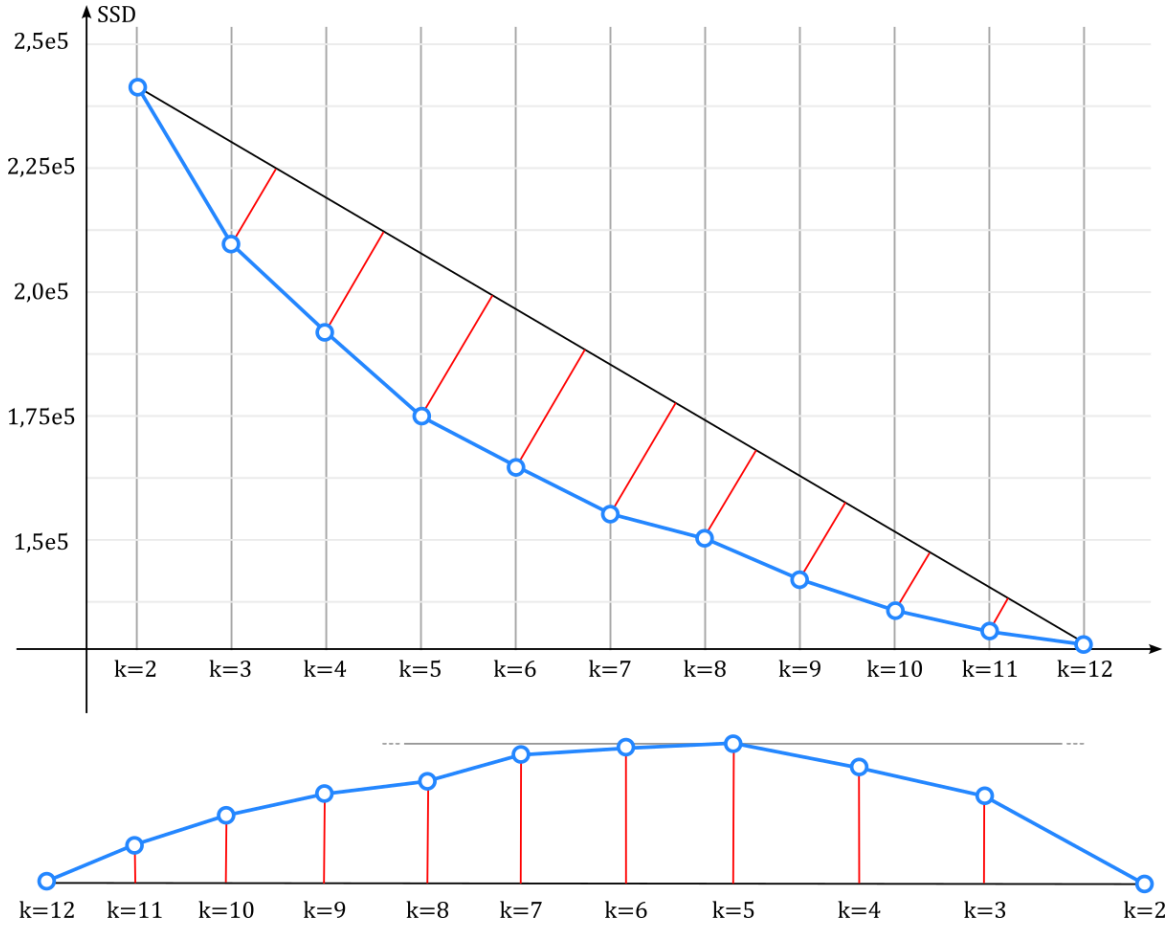
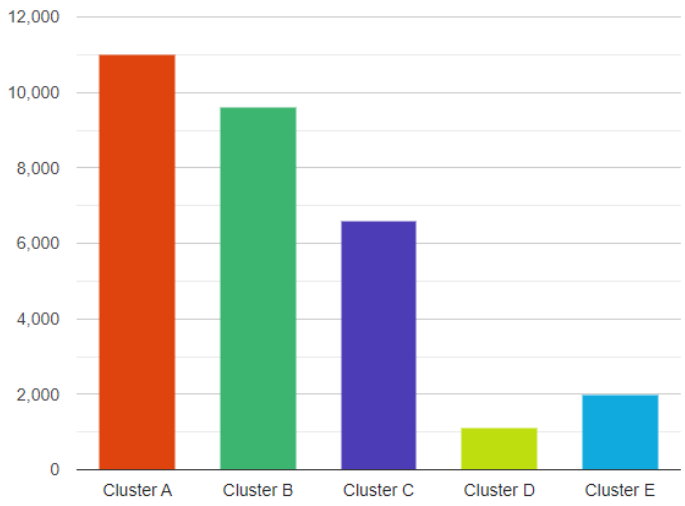


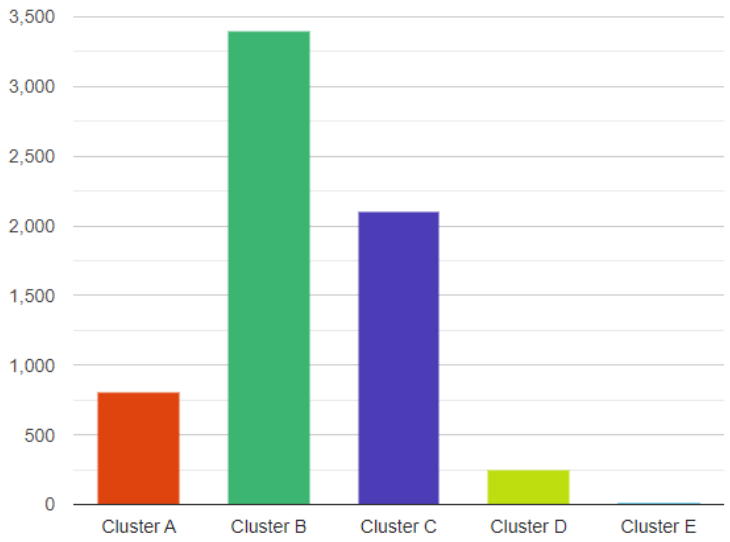
Figura 11. Separação em Clusters por Requisições sem User-Agent



Além disso, é o Cluster onde há o maior número de requisições oriundas de outros países, fato pouco comum em aplicações de viés governamental. Estas características, aliadas ao tipo de aplicação onde o bloqueio é mais frequente (*backends* de aplicações móveis de uso intensivo), permitem que possamos inferir alta frequência de acessos a partir de clientes alternativos, que não os *frontends* de aplicações móveis. A severidade aplicada pelas regras fica um pouco acima da média geral, assim como os bloqueios, posicionando o Cluster como intermediário do ponto de vista da criticidade.

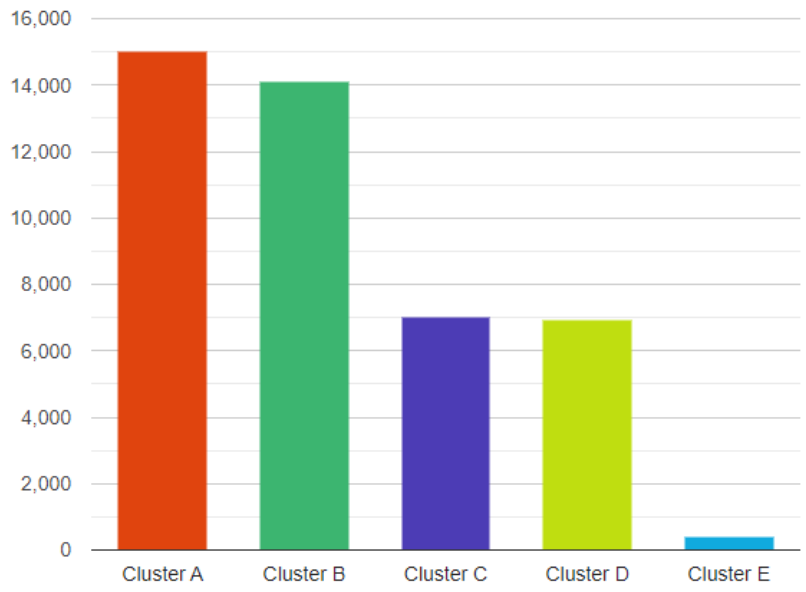
O Cluster B é caracterizado pela quantidade de registros (34,5% do total), e pela classificação média de severidade mais alta dentre todos os grupos, como pode ser observado na Figura 12.

Figura 12. Separação em Clusters por Severidade



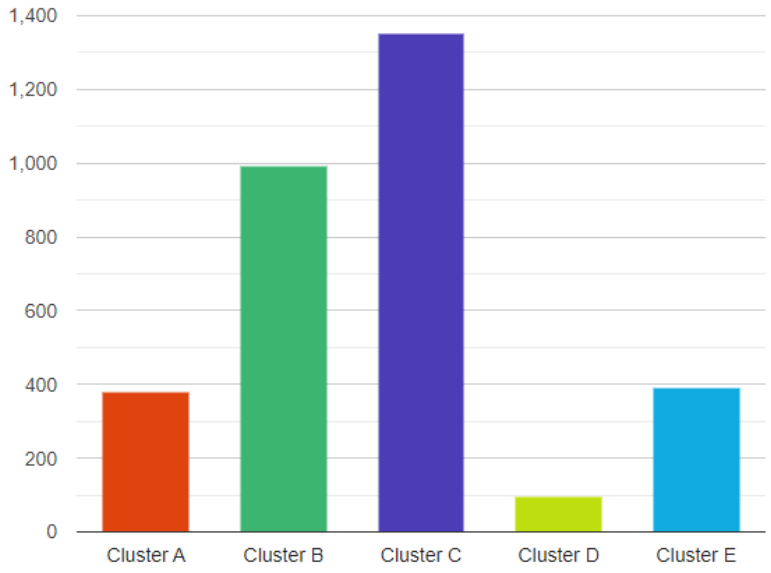
Entre as características predominantes deste conjunto, identificamos o grande número de apontes relacionados de listas de bloqueio de IP customizadas e serviços de identificação de *proxies* anônimos, ambos categorizados com criticidade elevada pela ferramenta, destacando a necessidade de adequação das políticas de bloqueio às fontes de dados sobre IPs de origem maliciosos. Além disso, se aproxima do Cluster A na quantidade de requisições originadas de IPs geolocalizados fora do país, conforme visto na Figura 13.

Figura 13. Separação em Clusters por Acessos a Partir do Exterior



O Cluster C apresenta grande parte das características bem próximas à média geral, com algum destaque na alta proporção de IPs de origem situados no Brasil. Dentre as verificações mais frequentes deste grupo, predominam as regras sobre aplicações abertas e sistemas de autenticação a partir de heurísticas customizadas, criadas a partir do monitoramento constante destes sistemas, como pode ser observado na Figura 14.

Figura 14. Separação em Clusters por Aplicações Sem Autenticação

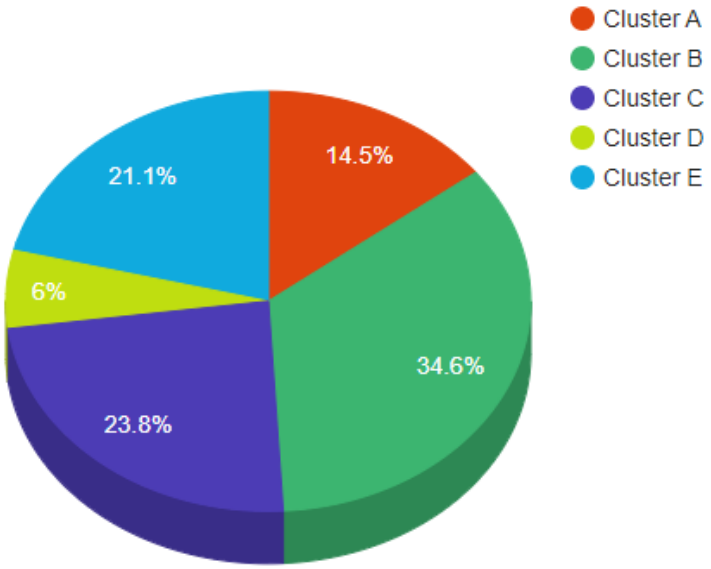


Em uma aplicação de alta relevância que não exige autenticação do usuário, o grupo representou o maior número de bloqueios (quase metade dos bloqueios para esta aplicação), mesmo

com o grupo correspondendo a cerca 20% do número total. O grupo também apresentou o maior desvio-padrão sobre os dados de *Timestamp* coletados, indicando grande variação entre os horários de bloqueio, ao mesmo tempo em que, para as demais características, apresentou resultados consistentes com a média geral (ou um pouco menores).

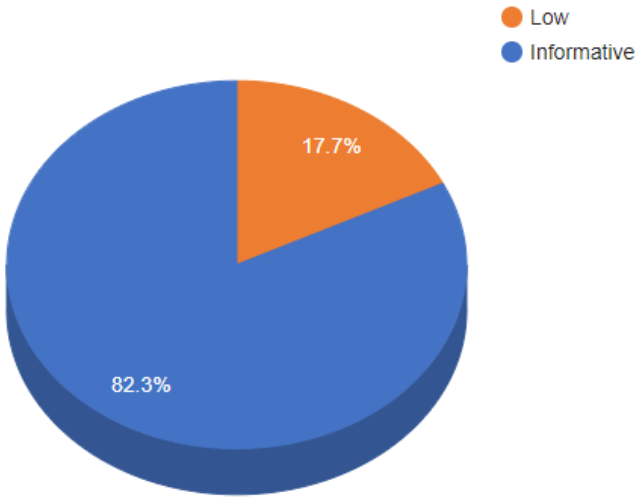
O Cluster D possui predominância de acessos de *bots* "legítimos", identificados como rastreadores de mecanismos de pesquisa através do *User-Agent*. É também o menor Cluster de todos, representando menos de 10% do total, como podemos observar na Figura 15, e o que apresenta maior proporção de acessos a partir de IPs identificados pelo serviço de geolocalização como oriundos de fora do Brasil. Como a maior parte dos acessos é claramente definida como automatizado, o grupo apresenta a maior média de bloqueio (acima de 77%).

Figura 15. Distribuição Percentual por Clusters



Por fim, o Cluster E apresentou os menores valores de severidade e bloqueio, como pode ser observado na Figura 16, indicando os apontes que apresentam o menor risco em geral (predominando as severidades *low* e *informative*). É também o grupo em que predominam em maior proporção os dados de *User-Agent* bem definidos, a maior proporção de IPs brasileiros (menos de 1% de IPs externos), e o grupo que mais utilizou o protocolo HTTPS em suas requisições. Menos de 1% dos apontes deste grupo resultou em bloqueio, sendo predominante a indicação de padrões suspeitos nas requisições, mas insuficientes para a tomada de decisão pela interrupção do acesso desta origem.

Figura 16. Distribuição de Severidade nos Apontes do Cluster E



Dentro das atividades de classificação, foi possível identificar características comuns aos ataques, considerando seus níveis de criticidade a partir da observação de ativos de proteção, e a partir deste conhecimento, direcionar os próximos passos da pesquisa quanto à implementação das heurísticas de contenção. A partir deste trabalho, a solução deverá considerar as informações assimiladas a partir desta base de conhecimento, auxiliando na construção do módulo de classificação, que irá avaliar os *requests* com base nas informações determinantes para a classificação, identificadas na etapa anterior.

Através desta implementação, a solução será capaz de considerar apenas as informações mais relevantes no tratamento das requisições, minimizando o tempo de processamento necessário e obtendo resultados mais confiáveis a partir do seu mecanismo de classificação. A implementação de heurísticas de agrupamento avançadas foi um passo fundamental para estruturar um conjunto de dados altamente relevante, utilizado como base para o modelo de aprendizado de máquina focado em classificação. Esse processo visa resultar em um treinamento mais eficaz do modelo e, por consequência, uma detecção mais precisa de requisições maliciosas.

5.2. Cenários Avaliados

Para os cenários avaliados, a proposta desenvolvida (denominada Algoritmo A) foi submetida a treinamento a partir de dados reais obtidos através do serviço de *log* centralizado, considerando-se os mesmos utilizados na etapa de agrupamento, divididos na proporção de 80% para a fase de treinamento e 20% para a fase de testes. A forma como esta distribuição é separada será levada em consideração: as simulações irão avaliar o comportamento dos algoritmos de classificação no caso da separação baseada em tempo, ou realizada aleatoriamente. Esta definição pretende avaliar a

eficácia dos mecanismos considerando a distribuição ao longo de um dia de operação, bem como a tendência da eficácia do método em um cenário menos previsível.

De modo a validar o resultado do módulo de classificação com base nos atributos selecionados na fase de agrupamento, foram implementados dois algoritmos de controle, B e C: O Algoritmo B é baseado apenas no IP de origem, de modo a simular o comportamento de uma heurística de lista de bloqueio (*block list*) de um *firewall* de camada de rede; o Algoritmo C, por sua vez mais abrangente, realizou avaliações considerando todo o escopo dos atributos capturados pelos registros de *log* do WAF e será a referência de controle. O objetivo é comparar a efetividade da seleção de parâmetros nos resultados da execução do classificador em relação aos dois modelos, considerando-se a simplicidade no desenvolvimento de ambos e avaliando o cenário vantajoso da seleção de parâmetros no índice de acerto sobre as classificações.

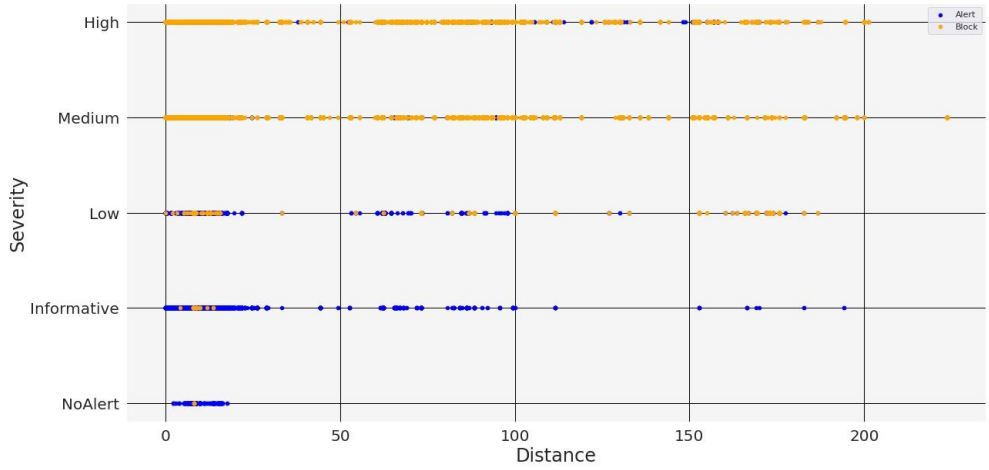
Os modelos também utilizaram mecanismos diferentes de classificação. A princípio, desejava-se adotar um modelo uniforme para os três algoritmos, e foi selecionado o *Random Forest*. No entanto, a utilização do TF-IDF (*Term Frequency – Inverse Document Frequency*) provou-se de baixa eficácia no cenário do Algoritmo C, que necessitava de um mecanismo para a classificação dos campos textuais. Neste caso, especificamente para o Algoritmo C, a implementação utilizou o *Multinomial Naïve-Bayes*, um método simplificado, porém substancialmente mais rápido e eficiente. Ambas as escolhas atenderam critérios de flexibilidade e simplicidade na implementação, relegando uma posterior seleção baseada em eficiência do algoritmo aos trabalhos futuros.

5.2.1. Geolocalização

A implementação do mecanismo de geolocalização na arquitetura do sistema representou um desafio significativo, especialmente devido à necessidade de um controle rigoroso e eficaz para determinar a origem geográfica dos IPs em cada requisição recebida. Durante a fase inicial de estudos, que focou no desenvolvimento de heurísticas de agrupamento, faz-se necessário ressaltar a importância e a clareza das informações de geolocalização na identificação de requisições maliciosas.

A análise dos dados obtidos a partir das heurísticas de agrupamento permitiu que se identificasse uma correlação entre a distância da origem das requisições e o risco associado a elas, conforme pode ser visto na Figura 17. Os pontos em azul representam alertas da ferramenta, enquanto os pontos laranjas, bloqueios executados com base em suas regras.

Figura 17. Correlação entre a Distância Geográfica e o Risco Associado à Requisição



Os dados obtidos podem representar com clareza a relevância da informação para agentes legítimos situados em solo brasileiro, considerando-se a aplicabilidade no escopo de sistemas governamentais, ao passo em que agentes maliciosos situados em regiões mais distantes do planeta apresentam comportamentos predominantemente maliciosos, uma vez que os sistemas governamentais brasileiros não contemplam, em sua maioria, recursos destinados a pessoas que habitam estas regiões.

No entanto, na fase de agrupamento, a implementação lidou com um conjunto de dados estático, onde os registros já possuíam as informações de geolocalização a partir do pré-processamento dos dados. Esta abordagem simplificava o processo, pois os dados necessários já estavam disponíveis e não requeriam processamento adicional em tempo real.

No entanto, na aplicação operacional da solução, a situação foi bastante diferente. As requisições de usuários chegavam sem informações prévias de geolocalização, o que impunha o desafio de identificar a origem geográfica de cada IP em tempo real. Para resolver este problema, era essencial integrar uma base de dados de geolocalização de maneira eficiente na aplicação, garantindo que as consultas a esta base fossem rápidas e precisas.

O processo iniciou com a aquisição de uma base de dados extensa e gratuita, disponível no site db-ip.com. Esta base de dados é composta por faixas de endereços IP, organizadas por região geográfica, e inclui informações como latitude e longitude médias de cada região. O arquivo, em formato CSV, possui um tamanho considerável de aproximadamente 481 MB. Devido a essa grande dimensão e à necessidade de desenvolver uma aplicação capaz de otimizar os recursos alocados, tornou-se impraticável carregar toda a base de dados na memória para processamento, o que exigiu uma abordagem mais eficiente. Ressalta-se que devido a limitações de tempo decorrentes da pesquisa, soluções mais sofisticadas como o uso de um banco de dados não-relacional foram descartadas.

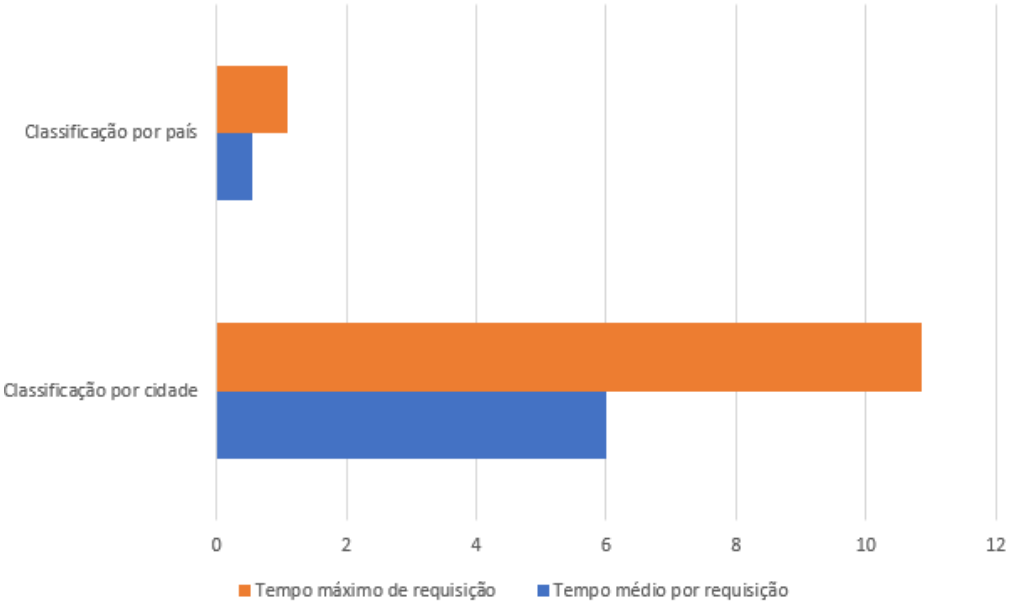
Para contornar essa limitação, desenvolvemos um algoritmo especializado capaz de realizar buscas rápidas e eficientes dentro dessa base de dados. O objetivo principal do algoritmo era determinar, de maneira ágil, a qual faixa de IP um determinado endereço de entrada pertence. Este passo foi crucial para associar cada requisição à sua localização geográfica correspondente.

Além disso, o algoritmo foi projetado para transformar os dados de geolocalização. Isso envolveu o cálculo da distância média entre a localização de origem da requisição (baseada nas coordenadas de latitude e longitude fornecidas) e as coordenadas médias da região geográfica correspondente no país. Este cálculo de distância foi um componente vital na fase de agrupamento dos dados, permitindo uma análise mais refinada e geograficamente contextualizada das requisições.

A capacidade de processar essa informação geográfica de maneira eficiente e precisa não só otimizou o uso de recursos computacionais, mas também enriqueceu significativamente a análise de dados, tornando-se fundamental para a detecção de atividades suspeitas ou anômalas nas requisições.

Mesmo com a supressão de alguns dados desnecessários para o desenvolvimento, ao se manter apenas as faixas de IP e a distância calculada para cada linha, o arquivo de consulta continuava com 257 MB. Nos testes iniciais, com uma estação de trabalho Intel Core i5 12600k e 32GB de RAM, as buscas sequenciais nesta base demoravam aproximadamente 5,99 segundos, em média, calculados após uma simulação de 1000 endereços IP gerados aleatoriamente. Optou-se, então, pelo uso de uma nova base, contendo apenas os dados de latitude e longitude médias de cada país, de modo a reduzir o universo de dados a serem pesquisados e, consequentemente, o tempo médio da requisição. Neste momento, a mesma simulação anterior apresentou um resultado significativamente melhor, de 0,559 segundo em média para cada busca, como apresentado na Figura 18.

Figura 18. Comparação entre Modelos de Dados



Com esses ajustes e a otimização dos dados, foi possível diminuir o tamanho do arquivo da nova base de 27 MB para apenas 10 MB. Essa redução tornou viável o carregamento total dos dados em memória, proporcionando uma consulta ainda mais rápida e mantendo a coesão do modelo arquitetural. Esse resultado foi essencial para evitar a necessidade de segmentar a funcionalidade de geolocalização em um serviço adicional, o que poderia ser requerido se o desempenho continuasse abaixo do esperado. As simulações desenvolvidas envolvem comparação entre os resultados de utilização da base de geolocalização em memória e em arquivo nos dois formatos, e a comparação não representou queda considerável da eficácia, justificando a opção pela base em memória.

5.3. Resultados Obtidos

O Algoritmo de Controle B, baseado apenas no IP de origem da requisição, foi, dentro do esperado, aquele que apresentou os resultados menos satisfatórios em sua capacidade preditiva. O quadro refletido na Tabela 7 representa seu grau de precisão, recall e F1-Score com relação aos rótulos baseados no campo Severidade (Alto, Médio, Baixo, Informativo e Sem Alerta) na massa de dados temporal. Os melhores resultados de cada comparativo entre os três algoritmos estão marcados com o asterisco.

Tabela 7. Resultados Obtidos Através da Classificação Temporal

Rótulo	**Precisão**			**Recall**			**F1-Score**			**Suporte**
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	
Alto	***0.88***	*0.73*	*0.77*	*0.85*	*0.60*	***0.86***	***0.86***	*0.66*	*0.81*	*23278*
Médio	**0.75**	0.48	0.57	0.63	0.55	**0.65**	**0.68**	0.51	0.61	10646

Baixo	**0.79**	0.66	0.65	**0.75**	0.71	0.74	**0.77**	0.68	0.69	2579
Informativo	0.80	0.61	**0.87**	**0.89**	0.67	0.69	**0.84**	0.64	0.77	21590
Sem Alerta	0.52	**0.78**	0.29	0.49	**0.70**	0.32	0.51	**0.74**	0.30	114
Média Total	**0.75**	0.50	0.63	**0.72**	0.51	0.65	**0.73**	0.50	0.64	58207
Média Ponderada	**0.82**	0.64	0.76	**0.82**	0.62	0.75	**0.82**	0.63	0.75	58207

Optou-se também por uma simulação baseada em uma distribuição aleatória dos dados entre as massas de treinamento e teste. Tal opção visa compreender com maior precisão o comportamento das soluções quando submetidas a um período maior de avaliação, que reflita o treinamento baseado em um conjunto de dados mais diversos e sem vinculação direta com o recorte temporal. Por este motivo, não foi levada em consideração a implementação B, já que a efetividade de seu mecanismo análogo à *block list* é correlacionada à prévia identificação de risco a partir de um IP, lógica pela qual a distribuição aleatória não representa um cenário factível. A escolha aleatória dos dados se deu a partir da mesma semente, de modo a reduzir inconsistências entre os modelos produzidos, e seus resultados são visualizados na Tabela 8. Pode-se observar nesta tabela a predominância do Algoritmo A nos resultados, enquanto o Algoritmo C desempenha em valores levemente superiores aos da série temporal.

Tabela 8. Resultados com a Classificação Aleatória dos Dados de Treinamento

	Precisão		**Recall**		**F1-Score**		**Suporte**
	A	C	**A**	C	**A**	C	
Alto	**0.97**	0.83	**0.96**	0.79	**0.97**	0.81	23419
Médio	**0.95**	0.78	**0.88**	0.66	**0.92**	0.72	12499
Baixo	**0.95**	0.62	**0.91**	0.72	**0.93**	0.67	3374
Informativo	**0.91**	0.74	**0.97**	0.84	**0.94**	0.79	18813
Sem Alerta	**0.91**	0.53	**0.98**	0.72	**0.94**	0.61	102
Média Total	**0.94**	0.70	**0.94**	0.75	**0.94**	0.72	58207
Média Ponderada	**0.95**	0.78	**0.95**	0.78	**0.94**	0.78	58207

5.4. Análise dos Resultados

Os resultados indicam o melhor desempenho do Algoritmo A, apontando para a relevância das categorias destacadas durante a fase de agrupamento, em comparação aos dois outros algoritmos de controle. Ao apresentar uma acurácia de 82% no primeiro caso de teste, o Algoritmo A se destaca frente aos demais considerando-se efetividade dos bloqueios realizados por ele. Este fato torna-se

ainda mais evidente quando se percebe a efetividade elevada dos apontes classificados como risco Alto, onde sua precisão sobe para 88%, com *recall* de 85%, o que indica também uma baixa taxa de falsos positivos.

Tal confiança no resultado é fundamental no contexto das análises de segurança em dois aspectos: ao considerar o falso negativo, um atacante poderá explorar a inefetividade do bloqueio para realizar um ataque bem-sucedido; por outro lado, um falso positivo representa o bloqueio de um usuário legítimo, com impacto ao correto funcionamento da aplicação. Este número, caso seja elevado, poderá incorrer em indisponibilidade reportada pelo usuário final, com impacto no Acordo de Nível de Serviço ofertado, além daquele associado à imagem do fornecedor do serviço. Por este motivo, os números de *recall* e F1-Score apresentados pelo Algoritmo A são promissores.

Os algoritmos de controle desempenham de forma razoavelmente inferior. O Algoritmo B detém os piores resultados, com uma precisão média de 64% e um *recall* ainda mais baixo, de 62%, que chega a 60% nas requisições de risco Alto, ou seja, indicando uma capacidade inferior de identificar verdadeiras ameaças. No entanto, trata-se de um modelo de mais simples implementação, e prevalece nas requisições Sem Alerta (que na massa de dados apresentaram o menor número de requisições, como visto na coluna Suporte). Tal informação pode representar um certo grau de aplicabilidade em contextos de baixo volume de dados para treinamento.

Por fim, o Algoritmo C desempenha uma precisão geral de 76%, próximo dos 82% do Algoritmo A, que representa moderado grau de sucesso na classificação das requisições. Ressalta-se que esta implementação foi baseada na classificação oriunda dos campos textuais da requisição, apresentando uma menor dependência do trabalho de agrupamento, e pode ser aplicável quando não houver o trabalho de seleção de características associado. No entanto, em um contexto de segurança, o fato de desempenhar com precisão de 11% abaixo do Algoritmo A em requisições de nível alto, embora com índice de *recall* ligeiramente superior, indica um algoritmo com maior probabilidade de incorrer em falsos negativos, e consequentemente uma menor resiliência a ataques.

A comparação entre a eficácia dos Algoritmos A e C apresenta uma disparidade ainda maior quando os dados de treinamento e teste são selecionados aleatoriamente, representando uma massa de dados menos previsível. Enquanto o Algoritmo C tende a apresentar uma precisão muito próxima àquela apresentada pelo conjunto de dados baseado no recorte temporal, o Algoritmo A melhora substancialmente seus índices de precisão, *recall* e F1-Score, apresentando um número maior de acertos em todas as categorias e uma precisão geral de 95%. Tal índice sinaliza a eficácia do método no longo prazo, quando treinado com requisições ainda mais diversas.

Pode-se perceber, portanto, que a efetividade da meta-heurística na seleção de características do classificador do Algoritmo A é capaz de aliviar a carga de processamento do WAF. Ao descentralizar a aplicação de políticas de bloqueio, antecipando-as à aplicação das regras do WAF, o *appliance* pode usar estes recursos extras para ser utilizado em uma quantidade maior de

aplicações, ampliando a superfície de captura de ataques, ao mesmo tempo em que gera insumos para que o Algoritmo A seja adaptável a novos cenários de proteção. A capacidade agregada pela implementação do classificador torna os cenários mais resilientes, ao passo em que pode ser aplicado de maneira complementar ao WAF no tratamento das tentativas de exploração de aplicações *web.*

Da mesma forma que a aplicação de regras de bloqueio representa uma redução de custo de processamento para o WAF, este impacto pode ser sentido no próprio dimensionamento da aplicação *web*, bem como de seus recursos de proteção próprios. Ao reduzir a demanda, antecipando a aplicação dos bloqueios baseados no *firewall*, serviços como CAPTCHA são também menos acionados, representando uma economia considerável.

Cabe, para fins de comparação, a análise dos custos associados a uma requisição de uma aplicação *web* ao serviço de CAPTCHA. Com base nos preços praticados pelo Google durante a elaboração deste trabalho, um *assessment* à API do ReCAPTCHA Enterprise custa em torno de 0,5 centavos de dólar. Em uma aplicação que trata 50 milhões de acessos mensais, cenário conservador para funcionalidades que atendem a um país de 200 milhões de habitantes, tal funcionalidade representa um custo de 250 mil dólares. Caso seja aplicado o Algoritmo A para contenção dos acessos de risco Alto, seus 88% de eficácia, em uma distribuição de 10% de acessos maliciosos, reduziria a dependência do CAPTCHA em 4 milhões e 400 mil acessos, representando uma economia de 22.000 dólares, aproximadamente 100.000 reais, apenas para uma aplicação.

Tal cenário tende a multiplicar sua efetividade no longo prazo, através da instanciação da solução em múltiplos contextos, somando-se às soluções disponíveis no contexto de proteção a aplicações *web* de forma a reduzir a dependência e o custo associados à infraestrutura de proteção. Considerando-se a possibilidade de segregar os módulos da aplicação e escalá-la individualmente, este modelo pode ser aplicado para otimizar o consumo dos recursos de proteção, permitindo a oferta dos serviços à população de maneira eficaz e consciente.

Outro aspecto identificado durante os testes diz respeito à especificidade de cenários fornecida pelo uso em um contexto governamental. A utilização do Google reCAPTCHA em ambientes com conectividade restrita, como redes governamentais, pode ser comprometida por especificidades da rede, impactando sua efetividade com relação a ataques internos.

Um sistema de proteção para um cenário totalmente online usa características como IPs de origem diferentes e *user agent*. Já em ambientes corporativos, muitas vezes há a restrição de IPs disponíveis para conexões externas, obrigando o uso de tecnologias como NAT para mapear mais unidades. Esta característica, por si só, resulta em um grande desafio para treinamento da ferramenta do Google, que pode identificar erroneamente os acessos do mesmo IP como uma tentativa de força bruta sobre um site. Além disso, uma única estação comprometida pode gerar baixa reputação para o serviço *online* do Google, impedindo que toda uma rede acesse o serviço de verificação. A

possibilidade de treinar um serviço especificamente para este cenário é uma das vantagens da solução proposta, que pode ser aplicada a estes cenários para diminuição das ocorrências de falsos positivos, resultando em uma experiência de usuário mais amigável.

Outro cenário específico que deve ser considerado diz respeito à indisponibilidade de rede em agências onde o acesso à Internet é restrito por questões geográficas, ou por apresentarem alta criticidade no tratamento de informações sensíveis.

Ambientes com grandes restrições de acesso a conteúdo externo podem ser impedidas de acessar qualquer serviço *web*, visando conter o vazamento de informações sensíveis. Este tipo de ambiente, comum em centrais de suporte básico a usuários, apresenta mais um desafio no acesso a sistemas de verificação na nuvem. A possibilidade de implantar um sistema local, com regras específicas para treinamento, implica em um caso de uso onde é possível proteger este acesso a informações sensíveis, visando conter possíveis extrações de grandes volumes de dados, sem que haja a dependência da conexão externa a um serviço com este fim.

Da mesma forma, outros serviços governamentais são fornecidos em condições precárias de acesso à Internet. Devido a este cenário, alguns sistemas são desenvolvidos com módulos 100% *offline*, para que seja possível fornecer um serviço localizado através da implantação em um ambiente local, como um servidor que atende a estações de trabalho de uma agência específica. Ao utilizar um mecanismo de proteção dos acessos dentro deste servidor, é possível impedir que o grande volume de requisições maliciosas torne instável, ou até mesmo indisponível, este servidor centralizado, resultando na operação adequada da agência.

6. CONSIDERAÇÕES FINAIS

Esta pesquisa consistiu na elaboração de uma proposta de arquitetura de um classificador baseado em Aprendizagem de Máquina que servisse como complemento a um conjunto de proteções aplicada em um contexto de aplicações *web*. Como base para a elaboração desta solução, o trabalho aplicou meta-heurísticas de agrupamento sobre os registros de *log* dos recursos de proteção atuais, gerando modelos que permitiram a visualização da relevância das características na diferenciação dos registros.

6.1. Conclusões

Existe uma grande diversidade de ameaças às quais uma aplicação *web* está frequentemente exposta. Cenários de ataque, bem como automações indevidas através de mecanismos de raspagem de dados são ocorrências frequentes, especialmente em contextos que envolvem informações de alto valor. Encontrar o equilíbrio ideal entre a implementação de uma proteção eficaz e a disponibilidade das informações respeitando o propósito dos sistemas que as acessam é o grande desafio de uma equipe de segurança neste contexto, considerando que a reação a novos métodos de ataque não pode depender da implementação manual de novos controles. Ou seja, há um cenário favorável ao uso de tecnologias de aprendizagem não-supervisionada.

A revisão sistemática sobre os trabalhos acadêmicos envolvendo os tópicos de Segurança da Informação e Aprendizagem de Máquina identificou uma lacuna no uso dos recursos de Inteligência Artificial para identificação de ameaças a partir dos registros de ativos de segurança. Tal aplicabilidade é fundamental para a correta utilização destes recursos, uma vez que realizam um trabalho de alta especificidade e, consequentemente, representam um custo de operação elevado. Além deste fator, outro diferencial identificado a partir da revisão consistiu na possibilidade do uso de dados reais para a classificação, uma vez que boa parte dos artigos representa soluções funcionais sob tráfego simulado, mas não testadas em um contexto real, com suas nuances e imprevisibilidades.

A partir deste estudo inicial, optou-se pela avaliação da eficácia de meta-heurísticas de agrupamento para compreender as características relevantes em um processo de classificação das informações obtidas durante o uso dos ativos de proteção. Este trabalho originou um artigo, publicado em 2021 no Simpósio Brasileiro de Pesquisa Operacional, sob o título “Meta-heurísticas Aplicadas ao Perfilamento de Acesso Indevido a Sistemas Operacionais”.

Após a identificação das características de destaque, partiu-se à implementação do algoritmo de classificação baseado nestes campos. Para efeito de comparação, foram desenvolvidas duas outras soluções de comparação, uma representando uma solução simplificada, baseada apenas no IP de origem (em um contexto similar ao de uma lista de bloqueios), e outra mais extensa, que utiliza

todos os campos do *log*, inclusive os textuais, e itera sobre eles a partir de algoritmos que avaliam a relevância de seus termos em um conjunto de palavras.

Quando comparado aos algoritmos de controle, o algoritmo baseado nas características oriundas do agrupamento apresentou resultados convincentes. Com uma precisão geral de 82%, elevada a 88% sobre os apontes de risco Alto e um número relativamente baixo de falsos positivos e negativos, seu funcionamento em uma série temporal representaria uma economia significativa de recursos financeiros.

Como estudo de caso, uma vez que a solução seja adotada em uma aplicação *web* com 50 milhões de acessos mensais, o custo seria reduzido em aproximadamente 100 mil reais/mês considerando o consumo do serviço do ReCAPTCHA associado a este cenário[13]. Este impacto, em comparação às demais implementações pesquisadas, representa uma economia de pelo menos 15 mil reais a mais do que o segundo algoritmo mais eficaz, constatando o valor da aplicação das meta-heurísticas na seleção de informações relevantes à análise.

Portanto, considerando os objetivos iniciais do trabalho, pôde-se observar o valor da aplicação da Aprendizagem de Máquina na otimização dos recursos computacionais e financeiros da operação de ativos de segurança em um contexto de aplicações *web* governamentais, com redução de custos e aumento da efetividade da proteção através da utilização do Algoritmo resultante da pesquisa.

6.2. Propostas de Trabalhos Futuros

Na expansão do uso da solução, pretende-se adotar uma estratégia de descentralização dentre as propostas arquiteturais mencionadas ao longo do trabalho. Um modelo potencialmente ideal para esta expansão deve considerar, além da descentralização em módulos, também a descentralização dos dados de treinamento. Esta abordagem permite a aprendizagem federada, compartilhada entre vários contextos diferentes, e sua eficácia poderá ser avaliada em outros trabalhos científicos. A escalabilidade do modelo de Aprendizagem de Máquina depende da possibilidade de aumentar a capacidade de treinamento do modelo central, enquanto o módulo responsável pela aplicação do modelo pode ser implantado em instâncias, dispondo de um *bucket* de objetos S3 para distribuição do arquivo PKL resultante do treinamento.

A evolução do modelo também pretende envolver o aprendizado a partir de diversas fontes de informação, como os próprios registros de acesso de ferramentas de CAPTCHA, para enriquecimento das características adotadas na classificação de risco realizada pela ferramenta.

---

[13] https://cloud.google.com/recaptcha-enterprise/docs/compare-tiers?hl=pt-br. Acessado em maio de 2024.

Outro aspecto a ser avaliado futuramente diz respeito à metodologia dos tempos de bloqueio. A definição inicial consiste em um bloqueio por bloco de tempo fixo, no entanto pode ser avaliada a eficácia da criação de uma regra que aplique bloqueios em um aumento exponencial do tempo. O ônus desta proposta sobre bloqueios muito longos deve ser considerado quanto à persistência dos agendamentos e a possível ocorrência de falsos positivos (bloqueio de um IP posteriormente atribuído a outro usuário, por exemplo). O código da solução atual já dispõe de um mecanismo adaptativo baseado em identificador de usuário, que pode ser aplicado em sistemas que exigem um usuário autenticado, mitigando o impacto de um potencial falso positivo. Para o ajuste adequado do tempo de bloqueio, pode ser avaliada a otimização baseada em outros mecanismos.

Por fim, a implementação baseada no modelo de arquitetura distribuída pode render novas comparações quando avaliada a efetividade do Algoritmo em cenários diversos. APIs, por exemplo, podem representar um desafio específico, que exigiria a construção de uma base de conhecimento e treinamento específica para estes cenários. A identificação de novas aplicações para o uso da Aprendizagem de Máquina na alocação adequada de recursos em contextos arquiteturais diversos deve ser fruto de um novo trabalho acadêmico, comparando a efetividade da solução em um contexto diferente do aqui apresentado.

REFERÊNCIAS BIBLIOGRÁFICAS

ALAHMADI, B.; MARICONTI, E.; SPOLAOR, R.; STRINGHINI, G.; MARTINOVIC, I. BOTection: Bot Detection by building Markov Chain Models of Bots Network Behavior. **ASIA CCS'20: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security**, pages 652-664, 2020. Citado na página 36.

ATEFINIA, R.; AHMADI, M. Network intrusion detection using multi-architectural modular deep neural network. **The Journal of Supercomputing**, 77, 2021. Citado na página 35.

AZAD, B. A.; STAROV, O.; LAPERDRIX, P.; NIKIFORAKIS, N. Web runner 2049: Evaluating third- party anti-bot services. **In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment**, 2020. Citado na página 19.

BOCK, H.-H. Clustering methods: a history of k-means algorithms. **Selected contributions in data analysis and classification, 2007**. Citado nas páginas 19 e 29.

EASTTOM, C. A Methodological Approach to Weaponizing Machine Learning. **AIAM 2019: Proceedings of the International Conference on Artificial Intelligence and Advanced Manufacturing**, pages 1-5, 2019. Citado na página 38.

GELLI, J. G. M.; BAIAO, F.; Clustering match-level statistics to determine playing styles in soccer**. LII Simpósio Brasileiro de Pesquisa Operacional**, 2020. Citado na página 37.

GOßEN, D.; JONKER, I. H.; e POLL, I. E. Design and implementation of a stealthy openwpm web scraper**. Bachelor thesis in Computing Science**, 2020. Citado na página 37.

GUO, W.; TONDI, B.; BARNI, M. Universal Detection of Backdoor Attacks via Density-Based Clustering and Centroids Analysis. IEEE Transactions on Information Forensics and Security, 2024. Citado na página 37.

GUERREIRO, M.; CASTANHO, D. S.; MARTINS, M. S. R.; CORREA, F.; TROJAN, F.; SIQUEIRA, H. Meta-heurísticas bio-inspiradas de clusterização para escolha de componentes na indústria automobilística**. LI Simpósio Brasileiro de Pesquisa Operacional**, 2019. Citado na página 57.

GUERREIRO, M. T. Análise de métodos de agrupamento de dados para detecção de anomalias na precificação e categorização de peças da indústria automotiva. **Tese de Mestrado**, Universidade Tecnológica Federal do Paraná, 2021. Citado nas páginas 37 e 57.

HARIFI, S.; KHALILIAN, M.; MOHAMMADZADEH, J.; EBRAHIMNEJAD, S. Using metaheuristic algorithms to improve k-means clustering: A comparative study. **Revue d'Intelligence Artificielle**, 34, 2020. Citado na página 37.

HAZAN, I.; MARGALIT, O.; ROKACH, L. Keystroke dynamics obfuscation using key grouping. **Expert Systems With Applications** 143, Elsevier, 2020. Citado na página 36.

HAZAN, I.; MARGALIT, O.; ROKACH, L. Supporting unknown number of users in keystroke dynamics models. **Knowledge-Based Systems** 221, Elsevier, 2021. Citado na página 36.

IKOTUN, A.; EZUGWU, A.; ABUALIGAH, L.; ABUHAIJA, B.; HEMING, J.; K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. **Information Sciences** 622, Elsevier, 2023. Citado na página 19.

JIN, D.; LU, Y.; QIN, J.; CHENG, Z.; MAO, Z. SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. **Computers & Security** 97, Elsevier. 2020. Citado na página 34.

LIU, J.; GAO, Y.; HU, F. A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. **Computers and Security** 106, Elsevier, 2021. Citado na página 34.

LIU, L.; CHEN, C.; ZHANG, J.; VEL, O.; XIANG, Y. Insider Threat Identification Using the Simultaneous Neural Learning of Multi-Source Logs. **IEEE Access**, 2019. Citado na página 38.

LIU, W. Computer Network Confidential Information Security Based on Big Data Clustering Algorithm. **Wireless Communications and Mobile Computing**, 2022. Citado na página 38.

MACULAN, N.; NEGREIROS, M.; PINTO, R. V. Two optimization models for clustering problems**. LII Simpósio Brasileiro de Pesquisa Operacional**, 2020. Citado na página 37.

MANJUSHREE, B.; SHARVANI, G. Survey on web scraping technology. **Wutan Huatan Jisuan Jishu**, 2020. Citado na página 19.

MCGAHAGAN IV, J.; BHANSALI, D.; PINTO-COELHO, C.; CUKIER, M.: Discovering features for detecting malicious websites: An empirical study. **Computers & Security** 109, Elsevier, 2021. Citado na página 34.

PAWLICKI, M.; CHORAS, M.; KOZIK, R.; HOLUBOWICK, W. On the Impact of Network Data Balancing in Cybersecurity Applications. **ICCS – International Conference on Computational Science**, 2020, pages 196-210. Citado na página 34.

PRASAD, M.; TRIPATHI, S.; DAHAL, K. Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection. **Computers & Security** 99, Elsevier, 2020. Citado na página 34.

QUEZADA, A. I.; OLIVEIRA, W. A. A k-means clustering approach based in mathematical programming: a bovine animal application. **LII Simpósio Brasileiro de Pesquisa Operacional**, 2020. Citado na página 37.

RAHAL, B.; SANTOS, A.; NOGUEIRA, M. A Distributed Architecture for DDoS Prediction and Bot Detection. **IEEE Access**, Volume 8, 2020. Citado na página 36.

REZAEI, T.; MANAVI, F.; HAMZEH, A.. A pe header-based method for malware detection using clustering and deep embedding techniques. **Journal of Information Security and Applications**, 2021. Citado na página 37.

ROBERTSON, S. P.; VATRAPU, R. K. Digital government. **Annual review of information science and technology**, 2010. Citado na página 16.

SAMTANI, S.; KANTARCIOGLU, M.; CHEN, H. Trailblazing the Artificial Intelligence for Cybersecurity Discipline: A Multi-Disciplinary Research Roadmap. **ACM Transactions on Management Information Systems**, Volume 11, Issue 4, 2020. Citado na página 38.

SAMTANI, S; LI, W.; BENJAMIN, V.; CHEN, H. Informing Cyber Threat Intelligence through Dark Web Situational Awareness: The AZSecure Hacker Assets Portal. **Digital Threats: Research and Practice**, Volume 2, Issue 4, 2021. Citado na página 38.

SHAMS, E.; RIZANER, A.; ULUSOY, A. A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems**. Neural Computing and Applications 33, 2021**. Citado na página 35.

STALLINGS, W.; BRESSAN, G.; BARBOSA, A. Criptografia e segurança de redes. **Pearson Education, 2008**. Citado na página 23.

WU, T.; MA, W.; WEN, S.; XIA, X.; PARIS, C.; NEPAL, S.; XIANG, Y. Analysis of Trending Topics and Text-based Channels of Information Delivery in Cybersecurity. **ACM Transactions on Internet Technology**, Volume 22, Issue 2, 2022. Citado na página 39.

YOUSEFNEZHAD, M.; HAMIDZADEH, J.; ALIANNEJADI, M. Ensemble classification for intrusion detection via feature extraction based on deep Learning. **Soft Computing** 25, 2021. Citado na página 36.

YUAN, S.; WU, X. Deep learning for insider threat detection: Review, challenges and opportunities. **Computers & Security** 104, Elsevier, 2021. Citado na página 35.

ZHAO, B. Web scraping. **Encyclopedia of big data**. 2017. Citado na página 16.

ZOPPI, T.; CECARELLI, A.; CAPECCHI, T.; BONDAVALLI, A. Unsupervised Anomaly Detectors to Detect Intrusions in the Current Threat Landscape. **ACM/IMS Transactions in Data Science**, Volume 2, Issue 2, 2021. Citado na página 35.