ANÁLISE DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ATAQUES WEB EM AMBIENTE DE WEB APPLICATION FIREWALL (WAF)

Universidade Presbiteriana Mackenzie
Pró-Reitoria de Pesquisa e Pós-Graduação - Coordenadoria de Fomento à Pesquisa

Leonardo Rodrigues Hofjud[1]
Apoio: PIVIC Mackenzie

Charles Boulhosa Rodamilans[2]

[1] Faculdade de Computação e Informática (FCI)
[2] Faculdade de Computação e Informática (FCI)

Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

<leonardo.hofjud@mackenzista.com.br>[1]
<charles.rodamilans@mackenzie.br>[2]

2025

Resumo

O aumento contínuo das ameaças cibernéticas tem intensificado a complexidade dos processos de detecção de ataques direcionados a aplicações web, tornando imprescindível a utilização de mecanismos de defesa mais robustos e eficazes. Ferramentas tradicionais de segurança, embora amplamente empregadas, apresentam limitações significativas, sobretudo relacionadas à elevada incidência de falsos positivos, o que compromete a identificação precisa de agentes maliciosos e, consequentemente, a eficácia das ações de resposta a incidentes. Nesse contexto, a aplicação de técnicas de aprendizado de máquina revela-se uma alternativa promissora, uma vez que possibilita a adaptação a diferentes cenários e aprimora a capacidade de reconhecimento de padrões complexos associados a atividades hostis. Neste trabalho, propõe-se a integração de algoritmos de aprendizado de máquina a um Web Application Firewall (WAF), sistema voltado à proteção de aplicações web, com o propósito de ampliar a acurácia na detecção de ataques. Para tal, foram coletados artefatos pertinentes de requisições web, notadamente URLs e dados enviados por usuários, que podem conter indicadores relevantes de comportamento malicioso. O processo de análise envolveu a aplicação de técnicas de pré-processamento de texto, especificamente Bag of Words e Term Frequency-Inverse Document Frequency (TF-IDF), destinadas a identificar termos potencialmente relacionados a ataques e a atribuir pesos proporcionais à sua relevância estatística. A partir dos dados processados, foram treinados e avaliados três algoritmos de classificação: K-Nearest Neighbour (KNN), Random Forest (RF) e Support Vector Machine (SVM). Os resultados obtidos evidenciam o destaque da combinação do Random Forest com a técnica TF-IDF, a qual apresentou acurácia de 95,24%, demonstrando, assim, o potencial dessa abordagem para o fortalecimento dos mecanismos de defesa em aplicações web.

Palavras-chave: Cibersegurança, Web Application Firewall, Aprendizado de Máquina, Segurança Web.

Abstract

The continuous increase in cyber threats has intensified the complexity of detecting attacks targeting web applications, making the adoption of more robust and effective defense mechanisms indispensable. Traditional security tools, although widely employed, present significant limitations, particularly concerning the high incidence of false positives, which undermines the accurate identification of malicious agents and, consequently, the effectiveness of incident response actions. In this context, the application of machine learning techniques emerges as a promising alternative, as it enables adaptation to different scenarios and enhances the ability to recognize complex patterns associated with hostile activities. In this study, the integration of machine learning algorithms into a Web Application Firewall (WAF), a system designed to protect web applications, is proposed with the objective of improving the accuracy of attack detection. For this purpose, relevant artifacts from web

requests were collected, specifically URLs and user-submitted data, which may contain meaningful indicators of malicious behavior. The analysis process involved the application of text preprocessing techniques, namely Bag of Words and Term Frequency-Inverse Document Frequency (TF-IDF), aimed at identifying terms potentially related to attacks and assigning weights proportional to their statistical relevance. Based on the processed data, three classification algorithms were trained and evaluated: K-Nearest Neighbour (KNN), Random Forest (RF), and Support Vector Machine (SVM). The results highlight the superior performance of the Random Forest model combined with the TF-IDF technique, which achieved an accuracy of 95.24%, thereby demonstrating the potential of this approach to strengthen defense mechanisms in web applications.

Keywords: Cybersecurity, Web Application Firewall, Machine Learning, Web Security.

1 INTRODUÇÃO

Com o crescente número de ataques cibernéticos, novas vulnerabilidades críticas têm ganhando mais visibilidade (SVIATUN et al., 2021). Essas falhas servem como pontos de entrada para que atacantes comprometam sistemas, muitas vezes sendo combinadas para realizar ataques de grande escala. (SHARIF; MOHAMMED, 2022)

Estes ataques visam principalmente segmentos de mercado de alto valor, buscando causar grandes prejuízos financeiros para às empresas e obter um grande retorno para os criminosos, por meio de *ransomware*, vazamentos e vendas de informações confidenciais e pessoais (SEGUNDO, 2019). Além disso, tem-se observado um aumento nos ataques patrocinados por governos, com o objetivo de prejudicar nações rivais (ROCHA, 2022).

Tais incidentes geram problemas que vão além da perda financeira, como a falta de conformidade com a LGPD e sérios danos à reputação da empresa (SVIATUN et al., 2021). Um exemplo recente é o ataque à C&M Software, empresa responsável pela integração de bancos ao sistema PIX. O incidente resultou no desvio de aproximadamente R$ 541 milhões, demonstrando o impacto financeiro devastador desses ataques (MAIA, 2025).

Ataques cibernéticos bem-sucedidos frequentemente empregam uma combinação de táticas, técnicas e procedimentos para contornar as defesas de um sistema (CHINWE; ALOZIE, 2025). Isso se deve ao fato de muitos sistemas de defesa, como Firewalls ou Web Application Firewalls (WAFs), se basearem em regras fixas. Essa abordagem compromete a detecção de ataques sofisticados e planejados, resultando em altas taxas de falso-positivo e falso-negativo que afetam negativamente a eficácia do controle de ameaças do sistema (IŞIKER; SOĞUKPINAR, 2021). O objetivo deste trabalho é analisar o uso de técnicas de aprendizado de máquina para detecção de ataques cibernéticos no contexto de Web Application Firewall (WAF).

O trabalho está organizado da seguinte forma: o Referencial Teórico (2) introduz os diferentes modelos de aprendizado de máquina utilizados e os principais conceitos de segurança cibernética relevantes aos estudo. A seção de Metodologia (3) apresenta os *datasets*, as técnicas de pré-processamento e a forma como foram aplicados em conjunto com os algoritmos de aprendizado selecionados. A seção de Resultados e Discussão (4) apresenta e analisa os resultados encontrados. Por fim, a seção de Conclusão (5) resume os resultados mais importantes do estudo.

2 REFERENCIAL TEÓRICO

2.1 Web Application Firewall vs Firewall

O Web Application Firewall (WAF) (RAZZAQ et al., 2013), (SILVA et al., 2021) é uma ferramenta efetiva para implementar na análise de possíveis ameaças a aplicações web. Esse tipo de firewall consegue coletar e analisar o conteúdo inserido no site por um usuário

externo. Baseado em um conjunto de regras o WAF deve decidir se uma requisição é maliciosa ou não, para assim, decidir uma contramedida. Para combater uma invasão, o WAF pode bloquear requisições, suas origens ou até mesmo guardar os dados da invasão para uma futura análise e consequente manutenção do sistema.

O firewall (LIANG; KIM, 2022) é um sistema de detecção de ataques focado em toda estrutura da rede. Ele pode detectar e agir contra ataques que visam prejudicar a parte interna do sistema, como o ataque a protocolos de rede, por exemplo, como demonstrado na Figura 1. Os WAFs se especializam na proteção de aplicações web para detectar, por exemplo, injeções de código na página ou tentativas de acesso não autorizadas a áreas restritas em um site, como demonstrado na Figura 2. As funções dos WAFs e Firewalls são parecidas e sua principal diferença é sua especialização, sendo assim, a criação do WAFs visava separar o escopo da rede da web e atua diretamente na camada 7 (camada de aplicação) do modelo Open Systems Interconnection model (OSI) (HOWSER, 2020).

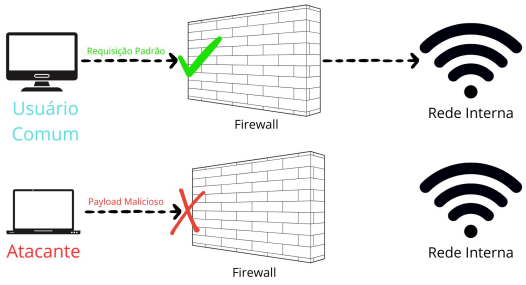

Figura 1 – Funcionamento do Firewall de Rede.

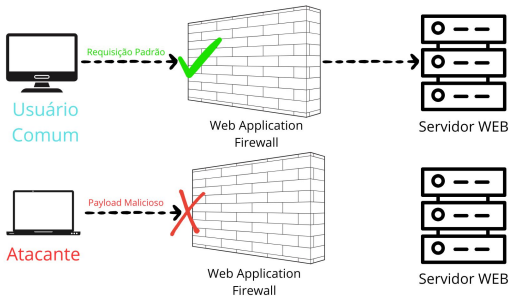

Figura 2 – Funcionamento do Web Application Firewall.

2.2 Aprendizado de Máquina

O aprendizado de máquina e suas técnicas podem ser usados para aprimorar os resultados das métricas de avaliação em várias áreas (SARKER, 2021). Na segurança digital, por exemplo,

esses modelos são aplicados para detectar anomalias e padrões que indicam um ataque (HANDA; SHARMA; SHUKLA, 2019).

A aplicação dessa abordagem já pode ser encontrada em diversos ambientes e trabalhos voltados à segurança digital (ATHIEF et al., 2024). Dentre as técnicas empregadas, temos:

- K Nearest Neighbour(KNN) (ALNUAIMI; ALBALDAWI, 2024): algoritmo usado para regressão e classificação, onde os dados são agrupados com base nas suas características. Suas vantagens incluem bom desempenho em pequenos conjunto de dados (datasets) e facilidade de retreinamento. No entanto, sua desvantagem é a dificuldade de encontrar o número ideal de vizinhos para otimizar o desempenho.
- Support Vector Machine(SVM)(NASTESKI, 2017): é um algoritmo supervisionado usado para classificação e regressão, onde é encontrado o melhor hiperplano para dividir as diferentes classes em que um dado pode ser classificado. Ele se destaca pela capacidade de seus *kernels* em identificar padrões lineares e não-lineares. Como desvantagem, sua baixa escalabilidade o torna menos adequado para ambientes de alto volume de dados.
- Random Forest(RF)(SALMAN; KALAKECH; STEITI, 2024): é um algoritmo que utiliza múltiplas árvores de decisão. Para classificação, faz uma previsão de acordo com os resultados da maior parte das árvores. O Random Forest consegue suportar textos vetorizados, como no caso deste projeto o TF-IDF, e também pode ajudar na redução do *overfitting*. Contudo, pode consumir uma grande quantidade de memória, e, dependendo do tamanho do modelo, pode trazer uma alta latência em um ambiente real.

2.3 Ataques web

Existem diversos ataques que podem ocorrer em uma aplicação web, cada um explorando uma diferente vulnerabilidade em uma diferente tecnologia instalada naquele sistema (MUSCAT, 2016). Além da diversidade de ataques, também há diversas técnicas de ofuscação que visam burlar medidas de segurança e esconder esses ataques dos sistemas de defesa presentes (HEIDERICH et al., 2010). Para essa pesquisa, os principais ataques analisados foram:

- *Cross-site Scripting* (GROSSMAN, 2007): esses ataques procuram explorar falhas na implementação do código ou de tecnologias presentes na aplicação para executar comandos JavaScript . Apesar desse ataque não afetar diretamente os servidores e a máquina onde o site está hospedado, um *payload* pode ser construído para manipular e atacar usuários e administradores do site, roubando suas credenciais ou forçando a vítima a realizar ações maliciosas. Esse ataque pode ser encontrado em forma de comandos de JavaScript, geralmente mesclados com algum elemento HTML.
- *SQL Injection* (NASEREDDIN et al., 2023): tem como o principal objetivo explorar vulnerabilidades ou falhas de implementação de código para executar comandos SQL no

banco de dados sem permissão. Esse ataque pode ser considerado uma falha crítica em um sistema, pois a partir do momento que esse ataque é identificado, um ator malicioso pode construir *payloads* para exfiltração de dados, podendo conter dados de cliente, documentos ou outras informações confidenciais, e também pode ganhar acesso à máquina onde o site está hospedado, e assim aumentar o alcance do ataque para outras máquinas na rede.

2.4 Conjunto de dados (*datasets*)

Os conjuntos de dados (*datasets*) (RENEAR; SACCHI; WICKETT, 2010) são dados coletados e agrupados em algum contexto que representam uma informação. Esses conjuntos de dados são utilizados para treinar o modelo a fazerem o proposto de forma mais precisa possível. Geralmente esses dados contém informações corretas e falsas que ajudam a guiar a aprendizagem e melhorar a performance do modelo.

Nesse projeto, um dos conjuntos de dados utilizado é o CSIC 2010 (GIMÉNEZ; VILLEGAS; MARAÑÓN, 2010). Ele foi criado pelo Instituto de Segurança da Informação do Conselho Nacional de Pesquisa da Espanha, onde foram gerados, automaticamente, milhares de exemplos de requisições HTTP, nas quais, dentre essas, pode haver possíveis ataques.

Também foi utilizado um conjunto de dados composto pela junção do CSIC 2010, apresentado anteriormente, e o ECML PKDD 2007 (RAïSSI et al., 2007), criado e demonstrado no evento "European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases", realizado em 2007.

2.5 Métricas

O desempenho de modelos de aprendizagem de máquina é avaliado por meio de métricas como acurácia, precisão e *recall* (IŞIKER; SOĞUKPINAR, 2021) (SHAHEED; KURDY, 2022) (BADRI; ALOUNEH, 2023). Essas métricas são calculadas usando os resultados do modelo: Verdadeiro-Positivo (VP), Falso-Positivo (FP), Verdadeiro-Negativo (VN) e Falso-Negativo (FN).

A acurácia é uma medida que procura avaliar a performance geral do modelo, levando em consideração todas as classificações.

$$Acurcia = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

A medida precisão procura quantificar o número de acertos, dentre as medidas positivas, que o modelo fez. Essa métrica é usada quando os Falsos Positivos são considerados mais prejudiciais que os Falsos Negativos, o que se encaixa nesse projeto. Assim, observa-se o número de requisições benignas que são bloqueadas, o que pode afetar o fluxo de usuários no site.

$$Preciso = \frac{VP}{VP + FP} \tag{2}$$

O *recall* é uma métrica que é usada quando o número de Falsos Negativos é mais prejudicial que o de Falsos Positivos. Com esta métrica podemos observar o número de ataques que não são detectados e consequentemente entrariam no sistema.

$$Recall = \frac{VP}{VP + FN} \tag{3}$$

Além destas métricas, a matriz de confusão é uma ferramenta para avaliar o desempenho de um modelo de classificação. Ela permite visualizar, de forma detalhada, o número de previsões corretas e incorretas. Os dados são organizados e apresentados em uma tabela, conforme a Tabela 1.

	Dados Classificados Como Positivo	Dados Classificados Como Negativo
Resultados Retornados Positivos	Verdadeiro-Positivo (VP)	Falso-Positivo (FP)
Resultados Retornados Negativos	Falso-Negativo (FN)	Verdadeiro-Negativo (VN)

Tabela 1 – Formatação da Matriz de Confusão.

2.6 Estrutura de uma Requisição HTTP

A comunicação entre um cliente e um servidor é feita através de mensagens, utilizando um protocolo específico para cada serviço. No contexto de aplicações web, o protocolo utilizado é o HTTP (Hypertext Transfer Protocol) (FIELDING et al., 1999).

O entendimento da estrutura de uma requisição (mensagem) HTTP é extremamente importante para detecção de ataques web pois é nela onde se encontram todos os dados referentes ao cliente que faz uma requisição, incluindo os dados inseridos e informações sobre àquela requisição (FIELDING et al., 1999). Alguns campos que podemos destacar e que possuem grande relevância na detecção de ameaças digitais são (FIELDING et al., 1999):

- Data: este campo armazena todos os dados que serão passados para o servidor processar que foram inseridos pelo usuário;
- Content-Length: Este campo informa o tamanho dos dados passado no campo Data;

A coleta da operação sendo enviada também é extremamente importante, pois é ela que irá identificar a intenção de quem faz uma determinada requisição (Leitura, Criação, Modificação ou Exclusão de uma informação).

A Figura 3 detalha os componentes de uma requisição web. A estrutura é composta por dois cabeçalhos: o cabeçalho da requisição (Request Headers), que contém informações básicas como o endereço do site e informações do navegador do cliente; e o cabeçalho de representação

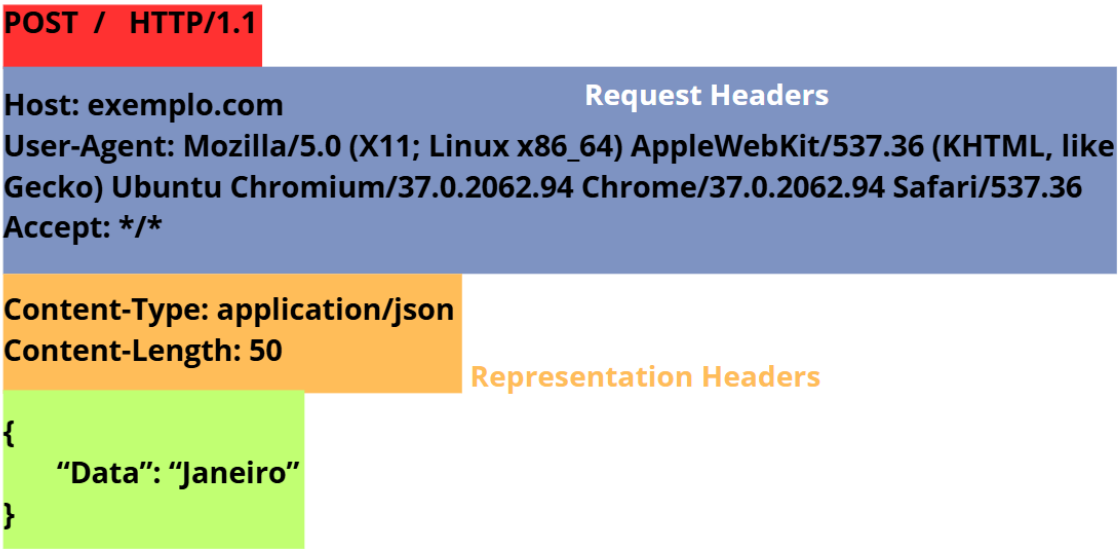

Figura 3 – Representação da Estrutura HTTP.

(Representation Headers), que especifica informações sobre os dados da comunicação, como o formato e o tamanho. Além disso, a Figura 3 mostra o método e a versão do protocolo HTTP utilizados (parte superior) e os dados enviados pelo cliente ao servidor (parte inferior).

2.7 Trabalhos Relacionados

Em (IŞIKER; SOĞUKPINAR, 2021) foi utilizado o conjunto de dados CSIC 2010, com a técnicas n-gram de NLP para quebrar e determinar a frequência das palavras do cabeçalho HTTP. Como sugestão no artigo, ele propõe a tentar utilizar outras técnicas de NLP. Ataques citados no artigo e no OWASP Top 10 foram testados nessa pesquisa.

Em (SHAHEED; KURDY, 2022) foi utilizado o conjunto de dados CSIC 2010, HTTPParams 2015 e registros de servidor para a aprendizagem, onde foi coletado o protocolo HTTP, a URL absoluta, as entradas dos usuários, o cabeçalho e os arquivos de uma requisição HTTP e onde houve o extrato do tamanho da requisição, da porcentagem de caracteres especiais e permitidos e o peso do ataque. Usando essas informações de parâmetro, foi usado alguns algoritmos como regressão linear, Naive Bayes, onde houve o foco, e árvore de decisão.

Em (BADRI; ALOUNEH, 2023) foi analisado o uso de aprendizagem de máquina para detecção de ameaças em servidores de nome de domínio (DNS) e aplicações web. Foram utilizados algoritmos como AdaBoost e Naive Bayes, juntamente de conjunto de dados como CSIC 2010, ECML/PKDD 2007, CICID 2017 e SQLi 2021, e os autores descobriram que o AdaBoost obteve a maior acurácia.

Em (KALARIYA; JETHVA; ALGINAHI, 2024), é criado um modelo de detecção de ameaças voltado para um ambiente de Web Application Firewall. Para criação desse modelo, Kalariya coleta diversas palavras referentes à ataques web e as aplica à um vetorizador TF-IDF.

Após isso, ele também calcula diversos pontos chaves que podem ser indicadores de ataque, assim, utilizando todas essas informações, ele testa 3 algoritmos de aprendizagem de máquina (KNN, Random Forest e Support Vector Machine) e aplica em seu ambiente WAF, tendo como melhor resultado o KNN.

Neste projeto utilizamos Bag of Words e TF-IDF com o objetivo de analisar a detecção de palavras chaves utilizadas em ataques cibernéticos (YANG et al., 2022), e suas possíveis variações e tentativas de ofuscamento.

3 METODOLOGIA

O fluxo de trabalho e as etapas de cada fase deste trabalho são apresentados na Figura 4 e são detalhados nas subseções seguintes.

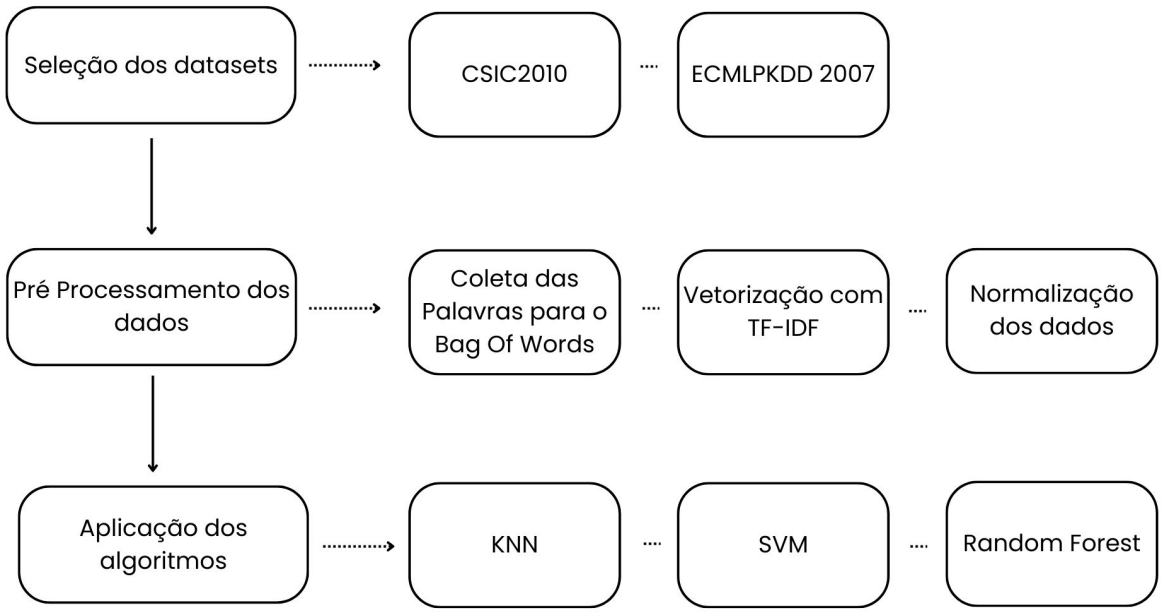

Figura 4 – Fluxo de trabalho.

3.1 Seleção dos datasets e dos atributos (*Features*)

Para o treinamento do modelo, foram escolhidos dois conjuntos de dados (*datasets*) principais: ECML/PKDD 2007 (RAïSSI et al., 2007) e CSIC 2010 (GIMÉNEZ; VILLEGAS; MARAÑÓN, 2010) . Ambos *datasets* possuem dados reais de diversos experimentos de ataques conduzidos em um ambiente simulado. Além deles, foi criado um terceiro conjunto de dados combinando as informações dos dois primeiros (ECML+CSIC).

O modelo foi treinado usando atributos (features) extraídos dos campos de requisição HTTP, que incluem:

- URL: armazena a URL completa que aquela requisição está sendo feita.
- *content*: armazena todos os dados enviados pelo clientes.
- *Content-Length*: onde armazena o tamanho dos dados.

Para identificar ataques, o algoritmo analisa os dados benignos e maliciosos de acordo com os seguintes critérios:

- URL: procura por ataques mais simples, geralmente implementados nos parâmetros da própria URL, e por tentativas de ofuscação através de caracteres especiais.
- Dados enviados pelo usuário (*Content*): busca por ataques mais complexos e bem desenvolvidos, procurando também identificar tentativas de ofuscação.
- Tamanho do conteúdo (*Content-Length*): analisa a discrepância de tamanho entre uma requisição normal e um ataque.

Os campos selecionados, sua descrição e exemplo de dados podem ser observados na Tabela 2.

Campo	Descrição	Exemplo
URL	Dados enviados/solicitados pela url	https://example.com/page.php=1
Content	Dados enviados no POST	{user:"admin",password:"123"}
Content-Length	Tamanho dos dados enviados	Content-Length: 50

Tabela 2 – Exemplo de dados chave encontrados no *dataset*.

3.2 Pré-Processamento dos Dados

Nesse trabalho, os testes foram feitos utilizando dois diferentes tipos de pré-processamento: *Bag of Words*(BoW), onde há a coleta manual ou automática de palavras chaves que serão utilizadas para a análise, e *Term Frequency - Inverse Document Frequency*(TF-IDF), onde será analisada a frequência de determinadas palavras e gerada uma pontuação a partir do número de aparições dessa palavra através dos dados (YANG et al., 2022).

No primeiro caso, com a utilização do modelo Bag of Words (BoW), foi realizada a coleta das principais palavras que podem indicar a ocorrência de um ataque. Como fonte principal, utilizou-se o repositório PayloadAllTheThings (SWISSKY, 2025), amplamente adotado por profissionais de cibersegurança para consulta e extração de exemplos de ataques, palavras-chave e seus respectivos *payloads*. A partir dessas palavras, foram extraídas características relevantes para serem aplicadas no modelo:

- Quantidade de palavras suspeitas na URL.

- Quantidade de palavras suspeitas no campo *Content*.
- Quantidade de caracteres especiais na URL.
- Quantidade de caracteres especiais no campo *Content*.
- Quantidade de caracteres alfanuméricos na URL.
- Quantidade de caracteres alfanuméricos no campo *Content*.
- Tamanho da URL.
- Tamanho do *Content*.

No segundo caso, adotou-se a técnica Term Frequency–Inverse Document Frequency (TF-IDF) para a vetorização dos dados textuais. Foram gerados dois vetores de TF-IDF, aplicados respectivamente aos campos de URL e *Content*, limitados a um máximo de 200 e 500 termos, nessa ordem. Os vetores resultantes foram convertidos para o formato esparso e, em seguida, combinados com as demais características selecionadas. Nessa versão, além dos vetores gerados, apenas a métrica *Content-Length* foi incluída como atributo adicional.

Nos dois casos foi utilizado o *Standard Scaler* para fazer a normalização dos dados.

3.3 Algoritmos de Aprendizado de Máquina utilizados

Nesse projeto, foram utilizados 3 algoritmos de aprendizado de máquina: Random Forest, K-Nearest Neighbour(KNN) e Support Vector Machine(SVM). Para cada um deles, os experimentos foram conduzidos utilizando diferentes divisões entre dados de treinamento e de teste, nas proporções de 90/10, 80/20 e 70/30, respectivamente.

Nos experimentos com o algoritmo K-Nearest Neighbours (KNN), foram testados diferentes valores para o número de vizinhos, sendo selecionados 11, 25 e 30, uma vez que apresentaram os melhores resultados de acurácia.

Já nos modelos que empregaram o Support Vector Machine (SVM), adotou-se de forma consistente o uso do *kernel* polinomial, por ter demonstrado desempenho superior em termos de acurácia em comparação com outras funções de *kernel*.

Também foi testado a aplicação do BoW e do TF-IDF em um modelo utilizando o algoritmo Random Forest, no qual retornou os melhores resultados.

4 RESULTADOS E DISCUSSÃO

4.1 Análise dos resultados

A análise dos resultados obtidos revelou que Random Forest com TD-IDF apresentou os melhores desempenho para ambos os *dataset* (CSIC2010 e ECML+CSI) (Tabela 3). Nos

experimentos realizados, observou-se que o algoritmo Random Forest apresentou acurácia de 95,24% no *dataset* ECML+CSIC , superando os demais modelos SVM e KNN que alcançaram valores de 93,13% e 92,35%.

Além disso, os resultados demonstram que o uso do TF-IDF contribuiu significativamente para a melhoria da detecção, uma vez que o método captura a frequência relativa de palavras-chave associadas a ataques.

Ao analisar as matrizes de confusão, nota-se que o modelo SVM com TF-IDF apresentou baixo número de falsos-negativos, o que indica alta capacidade de bloqueio de tráfego malicioso.

Por outro lado, o modelo Random Forest com TF-IDF apresentou equilíbrio entre as métricas, com baixos índices de falsos positivos e falsos negativos, resultando no melhor desempenho global.

Dataset	Modelo	TF-IDF	Acurácia	Precisão	Recall	Matriz de Confusão
CSIC2010	KNN	Não	89,93%	90%	89%	|6832 488| |741 4152|
CSIC2010	KNN	Sim	92,35%	93%	91%	|7018 302| |632 4261|
CSIC2010	SVM	Não	68,25%	73%	61%	|6991 329| |3548 1345|
CSIC2010	SVM	Sim	93,13%	94%	92%	|7179 141| |698 4195|
CSIC2010	RF	Não	89,86%	90%	89%	|6814 506| |732 4161|
CSIC2010	RF	Sim	95,24%	95%	95%	|7067 253| |340 4553|
ECML+CSIC	KNN	Não	81,88%	83%	81%	|5287 2375| |703 8627|
ECML+CSIC	KNN	Sim	83,25%	85%	82%	|5283 2379| |467 8863|
ECML+CSIC	SVM	Não	73,35%	77%	71%	|3904 3758| |770 8560|
ECML+CSIC	SVM	Sim	84%	86%	84%	|5239 2423| |295 9035|
ECML+CSIC	RF	Não	81,11%	82%	80%	|5298 2364| |839 8491|
ECML+CSIC	RF	Sim	86,83%	89%	86%	|5789 1873| |364 8966|

Tabela 3 – Tabela de Resultados.

4.2 Discussão

A análise da literatura mostra que os *datasets* ECMLPKDD 2007 e CSCIC 2010 são frequentemente utilizados, como pode ser observado em (IŞIKER; SOĞUKPINAR, 2021),

(SHAHEED; KURDY, 2022), (KALARIYA; JETHVA; ALGINAHI, 2024), (BADRI; ALOUNEH, 2023). Estes *datasets* possuem uma grande quantidade de dados essenciais para esse tipo de pesquisa, coletadas em um ambiente simulado, que contêm tanto requisições maliciosas e quantos benignas. Algumas pesquisas também utilizaram seus próprios conjuntos de dados, geralmente fornecidos por organizações ou coletados manualmente, e nem sempre esses dados são disponibilizados para serem utilizados.

Entre os trabalhos analisados é possível destacar o uso de algoritmos como KNN, Random Forest e Support Vector Machine (SVM). Esses algoritmos, combinados a seleção de atributos (*features*) apresentados em 3.1, pré-processamento e análise dos dados, resultaram em diversos modelos com performances variadas. Dessa forma, um modelo proposto pode ser capaz de detectar certo tipos de ataques, com diferentes formatos e possíveis tentativas de ofuscação, enquanto outro modelo pode falhar nesta detecção, mas ser capaz de detectar outros padrões.

Comparando os resultados com (KALARIYA; JETHVA; ALGINAHI, 2024), os modelos propostos neste trabalho retornaram as melhores medidas relacionadas à precisão quando aplicados no conjunto de dados CSIC 2010, utilizando o algoritmo Random Forest e TF-IDF. O modelo proposto foi capaz de retornar uma precisão de 95% enquanto o modelo de Kalariya et. al. retornou uma precisão de 89,45%.

Os resultados inferiores nos outros *datasets* podem ter ocorrido pois há uma grande diversidade de ataques. Alguns ataques não contêm palavras chaves que são o foco principal de detecção do modelo deste projeto, o que limita sua capacidade de detecção nesses cenários.

Os modelos e resultados obtidos neste projeto demonstram que a implementação do aprendizado de máquina em sistemas de segurança defensiva podem trazer melhorias para os processos e ferramentas utilizadas na defesa de aplicações. Para criar modelos ainda mais precisos e capazes de interromper ataques complexos, é necessário alimentá-los com mais dados de cenários e ataques reais, além de treiná-los com técnicas avançadas para detecção de palavras-chave e tentativas de ofuscação.

5 CONCLUSÃO

Este trabalho teve como objetivo avaliar o uso de modelos de aprendizado de máquina na detecção de ataques em aplicações web. Os experimentos realizados evidenciaram que o Random Forest em conjunto com a técnica TF-IDF apresentou os melhores resultados, alcançando 95,24% de precisão e obteve equilíbrio entre falsos-positivos e falsos-negativos, com 95% de precisão e *recall*. Resalta-se também o aprimoramento dos resultados utilizando a técnica de pré-processamento TF-IDF para todos os algoritmos de aprendizado de máquinas analisados (KNN, SVM e Random Forest).

A implementação de modelos de aprendizado de máquina em sistemas de proteção digital pode aprimorar significativamente as atividades de detecção e prevenção de ameaças e liberar

analistas de segurança para se concentrarem em tarefas mais complexas e estratégicas.

Além disso, um sistema de segurança mais robusto oferece à empresa maior controle sobre o *compliance* com a Lei Geral de Proteção dos Dados (LGPD). Isso mitiga o risco de grandes vazamentos de dados e, consequentemente, reduz os custos e danos reputacionais associados a esses incidentes.

Em um ambiente de Web Application Firewall (WAF), a implementação de um modelo de aprendizado de máquina pode ser extremamente vantajosa. Ele tem a capacidade de bloquear ataques e atacantes de forma proativa, antes mesmo que sejam executados. No entanto, um modelo com um grande número de falsos positivos pode prejudicar o fluxo de usuários legítimos, impactando negativamente o acesso e a operação do negócio.

Alguns aspectos podem ser explorados em trabalhados futuros com intuito de aprimorar os resultados dos modelos, tais como: ampliar a análise de outros ataques de aplicações web; investigar táticas e técnicas utilizadas por grandes grupos de criminosos para identificar padrões que possam ser aplicados ao modelo; explorar o uso de outras técnicas de aprendizado de máquina, como abordagens baseadas em *deep learning*; criar de um ambiente controlado para coleta centralizada de requisições web e de tentativas de ataques, para geração de base de dados que comportem cenários reais de ataques.

6 REFERÊNCIAS

ALNUAIMI, A. F.; ALBALDAWI, T. H. An overview of machine learning classification techniques. In: EDP SCIENCES. *BIO Web of Conferences*. [S.l.], 2024. v. 97, p. 00133.

ATHIEF, R. et al. Web application firewall using machine learning. In: IEEE. *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. Chennai, India, 2024. p. 1–7.

BADRI, A. M. A.; ALOUNEH, S. Detection of malicious requests to protect web applications and dns servers against sql injection using machine learning. In: IEEE. *2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS)*. Zhuhai, China, 2023. p. 5–11.

CHINWE, E.; ALOZIE, C. E. Adversarial tactics, techniques, and procedures (ttps): A deep dive into modern cyber attacks. *ICONIC RESEARCH AND ENGINEERING JOURNALS*, v. 8, n. 7, p. 552–561, 2025.

FIELDING, R. et al. *RFC2616: Hypertext transfer protocol–HTTP/1.1*. [S.l.]: RFC Editor, 1999.

GIMÉNEZ, C. T.; VILLEGAS, A. P.; MARAÑÓN, G. Á. *HTTP data set CSIC 2010*. Madrid, Espanha: Information Security Institute of CSIC (Spanish Research National Council), 2010. Disponível em: <https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks/>. Acesso em: 05 ago. 2025.

GROSSMAN, J. *XSS attacks: cross site scripting exploits and defense*. [S.l.]: Syngress, 2007.

HANDA, A.; SHARMA, A.; SHUKLA, S. K. Machine learning in cybersecurity: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 9, n. 4, p. e1306, 2019.

HEIDERICH, M. et al. *Web Application Obfuscation:'-/WAFs.. evasion.. filters//alert (/obfuscation/)-'*. [S.l.]: Elsevier, 2010.

HOWSER, G. *Computer Networks and the Internet*. [S.l.]: Springer, 2020.

IŞIKER, B.; SOĞUKPINAR, İ. Machine learning based web application firewall. In: IEEE. *2021 2nd International Informatics and Software Engineering Conference (IISEC)*. Bhubaneswar, Índia, 2021. p. 1–6.

KALARIYA, P.; JETHVA, M.; ALGINAHI, Y. Ml assisted web application firewall. In: IEEE. *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*. Texas, Estados Unidos, 2024. p. 1–6.

LIANG, J.; KIM, Y. Evolution of firewalls: Toward securer network using next generation firewall. In: IEEE. *2022 IEEE 12th annual computing and communication workshop and conference (CCWC)*. Las Vegas, Estados Unidos, 2022. p. 0752–0759.

MAIA, E. *Suspeito de ataque hacker a prestadora de serviços financeiros é preso*. Brasília, Brasil: CNN, 2025. Disponível em: <https://www.cnnbrasil.com.br/nacional/sudeste/sp/suspeito-de-ataque-hacker-a-prestadora-de-servicos-financeiros-e-preso>. Acesso em: 05 ago. 2025.

MUSCAT, I. Web vulnerabilities: identifying patterns and remedies. *Network Security*, Elsevier, v. 2016, n. 2, p. 5–10, 2016.

NASEREDDIN, M. et al. A systematic review of detection and prevention techniques of sql injection attacks. *Information Security Journal: A Global Perspective*, Taylor & Francis, v. 32, n. 4, p. 252–265, 2023.

NASTESKI, V. An overview of the supervised machine learning methods. *Horizons. b*, v. 4, n. 51-62, p. 56, 2017.

RAZZAQ, A. et al. Critical analysis on web application firewall solutions. In: IEEE. *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*. Mexico City, Mexico, 2013. p. 1–6.

RAïSSI, C. et al. *ECML PKDD 2007 dataset*. Varsóvia, Polônia: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2007. Disponível em: <https://github.com/msudol/Web-Application-Attack-Datasets>. Acesso em: 05 ago. 2025.

RENEAR, A. H.; SACCHI, S.; WICKETT, K. M. Definitions of dataset in the scientific and technical literature. *Proceedings of the American Society for Information Science and Technology*, Wiley Online Library, v. 47, n. 1, p. 1–4, 2010.

ROCHA, G. C. Caso stuxnet: os impactos do ataque cibernético ao programa nuclear do irã com a primeira arma cibernética à segurança internacional (2009-2010). Universidade Estadual Paulista (Unesp), 2022.

SALMAN, H. A.; KALAKECH, A.; STEITI, A. Random forest algorithm overview. *Babylonian Journal of Machine Learning*, v. 2024, p. 69–79, 2024.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, Springer, v. 2, n. 3, p. 160, 2021.

SEGUNDO, C. B. T. A defesa cibernética em ambientes de infraestrutura crítica e os riscos dos ataques cibernéticos. Escola Superior de Guerra (Campus Brasília), 2019.

SHAHEED, A.; KURDY, M. B. Web application firewall using machine learning and features engineering. *Security and Communication Networks*, Wiley Online Library, v. 2022, n. 1, p. 5280158, 2022.

SHARIF, M. H. U.; MOHAMMED, M. A. A literature review of financial losses statistics for cyber security and future trend. *World Journal of Advanced Research and Reviews*, v. 15, n. 1, p. 138–156, 2022.

SILVA, E. dos S. et al. Waf: Uma análise de desempenho e eficácia. In: SBC. *Simpósio Brasileiro de Sistemas de Informação (SBSI)*. [S.l.], 2021. p. 21–24.

SVIATUN, O. et al. Combating cybercrime: economic and legal aspects. *WSEAS Transactions on Business and Economics*, WSEAS, v. 18, p. 751–762, 2021.

SWISSKY. *Payloads All The Things*. 2025. Disponível em: <https://github.com/swisskyrepo/PayloadsAllTheThings>. Acesso em: 05 ago. 2025.

YANG, X. et al. A study of text vectorization method combining topic model and transfer learning. *Processes*, MDPI, v. 10, n. 2, p. 350, 2022.