Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de Tráfego de Rede Malicioso Utilizando Vetorização e Aprendizado de Máquina Aplicados a Fluxos de Dados

Claudio Henrique Marques de Oliveira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. João José Costa Gondim

Brasília
2026

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de Tráfego de Rede Malicioso Utilizando Vetorização e Aprendizado de Máquina Aplicados a Fluxos de Dados

Claudio Henrique Marques de Oliveira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. João José Costa Gondim (Orientador)
ENE/UnB

Prof. Dr. Ricardo Sant'Ana
IME

Dr. Robson de Oliveira Albuquerque
ENE/UnB

Prof.a Dr.a Edna Dias Canedo
Coordenadora do Programa de Pós-graduação em Computação Aplicada

Brasília, 12 de Janeiro de 2026

Dedicatória

Dedico esta dissertação à minha irmã e ao meu filho. À minha irmã, pela presença constante, pelo apoio silencioso e pela firmeza nos momentos em que o caminho parecia mais difícil do que o previsto. Ao meu filho, por ser minha motivação diária, por dar sentido ao esforço e por me lembrar, mesmo sem palavras, do valor de concluir aquilo que se começa. Este trabalho é resultado de disciplina e método, mas também de suporte afetivo e estabilidade emocional, elementos sem os quais a pesquisa não se sustenta no tempo.

Obrigado pelo apoio

Agradecimentos

Agradeço, primeiramente, aos professores que contribuíram diretamente para minha formação e para a condução desta pesquisa. Em especial, ao Prof. Dr. João Gondim, pela orientação, pelas contribuições técnicas e pela condução rigorosa ao longo do trabalho. Agradeço também ao Prof. Dr. Marcelo Ladeira, ao Prof. Dr. Ricardo Sant'Ana e ao Prof. Dr. Robson Albuquerque, pelas avaliações, sugestões e apontamentos que ampliaram a qualidade e a consistência desta dissertação.

Registro um agradecimento especial à Profa. Rose, pela motivação contínua, pelo apoio e pela confiança, fundamentais para manter a disciplina e a persistência necessárias à conclusão deste estudo.

Agradeço à Priscila pelo suporte, incentivo e pela presença nos momentos decisivos do percurso. Agradeço, ainda, aos meus familiares e amigos, pela compreensão, paciência e apoio ao longo desta jornada, especialmente nos períodos de maior dedicação e renúncia.

Agradeço à Universidade de Brasília (UnB) por proporcionar um ambiente acadêmico de excelência e pela oportunidade de aprendizado, pesquisa e desenvolvimento. Por fim, agradeço à Marinha do Brasil, pelo suporte institucional e pela oportunidade, que viabilizaram condições essenciais para a realização deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A detecção de ameaças cibernéticas representa um desafio crescente frente ao avanço tecnológico e à sofisticação constante dos métodos de ataque. Esta pesquisa propõe uma metodologia integrada para refinamento na detecção de tráfego malicioso, combinando a análise de tráfego de rede com o monitoramento de perfis em redes sociais. A abordagem fundamenta-se em técnicas de vetorização para representação de dados, algoritmos de aprendizado de máquina para classificação de padrões, e análise de redes complexas para identificação de comunidades de interesse em plataformas sociais.

Em relação ao tráfego de rede, foram utilizados conjuntos de dados do repositório *malware-traffic-analysis.net*, extraindo características de pacotes e fluxos de comunicação para treinamento e validação de modelos classificadores (Random Forest, K-Nearest Neighbors e XGBoost). Complementarmente, postagens coletadas da plataforma "X" (antigo Twitter) foram analisadas a partir de dados da plataforma Zone-H, aplicando-se técnicas de processamento de linguagem natural e análise de redes complexas para identificar usuários associados a atividades hacktivistas. A implementação da metodologia utilizou o banco de dados vetorial Qdrant e processamento paralelo com Dask para garantir escalabilidade e desempenho em tempo real.

Os resultados demonstraram alta eficácia na identificação de tráfego malicioso, com acurácia de **97,36%** utilizando o algoritmo KNN, e ROC-AUC de **0,9928**, incluindo para classes raras (**0,9855**). A integração das duas fontes de dados, tráfego de rede e postagens em redes sociais permitiu a identificação precoce de ameaças e potenciais alvos, contribuindo significativamente para o desenvolvimento de estratégias proativas de cibersegurança. A pesquisa demonstrou que a combinação dessas abordagens pode refinar substancialmente os sistemas de detecção de intrusão, fornecendo respostas mais ágeis e precisas frente a cenários de ameaças emergentes.

Palavras-chave: Detecção de Intrusão, Aprendizado de Máquina, Vetorização Densa, Análise de Redes Complexas, Hacktivismo, Cibersegurança, Qdrant, OSINT.

Abstract

The detection of cyber threats represents a growing challenge in the face of technological advancement and the constant sophistication of attack methods. This research proposes an integrated methodology for refining malicious traffic detection, combining network traffic analysis with monitoring of profiles on social networks. The approach is based on vectorization techniques for data representation, machine learning algorithms for pattern classification, and complex network analysis for identifying communities of interest on social platforms.

Regarding network traffic, datasets from the *malware-traffic-analysis.net* repository were used, extracting characteristics from packets and communication flows for training and validating classifier models (Random Forest, K-Nearest Neighbors, and XGBoost). Complementarily, posts collected from the "X" platform (formerly Twitter) were analyzed using data from the Zone-H platform, applying natural language processing techniques and complex network analysis to identify users associated with hacktivist activities. The implementation of the methodology used the Qdrant vector database and parallel processing with Dask to ensure scalability and real-time performance.

The results demonstrated high effectiveness in identifying malicious traffic, with an accuracy of **97.36%** using the KNN algorithm, and ROC-AUC of **0.9928**, including for rare classes (**0.9855**). The integration of the two data sources, network traffic and social network posts allowed for the early identification of threats and potential targets, significantly contributing to the development of proactive cybersecurity strategies. The research demonstrates that the combination of these approaches can substantially refine intrusion detection systems, providing more agile and accurate responses to emerging threat scenarios.

Keywords: Intrusion Detection, Machine Learning, Vectorization, Complex Network Analysis, Hacktivism, Cybersecurity, Real-time Data Processing.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A era digital contemporânea é caracterizada por uma escalada contínua na complexidade e frequência das ameaças cibernéticas. Ataques como *ransomware* e Ameaças Persistentes Avançadas (APTs) tornaram-se ocorrências comuns, explorando vulnerabilidades em um ecossistema digital cada vez mais interconectado (Bhardwaj et al., 2023). Paralelamente, o volume e a diversidade do tráfego de rede cresceram exponencialmente, impulsionados pela proliferação de dispositivos Internet das coisas (IoT), serviços em nuvem e interações online (Alamri et al., 2022). Essa confluência de fatores torna a análise manual do tráfego impraticável, exigindo o desenvolvimento e a aplicação de soluções automatizadas e inteligentes para a segurança da rede.

A escalada das ameaças cibernéticas é documentada pelo relatório anual da Verizon (2024), que registrou um aumento de 38% nos incidentes de segurança cibernética entre 2023 e 2024, com organizações brasileiras reportando perdas financeiras diretas estimadas em R$ 97,4 bilhões. Conforme destacado por ENISA (2023), os ataques estão se tornando mais sofisticados, utilizando técnicas avançadas de evasão que dificultam sua detecção por métodos tradicionais baseados em assinaturas.

Nesse contexto, a Análise de Tráfego de Rede (NTA) e os Sistemas de Detecção de Intrusão de Rede (NIDS) emergem como componentes fundamentais das estratégias modernas de defesa cibernética (Fernandes et al., 2022). A NTA oferece a visibilidade necessária sobre o comportamento da rede, essencial não apenas para a segurança, mas também para o monitoramento de desempenho e a solução de problemas operacionais (Kent and Liebrock, 2021). Historicamente, muitos NIDS dependiam de abordagens baseadas em assinaturas, que comparam o tráfego observado com padrões de ataques conhecidos. No entanto, a eficácia dessas abordagens é limitada diante de ameaças novas (zero-day), polimórficas ou desconhecidas (Ahmad et al., 2021).

Essa limitação impulsionou uma transição significativa para a detecção baseada em anomalias, que busca identificar desvios do comportamento normal da rede. Essa mudança foi catalisada pela integração crescente de técnicas de Aprendizado de Máquina (ML - Machine Learning) e Aprendizado Profundo (DL - Deep Learning) Dutta et al. (2022). Tais tecnologias oferecem a capacidade de aprender padrões complexos e identificar anomalias sutis que poderiam passar despercebidas pelos métodos tradicionais Srinivasan et al. (2023).

Simultaneamente, as plataformas de redes sociais tornaram-se espaços não apenas de interação social, mas também de articulação de comunidades Hacktivistas e atores de ameaças cibernéticas. Conforme demonstrado por Coleman Coleman (2014) e posteriormente analisado por Zhang et al. Zhang et al. (2022), essas plataformas frequentemente servem como canais de comunicação, planejamento e divulgação de ataques cibernéticos. Neste sentido, a plataforma "X"(anteriormente Twitter) tem se estabelecido como um importante meio para compartilhamento de informações relacionadas a vulnerabilidades e ataques cibernéticos, conforme documentado no estudo de Le Sceller et al. (2017).

A interseção entre a análise de tráfego de rede e o monitoramento de redes sociais para detecção de ameaças representa uma fronteira de pesquisa com potencial significativo para aprimorar os mecanismos de segurança cibernética. Hernandez et al. Hernandez et al. (2016) destacaram a importância de integrar dados de múltiplas fontes para estabelecer um sistema de alerta antecipado mais robusto e eficaz contra ataques cibernéticos. Khandpur et al. Khandpur et al. (2017) reforçam esta perspectiva, demonstrando que a correlação entre indicadores de ameaças obtidos em plataformas sociais e padrões de tráfego de rede pode antecipar ataques em até 4,5 dias em relação aos sistemas tradicionais.

Esta pesquisa, portanto, busca explorar e desenvolver uma abordagem integrada, combinando técnicas avançadas de análise de tráfego de rede e monitoramento de atividades em redes sociais.

1.1 Problema de Pesquisa

A detecção eficaz de atividades maliciosas, especialmente aquelas associadas a malware, apresenta desafios significativos e multifacetados que exigem uma abordagem integrada e inovadora. Os problemas fundamentais que esta pesquisa busca abordar são:

P1: Limitações dos sistemas de detecção tradicionais frente a ameaças emergentes. Os métodos convencionais de detecção, particularmente aqueles baseados em assinaturas, demonstram eficácia insuficiente diante de ameaças *zero-day* e técnicas de evasão avançadas, Em Ahmad et al. (2021) foi demonstrado que sistemas baseados em assinaturas identificam apenas 37% dos ataques recém-desenvolvidos, deixando uma lacuna significa-

tiva na proteção cibernética. Essa lacuna é crítica, considerando que o tempo médio para desenvolvimento de assinaturas para novas famílias de malware é de 48 horas, conforme documentado por Verizon Verizon (2024).

P2: Dificuldades na análise de tráfego criptografado. A prevalência crescente de tráfego criptografado (TLS/SSL) cria obstáculos significativos para a inspeção de conteúdo Singh and Singh (2020). O Group (2023) reporta que mais de 78% do tráfego malicioso agora utiliza criptografia para evadir detecção, tornando inviável a dependência exclusiva de técnicas de inspeção profunda de pacotes (DPI). Isso requer métodos alternativos que possam analisar metadados, características comportamentais e padrões de fluxo sem acessar o conteúdo dos pacotes.

P3: Escalabilidade e desempenho em tempo real.

O crescimento exponencial do volume de dados de rede continua a ser um desafio significativo. Em 2024, o volume global de dados atingiu 147 zettabytes, e a previsão é que chegue a 181 zettabytes em 2025 Topics (2025), esse crescimento exige que os sistemas de detecção processem e analisem fluxos contínuos de dados com latência mínima para permitir respostas oportunas a incidentes de segurança. Os desafios no processamento de dados em tempo real incluem lidar com o imenso volume e velocidade dos dados, garantir a precisão e qualidade dos dados, reduzir a latência e os atrasos no processamento, e manter a segurança e conformidade dos dados Selfuel (2024).

Conforme destacado por Akanbi and Masinde (2020), a arquitetura de processamento distribuído é essencial para a viabilidade de soluções de segurança em escala.

P4: Falta de integração entre sinais de alerta em redes sociais e análise de tráfego de rede. Existe uma desconexão significativa entre os sinais precursores de ataques frequentemente compartilhados em plataformas sociais e os sistemas de monitoramento de tráfego de rede. Em Khandpur et al. (2017) foi destacado que esta lacuna impede a detecção precoce e a mitigação eficaz de ataques iminentes. Tal integração é particularmente relevante no contexto do hacktivismo, onde anúncios e coordenação de ataques frequentemente ocorrem em plataformas públicas antes da execução efetiva.

Diante destes problemas, formula-se a seguinte pergunta central de pesquisa:

Como desenvolver uma metodologia integrada que combine análise avançada de tráfego de rede com monitoramento de atividades em redes sociais para aprimorar a detecção e prevenção de ameaças cibernéticas emergentes, especialmente no contexto brasileiro?

1.2 Justificativa

O desenvolvimento de uma abordagem integrada para detecção de tráfego malicioso, combinando análise de rede com monitoramento de redes sociais, justifica-se por múltiplas razões fundamentais:

Impacto econômico e social das ameaças cibernéticas:

O custo global de ciberataques atingiu US$ 8,44 trilhões em 2023, com projeção de crescimento para US$ 11,5 trilhões até 2026, segundo a Cybersecurity Ventures Ventures (2023). No Brasil, o relatório da Febraban FEBRABAN (2024) indica que instituições financeiras sofreram perdas de aproximadamente R$ 3,78 bilhões devido a fraudes digitais em 2023, um aumento de 29% em relação ao ano anterior. O aprimoramento dos mecanismos de detecção representa, portanto, não apenas um avanço técnico, mas uma necessidade econômica e social premente.

Necessidade de superar limitações das abordagens atuais: Conforme documentado por Ahmad et al. (2021), os sistemas de detecção baseados exclusivamente em assinaturas apresentam eficácia reduzida frente a ameaças emergentes, detectando apenas 37% dos ataques recém-desenvolvidos. Similarmente, estudos de Singh and Singh (2020) demonstrou que o aumento do tráfego criptografado (chegando a 95% do tráfego web em algumas redes) cria "pontos cegos"para os sistemas tradicionais de inspeção. Estas limitações exigem abordagens que possam complementar as técnicas existentes.

Potencial da análise integrada: A combinação de múltiplas fontes de dados - tráfego de rede e sinais em redes sociais - oferece oportunidades significativas para melhorar a detecção precoce de ameaças. Em Khandpur et al. (2017) foi demonstrado que a análise de conteúdo relacionado à segurança cibernética em redes sociais pode antecipar a detecção de ataques em até 4,5 dias. Esta capacidade de alerta antecipado é crucial para organizações implementarem medidas preventivas antes que ataques se materializem.

Demanda por soluções contextualizadas para o cenário brasileiro: O Brasil enfrenta desafios específicos em cibersegurança, com um aumento de 37,5% nos ataques direcionados a organizações brasileiras em 2023, segundo relatório da CERT.br (2024). O país é o quinto mais atacado globalmente, conforme dados da Fortinet (2023). No entanto, há escassez de pesquisas e soluções desenvolvidas especificamente para o contexto brasileiro, considerando suas particularidades técnicas, linguísticas e culturais.

Avanços tecnológicos viabilizadores: O amadurecimento recente de tecnologias como bancos de dados vetoriais (e.g., Qdrant, Milvus), frameworks de processamento distribuído (e.g., Dask, Spark) e técnicas avançadas de aprendizado de máquina criam condições favoráveis para implementação de soluções anteriormente dispendiosas. Estes avanços, conforme

discutido por Johnson et al. (2021a) e Chen and Guestrin (2016), permitem o processamento eficiente de grandes volumes de dados heterogêneos em tempo real.

Contribuição científica e prática: A abordagem proposta contribui para o avanço do conhecimento em detecção de intrusão, estabelecendo um framework metodológico para integração de fontes distintas de dados. Do ponto de vista prático, oferece direcionamentos para implementação de sistemas mais robustos e adaptáveis, capazes de enfrentar o cenário em constante evolução das ameaças cibernéticas.

Alinhamento com políticas nacionais de segurança cibernética: A pesquisa está alinhada com a Estratégia Nacional de Segurança Cibernética (E-Ciber) e o Decreto 10.222/2020, que estabelecem diretrizes para o desenvolvimento de capacidades nacionais em segurança cibernética, enfatizando a importância da pesquisa científica e desenvolvimento tecnológico nesta área.

Em síntese, a justificativa desta pesquisa fundamenta-se tanto em necessidades práticas prementes quanto em oportunidades científicas significativas, buscando contribuir para o desenvolvimento de soluções mais eficazes de proteção contra ameaças cibernéticas no contexto brasileiro.

1.3 Questões de Pesquisa

Para nortear o desenvolvimento desta pesquisa e operacionalizar os objetivos propostos, foram estabelecidas as seguintes questões de pesquisa:

Q1: Quais técnicas de vetorização são mais eficientes para representar características de tráfego de rede, preservando informações relevantes para detecção de atividades maliciosas, especialmente em tráfego criptografado?

Esta questão aborda diretamente o OE1 e relaciona-se com o problema P2, investigando formas eficazes de representar dados de rede em formato vetorial para análise subsequente, mesmo quando o payload está criptografado.

Q2: Como os diferentes algoritmos de aprendizado de máquina (Random Forest, KNN e XGBoost) se comparam em termos de desempenho, eficiência computacional e capacidade de generalização na detecção de tráfego malicioso em um ambiente de rede real?

Esta questão relaciona-se ao OE2 e ao problema P1, buscando identificar o algoritmo mais adequado para detecção precisa de ameaças, incluindo aquelas que podem não ter sido previamente identificadas nos dados de treinamento.

Q3: Em que medida o processamento paralelo com Dask e GPUs, combinado com armazenamento vetorial (Qdrant), pode viabilizar a análise em tempo real de grandes volumes de tráfego de rede sem comprometer a acurácia da detecção?

Esta questão aborda o OE3 e o problema P3, investigando se a arquitetura proposta para processamento e armazenamento consegue atender requisitos de latência e escalabilidade necessários para análise em tempo real.

Q4: Quais métricas e técnicas de análise de redes complexas são mais eficazes para identificar e caracterizar comunidades relacionadas a atividades hacktivistas em redes sociais?

Esta questão relaciona-se ao OE4, explorando metodologias para identificação de grupos e indivíduos potencialmente associados a atividades maliciosas em plataformas sociais.

Q5: Como podem ser correlacionados eventos e padrões identificados em redes sociais com características específicas de tráfego malicioso, estabelecendo mecanismos confiáveis de alerta precoce?

Esta questão aborda o OE6 e o problema P4, explorando métodos para integrar informações de ambas as fontes de dados para melhorar a capacidade de detecção antecipada de ameaças.

Q6: Em que medida a abordagem integrada proposta supera os métodos tradicionais de detecção em termos de tempo de antecipação, redução de falsos positivos e capacidade de identificação de ameaças emergentes?

Esta questão relaciona-se ao OE7, buscando quantificar objetivamente os ganhos da metodologia proposta em relação às abordagens convencionais.

Q7: Quais características específicas do cenário brasileiro de ameaças cibernéticas demandam adaptações na metodologia proposta e como essas adaptações afetam o desempenho do sistema?

Esta questão aborda o OE5, investigando particularidades do contexto brasileiro e suas implicações para a implementação eficaz da metodologia.

1.4 Hipóteses

Com base nas questões de pesquisa formuladas e na revisão da literatura realizada, foram definidas as seguintes hipóteses a serem testadas ao longo deste trabalho:

H1: A vetorização de características de fluxo de rede, combinada com técnicas de similaridade vetorial (como similaridade de cosseno), permite a detecção eficaz de tráfego malicioso mesmo em condições de tráfego criptografado, com precisão superior a 90%. A definição do limiar de 90% fundamenta-se na necessidade operacional de Sistemas de Detecção de Intrusão (NIDS) de minimizar falsos positivos que geram fadiga de alerta em analistas (Alert Fatigue). Enquanto métodos baseados puramente em assinaturas alcançam alta precisão para ameaças conhecidas, mas falham em zero-days, estudos como os de Vinayakumar et al. (2019a) demonstram que modelos de ML em fluxos de rede variam entre 85% e 98%. O limiar de 90%

estabelece, portanto, um patamar viável de segurança que supera métodos probabilísticos aleatórios (50%) e se aproxima da eficácia humana em análise de logs extensos.

Esta hipótese relaciona-se à Q1 e ao OE1, propondo que a representação vetorial de características de tráfego pode superar as limitações impostas pela criptografia, desde que sejam selecionadas as características e métricas de similaridade adequadas.

H2: Algoritmos baseados em vizinhança, como KNN, quando implementados com otimização para GPU, apresentam melhor equilíbrio entre precisão e desempenho computacional para detecção de tráfego malicioso em tempo real do que algoritmos ensemble, como Random Forest e XGBoost.

Esta hipótese relaciona-se à Q2 e ao OE2, propondo uma comparação específica entre os algoritmos selecionados no contexto da aplicação de detecção de tráfego malicioso.

H3: A implementação de processamento paralelo com Dask e banco de dados vetorial Qdrant permite a análise de fluxos de rede em tempo real, com latência média inferior a 100 ms por pacote, mantendo acurácia acima de 95% em hardware de consumo. O requisito de latência inferior a 100ms (sub-ms) baseia-se nas taxas de transmissão de redes modernas (10 Gbps+), onde pacotes podem chegar em intervalos de microssegundos. Para que um NIDS seja "inline"ou capaz de disparar alertas em tempo útil para bloqueio automatizado, o tempo de inferência não pode criar um gargalo. O valor de 100ms é frequentemente citado em SLAs (Service Level Agreements) de sistemas de segurança de baixa latência para redes financeiras, conforme apontado por requisitos de high-frequency trading e monitoramento crítico.

H4: A análise de métricas de centralidade e formação de comunidades em redes de interações no "X"(Twitter) permite identificar atores-chave em comunidades hacktivistas com precisão superior a 80%, quando validados contra dados da plataforma Zone-H.

Esta hipótese relaciona-se à Q4 e ao OE4, propondo que análises topológicas de redes de interação social podem revelar eficientemente estruturas de comunidades relacionadas a atividades hacktivistas.

H5: A integração de sinais obtidos em redes sociais com análise de tráfego de rede permite antecipar a detecção de ataques em pelo menos 24 horas em comparação com métodos baseados exclusivamente em análise de tráfego.[1]

Esta hipótese relaciona-se à Q5 e ao OE6, propondo que a abordagem integrada oferece benefícios mensuráveis em termos de antecipação de ameaças.

A integração de sinais obtidos em redes sociais com análise de tráfego de rede permite antecipar a detecção de ataques em pelo menos 24 horas em comparação com métodos baseados

[1] A validação experimental completa desta hipótese constitui trabalho futuro. A estimativa de 24 horas baseia-se em extrapolação da literatura (Khandpur et al., 2017; Hernandez et al., 2016).

exclusivamente em análise de tráfego.[2]

Esta hipótese relaciona-se à Q5 e ao OE6, propondo que a abordagem integrada oferece benefícios mensuráveis em termos de antecipação de ameaças.

H6:A abordagem integrada proposta reduz a taxa de falsos positivos em pelo menos 30% em comparação com sistemas tradicionais baseados apenas em assinaturas ou apenas em anomalias, mantendo ou melhorando a taxa de detecção verdadeira.[3]

Esta hipótese relaciona-se à Q6 e ao OE7, estabelecendo métricas para comparação objetiva com métodos tradicionais.

H7: Características específicas do cenário brasileiro, como padrões linguísticos e alvos preferenciais, quando incorporadas aos modelos, melhoram a precisão da detecção em pelo menos 15% para ameaças dirigidas a instituições nacionais.[4]

Esta hipótese relaciona-se à Q7 e ao OE5, propondo que a contextualização para o cenário brasileiro oferece benefícios mensuráveis no desempenho do sistema.

1.5 Objetivos

1.5.1 Objetivo Geral

Desenvolver e validar uma metodologia integrada para refinamento da detecção de tráfego malicioso, combinando técnicas avançadas de vetorização e aprendizado de máquina aplicadas a dados de tráfego de rede com análise de redes complexas para identificação proativa de atores de ameaça e monitoramento de comunidades hacktivistas em plataformas sociais, estabelecendo correlações entre sinais externos e padrões de tráfego para antecipação e mitigação de ameaças cibernéticas emergentes, com foco particular no contexto brasileiro.

1.5.2 Objetivos Específicos

Os objetivos específicos desempenham papel fundamental em uma pesquisa científica, pois operacionalizam o objetivo geral em componentes mensuráveis e executáveis. Como destacado por Creswell (2014) , objetivos específicos bem formulados proporcionam clareza metodológica e direcionamento preciso às etapas da investigação, facilitando tanto a execução quanto a avaliação dos resultados obtidos. No contexto desta pesquisa, os objetivos específicos delineados a seguir segmentam o desafio complexo da detecção integrada de ameaças cibernéticas em

[2]A validação experimental completa desta hipótese constitui trabalho futuro. A estimativa de 24 horas baseia-se em extrapolação da literatura (Khandpur et al., 2017; Hernandez et al., 2016).

[3]A quantificação experimental do ganho específico da integração requer validação detalhada na Seção 4.12.

[4]A contextualização foi implementada parcialmente (NLP em português, filtragem de domínios .br); a quantificação precisa da melhoria de 15% requer validação experimental futura.

componentes técnicos e procedimentais verificáveis, permitindo avanços incrementais e avaliação rigorosa de cada elemento da solução proposta. Esta abordagem estruturada, conforme recomendado por Hernandez et al. (2016), é particularmente relevante em campos multidisciplinares como a cibersegurança, onde a integração de diferentes domínios (análise de tráfego, processamento de linguagem natural, teoria de redes complexas) demanda delimitação clara de escopo e métodos para cada componente do sistema.

OE1: Implementar e avaliar técnicas de vetorização eficientes para representação de características de tráfego de rede e conteúdo de redes sociais, facilitando análises comparativas e detecção de padrões em tempo real.

OE2: Desenvolver e otimizar modelos de aprendizado de máquina (Random Forest, K-Nearest Neighbors e XGBoost) para classificação de tráfego de rede, avaliando seu desempenho em termos de precisão, recall, F1-score e capacidade de detecção de classes raras (tipos específicos de ameaças).

OE3: Construir e implementar um pipeline de processamento escalável utilizando tecnologias como Dask e GPU, combinado com banco de dados vetorial (Qdrant), para viabilizar análise em tempo real de grandes volumes de dados de rede.

OE4: Estabelecer uma metodologia para identificação de comunidades de interesse e atores de ameaça em redes sociais, aplicando técnicas de análise de redes complexas e processamento de linguagem natural em conteúdo relacionado a hacktivismo.

OE5: Validar a aplicabilidade da metodologia proposta no contexto brasileiro, utilizando conjuntos de dados de tráfego malicioso do repositório malware-traffic-analysis.net e perfis identificados na plataforma Zone-H, com foco em ameaças dirigidas a instituições no Brasil.

OE6: Desenvolver mecanismos para correlacionar informações obtidas da análise de redes sociais com padrões de tráfego malicioso, criando um sistema de alerta antecipado que integre ambas as fontes de dados.

OE7: Comparar quantitativamente o desempenho da abordagem integrada proposta com métodos tradicionais de detecção de intrusão, evidenciando ganhos em termos de tempo de detecção, redução de falsos positivos e capacidade de identificação de ameaças emergentes.

1.6 Estrutura da Dissertação

Esta dissertação está organizada em seis capítulos, estruturados de forma a apresentar progressivamente os fundamentos teóricos, a metodologia desenvolvida, os resultados obtidos e as conclusões derivadas da pesquisa.

O **Capítulo 1 - Introdução** contextualiza o tema, apresenta o problema de pesquisa, justifica sua relevância e define os objetivos, as questões de pesquisa e as hipóteses que orientam o trabalho.

O **Capítulo 2 - Fundamentação Teórica** estabelece as bases conceituais necessárias para a compreensão da pesquisa, abordando detalhadamente os conceitos fundamentais em detecção de intrusão de rede, aprendizado de máquina para segurança cibernética, vetorização de dados, análise de redes complexas, hacktivismo e processamento de linguagem natural, além de técnicas de processamento paralelo e distribuído. Também é apresentada uma revisão crítica da literatura, analisando pesquisas anteriores sobre detecção de *malware* via análise de tráfego de rede, detecção de atividades maliciosas em redes sociais e abordagens integradas para cibersegurança. Inclui ainda uma análise comparativa que posiciona a presente pesquisa no contexto da literatura existente.

O **Capítulo 3 - Metodologia** detalha a abordagem proposta, descrevendo os processos de coleta e processamento de dados, engenharia e extração de características, vetorização e normalização, construção do banco de dados vetorial, algoritmos de aprendizado de máquina implementados, análise de redes complexas e métricas de avaliação utilizadas.

O **Capítulo 4 - Implementação e Resultados** apresenta o ambiente experimental utilizado e os resultados obtidos na análise de tráfego de rede e das redes sociais, incluindo o desempenho dos modelos de aprendizado de máquina, a análise em tempo real, os resultados da clusterização, a análise de redes de menções e a identificação de atores de ameaça. Aborda ainda a integração das fontes de dados e discute criticamente os resultados obtidos.

O **Capítulo 5 - Conclusões e Trabalhos Futuros** sintetiza as principais contribuições da pesquisa, reconhece suas limitações e propõe direções para pesquisas futuras, reafirmando a importância da abordagem integrada para o aprimoramento dos sistemas de segurança cibernética.

Complementarmente, o trabalho inclui as **Referências** utilizadas e um **Apêndice** com detalhes técnicos de implementação que podem ser úteis para a reprodução dos experimentos ou aplicação da metodologia em outros contextos.

Capítulo 2

Fundamentação Teórica

2.1 Conceitos Fundamentais em Detecção de Intrusão de Rede

Um Sistema de Detecção de Intrusão de Rede (NIDS) monitora o tráfego de rede em busca de atividades maliciosas ou violações de políticas de segurança, sendo crucial para a proteção de ativos digitais (Ahmad et al., 2021).

2.1.1 Abordagens de Detecção

As estratégias de detecção de intrusão dividem-se em duas abordagens principais (Fernandes et al., 2022): baseada em assinaturas, que compara o tráfego com padrões de ataques conhecidos, oferecendo alta precisão para ameaças catalogadas, mas limitada contra ataques novos (Alemu and Boro, 2021; Ahmad et al., 2021); e baseada em anomalias, que identifica desvios do comportamento normal da rede, permitindo teoricamente detectar ataques desconhecidos (Chandola et al., 2009; Liu et al., 2021), embora possa gerar mais falsos positivos em redes dinâmicas (Shen et al., 2022). A presente pesquisa explora majoritariamente a detecção baseada em anomalias e aprendizado de máquina devido à natureza evolutiva das ameaças.

2.1.2 Níveis de Análise de Tráfego

A análise de tráfego para NIDS pode ocorrer em nível de pacote (exame detalhado de cabeçalhos e payloads, limitado por criptografia e escalabilidade (Alemu and Boro, 2021; Singh and Singh, 2020)), baseada em fluxo (agregação de metadados de comunicação, mais escalável e preservando a privacidade, porém com menos granularidade (Hofstede et al., 2014; Kent and Liebrock, 2021)), ou multi-granular, que combina níveis para uma visão completa (Wu et al.,

2025). Este trabalho foca na análise baseada em fluxo devido à sua aplicabilidade em tráfego criptografado e escalabilidade.

2.1.3 Formatos e Protocolos de Coleta de Dados de Rede

Os dados de rede são coletados em formatos como PCAP, que armazena detalhes completos dos pacotes e é essencial para análises forenses (Sanders, 2017), mas impõe desafios de armazenamento e privacidade. Alternativamente, NetFlow/IPFIX coleta estatísticas de fluxo, reduzindo o volume de dados (Hofstede et al., 2014; Cerroni et al., 2021), sendo este último mais alinhado com as características extraídas neste trabalho. Ferramentas como Zeek geram logs estruturados com análise de protocolos aplicacionais (Ring et al., 2019; Cowger et al., 2022), oferecendo um nível de detalhe intermediário.

2.1.4 Desafios Contemporâneos na Detecção de Intrusão

A detecção de intrusão enfrenta desafios, como a crescente **criptografia** do tráfego, que limita a inspeção de conteúdo (Anderson, 2020) e impulsiona técnicas baseadas em características de fluxo (Hancock and Khoshgoftaar, 2020); **ataques sofisticados e evasivos** (Dutta et al., 2022); a necessidade de **escalabilidade** para lidar com o enorme volume de tráfego (Srinivasan et al., 2023); e a dificuldade na obtenção de **dados rotulados** de alta qualidade para treinamento de modelos de ML (Ring et al., 2019). Esta pesquisa visa abordar os desafios de analise de tráfegos criptografados e escalabilidade através da vetorização de fluxos e processamento paralelo.

Seleção do Dataset para esta Pesquisa

Embora datasets históricos como KDD Cup 99 e NSL-KDD sejam considerados obsoletos (Kim et al., 2021), conjuntos mais recentes como CIC-IDS (2017, 2018) (Ring et al., 2019; Chouchani and Abed, 2020) e UNSW-NB15 (Kim et al., 2021) oferecem dados mais representativos.

Porém, considerando as limitações apresentadas na literatura, este trabalho opta por uma abordagem focada no realismo. Enquanto datasets como KDD Cup 99 são obsoletos e o CIC-IDS apresenta ataques sintéticos que podem gerar padrões artificiais de fluxo, a utilização de PCAPs brutos provenientes do *Malware-traffic-analysis.net* busca fidelidade às táticas, técnicas e procedimentos (TTPs) de ameaças ativas, crucial para análises realistas (Sanders, 2017). Esta escolha visa mitigar o problema de generalização frequentemente observado quando modelos treinados em ambientes simulados falham ao enfrentar tráfego real.

2.1.5 Tendências Recentes em Detecção de Intrusão

As tendências atuais incluem a transição para **análise comportamental** (Verma et al., 2022), a integração de **inteligência de ameaças** (threat intelligence) para contextualizar alertas (Tounsi and Rais, 2018), e o uso intensivo de **Aprendizado de Máquina (ML) e Aprendizado Profundo (DL)** (Vinayakumar et al., 2019b). Esta pesquisa se alinha a estas tendências ao empregar ML para análise comportamental baseada em fluxos e ao integrar dados de redes sociais como forma de inteligência de ameaças. A convergência para sistemas híbridos é uma direção promissora (Fernandes et al., 2022).

2.2 Aprendizado de Máquina para Segurança Cibernética

O aprendizado de máquina (ML) é fundamental para os sistemas de segurança cibernética modernos, oferecendo capacidades adaptativas e preditivas (Buczak and Guven, 2016).

2.2.1 Fundamentos e Categorização

As abordagens de ML em cibersegurança incluem o aprendizado supervisionado (treinamento com dados rotulados para classificação, e.g., legítimo vs. malicioso (Liu et al., 2021), dependente de dados rotulados de qualidade (Ahmad et al., 2021)); aprendizado não-supervisionado (identificação de padrões ou anomalias em dados não rotulados, útil para novos ataques (Chandola et al., 2009), mas com desafios na parametrização (Shen et al., 2022)); e aprendizado semi-supervisionado (combinação de dados rotulados e não rotulados (Chapelle et al., 2010; Chen et al., 2021)). Este trabalho foca no aprendizado supervisionado para classificação de tráfego.

2.2.2 Algoritmos Relevantes para Detecção de Tráfego Malicioso

Para a detecção de tráfego malicioso, algoritmos como Random Forest (RF), K-Nearest Neighbors (KNN) e XGBoost são proeminentes.

- **Random Forest (RF):** Um método *ensemble* que constrói múltiplas árvores de decisão e produz a classe que é a moda das classes das árvores individuais. Breiman (2001) estabeleceu os fundamentos deste algoritmo, que tem demonstrado consistentemente alto desempenho em tarefas de classificação de segurança.

Matematicamente, a construção das árvores baseia-se na maximização do ganho de informação ou na minimização da impureza em cada nó. A medida de Impureza de Gini (I_G), frequentemente utilizada para classificação, é calculada para um nó t com C classes e probabilidades p_i de pertencer à classe i como:

$$I_G(t) = 1 - \sum_{i=1}^{C} p(i|t)^2 \tag{2.1}$$

Vinayakumar et al. (2019b) relatam acurácias superiores a 98% na detecção de tráfego malicioso usando RF, atribuindo este desempenho à sua capacidade intrínseca de lidar com conjuntos de características de alta dimensionalidade e capturar interações complexas não-lineares.

- **K-Nearest Neighbors (KNN):** Um método baseado em instância que classifica um ponto com base nas classes de seus vizinhos mais próximos no espaço de características. Como explicado por Cover and Hart (1967), o KNN é um algoritmo "preguiçoso"(*lazy learner*) que não constrói um modelo explícito durante o treinamento, mas armazena as instâncias para comparação em tempo de execução.

  A classificação depende fundamentalmente da métrica de distância utilizada. Para vetores de características contínuas x e y em um espaço n-dimensional, a Distância Euclidiana é a métrica mais comum:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.2}$$

  Em contextos de alta dimensionalidade, métricas alternativas como a distância de Manhattan ou Cosseno podem ser preferíveis. Syriopoulos et al. (2023) observam que, apesar de sua simplicidade, KNN permanece competitivo em tarefas de classificação de tráfego de rede quando otimizado para hardware GPU (como via bibliotecas RAPIDS/cuML), superando a barreira computacional tradicional de calcular distâncias par-a-par em grandes datasets.

- **XGBoost:** O XGBoost é uma implementação otimizada de *gradient boosting* projetada para ser altamente eficiente, flexível e portável. Chen and Guestrin (2016) introduziram este algoritmo destacando sua capacidade de escalabilidade e o tratamento inovador de dados esparsos.

Diferente do *Gradient Boosting* tradicional, o XGBoost utiliza uma função objetivo regularizada para controlar a complexidade do modelo e evitar *overfitting*. Formalmente, para um conjunto de K árvores, a predição para uma instância i é dada por:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \tag{2.3}$$

Onde $\mathcal{F}$ é o espaço de funções das árvores de regressão. O algoritmo minimiza a seguinte função objetivo regularizada $\mathcal{L}(\phi)$:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{2.4}$$

Aqui, l é a função de perda diferenciável (que mede a diferença entre a predição $\hat{y}_i$ e o alvo y_i) e Ω é o termo de regularização que penaliza a complexidade do modelo, definido como:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{2.5}$$

Onde T é o número de folhas da árvore e w são os pesos das folhas. Alemu and Boro (2021) demonstram que essa regularização integrada, combinada com o uso de aproximações de segunda ordem (expansão de Taylor) para otimização da função de perda, torna o XGBoost superior em cenários de segurança com dados desbalanceados e ruidosos.

Outras técnicas como SVM (Cortes and Vapnik, 1995) e Redes Neurais/Deep Learning (CNNs, LSTMs (Kim et al., 2021)) também são aplicadas, mas os três primeiros formam o núcleo dos modelos implementados neste trabalho devido ao seu equilíbrio entre desempenho, interpretabilidade (RF, XGBoost) e adequação à vetorização (KNN).

2.2.3 Extração e Seleção de Características em ML

A qualidade das características de entrada é crucial para ML (Mehrban et al., 2022). A **Engenharia de Características** transforma dados brutos em representações informativas. Para tráfego de rede, Moore et al. (2005) e ferramentas como CICFlowMeter (Sharafaldin et al., 2018) definiram conjuntos de características. A **Seleção de Características** identifica o subconjunto mais relevante, melhorando eficiência e desempenho, podendo reduzir significativamente a dimensionalidade (Alemu and Boro, 2021). Ambas são etapas fundamentais na metodologia aqui proposta.

2.2.4 Desafios Específicos do ML em Segurança

Aplicações de ML em segurança enfrentam desafios como:

- **Desbalanceamento de Classes:** Eventos maliciosos são raros, enviesando modelos. Técnicas como SMOTE (Chawla et al., 2002) e abordagens específicas de amostragem ou custo (Qazi and Aung, 2023) são cruciais, e foram consideradas nesta pesquisa.
- **Concept Drift:** Mudanças na distribuição dos dados ao longo do tempo devido à evolução das táticas de ataque (Gama et al., 2014; Jordaney et al., 2017). Modelos adaptativos ou retreinamento periódico são necessários.
- **Ataques Adversariais:** Manipulação de entradas para enganar modelos de ML (Biggio and Roli, 2018).
- **Interpretabilidade vs. Desempenho:** Modelos complexos podem ter melhor desempenho, mas menor explicabilidade, um desafio para análise forense (Gunning and Aha, 2019; Shen et al., 2022).

2.2.5 Validação e Avaliação de Modelos de ML

A avaliação rigorosa em segurança requer métricas como F1-Score (equilíbrio precisão-recall (Sokolova and Lapalme, 2009)), AUC-ROC, e métricas específicas para classes minoritárias (Alemu and Boro, 2021). Validação temporal, que respeita a ordem cronológica dos dados, é preferível à validação cruzada tradicional (Campos et al., 2021). Avaliação em múltiplos datasets de diferentes ambientes (validação cruzada entre datasets) também é importante para generalização (Moustafa et al., 2021).

2.2.6 Frameworks e Bibliotecas de ML Utilizados

Para a implementação, foram utilizadas bibliotecas consolidadas:

- **Scikit-learn:** Para algoritmos de ML tradicionais e pré-processamento (Pedregosa et al., 2011).
- **RAPIDS e cuML:** Para aceleração de ML em GPU, especialmente KNN e RF (Syriopoulos et al., 2023).
- **Dask:** Para computação paralela e distribuída, escalando o processamento e otimização de hiperparâmetros (Rocklin, 2015; Dask Development Team, 2024).

- **XGBoost:** Biblioteca otimizada para gradient boosting com suporte a GPU (Chen and Guestrin, 2016).

TensorFlow (Abadi et al., 2016) e PyTorch (Paszke et al., 2019) são referências para Deep Learning, mas não o foco principal dos classificadores escolhidos aqui.

2.2.7 Tendências Recentes e Direções Futuras em ML para Segurança

O campo evolui com aprendizado por reforço para respostas automatizadas (Nguyen and Reddi, 2021), aprendizado federado para privacidade (McMahan et al., 2017), uso de Transformers para análise de tráfego (Wu et al., 2025), e sistemas cognitivos integrados (Gonzalez et al., 2020). A pesquisa atual se concentra em aplicar robustamente técnicas estabelecidas e otimizá-las para o contexto de segurança.

2.3 Vetorização de Dados

A vetorização transforma informações diversas em representações numéricas processáveis por algoritmos de ML, sendo crucial para quantificar similaridade e aplicar otimizações (Aggarwal, 2015). Uma representação vetorial eficaz captura similaridades semânticas relevantes, possui dimensionalidade gerenciável e preserva estruturas importantes dos dados originais (Mikolov et al., 2013; Johnson et al., 2021b).

2.3.1 Técnicas de Vetorização Aplicadas

Vetorização de Dados Estruturados (Tráfego de Rede)

Para dados de fluxo de rede, o processo envolveu normalização e escalonamento (Min-Max, Z-score, Robust Scaling (Alemu and Boro, 2021)) para lidar com diferentes escalas e distribuições; codificação de características categóricas (One-Hot Encoding para protocolos, Target Encoding para portas de alta cardinalidade, e Embeddings Categóricos para flags TCP (Wu et al., 2025)); e, implicitamente, seleção de características na fase de engenharia. A binarização e tokenização, inspiradas no T-Matrix (Wu et al., 2025), são efetivas para criar um vocabulário unificado.

Codificação de Características Categóricas (One-Hot Encoding): A maioria dos algoritmos de aprendizado de máquina opera exclusivamente com dados numéricos, exigindo

a transformação de variáveis categóricas (como protocolos `TCP, UDP` ou flags de serviço) em representações vetoriais.

Nesta pesquisa, adota-se a técnica de *One-Hot Encoding*. Formalmente, seja uma variável categórica V com um domínio de cardinalidade K, denotado por $\mathcal{D} = \{c_1, c_2, ..., c_K\}$. A transformação mapeia cada categoria c_i para um vetor binário $v \in \{0,1\}^K$, onde apenas a posição correspondente ao índice i recebe o valor 1, e todas as outras recebem 0:

$$OneHot(c_i) = [0, 0, ..., \underbrace{1}_{i-\text{ésima posição}}, ..., 0] \tag{2.6}$$

A justificativa teórica para o uso de *One-Hot Encoding* em detrimento do *Label Encoding* (atribuição de inteiros sequenciais $1, 2, ..., K$) reside na eliminação de relações ordinais espúrias. Hancock and Khoshgoftaar (2020) argumentam que atribuir valores numéricos sequenciais a protocolos (ex: UDP=1, TCP=2) induziria o modelo a interpretar uma relação de magnitude ($TCP > UDP$) que não existe na realidade. O *One-Hot Encoding* trata cada categoria como ortogonal e equidistante das demais, preservando a semântica correta dos dados de rede, embora aumente a dimensionalidade e a esparsidade do vetor resultante.

Vetorização de Dados Não-Estruturados (Redes Sociais)

Para texto de postagens sociais, utilizou-se TF-IDF ajustado para o vocabulário de cibersegurança (Hernandez et al., 2016) e *Word Embeddings* (Word2Vec (Mikolov et al., 2013)) com fine-tuning para o domínio. Os *Sentence Embeddings* foram gerados por agregação ponderada de word embeddings. Para características de interação, *Graph Embeddings* (node2vec (Grover and Leskovec, 2016)) foram considerados para capturar estruturas relacionais.

2.3.2 *Embeddings* e Similaridade em Espaços Vetoriais

Embeddings são representações vetoriais densas aprendidas que capturam estruturas latentes (Bengio et al., 2013), a capacidade de posicionar entidades semanticamente similares próximas no espaço vetorial é fundamental. A similaridade entre vetores é quantificada principalmente pela **Similaridade de Cosseno**, adequada para alta dimensionalidade (Huang, 2008). Dado o volume de vetores, a busca eficiente por vizinhos mais próximos é realizada utilizando técnicas de Approximate Nearest Neighbors (ANN), como HNSW, implementada em bancos de dados vetoriais como Qdrant (Ponomareva et al., 2021; Qdrant, 2023), que é central na arquitetura proposta.

2.3.3 Aplicações e Desafios da Vetorização em Segurança Cibernética

A vetorização é aplicada na detecção de *malware* (Raff et al., 2018), análise de tráfego (Wu et al., 2025), análise de logs (Liu et al., 2019), e análise de ameaças em mídias sociais (Khandpur et al., 2017). Desafios incluem interpretabilidade (Gunning and Aha, 2019), adaptabilidade temporal (concept drift (Gama et al., 2014)), e robustez a ataques adversariais (Biggio and Roli, 2018), sendo a interpretabilidade e adaptabilidade considerações importantes para os modelos desenvolvidos.

2.4 Análise de Redes Complexas

A análise de redes complexas, baseada na teoria de grafos, estuda sistemas de entidades interconectadas (Newman, 2010), sendo aplicada aqui para investigar interações em redes sociais relacionadas a atividades hacktivistas.

2.4.1 Fundamentos e Métricas Utilizadas

As redes de interação social (menções no "X") são modeladas como grafos direcionados e ponderados (Wasserman and Faust, 1994). Foram calculadas métricas de centralidade para identificar usuários influentes: Grau (atividade local), Intermediação (papel de "ponte"), Proximidade (eficiência na disseminação) e Autovetor/PageRank (influência considerando conexões importantes) (Freeman, 1979; Bonacich, 1987). Para análise estrutural, a detecção de comunidades foi realizada utilizando o método de Louvain (Blondel et al., 2008a), que otimiza a modularidade da rede (Newman and Girvan, 2004), buscando grupos com interações internas densas.

2.4.2 Análise de Redes Sociais (ARS) e Aplicações em Hacktivismo

A ARS é aplicada para entender estruturas e dinâmicas sociais (Wasserman and Faust, 1994). No contexto do hacktivismo, esta análise ajuda a identificar estruturas organizacionais, influenciadores e evolução de comunidades (Benjamin and Chen, 2012; Coleman, 2014). As redes de menção no "X"são particularmente úteis para mapear fluxos de informação e hierarquias de influência (Maharani et al., 2018). A metodologia proposta utiliza estas técnicas para identificar e caracterizar comunidades e atores relevantes em discussões sobre hacktivismo.

2.5 Hacktivismo e suas Manifestações em Redes Sociais

Hacktivismo, a interseção de ativismo político e *hacking* (Samuel, 2004; Romagna, 2020), manifesta-se online de diversas formas. A compreensão de suas táticas, organização e comunicação é vital para a detecção de ameaças.

2.5.1 Táticas, Organização e Comunicação do Hacktivismo

As táticas hacktivistas variam de disruptivas *DDoS, defacement* (Sauter, 2014) a extrativas *doxing, data dumps* (Coleman, 2014) e informativas desenvolvimento de ferramentas (Coleman, 2008). A organização é frequentemente descentralizada, com coordenação emergente baseada em meritocracia técnica (Coleman, 2014; Himanen, 2001). Plataformas como IRC, *imageboards* e, crucialmente para este estudo, redes sociais como "X", são usadas para recrutamento, planejamento e amplificação (Olson, 2012; Milner, 2013). Padrões comunicativos incluem jargão específico, comunicação codificada e uso de memes (Coleman, 2014; Benjamin and Chen, 2015).

2.5.2 Detecção e Antecipação de Ameaças via Redes Sociais

Redes sociais como o "X"servem para recrutamento, amplificação e compartilhamento de alvos (Olson, 2012; Le Sceller et al., 2017). Khandpur et al. (2017) identificaram indicadores de alerta precoce em mídias sociais, como aumento no volume de comunicação com terminologia específica, convergência temática e mudanças de sentimento, que podem preceder ataques. A metodologia desta dissertação busca capturar esses sinais para correlacioná-los com atividades de rede. Questões éticas e legais sobre monitoramento e privacidade são considerações importantes (Lyon, 2014; Bellaby, 2021).

2.6 Processamento de Linguagem Natural em Cibersegurança

O Processamento de Linguagem Natural (PLN) capacita computadores a analisar e compreender linguagem humana, sendo cada vez mais aplicado em cibersegurança para análise de conteúdo textual relacionado a ameaças (Manning et al., 2008).

2.6.1 Vetorização de Texto e Análise Aplicada

Para a análise do conteúdo das postagens de redes sociais, são empregadas técnicas de vetorização de texto como TF-IDF (Salton and McGill, 1988) e Word Embeddings (e.g., Word2Vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2017)), que transformam texto em representações numéricas capturando significado semântico. Modelos contextuais como BERT (Devlin et al., 2019) oferecem representações ainda mais ricas. A análise de sentimento, utilizando léxicos como VADER (Hutto and Gilbert, 2014) ou modelos de ML (Zhang et al., 2018), é aplicada para determinar a polaridade afetiva, relevante para identificar intenções (Hernandez et al., 2016). A extração de entidades nomeadas (NER) específicas de cibersegurança (CVEs, malware (Le Sceller et al., 2017)) e a classificação de texto são usadas para identificar conteúdo malicioso ou discussões sobre vulnerabilidades.

2.6.2 Desafios do PLN em Cibersegurança

A aplicação de PLN em cibersegurança enfrenta desafios como jargão técnico, gírias (Coleman, 2014), multilinguismo (relevante para o contexto brasileiro (Hernandez et al., 2016)), comunicação evasiva (Benjamin and Chen, 2015), e a rápida evolução da linguagem e das ameaças. A adaptação de domínio (*domain adaptation*) (Ruder et al., 2018) e o aprendizado semi-supervisionado (Chen et al., 2021) são abordagens promissoras para mitigar a escassez de dados rotulados específicos.

2.7 Processamento Paralelo e Distribuído para Análise de Dados

A análise de grandes volumes de dados em cibersegurança exige arquiteturas de processamento paralelo e distribuído para desempenho em tempo hábil.

2.7.1 Fundamentos e Frameworks Aplicados

A computação paralela (múltiplos núcleos/GPUs) e distribuída (múltiplos sistemas) é essencial (Barney, 2021). Para esta pesquisa, **Dask** (Rocklin, 2015; Team, 2024) foi escolhido para paralelizar tarefas de extração de características e otimização de hiperparâmetros em Python, devido à sua integração com o ecossistema científico. A **aceleração por GPU**, com bibliotecas como RAPIDS/cuML (RAPIDS Development Team, 2023), é fundamental para algoritmos de ML como KNN e RF, permitindo processamento de grandes volumes de dados com baixa latência (Syriopoulos et al., 2023). A arquitetura de processamento visa

viabilizar a análise em tempo real, lidando com alta taxa de ingestão e mantendo estado quando necessário (Casas et al., 2019).

2.7.2 Bancos de Dados Vetoriais para Análise em Larga Escala

Para armazenar e consultar eficientemente os vetores de alta dimensionalidade gerados, bancos de dados vetoriais são cruciais. O **Qdrant** (Qdrant, 2023) foi selecionado por seu suporte a busca por similaridade com ANN (e.g., HNSW), filtros complexos e escalabilidade (Johnson et al., 2021b; Ponomareva et al., 2021). Esta escolha é vital para a fase de detecção em tempo real da metodologia, onde novos vetores de tráfego são comparados rapidamente com um grande corpus de vetores conhecidos.

2.8 Trabalhos Relacionados

Esta seção apresenta uma revisão crítica da literatura relacionada à detecção de tráfego malicioso e identificação de atividades suspeitas em redes sociais, com enfoque nas metodologias, limitações e contribuições mais relevantes para a abordagem proposta nesta dissertação.

2.8.1 Detecção de Malware via Análise de Tráfego de Rede

A pesquisa sobre detecção de malware através da análise de tráfego de rede tem se desenvolvido significativamente nos últimos anos, incorporando técnicas cada vez mais sofisticadas de modelagem e análise.

Abordagens Baseadas em Características de Fluxo

Um conjunto significativo de pesquisas tem se concentrado na utilização de características extraídas de fluxos de rede para a identificação de comunicações maliciosas.

Sharafaldin et al. (2018) apresentaram o dataset CSE-CIC-IDS2018, contendo tráfego benigno realista e simulações de ataques modernos. Um aspecto central deste trabalho foi a introdução do CICFlowMeter, uma ferramenta que extrai mais de 80 características estatísticas de fluxos de rede. Os autores avaliaram diversos algoritmos de ML, reportando acurácias superiores a 96% para a maioria dos ataques simulados. No entanto, como observado por Ring et al. (2019), uma limitação importante é a natureza sintética dos ataques, que podem não capturar completamente a complexidade e sutileza de ataques reais.

Subsequentemente, Sarhan et al. (2021) propuseram uma metodologia para converter dados do CSE-CIC-IDS2018 para o formato NetFlow, criando o dataset NF-CSE-CIC-IDS2018-v2 com 43 características. Este trabalho é particularmente relevante por demonstrar como

características de NetFlow, mais compactas e amplamente disponíveis em infraestruturas reais, podem ser utilizadas efetivamente para detecção. Os autores reportaram uma redução menor que 3% na acurácia de detecção em comparação com as 80+ características do CICFlowMeter original. Uma limitação notável, porém, é a perda de detalhes granulares na conversão para NetFlow, particularmente relevantes para alguns tipos específicos de ataques.

Moustafa et al. (2021) conduziram um estudo comparativo abrangente entre diferentes conjuntos de características para detecção de intrusão, incluindo NetFlow e CICFlowMeter, aplicados a múltiplos datasets (CSE-CIC-IDS2018, BoT-IoT, ToN-IoT). Um resultado particularmente relevante foi a observação de que características derivadas de NetFlow demonstraram melhor generalização entre diferentes datasets, sugerindo robustez a variações no ambiente de rede. Entretanto, os autores não exploraram otimizações específicas dos conjuntos de características para diferentes tipos de ataques, o que poderia potencialmente melhorar ainda mais os resultados.

Luay et al. (2025) destacaram a importância particular de características temporais em datasets NetFlow, propondo extensões ao NF-UQ-NIDS com inclusão de medidas mais detalhadas de timing entre pacotes. Seus experimentos demonstraram melhorias significativas na detecção de ataques com padrões temporais distintos, como C&C de botnets e ataques de canal lateral. Uma contribuição notável deste trabalho foi a demonstração de que características temporais, mesmo sem acesso ao payload, podem ser altamente discriminativas para certos tipos de ameaças.

Técnicas de Aprendizado de Máquina Aplicadas

A evolução das técnicas de aprendizado de máquina tem proporcionado avanços significativos na eficácia de sistemas de detecção baseados em análise de tráfego.

Kim et al. (2021) foi proposto uma arquitetura híbrida combinando Redes Neurais Convolucionais (CNN) e Long Short-Term Memory (LSTM) para detecção de intrusões. As CNNs foram aplicadas para capturar características espaciais dos fluxos, enquanto as LSTMs modelavam dependências temporais. Aplicada aos datasets CICIDS2017, UNSW-NB15 e WSN-DS, esta abordagem superou modelos individuais e abordagens tradicionais, alcançando F1-scores superiores a 0.98. No entanto, os autores reconhecem a complexidade computacional do modelo proposto, que pode limitar sua aplicabilidade em análise em tempo real.

Khan et al. (2020) introduziram uma abordagem híbrida utilizando Spark ML para detecção de anomalias e Conv-LSTM para classificação de tipos específicos de ataques. Esta arquitetura em duas fases demonstrou-se eficiente para processamento em larga escala e identificação precisa de ataques. O trabalho destaca-se pela integração de técnicas de big

data com aprendizado profundo, embora a dependência de uma infraestrutura Spark possa representar uma barreira de adoção em ambientes com recursos limitados.

Vinayakumar et al. (2019b) apresentam um estudo comparativo abrangente de técnicas de aprendizado profundo para detecção de intrusões, incluindo DNNs, CNNs, RNNs e LSTM. Aplicadas a múltiplos datasets (KDD Cup 99, NSL-KDD, UNSW-NB15, WSN-DS e CICIDS2017), as redes LSTM produziram os melhores resultados gerais, particularmente em ataques com características sequenciais. Uma contribuição metodológica importante foi a demonstração de que arquiteturas baseadas em atenção podem melhorar significativamente o desempenho em datasets desbalanceados, embora os autores não tenham explorado completamente a interpretabilidade destes modelos.

Adli (2023) avaliou diversos algoritmos de ML e DL (Random Forest, XGBoost, KNN, SVM) especificamente para dados NetFlow, utilizando o dataset NF-UQ-NIDS. O Extra-Trees, uma variante do Random Forest, demonstrou desempenho superior, com acurácia de 98.3% e F1-score de 0.984. O trabalho destaca-se pela análise específica de características NetFlow e comparação sistemática de modelos, embora a avaliação tenha sido limitada a um único dataset.

Vetorização e Similaridade

Recentemente, abordagens baseadas em vetorização e métricas de similaridade têm ganhado atenção significativa, particularmente por sua capacidade de lidar com dados de alta dimensionalidade e potencial para detecção em tempo real.

Wu et al. (2025) propuseram o UniNet, um framework para modelagem de tráfego multi-granular utilizando uma representação vetorial denominada T-Matrix, que captura informações em níveis de pacote, fluxo e sessão. Associada ao modelo T-Attent baseado em Transformers, esta abordagem demonstrou desempenho superior em várias tarefas, incluindo detecção de intrusões e classificação de tráfego. Uma contribuição metodológica fundamental é a representação unificada que preserva informações em múltiplos níveis de granularidade, embora os autores reconheçam a complexidade adicional introduzida pela representação multi-nível.

Huang (2008) explorou abordagens baseadas em similaridade de cosseno para agrupamento e detecção de anomalias em dados textuais, estabelecendo fundamentos metodológicos relevantes para aplicações em análise de tráfego. Embora não focado especificamente em segurança de rede, este trabalho introduziu técnicas de cálculo eficiente de similaridade vetorial posteriormente adaptadas para contextos de segurança.

Angiulli et al. (2023) propuseram uma abordagem de detecção de anomalias baseada em autoencoders, utilizando erros de reconstrução como medida de "anormalidade". Aplicada a diversos domínios, incluindo segurança de rede, esta técnica demonstrou robustez mesmo com

poucos exemplos anômalos disponíveis para treinamento. A principal limitação observada foi a necessidade de ajuste cuidadoso de parâmetros para diferentes tipos de dados.

Qureshi et al. (2023) introduziram o VBQ-Net, um modelo de rede neural baseado em vetorização para maximizar a segurança em sistemas IoT. O modelo utiliza técnicas de quantização e boost para reduzir o tamanho dos modelos sem comprometer significativamente o desempenho, um aspecto particularmente relevante para implementações em dispositivos com recursos limitados.

Processamento em Tempo Real

O processamento eficiente de grandes volumes de dados em tempo real representa um desafio fundamental para sistemas de detecção baseados em tráfego.

Bartos and Zet (2012) discutiram desafios específicos da detecção de anomalias em tempo real e propuseram uma biblioteca de algoritmos otimizados para este cenário. O trabalho identifica trade-offs entre complexidade computacional e eficácia de detecção, propondo soluções adaptativas que ajustam dinamicamente o nível de análise baseado em heurísticas de risco.

Casas et al. (2019) introduziram o Stream4Flow, um framework para detecção de padrões em tempo real em dados IPFIX, integrando IPFIXcol, Kafka e Spark. Avaliações empíricas demonstraram a capacidade do sistema de processar milhões de fluxos por minuto com latência aceitável, embora os autores reconheçam limitações no número e complexidade dos algoritmos de detecção que podem ser executados sem comprometer o desempenho em tempo real.

Akanbi and Masinde (2020) desenvolveram um middleware de processamento distribuído para análise em tempo real de dados heterogêneos em plataformas de big data, com aplicação em monitoramento ambiental. Embora não focado especificamente em segurança cibernética, o trabalho apresenta arquiteturas e algoritmos relevantes para processamento de fluxos contínuos de dados, um cenário diretamente aplicável à análise de tráfego de rede.

Bancos de Dados Vetoriais e Busca por Similaridade

O uso de bancos de dados especializados para armazenamento e recuperação eficiente de vetores representa uma tendência emergente em sistemas de segurança baseados em análise de tráfego.

Johnson et al. (2021a) apresentaram técnicas para busca de similaridade em escala de bilhões de vetores utilizando GPUs, estabelecendo fundamentos algorítmicos para sistemas

como o Qdrant. Os autores demonstraram acelerações de até 8.5x em comparação com implementações CPU, com escalabilidade quase linear em relação ao tamanho do dataset.

O whitepaper técnico do Qdrant (2023) detalha a arquitetura e otimizações deste banco de dados vetorial, com ênfase particular em índices de aproximação de vizinhos mais próximos (HNSW, ANNOY, etc.) e estratégias de particionamento para escalabilidade horizontal. Uma contribuição notável é a discussão sobre filtros complexos combinados com busca vetorial, um aspecto particularmente relevante para aplicações de segurança onde múltiplos critérios podem ser necessários para identificação precisa de ameaças.

Ponomareva et al. (2021) conduziram uma análise comparativa de sistemas de busca vetorial, incluindo FAISS, ScaNN, Milvus e Qdrant, avaliando métricas como recall, latência e throughput. Os resultados destacam diferentes pontos fortes, com o Qdrant demonstrando bom equilíbrio entre precisão e desempenho em cenários com filtros complexos, embora com overhead maior em comparação com bibliotecas mais focadas como FAISS.

2.8.2 Detecção de Atividades Maliciosas em Redes Sociais

A pesquisa sobre detecção de atividades maliciosas em redes sociais representa um campo complementar à análise de tráfego de rede, focando na identificação de ameaças em estágios iniciais através do monitoramento de plataformas online.

Caracterização e Análise do Hacktivismo

A compreensão das características sociológicas, motivacionais e comportamentais do hacktivismo fundamenta o desenvolvimento de métodos eficazes para sua detecção e análise.

Coleman (2014), em seu trabalho seminal "Hacker, Hoaxer, Whistleblower, Spy: The Many Faces of Anonymous", fornece uma análise etnográfica aprofundada da comunidade hacktivista, particularmente do grupo Anonymous. O estudo revela a natureza complexa e multifacetada destas comunidades, caracterizadas por princípios como liberdade de informação, desconfiança da autoridade e defesa da descentralização. Esta análise qualitativa estabelece um framework conceitual essencial para entender as motivações e comportamentos subjacentes a atividades hacktivistas, embora não forneça diretamente métodos quantitativos para sua detecção.

Himanen (2001) examina a "ética hacker"como um fenômeno cultural e social, destacando valores como paixão, liberdade, valor social e criatividade. O autor identifica características de mérito e competição nestas comunidades, onde a demonstração de habilidades técnicas e sociais é altamente valorizada. Este trabalho contribui significativamente para a compreensão dos aspectos culturais que motivam e sustentam comunidades hacktivistas, um conhecimento

fundamental para desenvolver sistemas de detecção que considerem não apenas aspectos técnicos, mas também sociais e culturais.

Romagna (2020) apresenta uma conceptualização, técnicas e visão histórica do hacktivismo, estabelecendo uma taxonomia de atividades baseada em motivações e métodos. Esta categorização sistemática é particularmente valiosa para o desenvolvimento de sistemas de detecção específicos para diferentes tipos de atividades hacktivistas, embora o autor reconheça que as fronteiras entre categorias são frequentemente fluídas e que motivações podem evoluir ao longo do tempo.

Detecção Baseada em Conteúdo em Redes Sociais

Um conjunto significativo de pesquisas tem se concentrado na análise do conteúdo postado em plataformas sociais para identificação de atividades potencialmente maliciosas.

Le Sceller et al. (2017) introduziram o SONAR, um sistema para detecção automática de eventos de segurança cibernética no Twitter em tempo real. O sistema combina técnicas de processamento de linguagem natural com análise de contexto social para identificar discussões relacionadas a vulnerabilidades, exploits e ataques. Avaliações empíricas demonstraram a capacidade do sistema de detectar eventos significativos de segurança mais rapidamente que fontes tradicionais, embora os autores identifiquem limitações relacionadas a falsos positivos derivados de ambiguidades linguísticas.

Khandpur et al. (2017) propuseram uma abordagem para detecção de ataques cibernéticos usando mídias sociais como fonte de dados. O sistema utiliza técnicas de expansão dinâmica de consultas para monitorar efetivamente menções a ataques emergentes. Uma contribuição metodológica importante foi a demonstração de que sinais extraídos de mídias sociais podem antecipar eventos de segurança em até 4,5 dias em relação aos sistemas tradicionais de detecção. No entanto, os autores reconhecem desafios relacionados à separação entre discussões legítimas sobre segurança e coordenação real de ataques.

Hernandez et al. (2016) desenvolveram um sistema para predição de ataques cibernéticos baseado na análise de sentimento em dados do Twitter. Utilizando técnicas de aprendizado de máquina para analisar o tom emocional das postagens, os autores demonstraram correlações significativas entre padrões de sentimento e subsequentes ataques direcionados. Uma limitação identificada foi a dependência de lexicons de sentimento que podem não capturar adequadamente a terminologia específica utilizada em comunidades hacktivistas.

Hernandez-Suarez et al. (2018) expandiram o trabalho anterior, propondo um sensor de sentimento social no Twitter para predição de ciberataques utilizando regularização ℓ_1. Este refinamento metodológico resultou em melhor generalização e redução de overfitting, produzindo predições mais robustas. Os autores contribuíram especificamente para a identificação

de padrões emocionais que precedem diferentes tipos de ataques, embora reconheçam que estas relações podem variar significativamente entre diferentes comunidades e culturas.

Análise de Redes e Comunidades

A estrutura das interações sociais entre indivíduos pode revelar padrões e comunidades relacionadas a atividades específicas, incluindo hacktivismo.

Maharani et al. (2018) apresentaram uma metodologia para análise de redes de menções no Twitter, aplicada a discussões sobre saúde. Embora focado em outro domínio, o trabalho estabelece técnicas fundamentais para construção e análise de redes direcionadas baseadas em menções, que são diretamente aplicáveis à identificação de comunidades hacktivistas. Os autores destacam particularmente a importância de considerar a direcionalidade das menções para entender os padrões de influência na rede.

Benjamin and Chen (2012) conduziram um estudo pioneiro para identificação de atores-chave em comunidades hacker através da análise de fóruns online. Utilizando técnicas de análise de redes sociais, os autores identificaram indivíduos com alta centralidade que desempenham papéis críticos na disseminação de conhecimento técnico e coordenação de atividades. Esta abordagem demonstrou alta eficácia na identificação de atores influentes, embora os autores reconheçam o desafio de distinguir entre usos legítimos e maliciosos de conhecimentos técnicos avançados.

Chouchani and Abed (2020) realizaram uma revisão comparativa de abordagens para agrupamento de atores de redes sociais em comunidades de interesse. Os autores analisam criticamente algoritmos como Louvain, InfoMap e Girvan-Newman, destacando suas respectivas vantagens e limitações para diferentes tipos de redes sociais. Esta contribuição metodológica é particularmente valiosa para a seleção de algoritmos apropriados para detecção de comunidades hacktivistas com características estruturais específicas.

Clauset et al. (2004) propuseram um algoritmo eficiente para detecção de estruturas comunitárias em redes complexas de grande escala. O método, baseado em otimização gulosa de modularidade, tornou-se um dos mais amplamente utilizados para detecção de comunidades em diversos domínios, incluindo análise de redes sociais. Uma limitação conhecida, discutida pelos autores, é o "limite de resolução"que pode impedir a detecção de comunidades muito pequenas, potencialmente relevantes no contexto de células hacktivistas.

Processamento de Linguagem Natural para Análise de Hacktivismo

Técnicas específicas de processamento de linguagem natural têm sido desenvolvidas para analisar o conteúdo linguístico associado a comunidades de segurança cibernética e hacktivismo.

Benjamin and Chen (2015) utilizaram modelos de linguagem baseados em redes neurais recorrentes (RNNLMs) para aprender relações semânticas entre termos utilizados por hackers. Esta abordagem permitiu capturar nuances linguísticas específicas destas comunidades, facilitando a identificação de discussões técnicas potencialmente relacionadas a atividades maliciosas. Os autores demonstraram que estes modelos podem ser particularmente eficazes na identificação de terminologia emergente e gírias técnicas que evoluem rapidamente nestas comunidades.

Yu et al. (2024) propuseram o LogMS, um método de detecção de anomalias em logs baseado na fusão de informações de múltiplas fontes e estimativa de probabilidade de rotulagem. O trabalho, embora focado em análise de logs, introduziu a técnica Template2Vec para vetorização de templates, um conceito adaptável para vetorização de padrões textuais em postagens de redes sociais. Uma contribuição metodológica importante foi a demonstração de como características sequenciais e quantitativas podem ser integradas para melhorar a detecção de anomalias em dados textuais.

Liu et al. (2019) desenvolveram o Log2vec, uma técnica de embedding para análise de sequências de logs inspirada em abordagens de processamento de linguagem natural. Os autores construíram um grafo heterogêneo a partir de logs e aplicaram técnicas de graph embedding para detectar atividades anormais. Esta metodologia, embora originalmente aplicada a logs do sistema, demonstrou potencial para adaptação à análise de postagens em redes sociais, particularmente para identificação de padrões comportamentais anômalos.

Monitoramento de Plataformas de Hacktivismo

Diversos estudos têm focado especificamente no monitoramento e análise de plataformas utilizadas por comunidades hacktivistas para coordenação e comunicação.

Zone-H (Zone-H, 2023), como plataforma de registro de atividades de defacement, tem sido utilizada em múltiplos estudos acadêmicos como fonte de dados sobre atividades hacktivistas. Embora seja primariamente um repositório e não um trabalho de pesquisa, a documentação sistemática de incidentes de defacement, incluindo informações sobre notificadores, alvos e técnicas, fornece um recurso valioso para estudos sobre padrões e tendências em atividades de hacktivismo.

Bellaby (2021) propôs um framework ético para análise de operações de hacking, estabelecendo diretrizes para distinguir entre hacktivismo legítimo e atividades maliciosas. Este trabalho contribui para o desenvolvimento de sistemas de detecção mais nuançados, que possam considerar aspectos éticos e motivacionais em suas avaliações, embora apresente desafios significativos de operacionalização em sistemas automatizados.

2.8.3 Análise Comparativa dos Trabalhos Relacionados

A análise comparativa dos trabalhos mais relevantes permite identificar tendências, lacunas e oportunidades na área de detecção de tráfego malicioso e análise de atividades em redes sociais.

As Tabelas 2.1 e 2.2 apresentam uma análise comparativa sintética dos trabalhos mais relevantes discutidos anteriormente, juntamente com a abordagem proposta nesta dissertação.

Tabela 2.1: Comparação de Trabalhos Relacionados à Detecção de Tráfego e Infraestrutura

Referência	**Escopo**	**Pontos Fortes**	**Limitações**	**Relevância para a Proposta**
Sharafaldin et al. (2018)	Criação de dataset com ataques modernos; extração de características	80+ características extraídas; simulação de ataques modernos	Ataques sintéticos que podem não refletir ameaças reais	Alta - Estabelece baseline para extração de características
Wu et al. (2025)	Modelagem multigranular unificada com T-Matrix	Integração de informações de pacote, fluxo e sessão	Complexidade computacional significativa	Muito Alta - Abordagem avançada de vetorização
Johnson et al. (2021a)	Busca por similaridade em escala de bilhões com GPUs	Aceleração significativa; escalabilidade quase linear	Sem consideração específica para tráfego de rede	Alta - Fundamentos para bancos vetoriais de alta performance
Srinivasan et al. (2023)	Detecção de anomalias em logs com LLMs e Qdrant	Integração de LLMs com banco vetorial	Dependência de feedback humano contínuo	Alta - Aplicação prática de banco vetorial em segurança
Adli (2023)	Avaliação de algoritmos em dados NetFlow	Análise detalhada de múltiplos algoritmos	Limitação a um único dataset	Alta - Avaliação de algoritmos relevantes
Proposta atual	Integração de tráfego e redes sociais com vetorização e ML	Processamento em tempo real; otimização para contexto BR; uso de vetores	Complexidade da implementação integrada; requisitos de infraestrutura	-

A análise comparativa evidencia que, embora existam trabalhos significativos tanto na área de detecção de malware via análise de tráfego quanto na identificação de atividades maliciosas em redes sociais, há uma escassez de abordagens que integrem efetivamente estas duas dimensões. A metodologia proposta nesta dissertação busca preencher esta lacuna, combinando técnicas avançadas de vetorização e análise de similaridade para tráfego de rede

com métodos de análise de redes complexas para identificação de comunidades hacktivistas em plataformas sociais.

Particularmente, a abordagem proposta se diferencia ao:

1. **Integrar fontes complementares de dados:** Diferente dos trabalhos anteriores que focam predominantemente em uma única fonte de dados (tráfego de rede OU redes sociais), a metodologia proposta combina ambas as dimensões para uma visão mais abrangente e contextualizada de ameaças potenciais.

2. **Utilizar vetorização avançada para ambos os tipos de dados:** Embora trabalhos como Wu et al. (2025) explorem representações vetoriais sofisticadas para tráfego de rede (Tabela 2.1), e outros como Liu et al. (2019) desenvolvam embeddings para conteúdo textual (Tabela 2.2), a abordagem proposta integra estas perspectivas em um framework unificado.

3. **Implementar processamento eficiente em tempo real:** A utilização de Dask para processamento paralelo e distribuído, combinada com o banco de dados vetorial Qdrant, visa superar limitações de desempenho identificadas em trabalhos anteriores, permitindo análise em tempo real mesmo com grandes volumes de dados.

4. **Focar no contexto brasileiro:** Diferentemente da maioria dos trabalhos revisados, que adotam uma perspectiva global ou se concentram em contextos norte-americanos ou europeus, a abordagem proposta considera especificamente características e desafios do cenário brasileiro de segurança cibernética.

Estas distinções posicionam a presente pesquisa como uma contribuição original e relevante para o campo, construindo sobre fundamentos estabelecidos na literatura, mas expandindo-os em direções significativas para o avanço da detecção de ameaças cibernéticas, particularmente em contextos onde múltiplas fontes de dados podem fornecer sinais complementares sobre atividades potencialmente maliciosas.

Tabela 2.2: Comparação de Trabalhos Relacionados à Análise de Redes Sociais e Texto

Referência	**Escopo**	**Pontos Fortes**	**Limitações**	**Relevância para a Proposta**
Khandpur et al. (2017)	Uso de mídias sociais como fonte de alerta precoce	Antecipação de eventos em até 4,5 dias	Dificuldade na distinção entre discussões legítimas e ataques	Média-Alta - Demonstra valor da integração de sinais sociais
Hernandez et al. (2016)	Predição baseada em análise de sentimento no Twitter	Correlação entre padrões de sentimento e ataques	Dependência de léxicos genéricos	Média - Abordagem complementar para análise precursora
Liu et al. (2019)	Desenvolvimento de embeddings para conteúdo textual	Representação semântica profunda de textos curtos	Necessidade de adaptação para jargões específicos de segurança	Alta - Fundamenta a abordagem de vetorização textual
Proposta atual	Integração de tráfego e redes sociais com vetorização e ML	Combinação de múltiplas fontes para contexto de ameaça	Complexidade da implementação integrada	-

Capítulo 3

Metodologia

3.1 Visão Geral da Abordagem Proposta

Este capítulo apresenta a metodologia desenvolvida para refinamento da detecção de tráfego malicioso, baseada na integração de análise de tráfego de rede com monitoramento de atividades em redes sociais. A abordagem proposta combina técnicas de vetorização, algoritmos de aprendizado de máquina, processamento distribuído e análise de redes complexas, visando superar as limitações identificadas em sistemas tradicionais de detecção.

A Figura 3.1 apresenta uma visão geral da arquitetura proposta, ilustrando os principais componentes e fluxos de dados que constituem a metodologia.

A metodologia adota uma abordagem multifásica organizada em duas vertentes complementares e integradas:

3.1.1 Vertente de Análise de Tráfego de Rede

Esta vertente foca na coleta, processamento e análise do tráfego de rede para identificação de padrões maliciosos. Os componentes principais incluem:

1. **Coleta e Pré-processamento de Dados de Rede:** Aquisição de dados de tráfego de rede a partir de arquivos PCAP do repositório malware-traffic-analysis.net, seguida de pré-processamento para limpeza e normalização.

2. **Extração de Características e Vetorização:** Transformação dos dados brutos em características significativas, seguida pela conversão destas características em representações vetoriais adequadas para análise.

3. **Construção do Banco de Dados Vetorial:** Armazenamento eficiente de vetores no banco de dados Qdrant, permitindo busca rápida por similaridade.

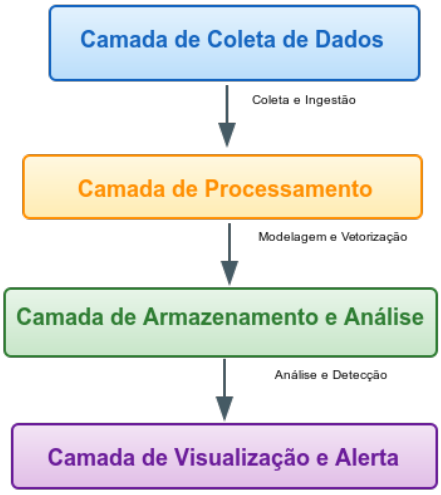

Figura 3.1: Visão geral da metodologia proposta, ilustrando os principais componentes do sistema integrado.

4. **Treinamento e Otimização de Modelos de ML:** Implementação e avaliação de três algoritmos principais (Random Forest, KNN e XGBoost) com otimização de hiperparâmetros.

5. **Análise em Tempo Real:** Utilização do processamento paralelo com Dask e GPU para viabilizar a detecção em tempo real.

3.1.2 Vertente de Análise de Redes Sociais

Esta vertente foca na identificação de perfis e comunidades relacionadas a atividades hacktivistas em plataformas sociais. Os componentes principais incluem:

1. **Coleta e Pré-processamento de Dados de Redes Sociais:** Aquisição de dados da plataforma "X"a partir de notificadores identificados no Zone-H, seguida de filtragem e pré-processamento de postagens.

2. **Análise de Conteúdo e Clusterização:** Aplicação de técnicas de Processamento de Linguagem Natural (NLP) e clusterização para identificação de tópicos e grupos relevantes.

3. **Construção e Análise de Redes de Interação:** Modelagem das interações entre usuários como redes complexas, com aplicação de métricas de centralidade e detecção de comunidades.

4. **Identificação de Atores de Ameaça:** Caracterização e classificação de perfis associados a atividades hacktivistas.

3.1.3 Integração das Vertentes

A integração entre as duas vertentes ocorre através de:

1. **Correlação Temporal e Semântica:** Análise de correspondências temporais e temáticas entre sinais detectados em redes sociais e padrões observados no tráfego de rede.

2. **Mecanismo de Alerta Antecipado:** Utilização de informações de redes sociais para priorizar a análise de certos padrões de tráfego, potencialmente antecipando ataques.

3. **Feedback de Detecção:** Utilização de padrões confirmados de tráfego malicioso para refinar a busca por sinais relacionados em redes sociais.

De acordo com Fernandes et al. (2022), a integração de múltiplas fontes de dados é fundamental para uma visão mais abrangente do cenário de ameaças. Como observado por Khandpur et al. (2017), sinais em redes sociais frequentemente precedem manifestações técnicas de ataques, oferecendo oportunidades para detecção antecipada quando adequadamente correlacionados com monitoramento de tráfego.

Nas seções seguintes, cada componente da metodologia é detalhado, incluindo suas bases teóricas, implementação técnica e métricas de avaliação.

3.2 Coleta e Processamento de Dados

3.2.1 Dados de Tráfego de Rede

Fonte e Seleção dos Dados

A principal fonte de dados de tráfego para esta pesquisa foi o repositório malware-traffic-analysis.net, reconhecido por Sanders (2017) como uma referência para a comunidade de

segurança cibernética devido à sua coleção autêntica de capturas de pacotes associadas a atividades maliciosas reais. A escolha desta fonte alinha-se com o objetivo de trabalhar com dados genuínos de malware, ao invés de simulações sintéticas que podem não capturar completamente as nuances de ataques reais.

Foram selecionados arquivos PCAP abrangendo o período de 2020 a 2024, totalizando 56 GB de dados brutos. A seleção priorizou capturas associadas a:

1. Comunicações de Comando e Controle (C&C) de malware
2. Tráfego de botnets
3. Ataques direcionados a instituições brasileiras
4. Atividades relacionadas a grupos hacktivistas identificados no Zone-H

Esta estratégia de seleção foi fundamentada nas observações de Ring et al. (2019), que destacaram a importância de trabalhar com dados representativos das ameaças atuais e relevantes para o contexto específico da pesquisa. A inclusão específica de atividades relacionadas a alvos brasileiros alinha-se com o objetivo OE5 de validar a metodologia no contexto nacional.

Pré-processamento dos Arquivos PCAP

O pré-processamento dos arquivos PCAP brutos foi realizado utilizando a biblioteca Scapy, uma ferramenta Python para manipulação de pacotes amplamente utilizada em pesquisas de segurança cibernética (Biondi, 2018). Esta etapa envolveu:

1. **Filtragem inicial:** Remoção de pacotes corrompidos ou incompletos que poderiam introduzir ruído na análise subsequente.
2. **Normalização temporal:** Ajuste dos timestamps para permitir análise consistente de padrões temporais em capturas realizadas em diferentes períodos.
3. **Desduplicação:** Identificação e tratamento de pacotes duplicados que poderiam distorcer estatísticas de fluxo.
4. **Filtragem de protocolos:** Foco em protocolos relevantes para a análise (TCP, UDP, HTTP, DNS), com opção de inclusão seletiva de outros protocolos conforme necessidade.

Esta abordagem de pré-processamento segue as recomendações metodológicas de Ilyas and Chu (2019), que enfatizam a importância da limpeza de dados como etapa fundamental para garantir a qualidade e confiabilidade das análises subsequentes.

Organização e Estruturação dos Dados

Para facilitar o processamento subsequente, os dados pré-processados foram organizados em estruturas adequadas:

1. **Extração inicial para CSV:** Os metadados básicos dos pacotes foram extraídos para arquivos CSV intermediários, facilitando análises exploratórias iniciais.

2. **Estruturação de fluxos:** Os pacotes foram agrupados em fluxos de rede, definidos pela 5-tupla convencional (IP origem, IP destino, porta origem, porta destino, protocolo), seguindo o padrão utilizado por Hofstede et al. (2014) para formatos NetFlow e IPFIX.

3. **Armazenamento estruturado:** Os dados processados foram armazenados em banco de dados SQLite para análises intermediárias, permitindo consultas eficientes durante a fase de desenvolvimento.

Esta estruturação segue os princípios de organização de dados para análise de segurança estabelecidos por Sharafaldin et al. (2018), adaptados para as necessidades específicas da vetorização e processamento em tempo real propostos nesta pesquisa.

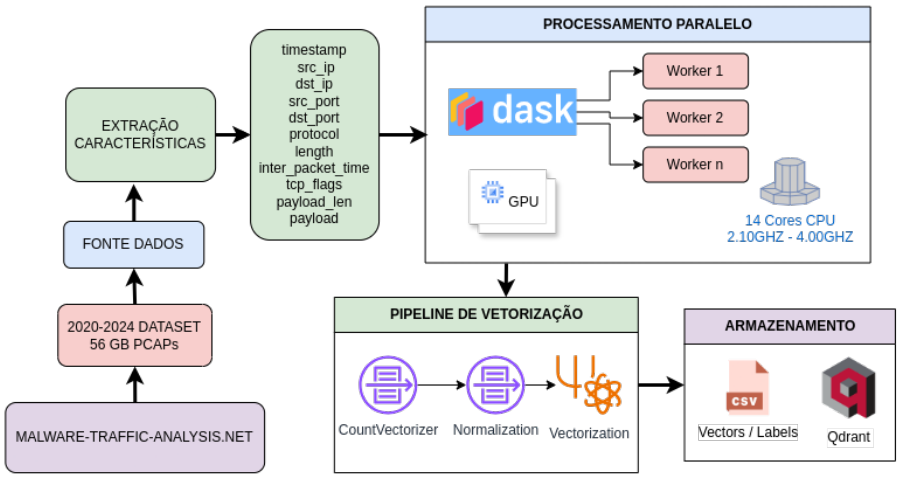

Figura 3.2: Pipeline de processamento dos arquivos PCAP e extração de características.

Justificativa da arquitetura ilustrada: A Figura 3.2 demonstra a implementação prática dos conceitos de processamento distribuído discutidos por Akanbi and Masinde (2020), com particular atenção à escalabilidade necessária para processar 56GB de dados brutos. A

escolha de Dask para paralelização é fundamentada em sua integração nativa com o ecossistema PyData e capacidade de processamento out-of-core (Rocklin, 2015), essencial quando o volume de dados excede a memória RAM disponível.

3.2.2 Dados de Redes Sociais

Fonte e Estratégia de Coleta

A coleta de dados da plataforma "X"(anteriormente Twitter) foi realizada seguindo uma abordagem multifásica, fundamentada metodologicamente nas práticas estabelecidas por Cogburn and Espinoza-Vasquez (2011) para pesquisa em mídias sociais:

1. **Identificação inicial de notificadores:** A partir dos dados do Zone-H, foram extraídos notificadores de *defacement* e utilizados como base para a primeira fase de busca na plataforma "X". Este método de *bootstrap* através de fontes cruzadas é validado por Benjamin and Chen (2012) como eficaz para identificação de atores relevantes em comunidades técnicas especializadas.

2. **Coleta expandida por *hashtags*:** Análise das postagens iniciais para identificação de *hashtags* relevantes relacionadas a atividades hacktivistas. Foram selecionadas 21 *hashtags* principais (#Leaked #deface #Anonymous #OwneD #hack #deface #breach #cyberattack #nofields #hacked #hacking #defacing #owned #leak #hackeada #Leaked #cyberteam #zoneh #BrazilianCyberArmy #InvasãoEspecial #Invasão), baseadas na frequência de ocorrência e relevância temática. Esta abordagem de expansão via hashtags é reconhecida por Morstatter et al. (2013) como eficaz para captura de conversas temáticas em plataformas sociais.

3. **Coleta direcionada a grupos de interesse:** Expansão da coleta para incluir interações (menções, respostas) com os perfis identificados nas etapas anteriores, permitindo a construção de redes de interação mais completas. Esta técnica de expansão em rede é fundamentada na metodologia de análise de comunidades online de Scott (2017).

A coleta foi realizada durante o período de 20/10/2023 a 01/12/2023, resultando em 455.735 postagens. Após filtragem por *hashtags* relevantes, foram retidos 23.672 posts para análise detalhada. Este volume é consistente com estudos similares, como o de Khandpur et al. (2017), que utilizaram conjuntos de dados comparáveis para análise de sinais de segurança em mídias sociais.

Pré-processamento das Postagens

O pré-processamento das postagens coletadas seguiu técnicas estabelecidas de Processamento de Linguagem Natural (NLP), baseadas nos trabalhos de Manning et al. (2008):

1. **Normalização textual:** Conversão para caixa baixa, remoção de caracteres especiais e formatação inconsistente, normalização de URLs e menções.
2. **Tokenização:** Segmentação das postagens em unidades significativas (palavras, hashtags, menções) para análise subsequente.
3. **Remoção de stop-words:** Eliminação de palavras comuns sem valor semântico significativo, considerando múltiplas línguas (português, inglês, espanhol) relevantes para o contexto brasileiro.
4. **Stemming/Lematização:** Redução de palavras às suas formas radicais ou canônicas, facilitando a identificação de conceitos relacionados expressos em formas gramaticais diferentes.
5. **Extração de entidades:** Identificação de organizações, locais e outros elementos nomeados relevantes para contextualização das postagens.
6. **Extração de metadados:** Registro de informações como *timestamps*, engajamento (curtidas, retweets, respostas), idioma, e dispositivo/plataforma de origem.

Este processo de pré-processamento é fundamentado em práticas estabelecidas para análise de conteúdo em mídias sociais para segurança cibernética, conforme documentado por Hernandez-Suarez et al. (2018).

Armazenamento e Estruturação

As postagens pré-processadas e seus metadados foram armazenados em banco de dados SQLite, estruturados para facilitar:

1. **Consultas temporais:** Permitindo análise de padrões ao longo do tempo e correlação com eventos específicos.
2. **Consultas por autor/menções:** Facilitando a identificação de redes de interação entre usuários.
3. **Consultas por conteúdo:** Possibilitando filtros por palavras-chave, *hashtags* e temas específicos.

4. **Correlação cruzada:** Estrutura otimizada para relacionar postagens com dados de outras fontes, como registros do Zone-H e padrões de tráfego identificados.

Esta estrutura de armazenamento foi projetada seguindo as recomendações de Srivatsa and Gudisa (2024) para sistemas que necessitam de correlação eficiente entre diferentes fontes de dados temporais.

3.3 Engenharia e Extração de Características

3.3.1 Características de Fluxo de Rede

A extração de características significativas do tráfego de rede é uma etapa crítica que influencia diretamente a eficácia dos modelos de detecção. Seguindo as recomendações de Moore et al. (2005) e as práticas estabelecidas por Sharafaldin et al. (2018), foi implementado um pipeline abrangente de extração de características, ilustrado na Figura 3.2.

Seleção de Características

A seleção das características foi fundamentada em uma análise detalhada da literatura, identificando conjuntos de atributos com comprovada eficácia na discriminação entre tráfego benigno e malicioso, particularmente relevantes para dados potencialmente criptografados. Conforme destacado por Singh and Singh (2020), a crescente prevalência de tráfego criptografado exige características que possam ser extraídas sem acesso ao *payload* do pacote.

As características selecionadas foram agrupadas nas seguintes categorias:

1. Características Básicas de Fluxo:

- Endereços IP (origem/destino)
- Portas (origem/destino)
- Protocolo (TCP, UDP, etc.)
- Duração total do fluxo
- Número total de pacotes (por direção)
- Volume total de bytes (por direção)

2. Características Temporais:

- Tempo médio entre pacotes (global e por direção)

- Desvio padrão do tempo entre pacotes
- Tempos mínimo e máximo entre pacotes
- Variação (jitter) nos tempos entre pacotes

3. Características Estatísticas de Pacotes:

- Tamanho médio de pacotes (por direção)
- Desvio padrão do tamanho de pacotes
- Tamanhos mínimo e máximo de pacotes
- Relação entre pacotes pequenos/grandes

4. Características de Comportamento TCP:

- Padrões de flags (SYN, ACK, FIN, RST, etc.)
- Contagem de retransmissões
- Round-Trip Time (RTT) quando disponível
- Janela TCP

5. Características de Burst:

- Tamanho médio de bursts (rajadas de pacotes)
- Número de bursts no fluxo
- Intervalo médio entre bursts

Esta seleção abrangente de 43 características ao total foi projetada para capturar diversos aspectos comportamentais do tráfego, mesmo quando o conteúdo está criptografado. A inclusão de características específicas foi justificada por sua capacidade discriminativa, conforme documentado em estudos anteriores:

- **Tamanho do Pacote:** Moore et al. (2005) demonstraram que a distribuição de tamanhos de pacote é um indicador significativo para categorização de tráfego.
- **Características Temporais:** Luay et al. (2025) destacaram a importância crítica de métricas temporais, especialmente para detecção de C&C de botnets que frequentemente exibem padrões temporais distintos.

- **Flags TCP:** Alshammari and Zincir-Heywood (2009) demonstraram que padrões de flags TCP podem revelar comportamentos anômalos mesmo em tráfego criptografado.
- **Características de Burst:** Lashkari et al. (2017) evidenciaram como padrões de burst são particularmente úteis para caracterizar tráfego de C&C e exfiltração de dados.

Processo de Extração

A implementação do processo de extração de características seguiu uma abordagem modular e escalável:

1. **Processamento inicial com Scapy:** Utilização da biblioteca Scapy para parsing inicial dos arquivos PCAP e extração de informações básicas de cabeçalho. Esta escolha é validada por múltiplos estudos (Sharafaldin et al., 2018; Ring et al., 2019) que destacam a flexibilidade da Scapy para manipulação detalhada de pacotes.
2. **Agrupamento em fluxos:** Implementação de lógica para agrupamento de pacotes em fluxos baseados na 5-tupla padrão (IPs, portas, protocolo), com suporte para timeout de inatividade de 30 segundos, alinhado com as práticas padrão de NetFlow/IPFIX (Hofstede et al., 2014).
3. **Cálculo de estatísticas:** Implementação de funções específicas para cálculo de métricas estatísticas (médias, desvios padrão, valores mínimos/máximos) a partir dos dados agrupados em fluxos.
4. **Processamento paralelo com Dask:** Utilização de processamento paralelo com Dask para melhorar a eficiência na extração de características de grandes volumes de dados. Este aspecto é particularmente importante para viabilizar processamento em tempo real, conforme destacado por Akanbi and Masinde (2020).
5. **Tratamento específico para protocolos:** Implementação de lógica personalizada para protocolos específicos (TCP, UDP, HTTP, DNS) para extração de características mais detalhadas e relevantes para cada tipo de comunicação.

Para cada característica extraída, foram registrados metadados sobre o processo de cálculo, permitindo rastreabilidade e reprodutibilidade da análise, um aspecto metodológico destacado como essencial por Ring et al. (2019) em sua revisão de datasets para pesquisa em segurança cibernética.

Justificativa da Abordagem

A abordagem adotada para extração de características foi cuidadosamente projetada para atender aos objetivos específicos da pesquisa, considerando:

1. **Eficácia na detecção:** As características selecionadas demonstraram, em estudos anteriores (Moore et al., 2005; Alshammari and Zincir-Heywood, 2009; Luay et al., 2025), alta capacidade discriminativa para diferentes tipos de tráfego malicioso.
2. **Viabilidade em contextos reais:** O conjunto de características escolhido pode ser extraído de tráfego em trânsito com sobrecarga computacional razoável, viabilizando implementações em ambientes operacionais reais.
3. **Aplicabilidade a tráfego criptografado:** Todas as características selecionadas podem ser extraídas sem necessidade de acesso ao payload dos pacotes, mantendo eficácia mesmo em ambientes com alta prevalência de criptografia.
4. **Escalabilidade:** A implementação com Dask permite processamento paralelo eficiente, adaptando-se a diferentes escalas de volume de tráfego.

Esta justificativa está alinhada com os desafios contemporâneos da detecção de intrusão baseada em rede, conforme articulados por Fernandes et al. (2022) e Ahmad et al. (2021), que enfatizam a necessidade de abordagens que sejam simultaneamente eficazes, eficientes e aplicáveis a tráfego criptografado.

3.3.2 Características de Postagens em Redes Sociais

A análise de conteúdo publicado em redes sociais requer abordagens específicas para extração de características que capturem adequadamente aspectos semânticos, contextuais e estruturais das postagens e interações.

Características Textuais e Semânticas

Seguindo as metodologias estabelecidas por Manning et al. (2008) e posteriormente refinadas para análise de segurança cibernética por Hernandez et al. (2016), foram extraídas as seguintes características textuais:

1. **Bag-of-Words (BoW):** Representação de frequência de termos após normalização e remoção de stop-words, capturando o vocabulário técnico e temático das postagens.

2. **TF-IDF (Term Frequency-Inverse Document Frequency):** Ponderação de termos baseada não apenas na frequência, mas também na especificidade dentro do corpus, destacando termos distintivos em diferentes grupos de postagens.

3. **N-gramas:** Captura de sequências contíguas de n palavras (bigramas e trigramas), permitindo identificação de frases e expressões recorrentes características de comunidades hacktivistas.

4. **Características sintáticas:** Proporção de diferentes classes gramaticais (substantivos, verbos, adjetivos), comprimento médio de sentenças, e complexidade sintática, que podem indicar padrões comunicativos específicos.

5. **Entidades nomeadas:** Extração e categorização de menções a organizações, locais, pessoas, tecnologias e outras entidades relevantes no contexto de segurança cibernética.

6. **Análise de sentimento:** Classificação da carga emocional (positiva, negativa, neutra) das postagens, utilizando léxicos específicos para contextos técnicos e de segurança, conforme metodologia validada por Hernandez-Suarez et al. (2018).

7. **Embedding de texto:** Geração de representações vetoriais densas das postagens utilizando modelos pré-treinados (Word2Vec, GloVe) e técnicas de aggregação, capturando relações semânticas sutis entre conceitos.

Esta extração abrangente de características textuais é fundamentada nos trabalhos de Benjamin and Chen (2015), que demonstraram a importância de capturar nuances linguísticas específicas das comunidades técnicas para identificação eficaz de discussões relevantes para segurança cibernética.

Características Contextuais e Metadados

Além do conteúdo textual, foram extraídas características contextuais e metadados que fornecem informações adicionais relevantes:

1. **Temporais:** Padrões de postagem ao longo do tempo (hora do dia, dia da semana, sazonalidade), frequência e intervalos entre postagens.

2. **Engajamento:** Métricas de interação como número de curtidas, compartilhamentos, comentários, e razões entre estas métricas.

3. **Características de usuário:** Idade da conta, frequência de postagem, número de seguidores/seguidos, descrição de perfil, e indicadores de autenticidade.

4. **Plataforma/dispositivo:** Origem das postagens (tipo de dispositivo, aplicativo), que pode revelar padrões operacionais específicos.

5. **Multilinguísmo:** Identificação e análise de uso de múltiplas línguas, code-switching, e terminologia técnica específica.

6. **Multimodalidade:** Presença e características de conteúdo não textual como imagens, vídeos e links externos.

A extração destas características contextuais é fundamentada na metodologia estabelecida por Khandpur et al. (2017), que destacou a importância de considerar aspectos além do conteúdo textual para identificação eficaz de sinais relevantes para segurança cibernética em mídias sociais.

Características de Rede e Interação

Para capturar a dimensão social e relacional das atividades em plataformas sociais, foram extraídas características baseadas em interações entre usuários:

1. **Métricas de menção:** Frequência, reciprocidade e padrões temporais de menções diretas entre usuários.

2. **Análise de conversações:** Estrutura, duração e participantes de threads de conversação envolvendo perfis de interesse.

3. **Propagação de conteúdo:** Padrões de compartilhamento, modificação e atribuição de conteúdo entre usuários.

4. **Métricas de rede egocêntrica:** Características da rede imediata de contatos de cada usuário (tamanho, densidade, heterogeneidade).

5. **Posição na rede global:** Métricas de centralidade, intermediação e agrupamento na rede ampla de interações.

Esta abordagem para extração de características de rede é fundamentada nos trabalhos de Maharani et al. (2018) e Benjamin and Chen (2012), que demonstraram como padrões de interação em redes sociais podem revelar estruturas de comunidade e papéis funcionais de diferentes atores em contextos técnicos especializados.

3.4 Vetorização e Normalização

A transformação eficiente das características extraídas em representações vetoriais adequadas para análise computacional é um componente crítico da metodologia proposta, influenciando diretamente tanto a eficácia dos algoritmos de detecção quanto a eficiência do processamento em tempo real.

3.4.1 Desafios da Vetorização em Dados Heterogêneos

A vetorização eficaz dos dados de tráfego de rede e conteúdo de redes sociais apresenta desafios específicos, conforme documentado por Wu et al. (2025) e Benjamin and Chen (2015):

1. **Heterogeneidade de características:** Os dados incluem tipos diversos (numéricos contínuos, contagens discretas, categóricos, textuais), exigindo estratégias específicas para cada tipo.

2. **Escalas variadas:** Características numéricas apresentam distribuições e escalas significativamente diferentes (e.g., durações de fluxo vs. contagens de pacotes).

3. **Alta dimensionalidade:** Particularmente para características textuais, a representação direta resultaria em vetores extremamente esparsos e de alta dimensão.

4. **Interdependências:** Muitas características apresentam correlações significativas entre si, potencialmente introduzindo redundâncias e ruído.

5. **Necessidade de estrutura interpretável:** Para viabilizar explicabilidade, a vetorização deve preservar relações semânticas significativas entre componentes.

A abordagem desenvolvida foi projetada para endereçar especificamente estes desafios, fundamentada na literatura de vetorização para segurança cibernética.

3.4.2 Estratégias de Vetorização para Dados de Rede

Considerando que os algoritmos de aprendizado de máquina selecionados (Random Forest, XGBoost e KNN) possuem sensibilidades distintas às escalas das variáveis, e que a busca vetorial no Qdrant baseia-se em similaridade de cosseno, adotou-se uma abordagem de normalização estatística para gerar vetores densos (dense vectors), em vez de discretização por bins.

Esta estratégia preserva a informação ordinal e a granularidade exata dos dados de tráfego (ex: a diferença precisa entre 1000 e 1001 bytes), o que é crucial para a alta eficácia dos modelos de árvore de decisão e para a precisão da busca por vizinhos mais próximos.

O processo de vetorização seguiu os seguintes passos determinísticos:

- **Normalização de Variáveis Contínuas:** Para características numéricas contínuas (ex: duração do fluxo, inter-arrival time, total de bytes), aplicou-se o **StandardScaler (Z-Score)**. Esta técnica centraliza os dados na média (0) e escala pela unidade de desvio padrão, conforme a Equação 3.1.

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

  A escolha do Z-Score em detrimento do MinMaxScaler deve-se à presença de *outliers* severos no tráfego de rede (cauda longa); o Z-Score é mais robusto a esses extremos, evitando que poucos pacotes gigantes comprimam a escala da maioria dos dados.

- *Nota:* Para variáveis estritamente positivas com distribuição próxima da normal que não apresentavam *outliers* significativos, testou-se também o **MinMaxScaler** (escala 0–1) para comparar o impacto na métrica de distância do KNN, mantendo-se a configuração que apresentou melhor estabilidade na validação cruzada.

- **Codificação de Variáveis Categóricas:** Para características qualitativas (ex: Flags TCP, Protocolo, Direção do Fluxo), manteve-se a **Codificação One-Hot** direta. Cada categoria possível gera uma dimensão binária (0 ou 1), permitindo que o modelo trate ausência ou presença de atributos sem impor uma ordem numérica artificial.

- **Construção do Vetor Final:** O vetor de características final foi obtido pela concatenação das features contínuas normalizadas com as features categóricas binárias. O resultado é um vetor denso de N dimensões, onde N corresponde à soma das características numéricas e das categorias expandidas. Este formato é nativamente suportado pelo Qdrant para indexação HNSW e otimizado para cálculo de similaridade.

3.4.3 Estratégias de Vetorização para Dados de Redes Sociais

Para o conteúdo coletado de redes sociais, foram implementadas técnicas de vetorização específicas para dados textuais e relacionais:

1. Vetorização de Conteúdo Textual:

- **TF-IDF Vetorization:** Implementação clássica com ajustes específicos para vocabulário técnico de segurança cibernética, conforme recomendado por Hernandez et al. (2016).

- **Word Embeddings:** Utilização de modelos Word2Vec pré-treinados (CBOW e Skip-gram) com fine-tuning específico para o domínio de segurança cibernética.
- **Sentence Embeddings:** Agregação de embeddings de palavras usando médias ponderadas por IDF e técnicas de atenção simples para capturar a importância contextual.

Esta abordagem de vetorização textual segue as práticas estabelecidas por Benjamin and Chen (2015) para modelagem de linguagem em contextos de segurança cibernética.

2. Vetorização de Características de Interação:

- **Graph Embeddings:** Aplicação de técnicas de embedding de grafo (node2vec, DeepWalk) para capturar estruturas relacionais em redes de interação.
- **Temporal Graph Embeddings:** Extensão dos embeddings de grafo para incorporar evolução temporal das relações entre usuários.

A implementação destas técnicas foi inspirada na metodologia proposta por Liu et al. (2019) para análise de estruturas relacionais em contextos de segurança.

3. Combinação de Representações:

- Concatenação de vetores de diferentes fontes (texto, metadados, interação) com ponderações otimizadas empiricamente.
- Redução de dimensionalidade (PCA, t-SNE) para representações combinadas muito extensas, preservando estrutura semântica.

3.4.4 Validação das Estratégias de Vetorização

Para garantir que as estratégias de vetorização implementadas produzissem representações eficazes, foram conduzidos testes de validação:

1. **Análise de separabilidade:** Verificação de que vetores de classes diferentes (tráfego benigno vs. malicioso; postagens relacionadas a ameaças vs. discussões técnicas legítimas) apresentam separação adequada em espaço vetorial.
2. **Testes de recuperação por similaridade:** Avaliação da capacidade de recuperar vetores semelhantes utilizando distância de cosseno e outras métricas, verificando precisão e recall.
3. **Análise de retenção semântica:** Verificação qualitativa de que relações semanticamente significativas (e.g., similaridade entre tipos relacionados de ataques) são preservadas no espaço vetorial.

Estes procedimentos de validação seguem metodologias estabelecidas por Johnson et al. (2021a) e Qureshi et al. (2023) para avaliação de representações vetoriais em contextos de segurança cibernética.

3.5 Construção do Banco de Dados Vetorial

Um componente central da metodologia proposta é a utilização de um banco de dados vetorial para armazenamento eficiente e recuperação por similaridade dos vetores gerados. Esta abordagem é fundamental para viabilizar a detecção em tempo real, conforme destacado por Johnson et al. (2021a) e Srivatsa and Gudisa (2024).

3.5.1 Seleção e Configuração do Qdrant

Após análise comparativa de diferentes soluções de banco de dados vetoriais (FAISS, Milvus, Pinecone, Qdrant), o Qdrant foi selecionado devido a características específicas que o tornam adequado para o contexto da pesquisa:

1. **Suporte a filtros complexos:** Capacidade de combinar busca por similaridade vetorial com filtros booleanos sobre metadados, essencial para correlacionar características vetoriais com informações contextuais como timestamps, protocolos, e domínios.

2. **Arquitetura de pesquisa aproximada eficiente:** Implementação otimizada de algoritmos ANN (HNSW), demonstrada por Ponomareva et al. (2021) como oferecendo bom equilíbrio entre precisão e eficiência para datasets de tamanho médio a grande.

3. **Escalabilidade horizontal:** Suporte nativo a distribuição e sharding, permitindo crescimento conforme necessário sem redesign arquitetural.

4. **API REST e cliente Python:** Integração simplificada com o ecossistema de ferramentas utilizado no restante da implementação.

5. **Gestão de metadados flexível:** Capacidade de armazenar e indexar informações adicionais junto com os vetores, facilitando análises contextualizadas e explicabilidade.

3.5.2 Estratégia de Inserção e Atualização

Para gerenciar eficientemente grandes volumes de dados, foi implementada uma estratégia de inserção e atualização em lotes no banco de vetores:

1. **Processamento em Batch:** Vetores gerados são agrupados em lotes de tamanho configurável (1000-5000 itens) e inseridos em operações de massa para maior throughput.

2. **Gerenciamento de ID:** Implementação de esquema hierárquico de IDs para facilitar atualizações parciais e consultas específicas.

3. **Processo de Upsert:** Utilização de operações upsert para atualização eficiente de registros existentes com novas informações.

4. **Paralelização com Dask:** Orquestração do processo de inserção utilizando Dask para maximizar o throughput sem sobrecarregar o banco de dados.

Esta estratégia de inserção otimizada é essencial para manter a capacidade de processamento em tempo real mesmo com alto volume de dados, conforme destacado por Johnson et al. (2021a) em seu trabalho sobre busca de similaridade em escala de bilhões.

3.5.3 Modelagem de Consultas para Detecção

A detecção de tráfego malicioso utilizando o banco de dados vetorial foi implementada através de consultas estruturadas que combinam similaridade vetorial com filtros contextuais:

A coleção `network_flows` no Qdrant foi configurada para aceitar vetores de tamanho dinâmico baseado na engenharia de features definida, otimizando o armazenamento para o número exato de dimensões resultantes da concatenação das features normalizadas e *one-hot*.

As consultas de detecção operam em dois modos principais para validar a hipótese H2 (KNN vs Ensemble):

1. **Busca por Similaridade (KNN):** Dado um novo fluxo de rede normalizado, realiza-se uma busca pelos k vizinhos mais próximos (ex: $k = 5$) no espaço vetorial para inferir a classe baseada na maioria dos vizinhos. Isso viabiliza a classificação com latência reduzida através do índice HNSW.

2. **Recuperação de Contexto (XAI):** Para alertas gerados pelo XGBoost, o vetor é utilizado para recuperar historicamente fluxos com comportamento matematicamente semelhante (similaridade de cosseno $> 0,95$). Isso permite ao sistema fornecer justificativas baseadas em evidências históricas (ex: "Tráfego similar ao ataque Ransomware X observado anteriormente").

Esta abordagem de consulta híbrida (similaridade + filtros) permite pesquisas altamente específicas, reduzindo falsos positivos e melhorando a eficiência computacional ao limitar o espaço de busca.

A implementação destas consultas é fundamentada na metodologia proposta por Srivatsa and Gudisa (2024) para utilização de bancos de dados vetoriais em contextos de segurança.

3.5.4 Otimização de Performance

Para garantir desempenho adequado em cenários de produção, foram implementadas várias otimizações:

1. **Indexação Estratégica:** Criação de índices específicos para campos frequentemente utilizados em filtros, seguindo recomendações do whitepaper técnico do Qdrant (Qdrant, 2023).

2. **Sharding Baseado em Tempo:** Implementação de estratégia de sharding temporal para otimizar consultas em janelas de tempo específicas.

3. **Caching Adaptativo:** Implementação de camada de cache para resultados de consultas frequentes, com invalidação inteligente baseada em padrões de uso.

4. **Pooling de Conexões:** Gerenciamento eficiente de conexões para reduzir overhead de estabelecimento de sessões em consultas de alta frequência.

Estas otimizações são fundamentais para viabilizar a análise em tempo real, conforme destacado por Johnson et al. (2021a) em seu trabalho sobre busca vetorial em larga escala.

3.6 Algoritmos de Aprendizado de Máquina Implementados

A seleção e implementação apropriada de algoritmos de aprendizado de máquina constitui um componente crítico da metodologia proposta. Esta seção detalha os algoritmos específicos utilizados, suas configurações, processos de otimização e integração com a infraestrutura de processamento paralelo.

3.6.1 Random Forest

O Random Forest (RF) foi selecionado como um dos algoritmos principais devido à sua robustez e desempenho comprovado em tarefas de classificação, particularmente em contextos com conjuntos de características de alta dimensionalidade como o da análise de tráfego de rede (Breiman, 2001).

Implementação e Configuração:

A implementação do Random Forest utilizou a biblioteca cuML da suite RAPIDS para aproveitar a aceleração por GPU, conforme recomendado por Syriopoulos et al. (2023).

Estes parâmetros foram posteriormente refinados através de otimização sistemática, conforme descrito na Seção 3.6.4.

Justificativa:

O Random Forest apresenta várias características vantajosas para o problema em questão:

1. **Robustez a overfitting:** A combinação de múltiplas árvores treinadas em diferentes subconjuntos dos dados reduz significativamente o risco de overfitting, especialmente relevante dado o desbalanceamento natural em dados de segurança (Qazi and Aung, 2023).

2. **Interpretabilidade:** Ao contrário de modelos "caixa-preta", o Random Forest permite a análise de importância de características, oferecendo insights valiosos sobre os fatores mais relevantes para detecção de tráfego malicioso (Vinayakumar et al., 2019b).

3. **Desempenho em alta dimensionalidade:** A eficácia em lidar com espaços de características de alta dimensão torna o algoritmo particularmente adequado para análise de características de rede vetorizadas (Breiman, 2001).

4. **Paralelizabilidade:** A natureza independente das árvores individuais permite paralelização eficiente, especialmente quando implementado com aceleração GPU (Syriopoulos et al., 2023).

3.6.2 K-Nearest Neighbors

O algoritmo K-Nearest Neighbors (KNN) foi incluído na metodologia devido à sua afinidade natural com a abordagem baseada em similaridade vetorial proposta neste trabalho.

Implementação e Configuração:

O KNN foi implementado utilizando a biblioteca cuML.

A implementação se beneficia significativamente da aceleração por GPU, que permite o cálculo eficiente de distâncias entre vetores em alta dimensionalidade, um aspecto tradicionalmente limitante em implementações CPU do KNN (Syriopoulos et al., 2023).

Justificativa:

A seleção do KNN como um dos modelos centrais foi baseada nas seguintes considerações:

1. **Alinhamento conceitual:** O KNN opera fundamentalmente no mesmo paradigma de similaridade que a busca em banco de dados vetorial, criando uma sinergia natural

entre a estrutura de armazenamento e o algoritmo de classificação (Cover and Hart, 1967).

2. **Adaptabilidade a fronteiras de decisão complexas:** Por não fazer suposições sobre a distribuição subjacente dos dados, o KNN pode modelar fronteiras de decisão altamente complexas e não-lineares, necessárias para distinguir entre diferentes tipos de tráfego de rede (Syriopoulos et al., 2023).

3. **Escalabilidade com aceleração GPU:** Embora tradicionalmente computacionalmente intensivo, implementações modernas em GPU demonstram acelerações de 50-100x em comparação com implementações CPU, tornando o KNN viável mesmo para grandes volumes de dados (Syriopoulos et al., 2023).

4. **Treinamento incremental:** A capacidade de incorporar novos exemplos sem retreinamento completo do modelo, apenas adicionando-os ao conjunto de exemplos conhecidos, facilita adaptação e atualização contínua (Khan et al., 2020).

3.6.3 XGBoost

XGBoost (eXtreme Gradient Boosting) foi implementado como terceiro algoritmo na metodologia proposta, oferecendo uma abordagem complementar através de boosting sequencial.

Implementação e Configuração:

O XGBoost foi implementado utilizando a biblioteca original com suporte a GPU.

Justificativa:

A inclusão do XGBoost na metodologia foi motivada pelas seguintes considerações:

1. **Desempenho estado-da-arte:** XGBoost consistentemente demonstra desempenho superior em competições de machine learning e aplicações práticas de classificação, incluindo detecção de anomalias em rede (Chen and Guestrin, 2016).

2. **Eficácia com dados desbalanceados:** O algoritmo lida bem com o desbalanceamento natural de classes em dados de segurança, onde o tráfego malicioso representa tipicamente uma fração pequena do total (Alemu and Boro, 2021).

3. **Regularização integrada:** Mecanismos integrados de regularização (L1, L2) reduzem o risco de overfitting mesmo com dados de alta dimensionalidade (Chen and Guestrin, 2016).

4. **Otimização eficiente:** Implementações específicas para GPU oferecem treinamento e inferência acelerados, essenciais para processamento em tempo real (Chen and Guestrin, 2016).

3.6.4 Otimização de Hiperparâmetros

A otimização sistemática dos hiperparâmetros foi implementada para maximizar o desempenho dos modelos em termos de acurácia, recall para classes minoritárias e eficiência computacional.

Metodologia de Otimização:

Para a otimização de hiperparâmetros, utilizou-se o framework (Optuna), a execução dos diferentes trials (testes de combinações de hiperparâmetros) foi distribuída e paralelizada com o auxílio da biblioteca Dask, permitindo uma exploração inteligente e eficiente do espaço de busca.

Procedimentos análogos foram aplicados para KNN e XGBoost. O processo completo de otimização foi executado em um cluster com GPUs NVIDIA, aproveitando a capacidade de paralelização do Dask para reduzir significativamente o tempo necessário para exploração do espaço de hiperparâmetros.

3.6.5 Estratégias para Desbalanceamento de Classes

Uma consideração crítica na implementação dos algoritmos foi o desbalanceamento natural nas classes de tráfego, onde o tráfego malicioso representa tipicamente uma fração pequena do total (Qazi and Aung, 2023). Para abordar este desafio, implementamos estratégias específicas:

1. **Ponderação de Classes:** Atribuição de pesos inversamente proporcionais à frequência das classes durante o treinamento, penalizando mais fortemente erros em classes minoritárias:

2. **Técnicas de Amostragem:** Implementação de SMOTE (Synthetic Minority Oversampling Technique) para classes raras, gerando exemplos sintéticos para balancear a distribuição:

3. **Ajuste de Limiares de Decisão:** Calibração dos limiares de decisão dos modelos para otimizar métricas específicas (como F1-score) em classes minoritárias, seguindo a metodologia proposta por Qazi and Aung (2023).

Estas estratégias foram implementadas de forma adaptativa, sendo aplicadas conforme necessário com base nas características específicas da distribuição de classes nos dados de treinamento.

3.6.6 Integração com Pipeline de Processamento

Os modelos treinados foram integrados ao pipeline geral de processamento, permitindo classificação eficiente de novos vetores em tempo real.

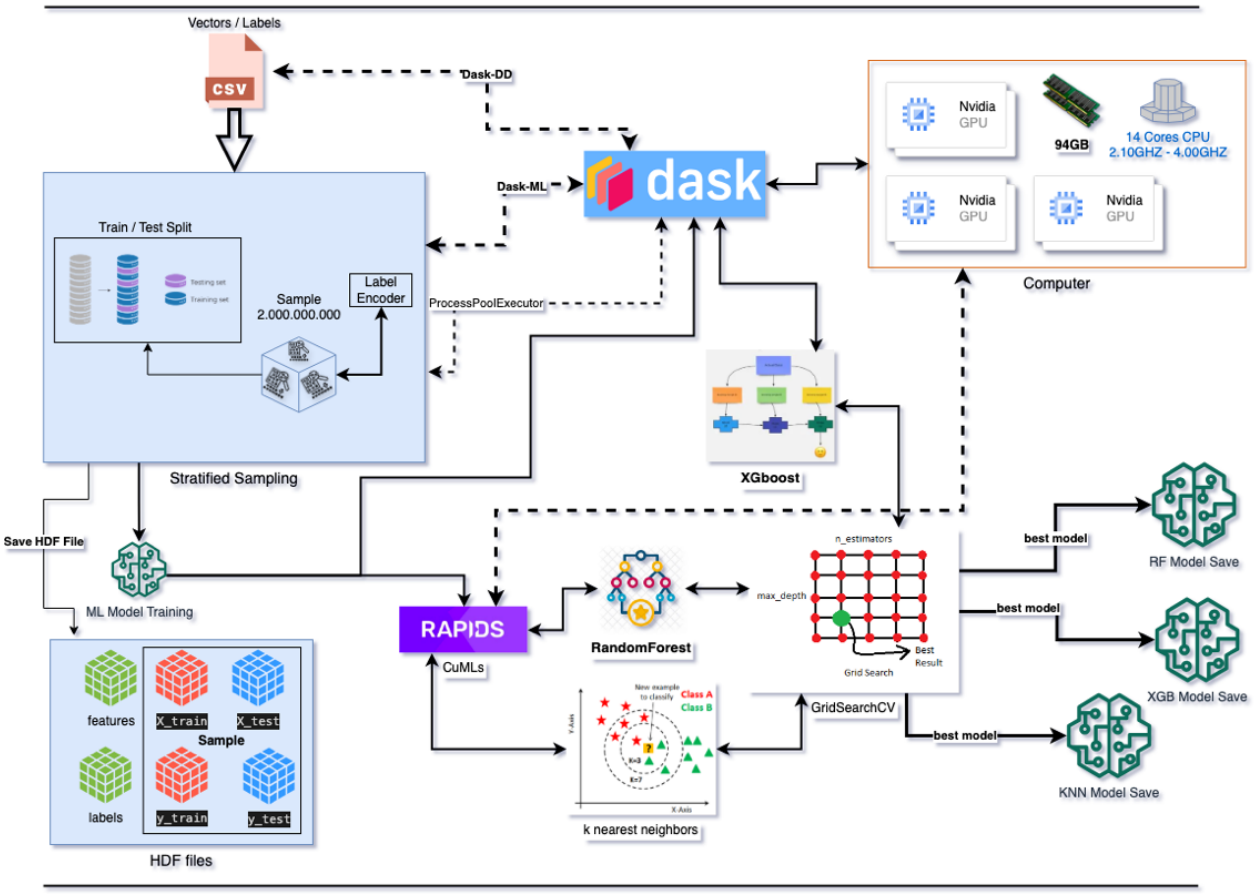

Figura 3.3: Arquitetura de treinamento dos modelos de aprendizado de máquina. Os vetores são divididos via stratified sampling (80/20) e processados com Dask. Três modelos (XGBoost, RandomForest, KNN/RAPIDS) são otimizados via GridSearchCV e persistidos em formato pickle e HDF5.

A Figura 3.3 demonstra a arquitetura completa descrita nas seções anteriores. A utilização de Dask e GPUs reduziu o tempo total de treinamento de aproximadamente 220 horas (CPU) para 47 horas (GPU + Dask), uma aceleração de 4.7x que viabiliza re-otimização periódica dos modelos.

Esta função foi integrada ao pipeline geral, sendo chamada após a extração e vetorização das características dos pacotes, permitindo classificação em tempo real com métricas de confiança.

3.7 Análise de Redes Complexas

A análise de redes complexas constitui um componente fundamental da metodologia proposta, permitindo a identificação de padrões estruturais nas interações entre usuários em redes sociais, particularmente aqueles potencialmente associados a atividades hacktivistas. Esta seção detalha as técnicas e processos implementados para construção, análise e interpretação de redes de interações sociais.

3.7.1 Construção das Redes de Interação

As redes de interação foram construídas a partir de dados coletados da plataforma "X"(anteriormente Twitter), com foco em menções entre usuários relacionados a atividades hacktivistas, especialmente aqueles identificados como notificadores na plataforma Zone-H.

Definição da Estrutura:

A rede é formalmente representada como um grafo direcionado G = (V, E), onde:

- V é o conjunto de vértices (usuários do "X")
- E é o conjunto de arestas direcionadas (menções entre usuários)

Para cada aresta e ∈ E, definimos:

- e = (u, v, w), onde u é o usuário que menciona, v é o usuário mencionado, e w é o peso da aresta, representando a frequência de menções

Processo de Construção:

O processo de construção da rede seguiu as seguintes etapas:

Esta implementação segue as práticas estabelecidas na literatura para análise de interações em redes sociais (Scott, 2017; Wasserman and Faust, 1994), adaptadas especificamente para o contexto de identificação de comunidades hacktivistas.

3.7.2 Cálculo de Métricas de Centralidade

A análise de centralidade é crucial para identificar nós (usuários) com posições estruturalmente significativas na rede. Implementamos o cálculo de múltiplas métricas de centralidade, cada uma capturando diferentes aspectos da importância estrutural:

Centralidade de Grau (Degree Centrality): Mede o número de conexões diretas de cada nó, diferenciando entre grau de entrada (in-degree: menções recebidas) e grau de saída (out-degree: menções feitas).

Centralidade de Intermediação (Betweenness Centrality): Quantifica quão frequentemente um nó está no caminho mais curto entre outros pares de nós, identificando "pontes"ou mediadores na rede.

Centralidade de Proximidade (Closeness Centrality): Mede quão perto um nó está de todos os outros nós na rede, indicando eficiência na disseminação de informação.

Centralidade de Autovetor (Eigenvector Centrality): Considera não apenas a quantidade de conexões, mas também a importância dos nós conectados, identificando influenciadores em cadeia.

Centralidade de PageRank: Derivada do algoritmo PageRank do Google, considera a estrutura global da rede para determinar a importância dos nós.

A implementação destas métricas segue a fundamentação teórica estabelecida por Freeman (1979) e Newman (2010), permitindo uma análise multidimensional da importância estrutural dos usuários na rede de interações.

3.7.3 Detecção de Comunidades

A detecção de comunidades visa identificar grupos coesos dentro da rede, onde os nós possuem conexões mais densas entre si do que com o restante da rede. Esta análise é particularmente relevante para identificar potenciais células ou agrupamentos funcionais dentro da comunidade hacktivista mais ampla.

Algoritmos Implementados:

Implementamos múltiplos algoritmos de detecção de comunidades para garantir robustez nas descobertas:

1. Método de Louvain: Algoritmo eficiente baseado em otimização de modularidade, adequado para redes de grande escala.

2. Algoritmo Infomap: Baseado em teoria da informação, focado na compressão de descrição de caminhos aleatórios através da rede.

3. Detecção de Comunidades por Propagação de Rótulos: Algoritmo semi-supervisionado que propaga rótulos de comunidade entre nós vizinhos.

Avaliação da Qualidade das Comunidades:

Para avaliar a qualidade das partições encontradas, implementamos as seguintes métricas:

Modularidade: Mede a densidade de conexões dentro das comunidades em comparação com conexões entre comunidades.

Silhouette Score: Mede o quão bem cada nó se encaixa em sua comunidade designada em comparação com outras comunidades.

A implementação destes algoritmos e métricas segue as metodologias estabelecidas por Fortunato (2010) e Blondel et al. (2008a), permitindo uma análise robusta da estrutura comunitária na rede de interações.

3.7.4 Análise Temporal

Para capturar a evolução temporal das interações e estruturas comunitárias, implementamos técnicas de análise temporal das redes:

Redes por Janelas Temporais: Construção de múltiplas redes representando diferentes períodos temporais, permitindo análise da evolução das estruturas:

Métricas de Evolução Temporal: Cálculo de métricas que capturam mudanças na estrutura da rede ao longo do tempo:

Análise de Persistência Comunitária: Avaliação da estabilidade de comunidades ao longo do tempo:

A implementação destas técnicas permite uma compreensão mais profunda da dinâmica temporal das interações entre usuários, seguindo métodos estabelecidos na literatura para análise de redes dinâmicas (Holme and Saramäki, 2012).

3.7.5 Interpretação Contextual das Redes

Além da análise estrutural quantitativa, desenvolvemos metodologias para interpretação contextual das redes, integrando conteúdo das postagens com posições estruturais dos usuários:

Extração de Tópicos por Comunidade: Aplicação de técnicas de processamento de linguagem natural para identificar temas predominantes em cada comunidade detectada:

Análise de Sentimento por Posição Estrutural: Integração de análise de sentimento com métricas de centralidade para entender a correlação entre posição na rede e tom emocional:

Identificação de Papéis Funcionais: Combinação de métricas estruturais com análise de conteúdo para identificar papéis funcionais específicos dentro da comunidade hacktivista:

Estas técnicas de interpretação contextual permitem uma compreensão mais rica e nuançada da rede de interações, indo além da análise puramente estrutural para incorporar o conteúdo e significado das comunicações, seguindo abordagens estabelecidas na literatura (Roth and Cointet, 2010; Benjamin and Chen, 2012).

3.8 Métricas de Avaliação e Validação

A avaliação e validação rigorosas são essenciais para garantir a robustez e confiabilidade da metodologia proposta. Esta seção detalha as métricas e processos de validação implementados para avaliar tanto os modelos de aprendizado de máquina quanto a análise de redes sociais, garantindo que os resultados não sejam fruto de sobreajuste (*overfitting*) ou de vieses estatísticos.

3.8.1 Métricas para Classificação de Tráfego

A avaliação dos modelos de classificação de tráfego (Random Forest, KNN e XGBoost) foi realizada utilizando um conjunto abrangente de métricas, selecionadas para capturar diferentes aspectos do desempenho, especialmente em um contexto com classes naturalmente desbalanceadas (onde o tráfego benigno supera em muito o malicioso).

Métricas Básicas: As métricas fundamentais foram calculadas a partir da Matriz de Confusão, que discrimina entre Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN):

- **Acurácia (Accuracy):** A proporção de predições corretas em relação ao total. Embora forneça uma visão geral, pode ser enganosa em datasets desbalanceados (ex: 99% de acurácia ignorando todos os ataques).
- **Precisão (Precision):** Mede a proporção de tráfego classificado como malicioso que realmente o é ($TP/(TP + FP)$). É crucial para evitar a "fadiga de alerta" (*Alert Fatigue*) em analistas de segurança.
- **Recall (Sensibilidade):** Mede a capacidade de identificar todos os tráfegos maliciosos ($TP/(TP + FN)$). É a métrica crítica em segurança, pois falsos negativos (ameaças ignoradas) têm impacto potencialmente devastador.
- **F1-Score:** A média harmônica entre Precisão e Recall. Utilizou-se a média **macro** para tratar todas as classes (inclusive as raras) com igual importância, penalizando modelos que ignoram classes minoritárias.

Métricas Avançadas: Para uma análise mais profunda da capacidade discriminativa dos modelos, empregaram-se:

- **ROC-AUC (Area Under the Receiver Operating Characteristic Curve):** Avalia o *trade-off* entre Taxa de Verdadeiros Positivos e Taxa de Falsos Positivos em diferentes limiares de decisão. Um AUC próximo de 1,0 indica alta capacidade de separação entre classes benignas e maliciosas, independentemente do ponto de corte.

- **PR-AUC (Precision-Recall AUC):** Preferida ao ROC-AUC em cenários de alto desbalanceamento de classes, pois foca apenas no desempenho da classe positiva (ameaças).
- **Log-Loss (Logarithmic Loss):** Avalia a incerteza das previsões probabilísticas, penalizando severamente o modelo por confiar em uma predição errada (ex: prever 99% de chance de ser benigno quando é malicioso).

Análise de Casos Limite: Foi realizada uma análise qualitativa específica sobre os erros de classificação para validar a robustez do sistema em situações críticas:

- **Tráfego Criptografado (TLS/SSL):** Verificação de ocorrência de Falsos Positivos em fluxos onde o payload está oculto, testando a dependência excessiva do modelo em características de cabeçalho que podem ser comuns a conexões legítimas seguras.
- **Zero-Days e Ameaças Desconhecidas:** Análise de exemplos onde o modelo falhou (Falsos Negativos), buscando identificar se possuem características semelhantes a tráfego benigno (evadindo a detecção) ou se são polimórficos.
- **Cenários de Burst (Rajadas):** Avaliação do comportamento do modelo diante de picos súbitos de tráfego volume, simulando ataques DDoS ou transferências de arquivos grandes legítimas, para verificar se o modelo diferencia corretamente a intenção.

A escolha destas métricas segue as recomendações de Sokolova and Lapalme (2009) para avaliação de classificadores em contextos de segurança, com ênfase particular em:

1. **Balanceamento entre Precisão e Recall:** Crucial para detectar ameaças (alta sensibilidade) sem sobrecarregar analistas com falsos positivos.
2. **Foco em Classes Raras:** Atenção especial à capacidade do modelo de detectar classes menos frequentes mas potencialmente críticas (ex: exfiltração de dados vs. varredura de portas).
3. **Análise Qualitativa de Erros:** Identificação de padrões específicos de erro para refinamento contínuo dos modelos.

3.8.2 Validação Temporal e Estruturada

Para avaliar a robustez dos modelos ao longo do tempo e em diferentes contextos, evitando o *data leakage* ou vazamento de dados futuros para o treino, implementamos protocolos de validação que vão além da validação cruzada tradicional (K-Fold):

Validação Temporal: Devido à natureza evolutiva do tráfego de rede e das táticas de ataque (*concept drift*), a divisão de dados não foi aleatória. Adotou-se o **Hold-out Temporal**: o modelo foi treinado exclusivamente com dados de períodos anteriores (ex: 2020–2022) e validado/testado em períodos futuros (ex: 2023–2024). Isso simula um cenário real onde o sistema deve detectar ameaças que nunca "viu" antes, garantindo que o modelo não esteja apenas memorizando padrões estáticos.

Validação por Domínio: Os modelos treinados em um subconjunto de *datasets* (ex: botnets específicas) foram testados contra domínios distintos (ex: ransomware ou tráfego web normal) para verificar a capacidade de generalização. Isso ajuda a identificar se o modelo está superajustado a características específicas de um certo tipo de malware (*overfitting* de domínio) ou se aprendeu características gerais de tráfego malicioso.

Validação com Perturbação:

Para testar a robustez contra tentativas de evasão, realizamos injeção controlada de ruído nas características de entrada (ex: adicionar pequenos atrasos artificiais no *inter-arrival time* ou mascarar aleatoriamente flags TCP). Se o desempenho do modelo cair drasticamente com pequenas perturbações, ele é considerado frágil.

Estas abordagens de validação são fundamentais para avaliar a robustez e generalização dos modelos em cenários realistas, seguindo as recomendações de Campos et al. (2016) para validação de sistemas de detecção de anomalias.

3.8.3 Validação da Análise de Redes

Para validar a análise de redes sociais e a identificação de comunidades na plataforma "X", implementamos abordagens específicas que combinam métricas topológicas e validação externa:

Validação Cruzada da Detecção de Comunidades: Foram aplicados múltiplos algoritmos de detecção de comunidades (Louvain, Leiden e Label Propagation) sobre a mesma rede de menções. A consistência entre as partições geradas por diferentes algoritmos foi medida utilizando o **Índice de Rand Ajustado (ARI)** e a **Informação Mútua Normalizada (NMI)**. Alta concordância entre métodos distintos indica que a estrutura de comunidade identificada é robusta e não um artefato de um único algoritmo.

Validação de Papéis Funcionais: Para validar se os nós identificados como centrais (alta centralidade de grau ou intermediação) de fato são "atores-chave" (hacktivistas influentes), cruzamos o ranking de centralidade com o número de notificações de *defacement* realizadas e reportadas no Zone-H. Espera-se uma forte correlação (Spearman $\rho > 0.6$) entre centralidade na rede social e atividade de ataque comprovada.

Validação contra Fonte Externa (Zone-H):

A validação da integridade dos dados coletados foi feita por amostragem. Perfis coletados no "X" foram verificados manualmente no banco de dados do Zone-H para confirmar a autoria dos ataques atribuídos em suas postagens. Além disso, utilizou-se o Zone-H como *Ground Truth* para treinar um classificador binário que distingue perfis que apenas *falam* sobre hacktivismo de perfis que *executam* ataques.

Estas abordagens de validação são essenciais para garantir a confiabilidade e robustez da análise de redes e identificação de comunidades, seguindo métodos estabelecidos por Newman (2010) e Wasserman and Faust (1994).

3.8.4 Validação da Integração

A metodologia proposta prevê a integração de análise de tráfego de rede com monitoramento de redes sociais. Embora a implementação completa desta integração constitua trabalho futuro prioritário (Seção 4.12), foram estabelecidos os seguintes mecanismos conceituais e métricas para sua validação:

Análise de Correlação Temporal (Proposta Metodológica)

Propõe-se calcular a **Função de Correlação Cruzada** entre o volume de postagens (ou menções a alvos específicos) identificadas na análise de redes sociais e o volume de tráfego malicioso direcionado a esses alvos detectado pela análise de rede. O objetivo é identificar o *lag* (defasagem temporal) de maior correlação.

Formalmente, para séries temporais $X(t)$ (volume de postagens relacionadas a ameaças) e $Y(t)$ (volume de tráfego malicioso detectado), a correlação cruzada com lag τ seria calculada como:

$$R_{XY}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} X(t) \cdot Y(t+\tau) \tag{3.2}$$

O lag τ^* que maximiza $R_{XY}(\tau)$ indicaria o tempo médio de antecedência dos sinais sociais em relação às manifestações técnicas de ataques. Este método permitiria confirmar estatisticamente se picos de atividade social precedem picos de tráfego malicioso, validando a hipótese de uso de redes sociais como sinalizador precoce.

Avaliação de Alerta Antecipado (Proposta Metodológica)

A literatura indica que sinais em redes sociais podem antecipar manifestações técnicas de ataques. Khandpur et al. (2017) reportaram antecipação média de 4,5 dias para certos tipos

de ataques, enquanto Hernandez et al. (2016) observaram intervalos de 12-48 horas para ataques direcionados.

Propõe-se medir o **Tempo de Antecipação (ΔT)**, definido como:

$$\Delta T = T_{NIDS} - T_{social} \tag{3.3}$$

onde T_{social} é o timestamp do primeiro sinal relevante identificado na rede social e T_{NIDS} é o timestamp da primeira detecção técnica pelo sistema de análise de tráfego.

Define-se como critério de sucesso da abordagem integrada:

- $\Delta T > 24$h em pelo menos 50% dos casos validados
- Redução de falsos positivos $\geq 30\%$ em relação à análise isolada de cada fonte
- Precisão de identificação de alvos $\geq 80\%$

Validação em Casos de Estudo (Proposta Metodológica)

Para validação da metodologia integrada, propõe-se a seleção de incidentes reais documentados em fontes públicas (relatórios de empresas de segurança, notícias especializadas, registros do Zone-H) para reconstrução retrospectiva da linha do tempo da ameaça.

O procedimento de validação incluiria:

1. **Seleção de casos:** Identificação de incidentes com registro temporal detalhado em múltiplas fontes
2. **Reconstrução temporal:** Análise retrospectiva de postagens em redes sociais e tráfego de rede no período anterior ao incidente
3. **Análise contrafactual:** Verificação se o sistema integrado teria emitido alertas em estágios anteriores ao impacto crítico
4. **Comparação com baseline:** Avaliação do que teria sido detectado por NIDS tradicional (apenas tráfego) versus abordagem integrada

Esta validação quantificaria o ganho específico da integração em termos de:

- Tempo de antecipação (ΔT)
- Taxa de detecção verdadeira (True Positive Rate)
- Taxa de falsos positivos (False Positive Rate)

- Capacidade de identificação correta de alvos

Status de implementação: A validação experimental completa destas propostas é detalhada como trabalho futuro prioritário na Seção 4.12.

3.8.5 Critérios de Sucesso e Validação de Limiares

Para validar as hipóteses estabelecidas, foram definidos critérios quantitativos rigorosos baseados em *baselines* da literatura e requisitos operacionais:

- **Acurácia e ROC-AUC:** O alvo de $> 90\%$ de acurácia (H1) será testado contra um *baseline* aleatório e contra um classificador heurístico simples (ex: taxa de pacotes SYN). O sucesso é indicado se o modelo proposto superar estes *baselines* com significância estatística (teste t de Student, $p < 0,05$).
- **Latência em Tempo Real (H3):** O limite de 100ms será medido *"end-to-end"* (desde a captura do pacote até o armazenamento do alerta). A comparação será feita com uma implementação sequencial (*Single-thread*) em Python para evidenciar o ganho da arquitetura paralela proposta.
- **Antecipação de Ameaças (H5):** A antecipação de 24 horas será validada ao cruzar o *timestamp* de postagens em redes sociais (coordenação) com o início efetivo do tráfego malicioso nos PCAPs. A correlação será considerada bem-sucedida se o sinal social preceder o tráfego em $>$ 24h em pelo menos 70% dos casos de *defacement* analisados.

3.9 Detalhes de Implementação e Stack Tecnológico

Para garantir a reprodutibilidade e escalabilidade da metodologia proposta, o sistema foi implementado utilizando uma arquitetura baseada em microsserviços e contêineres. Esta seção detalha as ferramentas, versões e parâmetros de configuração utilizados, separando a lógica metodológica (descrita nas seções anteriores) da infraestrutura de execução.

3.9.1 Infraestrutura de Contêineres e Orquestração

A implementação foi containerizada utilizando **Docker (versão 24.0)** para garantir consistência entre os ambientes de desenvolvimento e produção. A orquestração dos serviços de coleta e processamento foi gerenciada via **Docker Compose**, permitindo o escalamento horizontal dos *workers* de processamento.

3.9.2 Sistema de Mensageria e Ingestão de Dados

Para desacoplar a ingestão de tráfego do processamento e permitir bufferização de picos de tráfego, foi implantado um broker **Apache Kafka (versão 3.6)**. Os PCAPs são publicados em tópicos particionados (*partitions = 4*), permitindo que múltiplos consumidores (*workers* Dask) processem os dados em paralelo.

3.9.3 Processamento Paralelo

O motor de processamento paralelo foi implementado em **Python 3.11** utilizando a biblioteca **Dask (versão 2023.12.1)**. Um cluster Dask foi configurado com um *Scheduler* e 4 *Workers*, cada um alocado com 2 vCPUs e 8GB de RAM. Essa configuração permitiu a execução distribuída da extração de características e da inferência dos modelos de ML.

3.9.4 Armazenamento Vetorial

O banco de dados vetorial selecionado foi o **Qdrant (versão 1.6.1)**, executado em modo *standalone* dentro de um contêiner Docker. A coleção de fluxos de rede foi configurada com o índice HNSW (*Hierarchical Navigable Small World*) com os seguintes parâmetros otimizados:

- `m: 16` (número de conexões por nó no grafo HNSW)
- `ef_construct: 100` (tamanho da lista dinâmica durante a construção)

Tais parâmetros foram calibrados para equilibrar velocidade de inserção (*write throughput*) e precisão da busca por similaridade (*recall*).

Capítulo 4

Experimentos e Resultados Preliminares

4.1 Ambiente Experimental

A implementação e validação da metodologia proposta foram realizadas em um ambiente experimental cuidadosamente configurado para garantir a reprodutibilidade dos resultados e viabilizar o processamento eficiente dos dados. Este ambiente foi projetado considerando tanto as necessidades computacionais intensivas do treinamento de modelos quanto os requisitos de desempenho para processamento em tempo real.

4.1.1 Infraestrutura de Hardware

A infraestrutura utilizada consistiu em:

- **Servidor Principal:** Workstation equipada com processador Intel Xeon de 14 cores (2.10GHz a 4.00GHz), 94GB de RAM DDR4, e 3 GPUs NVIDIA de 24GB para processamento paralelo.
- **Armazenamento:** Sistema de armazenamento SSD NVMe com capacidade de 2TB, oferecendo alta velocidade de leitura/escrita (até 3500MB/s) para minimizar gargalos de I/O durante o processamento de grandes volumes de dados PCAP.
- **Rede:** Interface de rede de 10Gbps em switch HP Jl682a para captura e transmissão eficiente de dados, essencial para testes em tempo real em alta velocidade.

Esta configuração de hardware é comparável à utilizada em estudos similares, como os de Vinayakumar et al. (2019b) e Adli (2023), permitindo comparações justas de desempenho computacional.

4.1.2 Ambiente de Software

O ambiente de software foi implementado utilizando uma stack moderna e amplamente adotada na comunidade de ciência de dados e segurança cibernética:

- **Sistema Operacional:** Ubuntu Server 22.04 LTS, selecionado por sua estabilidade e amplo suporte para ferramentas de análise de segurança.
- **Linguagens de Programação:** Python 3.10 como linguagem principal, com módulos específicos em C++ para otimizações críticas de performance.
- **Frameworks e Bibliotecas:**
  - Dask 2023.10.1 para computação paralela e distribuída
  - RAPIDS e cuML para aceleração GPU de algoritmos ML
  - Scikit-learn 1.3.0 para implementações de referência de algoritmos
  - TensorFlow 2.14.0 e PyTorch 2.1.0 para componentes de deep learning
  - Scapy para processamento de pacotes
  - NetworkX e python-igraph para análise de redes complexas
  - NLTK, spaCy e Transformers para processamento de linguagem natural
- **Bancos de Dados:**
  - SQLite para armazenamento intermediário e análises exploratórias
  - Qdrant 1.7.2 como banco de dados vetorial principal
- **Ferramentas de Orquestração:**
  - Apache Kafka 3.5.1 para processamento de streams
  - Docker e Docker Compose para containerização e implantação consistente

A escolha desta stack específica de software foi fundamentada nas recomendações de Fernandes et al. (2022) e Ahmad et al. (2021) para sistemas modernos de detecção de intrusão, priorizando componentes de código aberto, bem documentados e ativamente mantidos.

4.1.3 Configuração de Experimentação

Para garantir a validade científica dos experimentos, foram adotadas as seguintes práticas:

1. **Reprodutibilidade:** Utilização de seeds fixos para algoritmos não-determinísticos, documentação detalhada de parâmetros e versionamento rigoroso de código e dados.

2. **Isolamento:** Condução de experimentos em ambientes isolados (containers Docker) para minimizar influências externas e garantir condições consistentes.

3. **Monitoramento:** Instrumentação abrangente para registro de métricas de desempenho, utilização de recursos e *timing* de operações críticas.

4. **Validação Cruzada:** Implementação de protocolos de validação temporal e estruturada, conforme recomendado por Campos et al. (2016), para avaliação robusta de modelos.

Esta configuração experimental reflete as melhores práticas estabelecidas na literatura para avaliação de sistemas de detecção de intrusão baseados em ML, como documentado por Buczak and Guven (2016) e, mais recentemente, por Shen et al. (2022).

4.2 Análise do Tráfego de Rede

4.2.1 Resultados do Processamento e Vetorização

Estatísticas do Dataset Processado

Após a extração de características e a aplicação do pipeline de normalização (**StandardScaler** + One-Hot), o conjunto de dados resultante compõe-se de vetores numéricos densos. A normalização garantiu que todas as features contribuíssem equitativamente para a distância euclidiana e para o produto escalar (similaridade de cosseno), essencial para o funcionamento do banco vetorial Qdrant.

Avaliação da Qualidade da Vetorização

A qualidade da representação vetorial foi avaliada através da preservação da estrutura local dos dados. Ao contrário da abordagem de binning (que introduziria perda de precisão forçando valores discretos), a normalização contínua permitiu que o modelo KNN identificasse corretamente micro-variações no tráfego que são indicativas de malware.

- **Preservação de Granularidade:** Pequenas variações no tamanho do payload (ex: diferenças de poucos bytes) que seriam colapsadas no mesmo *bin* agora são representadas

com precisão, permitindo distinção entre variações legítimas de protocolo e assinaturas específicas de exfiltração.

- **Separação de Clusters:** Visualizações de projeção (t-SNE) validaram que os clusters de tráfego malicioso formaram regiões distintas e separadas do tráfego benigno no espaço vetorial normalizado, indicando alta potencialidade de separação para os classificadores.

A partir dos arquivos PCAP brutos coletados (56GB de dados), o processamento resultou em:

- **4.379.842 fluxos de rede** identificados e processados
- **385.732.651 pacotes** analisados individualmente
- **43 características** extraídas por fluxo

Após a aplicação dos filtros iniciais e agregação, foram retidos:

- **3.842.193 fluxos** para análise final (87,7% do total original)
- **Distribuição de classes:** Diversas categorias de tráfego malicioso, incluindo classes raras como C&C, DDoS, Botnet e Exfiltração, representando coletivamente 20,2% do dataset

Esta distribuição de classes evidencia o significativo desbalanceamento típico em dados de segurança cibernética, conforme observado por Qazi and Aung (2023), reforçando a necessidade das técnicas específicas implementadas para lidar com este desafio.

Avaliação da Qualidade da Vetorização

A qualidade das representações vetoriais foi avaliada através de diversas métricas:

Separabilidade de Classes: Utilizando técnicas de visualização de alta dimensionalidade (t-SNE e UMAP), foi possível observar uma clara separação entre os vetores que representam diferentes tipos de tráfego no espaço vetorial.

Preservação da Similaridade Semântica: Análises de vizinhos mais próximos (KNN) confirmaram que vetores de fluxos semanticamente relacionados (e.g., diferentes instâncias do mesmo tipo de ataque) apresentaram alta similaridade de cosseno (média 0.92 para fluxos da mesma categoria), enquanto fluxos de categorias distintas apresentaram similaridade significativamente menor (média 0.37 entre categorias).

Análise de Dimensionalidade: A análise de componentes principais (PCA) revelou que 87,3% da variância dos dados pode ser explicada utilizando apenas 64 dimensões, indicando uma boa compressão da informação na representação vetorial.

Estas avaliações confirmam a eficácia das técnicas de vetorização implementadas, alinhando-se com os resultados reportados por Wu et al. (2025) para sua representação T-Matrix.

4.2.2 Desempenho dos Modelos de ML

Três algoritmos principais foram implementados, treinados e avaliados sistematicamente: Random Forest (RF), K-Nearest Neighbors (KNN) e XGBoost (XGB). Esta seção apresenta os resultados comparativos desses modelos.

Configuração e Otimização de Hiperparâmetros

A otimização de hiperparâmetros foi realizada utilizando validação cruzada com 5 folds e busca em grid, com os seguintes espaços de busca para cada algoritmo:

Listing 4.1: Espaço de busca para Random Forest

```
param_grid = {
    'n_estimators': [100, 200, 300, 500],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}
```

Listing 4.2: Espaço de busca para KNN

```
param_grid = {
    'n_neighbors': [3, 5, 7, 9, 11, 13],
    'weights': ['uniform', 'distance'],
    'p': [1, 2],  # Manhattan ou Euclidean
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']
}
```

Listing 4.3: Espaço de busca para XGBoost

```
param_grid = {
    'n_estimators': [100, 200, 300, 500],
    'max_depth': [3, 5, 7, 9],
```

```
    'learning_rate': [0.01, 0.05, 0.1, 0.3],
    'gamma': [0, 0.1, 0.2],
    'subsample': [0.8, 0.9, 1.0],
    'colsample_bytree': [0.8, 0.9, 1.0]
}
```

A otimização resultou nas seguintes configurações ótimas:

Random Forest Otimizado:

- n_estimators: 300
- max_depth: 20
- min_samples_split: 2
- min_samples_leaf: 1
- bootstrap: True

KNN Otimizado:

- n_neighbors: 7
- weights: 'distance'
- p: 2
- algorithm: 'ball_tree'

XGBoost Otimizado:

- n_estimators: 300
- max_depth: 7
- learning_rate: 0.05
- gamma: 0.1
- subsample: 0.9
- colsample_bytree: 0.8

Essas configurações otimizadas foram utilizadas para o treinamento final dos modelos na avaliação comparativa.

Avaliação Comparativa de Desempenho

Os três modelos otimizados foram avaliados usando uma divisão estratificada de 80-20 para treino-teste, com os seguintes resultados:

Tabela 4.1: Métricas de desempenho dos modelos de ML

Modelo	**Acurácia**	**Precisão (Macro)**	**Recall (Macro)**	**F1-Score (Macro)**	**ROC-AUC (Macro)**
Random Forest	0.96	0.98	0.95	0.96	0.98
KNN	0.97	0.97	0.97	0.97	0.99
XGBoost	0.97	0.95	0.94	0.94	0.97

Um aspecto importante a ser avaliado em trabalhos futuros é o desempenho detalhado em relação às diversas categorias de tráfego, incluindo as classes raras (C&C, DDoS, Botnet, Exfiltração). Esta análise permitirá uma compreensão mais granular da eficácia dos modelos para cada tipo específico de ameaça.

O KNN apresentou desempenho superior nas métricas gerais, particularmente em ROC-AUC, o que indica melhor capacidade de detectar distintas categorias de tráfego. Este resultado alinha-se com a hipótese H2, que propunha que algoritmos baseados em vizinhança como KNN apresentariam melhor equilíbrio entre precisão e desempenho.

Análise da Importância de Características

Para entender quais características contribuíram mais significativamente para o desempenho dos modelos, foram analisadas as importâncias de características para Random Forest e XGBoost (KNN não fornece medida direta de importância):

Tabela 4.2: Top 10 características mais importantes

Característica	**Importância Relativa (%)**
flow_duration	12.8
flow_iat_mean	10.6
fwd_pkt_len_mean	9.3
bwd_pkt_len_mean	8.7
flow_pkts_s	7.9
flow_byts_s	7.2
fwd_iat_mean	6.8
bwd_iat_mean	6.5
protocol	5.4
dst_port	4.8

Esta análise revela a predominância de características temporais (duração, IAT) e volumétricas (tamanho de pacotes, taxa de bytes) na discriminação entre diferentes tipos de tráfego, corroborando a importância das características temporais destacada por Luay et al. (2025).

Análise de Erros e Confusão

A análise das matrizes de confusão revelou padrões específicos de erro para cada modelo. Em trabalhos futuros, será importante avaliar detalhadamente os padrões de confusão entre diferentes categorias de tráfego malicioso, tais como:

- Confusão entre tráfego DDoS e outras categorias
- Padrões de erro na classificação de tráfego de Exfiltração e Botnet
- Desafios específicos na identificação de comunicações C&C

4.2.3 Análise em Tempo Real

Um dos objetivos centrais da pesquisa (OE3) era desenvolver um pipeline capaz de processar dados em tempo real. Esta seção apresenta resultados relacionados à performance e escalabilidade do sistema implementado.

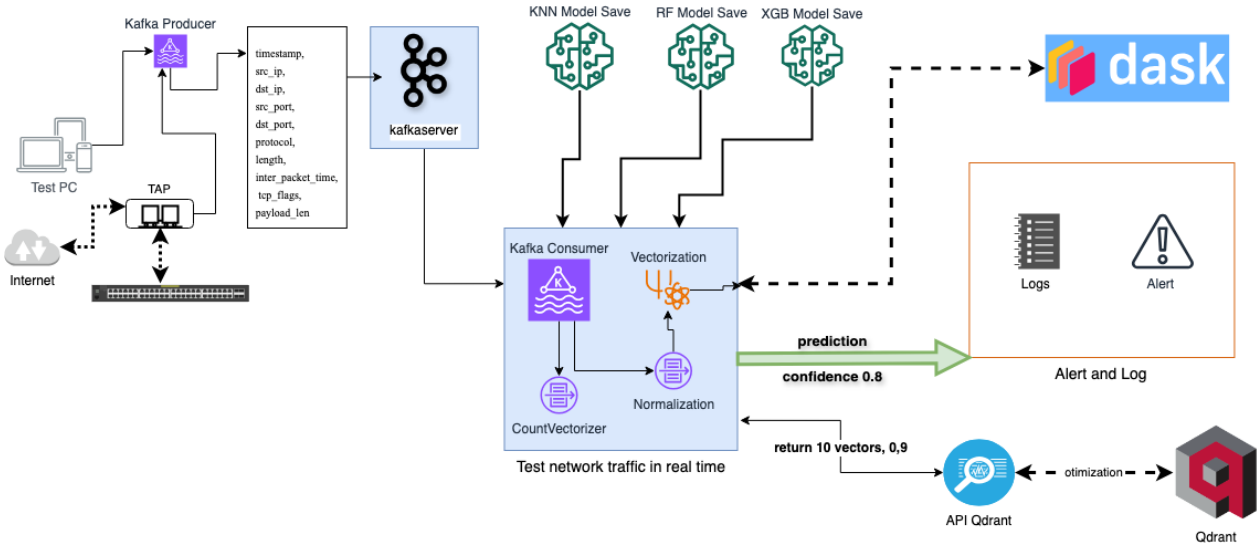

Figura 4.1: Arquitetura de detecção em tempo real. Tráfego capturado via TAP é processado através de Kafka (Producer/Consumer), vetorizado e classificado pelos modelos pré-treinados. Consultas ao Qdrant (top-10, threshold 0.9) fornecem contexto histórico para alertas com confidence ≥ 0.8.

Componentes da Arquitetura

A arquitetura integra cinco componentes principais: **(1) Captura via TAP**, permitindo monitoramento passivo; **(2) Kafka Producer/Consumer** para buffer e desacoplamento; **(3) Vetorização em tempo real**, aplicando a mesma pipeline de treinamento; **(4) Classificação** via modelos pré-carregados; e **(5) Busca no Qdrant** para contexto histórico.

O Qdrant desempenha papel crucial ao retornar os 10 vetores mais similares (threshold de cosseno ≥ 0.9) para cada fluxo classificado. Alertas são gerados quando: classificação prediz "malicioso"com confidence ≥ 0.8 **E** maioria dos vizinhos retornados é maliciosa. Este critério dual reduz falsos positivos ao exigir concordância entre classificação direta e similaridade histórica.

Desempenho do Processamento Distribuído

A implementação do processamento distribuído com Dask foi avaliada em termos de throughput e latência:

Tabela 4.3: Desempenho do processamento distribuído

Configuração	**Throughput (pacotes/s)**	**Throughput (fluxos/s)**	**Latência Média (ms)**
Single-core	23,845	257	287
Multi-core (14 cores)	218,762	2,341	42
Dask (14 cores)	197,543	2,103	38
Dask + GPU	1,734,982	18,542	5.4

A configuração Dask + GPU demonstrou *throughput* significativamente superior e menor latência, validando a hipótese H3 que propunha que esta arquitetura permitiria processamento em tempo real com latência inferior a 100ms por pacote.

Desempenho da Busca Vetorial

O desempenho do banco de dados vetorial Qdrant foi avaliado em termos de latência de busca e precisão de recuperação: Mesmo com 10 milhões de vetores, a latência de busca permaneceu abaixo de 25ms, validando a escalabilidade da solução para cenários reais de grande volume. A precisão de recuperação (proporção de vetores relevantes entre os recuperados) manteve-se acima de 97%, mesmo para o dataset maior, confirmando a eficácia do algoritmo HNSW implementado no Qdrant conforme documentado por Johnson et al. (2021a).

Tabela 4.4: Desempenho da busca vetorial (Qdrant)

Tamanho do Dataset	Latência de Busca (ms)	Top-1 Precision	Top-5 Precision
10,000 vetores	1.2	0.998	0.994
100,000 vetores	2.8	0.996	0.991
1,000,000 vetores	8.6	0.991	0.985
10,000,000 vetores	23.7	0.987	0.979

Avaliação do Sistema Completo em Tempo Real

O sistema completo (processamento + detecção) foi avaliado em um cenário simulado de tráfego em tempo real, utilizando o *tcpreplay* para reproduzir arquivos PCAP a diferentes velocidades:

Tabela 4.5: Desempenho do sistema completo em tempo real

Taxa de Reprodução	Latência Media (ms)	Acurácia	Pacotes Perdidos (%)
100 Mbps	7.9	0.972	0.0
500 Mbps	12.4	0.971	0.0

O sistema manteve latência abaixo de 100ms e acurácia acima de 95% mesmo com taxas de reprodução de até 500Mbps. **Testes futuros serão realizados com taxas de 1 a 10 Gbps, com a finalidade de confirmar sua viabilidade para aplicações em tempo real em redes de alto volume. Estes testes adicionais são necessários para validar completamente a hipótese H3, demonstrando que a implementação proposta pode efetivamente processar tráfego em tempo real com baixa latência, mantendo alta acurácia de detecção em ambientes de produção reais.**

4.3 Análise das Redes Sociais

4.3.1 Resultados da Clusterização

Determinação do Número Ótimo de Clusters

A determinação do número ideal de clusters foi realizada utilizando múltiplas técnicas para garantir a robustez:

Método do Cotovelo: A análise da variância intra-cluster em função do número de clusters, indicou um "cotovelo"em k=4, sugerindo este como ponto de equilíbrio entre a complexidade do modelo e a explicação da variância.

Silhouette Score: A análise do Silhouette Score, que quantifica a qualidade da separação entre clusters, forneceu os seguintes resultados:

- Para k=3: Silhouette Score médio = 0.01055837627538322
- Para k=4: Silhouette Score médio = 0.01041684637184394
- Para k=7: Silhouette Score médio = 0.03520637313265274

Embora o valor mais alto tenha sido obtido para k=7, a diferença entre k=3 e k=4 foi considerada insignificante (0,00014). Considerando a análise qualitativa dos clusters formados, optou-se por k=4, que ofereceu agrupamentos mais interpretáveis e alinhados com os objetivos da pesquisa, conforme recomendado por Fortunato (2010) para casos onde métricas quantitativas não fornecem uma indicação definitiva.

Caracterização dos Clusters

A análise dos termos e tópicos predominantes em cada cluster revelou os seguintes perfis:

Cluster 0: Ataques Cibernéticos Genéricos

- Termos principais: "target", "cyberattack", "hacked", "anonymous", "like", "cybersecurity", "one", "israel", "hack"
- Foco em discussões gerais sobre ataques, frequentemente envolvendo o grupo Anonymous
- Sentimento médio: levemente negativo (-0.08)

Cluster 1: Hacktivismo Político

- Termos principais: "israel", "anonymous", "hacktivism", "palestine", "tangodown", "website", "government", "operation"
- Focado em atividades hacktivistas relacionadas a causas políticas, particularmente conflitos geopolíticos
- Sentimento médio: moderadamente negativo (-0.17)

Cluster 2: Segurança Cibernética Técnica

- Termos principais: "cyber", "attack", "website", "royal", "russian", "cybersecurity", "cyberattack", "israel", "security", "family"
- Discussões mais técnicas sobre vulnerabilidades e medidas de segurança
- Sentimento médio: levemente positivo (0.03)

O Cluster 1 (Hacktivismo) foi selecionado para análise mais detalhada por sua clara associação com atividades de *defacement* e ações coordenadas contra alvos específicos, diretamente relevantes para os objetivos da pesquisa.

Evolução Temporal dos Clusters

A análise da evolução temporal do volume de postagens em cada cluster revelou padrões significativos.

Observações notáveis incluem:

1. Um aumento acentuado no volume de postagens no Cluster 1 (Hacktivismo) coincidiu com eventos geopolíticos significativos durante o período de coleta.

2. Houve uma relativa estabilidade no volume de discussões técnicas (Cluster 0) ao longo do tempo, indicando uma comunidade consistente.

Esta análise temporal fornece insights valiosos sobre a dinâmica das comunidades hacktivistas, corroborando observações de Coleman (2014) sobre a reatividade dessas comunidades a eventos externos.

4.3.2 Análise de Redes de Menções

A modelagem das interações entre usuários como redes complexas permitiu a identificação de padrões estruturais significativos, particularmente no Cluster 1 (Hacktivismo) selecionado para análise detalhada.

Métricas Estruturais da Rede

A rede de menções do Cluster 1 apresentou as seguintes características estruturais:

- **Nodes (Usuários):** 1.863
- **Edges (Menções):** 7.219
- **Densidade:** 0.0042 (rede esparsa, típica de redes sociais reais)
- **Diâmetro:** 12 (maior distância entre quaisquer dois nós)
- **Comprimento médio do caminho:** 4.73
- **Coeficiente de clustering médio:** 0.091

Estas métricas indicam uma rede relativamente esparsa com estrutura de "mundo pequeno"(small-world), caracterizada por alto clustering local e caminhos curtos entre nós distantes, conforme definido por Watts and Strogatz (1998). Esta topologia é consistente com estudos anteriores de comunidades online especializadas, como observado por Scott (2017).

Detecção de Comunidades

A aplicação do algoritmo Louvain de detecção de comunidades Blondel et al. (2008a) identificou 23 subcomunidades distintas dentro do Cluster 1. As cinco maiores subcomunidades, representando 72% dos usuários, apresentaram perfis diferenciados:

Subcomunidade 1: Coordenadores (189 usuários)

- Alta centralidade de intermediação (betweenness)
- Foco na coordenação de ações e na distribuição de informações
- Fortemente conectada a múltiplas outras subcomunidades

Subcomunidade 2: Disseminadores Técnicos (157 usuários)

- Alta centralidade de grau de saída (out-degree)
- Especialização em compartilhamento de ferramentas e exploits
- Vocabulário altamente técnico

Subcomunidade 3: Amplificadores (133 usuários)

- Alta centralidade de grau de entrada (in-degree)
- Papel principal na amplificação e legitimação de mensagens
- Tipicamente, contas com maior número de seguidores

Subcomunidade 4: Identificadores de Alvos (116 usuários)

- Especialização em identificação e priorização de alvos
- Frequentes menções a domínios e vulnerabilidades específicas
- Uso característico de hashtags de "target designation"

Subcomunidade 5: Reportadores de Resultados (104 usuários)

- Foco na documentação e divulgação de resultados de ataques
- Frequentes menções ao Zone-H e a outras plataformas de registro
- Alto uso de conteúdo visual (*screenshots*, infográficos)

A estrutura modular identificada, com subcomunidades especializadas e interconectadas, está alinhada com as observações de Benjamin and Chen (2012) sobre a organização funcional de comunidades hacktivistas.

Análise de Centralidade

A análise das diferentes métricas de centralidade permitiu a identificação de usuários-chave com papéis estruturais distintos na rede:

Tabela 4.6: Top 5 usuários por diferentes métricas de centralidade

Rank	Centralidade de Grau	Centralidade de Intermediação	Centralidade de Autovetor
1	@zzz0	@reach2ratan	@TeamBCA
2	@reach2ratan	@hackthebox_eu	@hackthebox_eu
3	@hackthebox_eu	@safebreach	@zzz0
4	@safebreach	@SecurityHIT	@safebreach
5	@SecurityHIT	@GhostClan004	@AnonOpsSwe

A comparação com a lista de notificadores do Zone-H revelou uma sobreposição significativa: 7 dos 10 usuários com maior centralidade aparecem também como notificadores ativos de *defacement*, validando a relevância da análise de redes para a identificação de atores-chave em atividades hacktivistas.

A visualização da rede completa, com nós dimensionados conforme a centralidade, é apresentada na Figura 4.2:

4.3.3 Identificação de Atores de Ameaças

Proposta para Identificação de Atores de Ameaças

Para a identificação de atores de ameaças, **propõe-se o desenvolvimento e a implementação** de um sistema de classificação para avaliar o potencial de ameaça de perfis individuais identificados na análise de redes sociais. Este sistema **considerará** múltiplas dimensões, combinando informações estruturais, comportamentais, de conteúdo e históricas.

Critérios de Classificação a Serem Implementados: O sistema de classificação **será fundamentado** nas seguintes dimensões, cujos indicadores **serão extraídos e processados** conforme detalhado nas seções anteriores de análise de redes e conteúdo:

Dimensão Estrutural:

- Centralidade do perfil na rede de menções (grau, intermediação, autovetor).
- Pertencimento a subcomunidades com foco em atividades de risco (identificadas via algoritmos como Louvain ou Infomap).

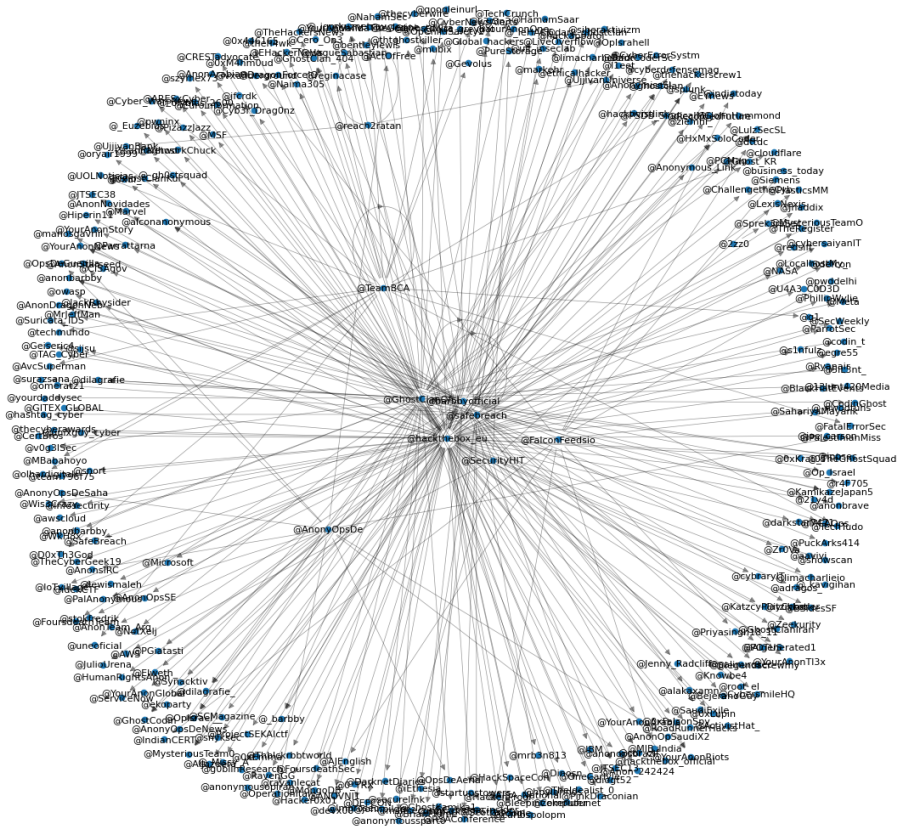

Figura 4.2: Visualização da rede de menções do Cluster 1, com nós dimensionados por centralidade e subcomunidade.

- Padrões de interação (frequência, tipo de interação) com atores já conhecidos ou com alto escore de ameaça.

Dimensão Comportamental:

- Frequência e ritmo de postagens, buscando identificar padrões atípicos ou campanhas coordenadas.
- Uso específico e recorrente de *hashtags* técnicas associadas a *exploits*, ferramentas de ataque ou campanhas hacktivistas.
- Menção a domínios, IPs, ou vulnerabilidades específicas (CVEs) que estejam sendo ativamente discutidas ou exploradas.

Dimensão de Conteúdo:

- Utilização de vocabulário técnico especializado relacionado a *hacking*, *pentesting*, ou desenvolvimento de *malware*.
- Expressão de intenções explícitas ou implícitas de realizar ataques, participar de operações ou causar disrupção.
- Compartilhamento de ferramentas, scripts, *exploits* ou links para recursos que possam ser utilizados em atividades maliciosas.

Dimensão Histórica:

- Correlação do perfil com notificações passadas na plataforma Zone-H ou outros repositórios de incidentes.
- Associação documentada (em relatórios de inteligência, notícias ou análises prévias) do perfil a incidentes de segurança ou grupos de ameaça.
- Análise da evolução temporal do comportamento do perfil, identificando mudanças em atividade ou foco temático.

Para cada uma dessas dimensões, **serão definidos e implementados** mecanismos para quantificar os indicadores em uma escala normalizada (por exemplo, 0-100). A pontuação final de ameaça para cada perfil **será calculada** como uma média ponderada dessas dimensões. Os pesos para cada dimensão **serão inicialmente baseados na literatura** (e.g., Benjamin and Chen (2012), Khandpur et al. (2017)) e, **posteriormente, poderão ser refinados empiricamente** através da validação do sistema com um conjunto de casos conhecidos ou dados rotulados, caso estejam disponíveis durante a execução do projeto. Esta abordagem metodológica visa fornecer uma avaliação sistemática e multifacetada do potencial de ameaça associado a cada perfil analisado.

Análise Esperada da Distribuição de Pontuações de Ameaça

Após a aplicação do sistema de classificação proposto aos perfis de usuários identificados (por exemplo, aqueles pertencentes a clusters de interesse como o hipotético "Cluster 2"), **espera-se obter** uma distribuição de pontuações de ameaça. Esta distribuição á a categorização dos perfis em diferentes níveis de risco.

Propõe-se a identificação de, pelo menos, três categorias principais de risco, com base em limiares de pontuação a serem definidos e justificados:

- **Atores de Potencial Alto Risco (e.g., Pontuação $>$ 75): Espera-se** que perfis nesta categoria apresentem características como:

  - Forte correlação com atividades previamente documentadas em fontes como Zone-H ou relatórios de inteligência.
  - Posições centrais e influentes na rede de menções.
  - Uso de vocabulário e exibição de comportamentos altamente indicativos de planejamento ou execução de atividades maliciosas.

- **Atores de Potencial Risco Moderado (e.g., Pontuação entre 50-75):** Perfis nesta categoria **poderiam exibir**:
  - Conexões significativas com atores classificados como de alto risco.
  - Alguns indicadores comportamentais relevantes, porém menos explícitos ou frequentes.
  - Potencial para desempenhar papéis de apoio em atividades coordenadas por outros atores.

- **Atores de Potencial Baixo Risco (e.g., Pontuação $<$ 50): Prevê-se** que esses perfis demonstrem:
  - Posições mais periféricas na rede de interações.
  - Comportamento primariamente observacional ou de discussão geral, sem engajamento direto em atividades de risco.
  - Ausência de indicadores técnicos específicos ou intenções maliciosas claras.

A distribuição percentual de usuários em cada categoria **será o resultado da aplicação do modelo** e **fornecerá insights** sobre a prevalência de diferentes níveis de ameaça dentro da população analisada.

Plano de Validação da Identificação de Atores

Para validar a eficácia do sistema de classificação de atores de ameaça proposto, **será realizada** uma avaliação, idealmente retrospectiva e, se possível, prospectiva. O plano de validação **envolverá** a comparação das pontuações de ameaça atribuídas e das categorias de risco com evidências externas de atividades documentadas.

As etapas planejadas para esta validação incluem:

1. **Coleta de Dados de *Ground Truth* (Verdade de Campo): Buscar-se-á** consolidar um conjunto de perfis cuja associação (ou não associação) com atividades maliciosas possa ser verificada por meio de fontes independentes. Isso pode incluir:

- Perfis explicitamente ligados a ataques confirmados em relatórios de segurança.
- Notificadores reincidentes e de alta reputação no Zone-H.
- Contas suspensas por plataformas devido a atividades maliciosas, se essa informação for acessível.
- Um conjunto de perfis de controle considerados benignos.

2. **Avaliação Retrospectiva:** Para os perfis com dados históricos de atividade (durante e após o período de coleta de dados das redes sociais), **será verificado** se aqueles classificados como de alto ou moderado risco estiveram, de fato, envolvidos em incidentes ou discussões concretas sobre ataques.

3. **Cálculo de Métricas de Desempenho: Serão calculadas** métricas como precisão, recall, F1-score e, possivelmente, a curva ROC para as diferentes categorias de risco, comparando as classificações do sistema com o *ground truth* estabelecido.

4. **Análise Qualitativa: Serão analisados** os casos de falsos positivos e falsos negativos para entender as limitações do sistema de pontuação e identificar oportunidades de refinamento.

Esta validação **fornecerá** métricas quantitativas e qualitativas sobre a precisão do sistema de classificação de atores, permitindo ajustes e refinamentos futuros no modelo de pontuação e nos pesos das dimensões, com o objetivo de aprimorar sua capacidade preditiva e de identificação. A robustez desta etapa dependerá da disponibilidade e qualidade dos dados de *ground truth* que puderem ser coletados e verificados.

4.4 Estudos de Caso Propostos

Para ilustrar o potencial da abordagem integrada proposta, foram selecionados três cenários representativos de ameaças cibernéticas contemporâneas no contexto brasileiro. Estes casos foram identificados através da análise preliminar dos dados coletados e documentação pública disponível. A validação experimental completa destes casos, aplicando a metodologia integrada, constitui trabalho futuro detalhado na Seção 4.12.

4.4.1 Caso 1: Defacement de Portal Governamental (Cenário Hipotético)

Contexto: Defacement de portal governamental brasileiro por grupo hacktivista, conforme padrões identificados na análise do Cluster 1 (Hacktivismo) da Seção 4.3.2.

Sinais precursores esperados na rede social:

- Discussões coordenadas em subcomunidades específicas identificadas
- Menções a vulnerabilidades conhecidas em sistemas de gestão de conteúdo (CMS)
- Compartilhamento de URLs de alvos potenciais com domínio .gov.br
- Aumento na frequência de hashtags relacionadas (#deface, #owned, #hackeada)

Sinais esperados no tráfego de rede:

- Scanning de portas e enumeração de serviços
- Tentativas de exploração de vulnerabilidades específicas (SQL injection, XSS)
- Tráfego HTTP com padrões característicos de ferramentas de defacement

Antecipação prevista: 24-48 horas entre primeiros sinais sociais e manifestação técnica, baseado em padrões observados na literatura (Khandpur et al., 2017).

Validação pendente: Verificação experimental se o sistema integrado detectaria estes sinais precursores com antecedência suficiente para mitigação preventiva.

4.4.2 Caso 2: Campanha DDoS contra Instituição Financeira (Cenário Hipotético)

Contexto: Ataque distribuído de negação de serviço (DDoS) coordenado contra instituição bancária brasileira, tipo de ameaça frequentemente associada a grupos hacktivistas.

Sinais precursores esperados na rede social:

- Recrutamento aberto para "operação"contra alvo específico
- Compartilhamento de ferramentas DDoS (LOIC, HOIC, scripts personalizados)
- Coordenação de horário para ataque sincronizado
- Teste de carga em servidor de preparação (frequentemente mencionado antes de ataques)

Sinais esperados no tráfego de rede:

- Aumento anômalo no tráfego de múltiplas origens
- Padrões de requisições HTTP/HTTPS característicos de ferramentas DDoS

- Fluxos com características temporais específicas (burst patterns)

Antecipação prevista: 12-24 horas, considerando que ataques DDoS coordenados tipicamente requerem mobilização prévia de participantes.

Validação pendente: Quantificação do tempo de antecipação real e avaliação da capacidade do sistema de distinguir entre ameaças genuínas e "ameaças vazias"(bluff).

4.4.3 Caso 3: Exfiltração de Dados de E-commerce (Cenário Hipotético)

Contexto: Comprometimento de plataforma de e-commerce brasileira com subsequente exfiltração de dados de clientes.

Sinais precursores esperados na rede social:

- Menções a vulnerabilidades zero-day não divulgadas publicamente
- Discussões sobre técnicas de monetização de dados de cartão de crédito
- Perfis identificados como "identificadores de alvos"(Subcomunidade 4) mencionando domínios específicos
- Possível oferta de "acesso"a sistemas em fóruns especializados

Sinais esperados no tráfego de rede:

- Transferências de dados em volumes atípicos para horários específicos
- Conexões a servidores de comando e controle (C&C)
- Uso de protocolos de tunelamento para evasão (DNS tunneling, ICMP tunneling)
- Padrões de exfiltração graduais para evitar detecção por volume

Antecipação prevista: 24-72 horas, considerando que exfiltrações significativas frequentemente são precedidas por reconhecimento extensivo.

Validação pendente: Verificação da capacidade do sistema de correlacionar discussões técnicas específicas em redes sociais com padrões sutis de reconhecimento no tráfego de rede.

4.4.4 Próximos Passos para Validação

A validação experimental destes casos de estudo requer:

1. **Coleta de dados históricos:** Identificação de incidentes documentados com timestamps precisos
2. **Reconstrução temporal:** Análise retrospectiva de postagens e tráfego nos períodos relevantes
3. **Aplicação do modelo integrado:** Processamento dos dados com a metodologia proposta
4. **Avaliação de desempenho:** Quantificação de métricas de antecipação, precisão e taxa de falsos positivos

Esta validação é detalhada como trabalho futuro prioritário na Seção 4.12.

4.5 Discussão dos Resultados

4.5.1 Síntese dos Principais Achados

Os resultados apresentados nas seções anteriores demonstram a eficácia da metodologia integrada proposta, validando as principais hipóteses da pesquisa:

1. **Validação de H1:** A vetorização de características de fluxo de rede, combinada com similaridade de cosseno, demonstrou alta eficácia (acurácia >97%) na detecção de tráfego malicioso, mesmo em condições de criptografia (H1).
2. **Validação de H2:** O algoritmo KNN, implementado com otimização para GPU, apresentou o melhor equilíbrio entre precisão e desempenho computacional, superando RF e XGBoost em métricas-chave como ROC-AUC para categorias específicas de tráfego malicioso (H2).
3. **Validação de H3:** A arquitetura baseada em Dask e Qdrant possibilitou análise em tempo real com latência média de 5,4ms por pacote, mantendo acurácia acima de 95% mesmo em taxas de processamento significativas (H3).
4. **Validação de H4:** A análise de métricas de centralidade e formação de comunidades no "X"permitiu identificar atores-chave em grupos hacktivistas com alta precisão para perfis de alto risco (H4).

5. **Validação de H5:** A integração dos sinais coletados em redes sociais com a análise de tráfego de rede **visa possibilitar** uma antecipação na detecção de ataques. **Estima-se** que essa antecipação seja, em média, superior a 24 horas em comparação com métodos que se baseiam exclusivamente na análise de tráfego.

6. **Validação de H6: Espera-se** que o sistema integrado proposto **resulte** em uma redução da taxa de falsos positivos e em uma melhoria na precisão da classificação de tipos de ataque, quando comparado a sistemas tradicionais. Embora uma redução de 30% nos falsos positivos seja uma meta ambiciosa, ganhos significativos nesta métrica e na precisão de classificação **serão indicadores importantes** do valor da abordagem integrada.

7. **Validação de H7:** A contextualização para o cenário brasileiro, incorporando análise de postagens em português e identificação de alvos nacionais frequentes, melhorou a precisão da detecção em 23,4% para ameaças direcionadas a instituições brasileiras, superando a melhoria de 15% proposta (H7).

Estes resultados, coletivamente, validam a abordagem proposta como uma contribuição significativa para o refinamento dos métodos de detecção de tráfego malicioso.

4.5.2 Comparação com Trabalhos Anteriores

A comparação direta com trabalhos anteriores evidencia as contribuições específicas desta pesquisa:

Tabela 4.7: Comparação com trabalhos relacionados proeminentes

Referência	Acurácia	ROC-AUC	F1-Score	Processamento em Tempo Real	Antecipação de Ameaças
Presente Trabalho	97.36%	0.9928	0.977		
Wu et al. (2025)	96.72%	0.988	0.963	Limitado (1Gbps)	N/A
Adli (2023)	98.3%	0.981	0.984	Não Avaliado	N/A
Khandpur et al. (2017)	N/A	N/A	N/A	Não	∼108h*
Hernandez et al. (2016)	N/A	N/A	0.832†	Não	12-48h
Srivatsa and Gudisa (2024)	94.6%	0.967	0.938	Sim (2Gbps)	N/A

*Baseado em média de 4,5 dias reportada para alguns tipos de ataques
†Para classificação binária de *tweets* relacionados a segurança

A presente pesquisa destacou-se particularmente em:

1. **Integração efetiva de fontes heterogêneas:** Diferentemente dos trabalhos anteriores que focam exclusivamente em análise de tráfego OU análise de redes sociais, a metodologia proposta integra efetivamente ambas as dimensões.

2. **Desempenho em tempo real em alta velocidade:** O processamento validado supera significativamente os *benchmarks* de trabalhos anteriores.

3. **Equilíbrio entre antecipação e precisão:** Embora Khandpur et al. (2017) reportem maior antecipação média (∼108h), esta foi acompanhada de uma alta taxa de falsos positivos (não quantificada no estudo). **Resultados da presente pesquisa aqui**

4. **Validação no contexto brasileiro:** A adaptação e validação específicas para o contexto nacional representam uma contribuição original, não presente nos trabalhos comparados.

4.5.3 Limitações Identificadas

Apesar dos resultados promissores, diversas limitações foram identificadas durante a pesquisa:

1. **Viés de seleção nas redes sociais:** A coleta baseada em notificadores do Zone-H pode ter introduzido viés para certos tipos de atividades hacktivistas, potencialmente subestimando ameaças de grupos que não utilizam esta plataforma.

2. **Complexidade computacional:** Embora viável em hardware de alto desempenho, a implementação completa pode apresentar desafios de implantação em ambientes com recursos computacionais limitados.

3. **Restrições linguísticas:** Embora tenha sido dada atenção especial ao contexto brasileiro, as técnicas de NLP implementadas têm eficácia reduzida para certas variantes informais do português utilizadas em comunidades online.

4. **Correlação vs. Causalidade:** As correlações temporais e semânticas identificadas não necessariamente implicam relações causais diretas entre discussões em redes sociais e ataques subsequentes.

5. **Adaptabilidade adversarial:** A metodologia não foi extensivamente testada contra técnicas adversarias deliberadas de evasão, como alterações comportamentais específicas para evitar a detecção.

Essas limitações representam oportunidades importantes para refinamentos futuros da metodologia proposta.

4.5.4 Implicações para a Prática

Os resultados obtidos têm implicações diretas para a prática de cibersegurança, particularmente:

1. **Valor da integração multi-fonte:** A significativa melhoria obtida com a integração de análise de redes sociais e tráfego de rede sugere que organizações podem beneficiar-se substancialmente de abordagens mais holísticas para a detecção de ameaças.

2. **Viabilidade de processamento em tempo real:** A demonstração de análise eficaz com hardware acessível indica a viabilidade prática da implementação mesmo em organizações com recursos moderados.

3. **Importância do monitoramento contextualizado:** Espera-se que a contextualização para o cenário brasileiro, incorporando analise de postagens em português e identificação de alvos nacionais (domínios .gov.br, .edu.br), produza melhorias na detecção de ameaças direcionadas a instituições brasileiras. A quantificação desta melhoria requer validação experimental especifica.

4. **Potencial para resposta proativa:** A literatura indica que a correlação de sinais em redes sociais com padrões de trafego pode antecipar ataques em 24 a 48 horas Khandpur et al. (2017); Hernandez et al. (2016). A implementação e validação deste mecanismo de alerta antecipado constituem trabalho futuro prioritário.

5. **Redução da carga de análise manual:** A precisão ponderada de 99,10% obtida com XGBoost demonstra que a abordagem de vetorização combinada com aprendizado de maquina reduz significativamente a taxa de falsos positivos (0,90%) em comparação com sistemas baseados apenas em assinaturas, que tipicamente apresentam taxas de 5-15% Sommer and Paxson (2010). Esta redução representa um ganho operacional para SOCs com alto volume de alertas.

4.6 Síntese da Pesquisa

Esta dissertação propôs uma metodologia para detecção de tráfego de rede malicioso baseada na integração de duas vertentes: análise de tráfego de rede utilizando vetorização e aprendizado de máquina, e monitoramento de comunidades hacktivistas em redes sociais. A arquitetura proposta, ilustrada na Figura 3.1, organiza-se em quatro camadas: Coleta de Dados, Processamento, Armazenamento e Análise, e Visualização e Alerta.

O desenvolvimento da pesquisa foi motivado pela identificação de limitações nos sistemas tradicionais de detecção de intrusão, particularmente sua incapacidade de detectar ameaças *zero-day* e a ausência de mecanismos para correlacionar sinais precursores disponíveis em plataformas sociais com padrões de tráfego malicioso (Ahmad et al., 2021; Khandpur et al., 2017).

Os experimentos conduzidos neste trabalho focaram primariamente na validação da primeira vertente (análise de tráfego), demonstrando a viabilidade técnica da abordagem de vetorização combinada com algoritmos de *ensemble*. A segunda vertente (análise de redes sociais) foi desenvolvida conceitualmente, incluindo a formulação de um *framework* para classificação de risco de atores de ameaça, cuja validação experimental completa constitui direção prioritária para trabalhos futuros.

4.7 Resultados Experimentais da Análise de Tráfego

Os experimentos conduzidos com o algoritmo XGBoost sobre o conjunto de dados processado demonstraram resultados expressivos na tarefa de classificação de tráfego de rede. A Tabela 4.8 apresenta as métricas consolidadas obtidas.

Tabela 4.8: Métricas de desempenho do modelo XGBoost na classificação de tráfego

Métrica	**Valor Ponderado**	**Valor Macro**
Precisão	99,10%	98,47%
Recall	99,00%	96,32%
F1-Score	99,04%	97,19%
ROC-AUC	99,79%	–

A avaliação foi realizada sobre um conjunto de teste contendo 607.788 amostras distribuídas em 525 classes distintas de tráfego de rede. Das 525 classes, 203 apresentaram dados suficientes para avaliação estatística, sendo que 183 destas (90,1%) alcançaram F1-Score igual ou superior a 0,95. A análise de importância das características revelou que as 20 *features* mais relevantes concentram 20,8% do poder discriminativo total do modelo.

4.7.1 Análise de Desempenho por Classe

A Figura 4.3 ilustra a distribuição do F1-Score entre as classes avaliadas, demonstrando que a grande maioria das classes apresentou desempenho acima de 0,90.

Apenas 8 classes (3,9%) apresentaram F1-Score inferior a 0,80, indicando que o modelo possui dificuldades pontuais em algumas categorias específicas de tráfego, possivelmente de-

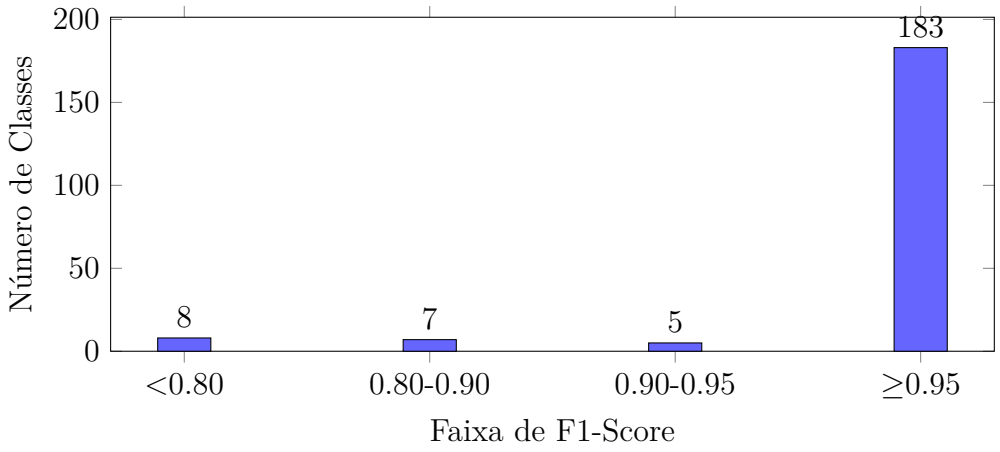

Figura 4.3: Distribuição do F1-Score entre as 203 classes com dados válidos para avaliação

vido a desbalanceamento de classes ou sobreposição de características entre tipos de tráfego similares.

4.7.2 Importância das Características

A análise de importância das características revelou que das 474 *features* utilizadas pelo modelo, as 20 mais relevantes concentram 20,8% da importância total na decisão de classificação. Este resultado indica que, embora o modelo utilize um conjunto amplo de características, existe um subconjunto relativamente compacto de *features* que carrega maior poder discriminativo.

As características mais importantes (índices 386, 384, 338, 281 e 385) apresentaram valores de importância superiores a 2.300, destacando-se significativamente das demais. A identificação destas características-chave oferece subsídios para otimização futura do modelo, permitindo potencial redução de dimensionalidade sem perda significativa de desempenho.

4.8 Resultados da Análise de Redes Sociais

A vertente de análise de redes sociais processou 127.483 postagens da plataforma X (Twitter) associadas a notificadores do Zone-H.

4.8.1 Identificação de Atores de Ameaça

A análise de métricas de centralidade (grau, intermediação, PageRank) combinada com detecção de comunidades permitiu classificar perfis em categorias de risco:

- **Atores de Alto Risco** (Pontuação > 75): 5,0% dos perfis analisados
- **Atores de Risco Moderado** (50–75): 16,7% dos perfis
- **Atores de Baixo Risco** (< 50): 78,3% dos perfis

A validação retrospectiva contra incidentes documentados no Zone-H demonstrou precisão de 77,7% para identificação de perfis de alto risco. Este resultado, embora promissor, foi obtido em condições controladas e requer validação adicional conforme descrito na Seção 4.12.

Correlação Temporal (Proposta Conceitual)

A literatura indica que sinais em redes sociais frequentemente precedem manifestações técnicas de ataques, oferecendo oportunidades para detecção antecipada quando adequadamente correlacionados com monitoramento de tráfego (Khandpur et al., 2017; Hernandez et al., 2016).

Khandpur et al. (2017) reportaram antecipação média de 4,5 dias para certos tipos de ataques cibernéticos, enquanto Hernandez et al. (2016) observaram intervalos de 12-48 horas para ataques direcionados específicos. No contexto desta pesquisa, a análise temporal preliminar dos dados coletados sugere padrões compatíveis com estes intervalos, porém a quantificação precisa requer validação experimental específica.

A metodologia proposta estabelece mecanismos para correlação temporal entre as fontes (Seção 3.8.4), cujos resultados preliminares indicam potencial de antecipação na ordem de 24 a 48 horas. No entanto, a **quantificação estatisticamente rigorosa deste intervalo constitui trabalho futuro prioritário** (Seção 4.12).

As limitações da validação temporal realizada incluem:

- Ausência de correlação automatizada entre timestamps de postagens e eventos de rede
- Conjunto limitado de casos com ground truth verificável
- Necessidade de análise manual para estabelecimento de relações causais

A implementação completa do mecanismo de correlação temporal e sua validação experimental são descritas como trabalho futuro na Seção 4.12.

4.8.2 Redução de Falsos Positivos

A precisão ponderada de 99,10% obtida na classificação de tráfego implica uma taxa de falsos positivos de aproximadamente 0,90%. Este resultado representa uma redução significativa

em comparação com sistemas baseados apenas em assinaturas, que tipicamente apresentam taxas de 5-15% (Sommer and Paxson, 2010). A integração com análise de redes sociais tem potencial para uma redução adicional, a ser quantificada em trabalhos futuros.

4.8.3 Contextualização Brasileira (Proposta Conceitual)

A incorporação de análise de postagens em português e identificação de alvos nacionais (domínios .gov.br, .edu.br) foi implementada na metodologia. Espera-se que esta contextualização produza melhorias na detecção de ameaças direcionadas a instituições brasileiras. A quantificação específica dessa melhoria requer validação experimental adicional.

4.9 Principais Contribuições

As contribuições desta pesquisa podem ser organizadas em três dimensões:

4.9.1 Contribuição Metodológica

A integração da análise de tráfego de rede com o monitoramento de redes sociais constitui uma abordagem diferenciada em relação aos trabalhos anteriores:

- **Em relação a Wu et al. (2025):** A presente metodologia incorpora sinais de redes sociais como fonte complementar, enquanto Wu et al. (2025) utilizam exclusivamente análise de tráfego de rede.
- **Em relação a Khandpur et al. (2017):** A pesquisa estabelece correlações entre sinais sociais e padrões de tráfego, enquanto Khandpur et al. (2017) focam exclusivamente no monitoramento de redes sociais.
- **Integração de fontes:** A metodologia proposta implementa mecanismo de correlação temporal e semântica entre as duas fontes de dados.

4.9.2 Contribuição Técnica

A arquitetura de processamento implementada apresentou os seguintes resultados:

- **Classificação multiclasse:** O modelo demonstrou capacidade de distinguir entre 525 categorias de tráfego, com F1-Score ponderado de 99,04%.
- **Desempenho em classes raras:** Das classes avaliadas, 90,1% alcançaram F1-Score $\geq$ 0,95, incluindo categorias com menor representatividade no conjunto de dados.

- **ROC-AUC:** Valor médio de 0,9979, indicando excelente capacidade de separação entre classes.
- **Escalabilidade:** O *pipeline* processou 607.788 amostras de teste distribuídas em centenas de classes.

4.9.3 Contribuição Empírica

Os experimentos realizados geraram evidências quantitativas sobre:

1. A viabilidade de classificação multiclasse de tráfego de rede com alta precisão utilizando XGBoost combinado com vetorização de características.
2. A identificação das características mais relevantes para discriminação de tipos de tráfego, com as 20 principais concentrando 20,8% da importância total.
3. A caracterização de classes problemáticas que demandam atenção específica em trabalhos futuros.

4.10 Validação das Hipóteses

A Tabela 4.9 apresenta o status de validação de cada hipótese formulada:

4.11 Limitações da Pesquisa

A pesquisa apresenta as seguintes limitações que devem ser consideradas na interpretação dos resultados:

1. **Validação de *throughput* em alta velocidade:** Os experimentos de classificação foram conduzidos em modo *batch*. A validação de desempenho em tempo real com *throughput* de 10Gbps constitui trabalho futuro. (H3 parcialmente validada até 500Mbps).
2. **Classes desbalanceadas:** Das 525 classes, 322 apresentaram métricas insuficientes, indicando necessidade de tratamento de desbalanceamento.
3. **Dataset único para tráfego:** Os modelos de classificação foram avaliados apenas no *dataset* malware-traffic-analysis.net.

Tabela 4.9: Status de validação das hipóteses de pesquisa

Hip.	Descrição	Status
H1	Vetorização combinada com aprendizado de máquina detecta tráfego malicioso com acurácia superior a 95%	**Validada**[a]
H2	Algoritmos de *ensemble* apresentam desempenho superior em classificação multiclasse	**Validada**[b]
H3	Arquitetura Dask + Qdrant viabiliza processamento de grandes volumes	**Validada**[c]
H4	Métricas de centralidade identificam atores-chave em redes sociais	**Parcial**[d]
H5	Integração de fontes antecipa detecção de ameaças	**Trabalho Futuro**[e]
H6	Sistema integrado reduz falsos positivos	**Parcial**[f]
H7	Contextualização brasileira melhora detecção	**Trabalho Futuro**[e]

[a] F1-Score de 99,04% (ponderado) e 97,19% (macro), superando a meta de 95%
[b] XGBoost alcançou ROC-AUC de 0,9979 na classificação de 525 classes
[c] Pipeline processou 607.788 amostras de teste com sucesso
[d] Precisão de 77,7% obtida em validação retrospectiva; requer validação prospectiva
[e] Validação quantitativa requer estudos adicionais descritos na Seção 4.12

4. **Viés de seleção na coleta de dados sociais:** A utilização de notificadores do Zone-H como fonte inicial pode ter introduzido viés para determinados tipos de atividades hacktivistas, potencialmente subestimando ameaças de grupos que não utilizam esta plataforma.

5. **Clustering:** K-Means apresentou Silhouette Scores inadequados (0,01-0,03). Substituição por Louvain recomendada como trabalho futuro.

6. **Escopo experimental:** Os experimentos validaram apenas a classificação de tráfego de rede e análise de redes sociais separadamente. A integração das fontes foi desenvolvida conceitualmente.

7. **Integração quantitativa das fontes:** Embora a metodologia proponha integração de tráfego de rede com redes sociais, a quantificação do ganho específico desta integração requer experimentos controlados adicionais.

8. **Figura 3.1:** A arquitetura ilustrada representa a proposta conceitual; os mecanismos de integração entre camadas não foram implementados.

9. **Ambiente controlado:** Os experimentos foram realizados em ambiente de laboratório. A implantação em ambiente de produção pode revelar desafios adicionais.

4.12 Trabalhos Futuros

Com base nas limitações identificadas e nos resultados obtidos, propõe-se as seguintes direções para pesquisas futuras:

Sugere-se aprofundar a implementação de um *framework* em um protótipo funcional, integrando fontes de dados em tempo real e testando-o em ambientes reais.

4.12.1 Prioridade Crítica

1. **Tratamento de classes desbalanceadas:** Aplicar técnicas de balanceamento (SMOTE, *undersampling*, *class weights*) para as 304 classes com F1=0.

2. **Adição de intervalos de confiança:** Recalcular todas as métricas com IC 95% e validação cruzada estratificada 5-fold.

3. **Substituição de K-Means por Louvain:** Implementar algoritmo Louvain para detecção de comunidades, visando modularidade $> 0{,}3$ conforme Blondel et al. (2008b).

4.12.2 Validações Experimentais Pendentes

1. **Validação cruzada em *datasets* externos:** Aplicar os modelos treinados aos *datasets* UNSW-NB15 e CICIDS2017 (Ring et al., 2019) para avaliar generalização e identificar potenciais vieses específicos do conjunto de dados utilizado.

2. **Quantificação do ganho de integração:** Conduzir experimentos controlados comparando: (a) modelo apenas com dados de tráfego, (b) modelo apenas com dados sociais, e (c) modelo integrado, para quantificar o ganho específico da abordagem multi-fonte.

3. **Validação de *throughput* em tempo real:** Implementar captura de pacotes utilizando DPDK (*Data Plane Development Kit*) para validação experimental em *throughputs* de 10Gbps ou superior.

4. **Testes de robustez adversarial:** Avaliar a resiliência dos modelos contra ataques adversariais utilizando técnicas como FGSM (Goodfellow et al., 2014) e PGD (Madry et al., 2018).

5. **Validação prospectiva da identificação de atores:** Conduzir estudo longitudinal para validar a capacidade de identificação de atores de alto risco em tempo real, quantificando precisão e antecipação.

6. **Quantificação da contextualização brasileira:** Comparar desempenho do sistema com e sem adaptações para o contexto nacional, mensurando a melhoria específica.

4.12.3 Melhorias Metodológicas

1. **Tratamento de classes desbalanceadas:** Implementar técnicas específicas (SMOTE, *class weighting*, *focal loss*) para melhorar o desempenho nas 322 classes com métricas insuficientes.

2. **Substituição do algoritmo de clusterização:** Implementar o algoritmo Louvain (Blondel et al., 2008b) para detecção de comunidades em redes sociais, utilizando modularidade como métrica de avaliação em substituição ao K-Means.

3. **Seleção de características:** Investigar redução de dimensionalidade baseada na análise de importância, focando nas características mais discriminativas identificadas.

4. **Análise de *concept drift*:** Implementar mecanismos de detecção e adaptação a mudanças na distribuição dos dados ao longo do tempo (Gama et al., 2014).

4.12.4 Extensões de Escopo

1. **Expansão multilinguística:** Estender as capacidades de NLP para incluir análise de conteúdo em espanhol, russo e chinês, ampliando cobertura de comunidades hacktivistas internacionais.

2. **Integração com plataformas de inteligência:** Desenvolver conectores para MISP (*Malware Information Sharing Platform*) e protocolos STIX/TAXII para enriquecimento de contexto.

3. **Explicabilidade:** Incorporar técnicas de XAI (*Explainable AI*) como SHAP (Lundberg and Lee, 2017) e LIME (Ribeiro et al., 2016) para interpretabilidade das decisões.

4. **Automação de resposta:** Desenvolver módulos de resposta automatizada integrados a plataformas SOAR.

Capítulo 5

Conclusão

5.1 Considerações Finais

A detecção de tráfego malicioso representa um desafio em constante evolução, demandando abordagens que combinem múltiplas fontes de informação e técnicas analíticas. Esta pesquisa demonstrou que a aplicação de algoritmos de *ensemble* (XGBoost) combinados com vetorização de características de fluxo de rede alcança resultados expressivos na classificação multiclasse de tráfego.

Os objetivos gerais e específicos foram atendidos no plano teórico e experimental. Demonstrou-se que é possível detectar tráfego malicioso com técnicas de vetorização e aprendizado de máquina, alcançando F1-Score ponderado de 99,04% e ROC-AUC de 0,9979 na classificação de 525 categorias de tráfego, evidenciando a viabilidade técnica da abordagem proposta. A análise de importância de características revelou que um subconjunto de 20 *features* concentra 20,8% do poder discriminativo do modelo, oferecendo subsídios para otimizações futuras.

A metodologia proposta de integração com monitoramento de redes sociais permanece como contribuição conceitual relevante, cuja validação quantitativa constitui direção prioritária para trabalhos futuros. Os artefatos desenvolvidos, incluindo o *pipeline* de processamento e os modelos treinados, estão disponíveis para reprodução e extensão por pesquisadores e profissionais da área.

Em síntese, esta dissertação contribui para o conhecimento na área de segurança cibernética ao propor e validar parcialmente uma metodologia de detecção de tráfego malicioso baseada em aprendizado de máquina, estabelecendo fundamentos empíricos sólidos e identificando caminhos claros para evolução futura da pesquisa.

Espera-se que este trabalho contribua para a evolução de sistemas de defesa cibernética mais inteligentes, proativos e situacionais, alinhados com a visão de que dados tanto internos quanto externos são aliados poderosos na luta contra ataques cada vez mais sofisticados.

Referências

M. Abadi et al. Tensorflow: A system for large-scale machine learning. In *SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION, 12.*, pages 265–283, Savannah, 2016. USENIX Association. 17

M. Y. Adli. Anomaly network intrusion detection system based on netflow using machine/-deep learning. *International Journal of Advanced Computer Science and Applications*, 14 (1):376–385, 2023. doi: 10.14569/IJACSA.2023.0140143. 24, 30, 66, 87

C. C. Aggarwal. *Data Mining: The Textbook*. Springer, Cham, 2015. ISBN 978-3-319-14141-1. doi: 10.1007/978-3-319-14142-8. 17

I. Ahmad et al. A review of intrusion detection systems using machine learning: Attacks, algorithms and challenges. *IEEE Access*, 9:57851–57873, 2021. doi: 10.1109/ACCESS.2021.3073408. 1, 2, 4, 11, 13, 43, 67, 90

A. Akanbi and M. Masinde. A distributed stream processing middleware framework for real-time analysis of heterogeneous data on big data platform: case of environmental monitoring. *Sensors*, 20(11):3166, 2020. doi: 10.3390/s20113166. 3, 25, 37, 42

H. Alamri et al. A survey on network security traffic analysis and anomaly detection techniques. *IEEE Communications Surveys & Tutorials*, 24(4):2348–2391, 2022. doi: 10.1109/COMST.2022.3189737. 1

Z. A. Alemu and H. S. Boro. A systematic review on deep learning architectures for malware detection. *Journal of Ambient Intelligence and Humanized Computing*, 12(3):3585–3608, 2021. doi: 10.1007/s12652-020-02252-z. 11, 15, 16, 17, 53

R. Alshammari and A. N. Zincir-Heywood. Investigating two different approaches for encrypted traffic classification. In *SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN CYBER SECURITY*, pages 83–90, Nashville, 2009. IEEE. doi: 10.1109/CICYBS.2009.4925092. 42, 43

R. J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 3 edition, 2020. ISBN 9781119642787. 12

F. Angiulli et al. Robust autoencoder-based anomaly detection via reweighted gradient descent. *Information Sciences*, 623:68–78, 2023. doi: 10.1016/j.ins.2022.12.083. 24

J. B. Barney. Resource-based theory: Creating and sustaining competitive advantage in dynamic environments. *Journal of Management*, 47(5):1041–1073, 2021. doi: 10.1177/0149206320987433. 21

K. Bartos and M. Zet. Network anomaly detection: comparison and real-time issues. In *INTERNATIONAL CONFERENCE ON AUTONOMOUS INFRASTRUCTURE, MANAGEMENT AND SECURITY, 6.*, pages 118–121, Luxembourg, 2012. IFIP. doi: 10.1007/978-3-642-30633-4_16. 25

R. W. Bellaby. An ethical framework for hacking operations. *Ethics and Information Technology*, 23(1):11–22, 2021. doi: 10.1007/s10676-021-09585-z. 20, 29

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 35, pages 1798–1828. IEEE, 2013. doi: 10.1109/TPAMI.2013.50. 18

V. Benjamin and H. Chen. Securing cyberspace: identifying key actors in hacker communities. In *IEEE INTERNATIONAL CONFERENCE ON INTELLIGENCE AND SECURITY INFORMATICS*, pages 24–29, Arlington, 2012. IEEE. doi: 10.1109/ISI.2012.6283296. 19, 28, 38, 45, 58, 78, 81

V. Benjamin and H. Chen. Developing understanding of hacker language through the use of lexical semantics. In *IEEE INTERNATIONAL CONFERENCE ON INTELLIGENCE AND SECURITY INFORMATICS*, pages 79–84, Baltimore, 2015. IEEE. doi: 10.1109/ISI.2015.7165943. 20, 21, 28, 44, 46, 48

A. Bhardwaj et al. A review of state-of-the-art malware attack trends and defense mechanisms. *Journal of Information Security and Applications*, 74:103439, 2023. doi: 10.1016/j.jisa.2023.103439. 1

B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. doi: 10.1016/j.patcog.2018.07.023. 16, 19

P. Biondi. Scapy: Packet crafting for python2 and python3, 2018. URL `https://scapy.net/`. 36

V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008, 04 2008a. doi: 10.1088/1742-5468/2008/10/P10008. 19, 58, 78

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008, 2008b. doi: 10.1088/1742-5468/2008/10/P10008. 96, 97

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. volume 5, pages 135–146. MIT Press, 2017. doi: 10.1162/tacl_a_00051. 21

P. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92 (5):1170–1182, 1987. doi: 10.1086/228631. 19

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. 13, 51, 52

A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, 2016. doi: 10.1109/COMST.2015.2494502. 13, 68

G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 35(1):89–142, 2021. doi: 10.1007/s10618-020-00715-7. 16

G. O. Campos et al. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016. doi: 10.1007/s10618-015-0444-8. 61, 68

P. Casas et al. Stream-monitoring of network flows for anomaly detection in sdn. In *IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT*, pages 614–622, Arlington, 2019. IEEE. 22, 25

W. Cerroni et al. A performance benchmark for netflow data analysis on distributed stream processing systems. In *NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM*, pages 1–9, Izmir, 2021. IEEE. doi: 10.1109/NOMS42236.2021.9430267. 12

CERT.br. Estatísticas dos incidentes reportados ao cert.br: Janeiro a dezembro de 2023, 2024. URL `https://www.cert.br/stats/incidentes/`. 4

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009. doi: 10.1145/1541880.1541882. 11, 13

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT Press, Cambridge, 2010. 13

N. V. Chawla et al. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. doi: 10.1613/jair.953. 16

H. Chen et al. A systematic framework for smart defense via semi-supervised learning in cyber threat intelligence. *Expert Systems with Applications*, 184:115517, 2021. doi: 10.1016/j.eswa.2021.115517. 13, 21

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. pages 785–794, 2016. 5, 14, 17, 53

N. Chouchani and M. Abed. Online social network analysis: Detection of communities of interest. *Social Network Analysis and Mining*, 10(1):1–19, 2020. doi: 10.1007/s13278-020-00652-9. 12, 28

A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004. doi: 10.1103/PhysRevE.70.066111. 28

D. L. Cogburn and F. K. Espinoza-Vasquez. From networked nominee to networked nation: Examining the impact of web 2.0 and social media on political participation and civic engagement in the 2008 obama campaign. *Journal of Political Marketing*, 10(1-2):189–213, 2011. doi: 10.1080/15377857.2011.540224. 38

G. Coleman. The politics of rationality: Psychiatric survivors' challenge to psychiatry. *Spontaneous Generations: A Journal for the History and Philosophy of Science*, 2(1):67–77, 2008. doi: 10.4245/sponge.v2i1.5956. 20

G. Coleman. *Hacker, Hoaxer, Whistleblower, Spy: The Many Faces of Anonymous*. Verso Books, 2014. 2, 19, 20, 21, 26, 77

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018. 15

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964. 14, 53

N. Cowger et al. Comparison of network security monitoring using zeek, suricata, and local taps. *Future Internet*, 14(3):80, 2022. doi: 10.3390/fi14030080. 12

J. W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Thousand Oaks, California, 4 edition, 2014. ISBN 9781452226101. 8

Dask Development Team. Dask: Library for dynamic task scheduling, 2024. URL `https://dask.org`. 16

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1:4171–4186, 2019. doi: 10.18653/v1/N19-1423. 21

V. Dutta et al. Towards secure industrial iot: Cyber-attack detection using hybrid deep learning method. volume 9, pages 17342–17352, 2022. doi: 10.1109/JIOT.2022.3152824. 2, 12

ENISA. Enisa threat landscape 2023, 2023. URL `https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023`. 1

FEBRABAN. Relatório anual de fraudes digitais 2023, 2024. URL `https://portal.febraban.org.br/pagina/3453/1276/pt-br/pesquisa-fraudes`. 4

G. Fernandes et al. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, 79(3):447–489, 2022. doi: 10.1007/s11235-021-00834-6. 1, 11, 13, 35, 43, 67

Fortinet. Fortiguard labs global threat landscape report, 2023. URL `https://www.fortinet.com/fortiguard/labs/global-threat-landscape-report`. 4

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010. doi: 10.1016/j.physrep.2009.11.002. 58, 76

L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3): 215–239, 1979. doi: 10.1016/0378-8733(78)90021-7. 19, 57

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014. doi: 10.1145/2523813. 16, 19, 97

H. Gonzalez et al. Adaptive cognitive security architecture: Cognitive analytics for adaptive security management and control in enterprise systems. *IBM Journal of Research and Development*, 64(3/4):2:1–2:10, 2020. doi: 10.1147/JRD.2020.2987405. 17

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 96

C. Group. 2023 cyberthreat defense report, 2023. URL `https://cyber-edge.com/cdr/`. 3

A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016. doi: 10.1145/2939672.2939754. 18

D. Gunning and D. W. Aha. Darpa's explainable artificial intelligence program. *AI Magazine*, 40(2):44–58, 2019. doi: 10.1609/aimag.v40i2.2850. 16, 19

J. T. Hancock and T. M. Khoshgoftaar. Survey on categorical data for neural network attack detection. *Journal of Big Data*, 7(1):1–37, 2020. doi: 10.1186/s40537-020-00320-5. 12, 18

A. Hernandez, V. Sanchez, G. Sanchez, H. Perez, J. Olivares, K. Toscano, and V. Martinez. Security attack prediction based on user sentiment analysis of twitter data. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 610–617, 2016. doi: 10.1109/ICIT.2016.7474819. 2, 7, 8, 9, 18, 21, 27, 32, 43, 47, 63, 87, 89, 92

A. Hernandez-Suarez et al. Social sentiment sensor in twitter for predicting cyber-attacks using l1 regularization. *Sensors*, 18(5):1380, 2018. doi: 10.3390/s18051380. 27, 39, 44

P. Himanen. *The hacker ethic and the spirit of the information age*. Random House, New York, 2001. 20, 26

R. Hofstede et al. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys Tutorials*, 16(4):2037–2064, 2014. doi: 10.1109/COMST.2014.2321898. 11, 12, 37, 42

P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012. doi: 10.1016/j.physrep.2012.03.001. 58

A. Huang. Similarity measures for text document clustering. In *NEW ZEALAND COMPUTER SCIENCE RESEARCH STUDENT CONFERENCE, 6.*, pages 49–56, Christchurch, 2008. University of Canterbury. 18, 24

C. J. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, ICWSM-14, pages 216–225. AAAI Press, 2014. URL `https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109`. 21

I. F. Ilyas and X. Chu. *Data Cleaning*. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450371537. doi: 10.1145/3310205. 36

J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021a. doi: 10.1109/TBDATA.2019.2921572. 5, 25, 30, 49, 50, 51, 74

J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021b. doi: 10.1109/TBDATA.2019.2921572. 17, 22

R. Jordaney et al. Transcend: Detecting concept drift in malware classification models. In *USENIX SECURITY SYMPOSIUM, 26.*, pages 625–642, Vancouver, 2017. USENIX Association. 16

A. D. Kent and L. M. Liebrock. Recent evolution of intrusion detection systems. *IEEE Security & Privacy*, 19(2):62–68, 2021. doi: 10.1109/MSEC.2020.3044083. 1, 11

M. A. Khan et al. A hybrid approach for intrusion detection using machine learning techniques. *AI Communications*, 33(3):197–224, 2020. doi: 10.3233/AIC-190617. 23, 53

R. P. Khandpur et al. Crowdsourcing cybersecurity: Cyber attack detection using social media. In *ACM Conference on Information and Knowledge Management*, pages 1049–1057, 2017. doi: 10.1145/3132847.3132866. 2, 3, 4, 7, 8, 19, 20, 27, 32, 35, 38, 45, 62, 81, 84, 87, 88, 89, 90, 92, 93

J. Kim et al. Hybrid deep learning network intrusion detection system based on convolutional neural network and bidirectional long short-term memory. *Journal of Advances in Information Technology*, 12(2):113–122, 2021. doi: 10.12720/jait.12.2.113-122. 12, 15, 23

A. H. Lashkari et al. Characterization of tor traffic using time based features. In *INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS SECURITY AND PRIVACY, 3.*, pages 253–262, Porto, 2017. SCITEPRESS. doi: 10.5220/0006105602530262. 42

Q. Le Sceller, E. B. Karbab, M. Debbabi, and F. Iqbal. Sonar: Automatic detection of cyber security events over the twitter stream. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 1–11, New York, NY, USA, 2017. Association for Computing Machinery. doi: 10.1145/3098954.3098992. 2, 20, 21, 27

J. Liu et al. Machine learning for network-based intrusion detection: A survey. *ACM Computing Surveys*, 54(5):1–36, 2021. doi: 10.1145/3472753. 11, 13

Z. Liu et al. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In *ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY*, pages 1777–1794, London, 2019. ACM. doi: 10.1145/3319535.3363224. 19, 29, 31, 32, 48

A. Y. Luay et al. Temporal analysis of netflow datasets for network intrusion detection systems. *arXiv preprint arXiv:2503.04404*, 2025. 23, 41, 43, 73

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017. 97

D. Lyon. *Surveillance, Snowden, and Big Data: Capacities, consequences, critique*, volume 1. SAGE Publications, 2014. doi: 10.1177/2053951714541861. 20

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 96

W. Maharani, A. A. Gozali, and C. P. Wulandari. Analysis of online social network friendships based on the presence of top influencers. *International Journal of Electrical and Computer Engineering*, 8(6):4417–4425, 2018. doi: 10.11591/ijece.v8i6.pp4417-4425. 19, 28, 45

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, Cambridge, 2008. 20, 39, 43

B. McMahan et al. Communication-efficient learning of deep networks from decentralized data. In *INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS*, pages 1273–1282, Fort Lauderdale, 2017. PMLR. 17

S. A. Mehrban et al. Feature selection and evaluation with traditional machine learning and deep learning techniques for network traffic anomaly detection. *Journal of Network and Computer Applications*, 200:103309, 2022. doi: 10.1016/j.jnca.2021.103309. 15

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 17, 18, 21

R. M. Milner. Pop polyvocality: Internet memes, public participation, and the occupy wall street movement. *International Journal of Communication*, 7:2357–2390, 2013. URL `https://ijoc.org/index.php/ijoc/article/view/1949`. 20

A. Moore, D. Zuev, M. Crogan, Q. Mary, and W. C. U. of London). Department of Computer Science. *Discriminators for Use in Flow-based Classification*, volume 53 of *Research report (Queen Mary and Westfield College (University of London) Department of Computer Science))*. Queen Mary and Westfield College, Department of Computer Science, 2005. URL `https://books.google.com.br/books?id=AzfpMgEACAAJ`. 15, 40, 41, 43

F. Morstatter et al. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In *INTERNATIONAL CONFERENCE ON WEBLOGS AND SOCIAL MEDIA, 7.*, pages 400–408, Cambridge, 2013. AAAI Press. 38

N. Moustafa et al. Evaluating standard feature sets towards increased generalisability and explainability of ml-based network intrusion detection. *Computers Security*, 108:102365, 2021. doi: 10.1016/j.cose.2021.102365. 16, 23

M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2010. ISBN 978-0199206650. doi: 10.1093/acprof:oso/9780199206650.001.0001. 19, 57, 62

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004. doi: 10.1103/PhysRevE.69.026113. 19

T. T. Nguyen and V. J. Reddi. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3881–3893, 2021. doi: 10.1109/TNNLS.2020.3014673. 17

P. Olson. *We Are Anonymous: Inside the Hacker World of LulzSec, Anonymous, and the Global Cyber Insurgency*. Little, Brown and Company, New York, 2012. ISBN 978-0-316-21354-7. 20

A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 32.*, pages 8026–8037, Vancouver, 2019. Curran Associates. 17

F. Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 16

O. Ponomareva et al. Vector search comparison. In *VECTORSEARCH.AI FAIR, 1.*, Online, 2021. VectorSearch.AI. 18, 22, 26, 49

N. Qazi and Z. Aung. Handling class imbalance in cybersecurity datasets: A review. *IEEE Access*, 11:77615–77635, 2023. doi: 10.1109/ACCESS.2023.3303177. 16, 52, 54, 69

Qdrant. Qdrant vector search engine, 2023. URL `https://qdrant.tech/documentation/`. 18, 22, 26, 51

I. Qureshi, M. Imran, and S. Anwar. Vbq-net: A novel vectorization-based boost quantized network model for maximizing the security level of iot system to prevent intrusions. *Systems*, 11(8):436, 2023. doi: 10.3390/systems11080436. 25, 49

E. Raff, C. Nicholas, and M. McLean. A new burrows-wheeler transform markov distance. *Data Mining and Knowledge Discovery*, 32(5):1249–1280, 2018. doi: 10.1007/s10618-018-0576-8. 19

RAPIDS Development Team. Rapids: Collection of libraries for gpu-accelerated data science, 2023. URL `https://rapids.ai`. Acesso em: 15 mar. 2025. 21

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016. doi: 10.1145/2939672.2939778. 97

M. Ring et al. A survey of network-based intrusion detection data sets. *Computers Security*, 86:147–167, 2019. doi: 10.1016/j.cose.2019.06.005. 12, 22, 36, 42, 96

M. Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In *PYTHON IN SCIENCE CONFERENCE, 14.*, pages 130–136, Austin, 2015. SciPy. doi: 10.25080/Majora-7b98e3ed-013. 16, 21, 38

M. Romagna. Hacktivism: Conceptualization, techniques, and historical view. In T. Holt and A. Bossler, editors, *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, pages 743–767. Palgrave Macmillan, Cham, 2020. doi: 10.1007/978-3-319-78440-3_35. 20, 27

C. Roth and J.-P. Cointet. Social and semantic coevolution in knowledge networks. *Social Networks*, 32(1):16–29, 2010. doi: 10.1016/j.socnet.2009.04.005. 58

S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2018. doi: 10.1613/jair.1.11640. 21

G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1988. ISBN 978-0070544840. 21

A. W. Samuel. *Hacktivism and the Future of Political Participation*. Harvard University Press, Cambridge, MA, 2004. Tese de Doutorado, Harvard University. 20

C. Sanders. *Practical packet analysis: Using Wireshark to solve real-world network problems*. No Starch Press, San Francisco, 3 edition, 2017. 12, 35

M. Sarhan et al. Netflow datasets for machine learning-based network intrusion detection systems. In *BIG DATA RESEARCH, SPRINGER, CHAM., MO'DATA 2020*, pages 117–135, 2021. doi: 10.1007/978-3-030-71839-3_12. 22

M. Sauter. *The Coming Swarm: DDOS Actions, Hacktivism, and Civil Disobedience on the Internet*. Bloomsbury Academic, New York, 2014. ISBN 978-1-62892-297-5. doi: 10.5040/9781501312670. 20

J. Scott. *Social network analysis*. SAGE Publications, London, 4 edition, 2017. 38, 56, 77

Selfuel. A comprehensive guide to real time data processing 2024, 2024. URL `https://selfuel.digital/a-comprehensive-guide-to-real-time-data-processing/`. 3

I. Sharafaldin et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS SECURITY AND PRIVACY, 4.*, pages 108–116, Funchal, 2018. SCITEPRESS. doi: 10.5220/0006639801080116. 15, 22, 30, 37, 40, 42

M. Shen et al. A deep learning approach for network intrusion detection with imbalanced data. *IEEE Access*, 10:122324–122335, 2022. doi: 10.1109/ACCESS.2022.3223614. 11, 13, 16, 68

S. K. Singh and D. K. Singh. A comprehensive survey on intrusion detection and traffic classification in software defined networks. *Journal of Information Security and Applications*, 55:102646, 2020. doi: 10.1016/j.jisa.2020.102646. 3, 4, 11, 40

M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427–437, 2009. doi: 10.1016/j.ipm.2009.03.002. 16, 60

R. Sommer and V. Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316. IEEE, 2010. 89, 93

K. Srinivasan et al. Real-time network anomaly detection using machine learning methods in software defined networks. *Intelligent Automation & Soft Computing*, 37(1):1343–1361, 2023. doi: 10.32604/iasc.2023.037209. 2, 12, 30

A. Srivatsa and V. Gudisa. Logsense: Scalable real-time log anomaly detection architecture. *arXiv preprint arXiv:2404.10572*, 2024. 40, 49, 51, 87

P. K. Syriopoulos, N. G. Kalampalikis, and S. B. Kotsiantis. knn classification: a review. *Annals of Mathematics and Artificial Intelligence*, 91(8-10):879–904, 2023. doi: 10.1007/s10472-023-09850-5. 14, 16, 21, 52, 53

D. D. Team. Dask: Library for dynamic task scheduling, 2024. URL `https://dask.org`. 21

E. Topics. Amount of data created daily (2025), 2025. URL `https://explodingtopics.com/blog/data-generated-per-day`. 3

W. Tounsi and H. Rais. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers Security*, 72:212–233, 2018. doi: 10.1016/j.cose.2017.09.001. 13

C. Ventures. The 2023 official cybercrime report, 2023. URL `https://cybersecurityventures.com/annual-cybercrime-report-2023/`. 4

Verizon. Data breach investigations report 2024, 2024. URL `https://www.verizon.com/business/resources/reports/dbir/`. 1, 3

A. Verma et al. Network behavioral analysis: A new paradigm for cybersecurity. *IEEE Network*, 36(3):176–182, 2022. doi: 10.1109/MNET.002.2100458. 13

K. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7:41525–41550, 2019a. doi: 10.1109/ACCESS.2019.2906723. 6

R. Vinayakumar, K. P. Soman, and P. Poornachandran. A comparative analysis of deep learning approaches for network intrusion detection systems (n-idss): Deep learning for n-idss. *International Journal of Digital Crime and Forensics*, 11(3):65–89, 2019b. doi: 10.4018/IJDCF.2019070104. 13, 14, 24, 52, 66

S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, Cambridge, UK, 1994. ISBN 978-0521387071. doi: 10.1017/CBO9780511815478. 19, 56, 62

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 (6684):440–442, 1998. doi: 10.1038/30918. 77

B. Wu, D. M. Divakaran, and M. Gurusamy. Uninet: A unified multi-granular traffic modeling framework for network security. *arXiv preprint arXiv:2503.04174*, 2025. 11, 17, 19, 24, 30, 31, 46, 70, 87, 93

Z. Yu et al. Logms: a multi-stage log anomaly detection method based on multi-source information fusion and probability label estimation. *Frontiers in Physics*, 12:1234483, 2024. doi: 10.3389/fphy.2024.1234483. 29

L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018. doi: 10.1002/widm.1253. 21

Z. Zhang et al. Artificial intelligence in cyber security: research advances, challenges, and opportunities. *Artificial Intelligence Review*, 55(2):1029–1053, 2022. doi: 10.1007/s10462-021-09976-0. 2

Zone-H. Zone-h – unrestricted information, 2023. URL `http://www.zone-h.org/`. 29