Mapeamento e Detecção de Indicadores de Comprometimento em Relatórios de *Threat Intelligence*

Charllynson Carvalho Caxias[1], **Brenda Salenave Santana**[1]

[1]Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)
Caixa Postal 354 – 96.010-610 – Pelotas – RS – Brasil

{charllynson, bssalenave}@inf.ufpel.edu.br

Abstract. *The increasing sophistication of cyberattacks demands automated, precise solutions to extract Indicators of Compromise (IoCs) from threat intelligence reports. This paper proposes a hybrid approach combining Regular Expressions (Regex) and Named Entity Recognition (NER) to identify IoCs in unstructured text. The methodology integrates structural validation of syntactic patterns with contextual analysis to detect artifacts such as IP addresses, URLs, and hashes, including those in obfuscated formats. Evaluated against a real-world corpus of CTI reports, the developed pipeline demonstrated that combining rigid rules with contextual models significantly reduces false positives and enhances the precision of contextualization of the extracted data.*

Resumo. *A crescente sofisticação dos ataques cibernéticos exige soluções automatizadas e precisas para a extração de Indicadores de Comprometimento (IoCs) contidos em relatórios de Threat Intelligence. Este trabalho propõe uma abordagem híbrida baseada em Expressões Regulares (Regex) e Reconhecimento de Entidades Nomeadas (NER) para a identificação de IoCs em textos não estruturados. A metodologia combina a validação estrutural de padrões sintáticos à análise contextual para detectar artefatos como endereços IP, URLs e hashes, inclusive sob formatos ofuscados. Avaliado sobre um corpus real de relatórios de CTI, o pipeline desenvolvido demonstrou que a convergência entre regras rígidas e modelos contextuais reduz significativamente os falsos positivos e amplia a precisão da contextualização dos dados extraídos.*

1. Introdução

O crescimento exponencial de ataques cibernéticos nas últimas décadas tem impulsionado a necessidade de mecanismos cada vez mais eficientes para identificação, análise e mitigação de ameaças digitais [Jo et al. 2022]. Nesse contexto, grande parte das informações necessárias para a detecção e resposta a incidentes encontra-se disponível em textos não estruturados, tais como relatórios de *Cyber Threat Intelligence* (CTI), fóruns especializados e outras fontes abertas, o que dificulta muito sua extração e análise automatizada [Long et al. 2019].

Além disso, nota-se que os estudos existentes concentram-se predominantemente na identificação isolada de indicadores, negligenciando alguns aspectos relevantes, tais como sua exposição em diferentes contextos e sua possível reutilização em múltiplos cenários de ataque [Gao et al. 2024]. Dessa forma, não apenas a extração, mas também

a interpretação contextual desses elementos torna-se fundamental para o avanço contínuo das soluções em segurança cibernética [Jo et al. 2022].

Diante desse cenário, este trabalho propõe e avalia um *pipeline* híbrido para a extração automatizada de Indicadores de Comprometimento (IoCs) em relatórios não estruturados de CTI. A abordagem integra técnicas de Reconhecimento de Entidades Nomeadas (do inglês *Named Entity Recognition* – NER) e Expressões Regulares (Regex) em um fluxo de processamento incremental. Como principal contribuição, destaca-se a sistematização e validação experimental desse modelo, que inclui refinamentos para o tratamento de formatos *defanged*, validação estrutural dos indicadores e a combinação entre reconhecimento contextual e regras heurísticas.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica relacionada aos principais conceitos abordados nesse trabalho. A Seção 3 descreve a metodologia proposta, incluindo a coleta de dados, extração de IoCs e treinamento dos modelos. A Seção 4 apresenta os experimentos realizados e a evolução dos testes conduzidos. Na Seção 5 são discutidos os resultados obtidos, e, por fim, a Seção 6 apresenta as conclusões e perspectivas de trabalhos futuros.

2. Fundamentação teórica

Nesta seção, apresentam-se os conceitos fundamentais que sustentam o desenvolvimento desta pesquisa, abrangendo desde os pilares da inteligência de ameaças até as técnicas de processamento de linguagem que permitem a automação da extração de dados.

2.1. Threat Intelligence e IoCs

A CTI é definida como o conhecimento baseado em evidências sobre ameaças existentes ou emergentes, incluindo contextos, mecanismos, indicadores e implicações [Gao et al. 2024]. O ciclo de inteligência visa transformar dados brutos em informações acionáveis para apoiar a tomada de decisão em diversos níveis: estratégico, operacional e tático [Tho et al. 2023].

No nível tático, destacam-se os IoCs. Estes consistem em artefatos técnicos observados em uma rede ou sistema que indicam, com alto grau de confiança, uma intrusão ou atividade maliciosa [Filho et al. 2023, Long et al. 2019]. Os IoCs são classificados conforme sua natureza, abrangendo desde indicadores de rede, como `endereço IP` (IPv4, IPv6 e nomes de domínio), até indicadores de host, como valores de `HASH` (MD5, SHA-1, SHA-256) de arquivos maliciosos [Filho et al. 2023]. A eficácia da resposta a incidentes depende da velocidade com que esses indicadores são compartilhados e consumidos pelas ferramentas de defesa [Jo et al. 2022, Filho et al. 2023].

2.2. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é a subárea da Inteligência Artificial que se dedica ao tratamento computacional da linguagem humana [Long et al. 2019, Fujii et al. 2023]. No contexto da segurança cibernética, o PLN é fundamental para converter o conhecimento contido em relatórios de inteligência não estruturados em dados legíveis por máquinas [Gao et al. 2024, Team 2025]. Através de algoritmos especializados, é possível identificar padrões sintáticos e anomalias na estrutura de textos

técnicos, permitindo uma classificação semântica posterior das orações [Long et al. 2019, Jo et al. 2022].

Diferentes abordagens podem ser empregadas para extrair sentido de sentenças complexas. Técnicas voltadas à análise de dependências entre palavras auxiliam no reconhecimento da intenção maliciosa e na identificação de relações entre entidades, como o vínculo entre uma `URL` específica e uma chamada para ação *(call to action)* em campanhas de ataque [Gao et al. 2024]. Além disso, o PLN fornece a base para o treinamento de modelos de aprendizado de máquina, como *Support Vector Machines (SVM)*, *Random Forest* e arquiteturas baseadas em transformadores, que são capazes de generalizar padrões de ameaças em grandes volumes de dados [Jo et al. 2022, Gao et al. 2024].

2.3. Reconhecimento de Entidades Nomeadas

O NER é uma tarefa de extração de informações que consiste em localizar e classificar elementos-chave em um texto em categorias predefinidas [Jo et al. 2022, Gao et al. 2024]. Enquanto em contextos genéricos o NER busca por nomes de pessoas ou organizações, na aplicação em CTI ele é adaptado para identificar entidades técnicas específicas, como nomes de malwares, grupos de adversários (APTs) e vulnerabilidades (CVEs) [Jo et al. 2022].

Modelos modernos de NER utilizam técnicas de aprendizado profundo para capturar o contexto em que uma palavra está inserida, o que é essencial para distinguir, por exemplo, quando uma sequência alfanumérica refere-se a uma `HASH` de arquivo ou apenas a um identificador interno de sistema [Gao et al. 2024, Jo et al. 2022]. A utilização de modelos pré-treinados, como o BERT *(Bidirectional Encoder Representations from Transformers)*, permite que o sistema compreenda as nuances semânticas de relatórios de segurança, aumentando a precisão na identificação de indicadores que não seguem um padrão fixo de caracteres [Jo et al. 2022].

2.4. Regex

O Regex constitui uma linguagem formal para a definição de padrões de busca e manipulação de cadeias de caracteres [Gao et al. 2024]. Trata-se de uma abordagem determinística que oferece alta eficiência e velocidade no processamento de volumes massivos de texto [Fujii et al. 2023]. Em sistemas de detecção de IoCs, as Regex são utilizadas para extrair indicadores que possuem estruturas sintáticas rígidas e bem definidas [Jo et al. 2022].

A aplicação de Regex é ideal para a captura de `endereço IP` (seguindo o padrão decimal pontuado) e `HASH`, cujos comprimentos são fixos de acordo com o algoritmo utilizado [Fujii et al. 2023, Filho et al. 2023]. Contudo, a eficácia das expressões regulares depende da sua robustez na lida com técnicas de ofuscação, como o uso de formatos *defanged* (ex.: 10[.]0[.]0[.]1), comuns em relatórios para evitar a ativação acidental de links maliciosos [Fujii et al. 2023]. Embora careçam da compreensão de contexto oferecida pelo NER, as Regex servem como uma camada de validação precisa e de baixo custo computacional em abordagens híbridas de extração de dados [Long et al. 2019].

2.5. Trabalhos relacionados

A identificação e análise automatizada de ameaças em textos não estruturados têm sido alvo de algumas pesquisas nas últimas décadas. Esses estudos exploram desde a

mineração de dados em fóruns clandestinos até o uso de modelos avançados de aprendizado profundo para contextualização. Nesse contexto, os trabalhos a seguir demonstram diferentes estratégias utilizadas para detectar IoCs, reconhecer entidades e auxiliar em atividades de CTI.

O trabalho de [Filho et al. 2023] dedica-se à extração empírica e análise de IoCs diretamente de fóruns abertos da *Dark Web*. Utilizando uma abordagem baseada em Regex e a ferramenta *ioc-finder*, os autores implementaram a extração de dez tipos distintos de indicadores (tais como `URL`, e-mails, domínios e `HASH`). O estudo constatou experimentalmente que os IoCs estão presentes em mais de 26% das postagens, ressaltando a relevância da análise do ambiente de exposição. Por exemplo, evidenciou-se que posts inseridos em categorias de fórum voltadas à computação possuem uma incidência de IoCs quase duas vezes maior em relação a outros temas.

[Long et al. 2019] abordam a identificação de IoCs em relatórios de cibersegurança formulando a extração como uma tarefa de anotação de sequência (*sequence labelling*), mitigando a dependência exclusiva de regras manuais e conhecimento de domínio. Os autores propuseram um modelo de rede neural fim-a-fim que combina uma arquitetura Bi-LSTM — capaz de processar dados bidirecionalmente para capturar contextos passados e futuros da entrada [Anishnama 2023] —, mecanismos de atenção (*multi-head self-attention*) e uma camada CRF, responsável por garantir que as predições finais sigam regras lógicas sequenciais [Team 2023]. Uma contribuição central dessa abordagem é a utilização conjunta de características ortográficas (extraídas via Regex) e características contextuais durante o treinamento, o que aprimora o reconhecimento dos tokens e ajuda a evitar falsos positivos na extração.

Com o intuito de buscar uma compreensão técnica e semântica mais ampla, [Jo et al. 2022] propuseram o sistema "Vulcan", que expande o escopo tradicional de busca de IoCs para extrair inteligência descritiva incluindo Táticas, Técnicas e Procedimentos (*Tactics, Techniques, and Procedures - TTPs*), identificando vetores de ataque, ferramentas, plataformas e vulnerabilidades. O sistema baseia-se no ajuste fino (*fine-tuning*) de modelos de linguagem como o BERT para realizar o NER e, na sequência, a Extração de Relações (RE) semânticas entre essas entidades. A plataforma facilita a compreensão comportamental e estrutural da ameaça, permitindo aplicações analíticas que constroem, por exemplo, linhas do tempo com a evolução de famílias de *ransomware*.

Por fim, a ferramenta CyNER, desenvolvida por [Fujii et al. 2023], combina modelos NER avançados (como o RoBERTa) para entidades conceituais e o uso eficiente de Regex para extrair IoCs formatados, objetivando estruturar os relatórios não estruturados no formato padrão STIX[1]. Um desafio notório resolvido por esta pesquisa é o problema dos "IoCs não contextuais", que são indicadores frequentemente listados isoladamente no final dos relatórios ou em formato de tópicos. O trabalho soluciona essa perda de contexto vinculando tais IoCs remotos às entidades nomeadas principais da publicação (como o nome de um *malware*) através de técnicas de extração de frases-chave, garantindo a restauração do sentido contextual para os indicadores.

Em resumo, as abordagens presentes na literatura evidenciam o forte potencial dos métodos híbridos que unem NER e Regex para extração e mapeamento de dependências.

[1] *Structured Threat Information eXpression:* Padrão mantido pela OASIS para intercâmbio de CTI.

É com base nessas evidências que o presente trabalho fundamenta sua proposta utilizando uma abordagem mista para detecção de IoCs em múltiplos cenários para gerar inteligência contextualizada.

3. Metodologia

A metodologia deste trabalho propõe uma abordagem incremental comparativa para a extração de IoCs, contrapondo e integrando o uso de Regex com modelos de NER. O processo metodológico compreende as fases de coleta de relatórios técnicos, extração e tratamento de dados textuais, desenvolvimento e refinamento iterativo de regras heurísticas baseadas em Regex, e o treinamento de modelos de aprendizado de máquina, inicialmente empregando a biblioteca spaCy[2] e, posteriormente, arquiteturas BERT, a partir de um dataset construído com anotação mista (manual e automatizada).

3.1. Coleta de dados

A base de dados empregada nesta pesquisa foi construída a partir da coleta de 40 relatórios de CTI publicados no período de março de 2023 a fevereiro de 2026. O corpus documental é constituído por 18 relatórios originados da CISA, 20 publicações da Unit 42, um relatório do CERT.PL e um relatório da Microsoft Defense.

Na etapa inicial de pré-processamento, o conteúdo dos arquivos PDF foi extraído para o formato .txt. Posteriormente, durante as fases de testes, optou-se pela extração direta do conteúdo em formato HTML, técnica que proporcionou uma melhoria estrutural significativa na identificação e na qualidade dos dados recuperados.

3.2. Extração de IoCs

A etapa de extração de IoCs foi conduzida por meio de uma abordagem baseada na combinação de Regex e modelos NER. O objetivo dessa estratégia foi explorar simultaneamente a precisão estrutural proporcionada pelas Regex e a capacidade de contextualização semântica oferecida pelos modelos de aprendizado de máquina.

As Expressões Regulares foram empregadas para identificar indicadores que apresentam estruturas sintáticas rígidas e bem definidas, como `endereço IP`, `URL`, `HASH` e `PORTA`. Para aumentar a robustez da detecção, foram incorporadas regras capazes de reconhecer formatos ofuscados (*defanged*), frequentemente utilizados em relatórios de CTI, como “10[.]0[.]0[.]1”, “hxxp” e “hxxps”.

Além disso, foram implementadas heurísticas de validação estrutural com o objetivo de reduzir falsos positivos. No caso de `endereço IPv4`, os octetos foram limitados ao intervalo válido de 0 a 255. Para domínios e `URL`, estabeleceram-se restrições que impedem o uso de hífens no início ou no final dos domínios. Já para `HASH`, foram adotados tamanhos estritos compatíveis com os algoritmos analisados, sendo 32 caracteres para MD5, 40 para SHA-1 e 64 para SHA-256.

Adicionalmente, utilizou-se a técnica de *negative lookahead* para descartar `HASH` inválidas compostas inteiramente por zeros, reduzindo ocorrências de correspondências artificiais ou sem significado prático. Também foram aplicadas etapas de normalização textual para minimizar problemas decorrentes da fragmentação de conteúdo durante a

[2]Disponível em: `https://spacy.io/`. Acesso em: 29 mai. 2026.

extração dos dados, especialmente em documentos originalmente disponibilizados em PDF.

3.3. Modelo NER

A modelagem baseada em NER iniciou-se com a etapa de rotulação dos dados. Parte do corpus foi anotado manualmente por meio da ferramenta Doccano[3], gerando um conjunto de dados inicial. Posteriormente, realizou-se uma etapa de anotação automatizada supervisionada com apoio de regras Regex, seguida de validação manual, resultando em um segundo conjunto de dados.

A união desses subconjuntos originou o dataset final de treinamento, contendo entidades dos tipos `HASH`, `endereço IP`, `URL` e `PORTA`. Além das sentenças contendo IoCs, o conjunto também incluiu exemplos negativos, isto é, trechos sem indicadores, visando aprimorar a capacidade discriminativa do modelo. Inicialmente, os experimentos utilizaram a biblioteca spaCy para treinamento de modelos NER. Posteriormente, também foram realizados experimentos utilizando arquiteturas baseadas em BERT, escolhidas pela capacidade de modelagem contextual e interpretação semântica do texto.

Durante o processo de preparação dos dados, observou-se desbalanceamento entre as classes anotadas.Para mitigar esse problema, foi realizado um aumento de dados (*oversampling*) por meio de geração sintética baseada em regras: para cada entidade da classe IP no conjunto de treinamento, foram geradas três novas amostras adicionando-se `PORTA` comuns a partir de uma lista pré-definida. As avaliações dos modelos foram realizadas por meio de comparação com um conjunto de referência previamente definido, utilizando métricas de *Precision*, *Recall* e *F1-score*.

4. Experimentos

Os experimentos foram conduzidos de forma incremental, de modo que cada teste introduzisse melhorias específicas no *pipeline* de extração de IoCs. O objetivo dessa abordagem foi avaliar o impacto individual de cada modificação sobre as métricas de precisão, revocação e F1-score.

4.1. Ambiente de execução

Na etapa inicial de pré-processamento, o conteúdo dos arquivos PDF foi extraído por meio da biblioteca *PyMuPDF*[4]. Posteriormente, motivado pelas fragmentações estruturais observadas nesse formato, o *pipeline* foi adaptado para a extração direta de código HTML utilizando *BeautifulSoup*[5]. Essa abordagem permitiu preservar a integridade sintática original das cadeias de caracteres dos IoCs, eliminando quebras de linha espúrias introduzidas pela renderização de páginas do PDF.

A extração dos IoCs foi conduzida a partir de duas abordagens distintas e complementares. A primeira baseou-se em Regex para identificar padrões léxicos e estruturais característicos de indicadores. A segunda empregou técnicas de NER, explorando tanto modelos estatísticos tradicionais quanto arquiteturas baseadas em aprendizado profundo.

[3]Ferramenta *open-source* para rotulagem de dados voltada a tarefas de aprendizado de máquina.
[4]Ferramenta *open-source* em Python para extração e manipulação de documentos PDF.
[5]Ferramenta *open-source* em Python para análise e *parsing* de documentos HTML e XML.

Para isso, foram utilizados modelos implementados com spaCy e uma variante do *Transformer* BERT (*bert-base-uncased*), ajustada por meio de treinamento supervisionado no domínio de CTI.

O processo de treinamento do BERT considerou um regime experimental utilizando 3 épocas, tamanho de lote igual a 8 e taxa de aprendizado de 2×10^{-5}. O dataset foi particionado em 80% para treinamento e 20% para teste. A execução dos experimentos em ambiente interativo possibilitou a análise incremental dos resultados, a inspeção qualitativa das entidades reconhecidas e o refinamento contínuo das estratégias de extração ao longo do desenvolvimento da pesquisa.

4.2. Conjunto de dados

O conjunto de dados foi composto por relatórios públicos de CTI contendo IoCs, como `HASH`, `URL`, `endereço IP` e `PORTA`. O conjunto final foi composto por 2.115 entidades anotadas, distribuídas da seguinte forma:

Tabela 1. Distribuição por classes (pós-*oversampling*)

Classe	Quantidade	Percentual
`PORTA`	844	39,9%
`HASH`	543	25,7%
`URL`	468	22,1%
`endereço IP`	260	12,2%
Total	**2.115**	**100%**

Para treinamento supervisionado dos modelos NER, foi construída uma base anotada manualmente em formato JSON contendo entidades dos tipos `HASH`, `IP`, `URL` e `PORTA`. Vale ressaltar que o conjunto de dados utilizado para os testes não compunha o dataset do treinamento.

4.3. Estratégia de avaliação e evolução do *pipeline*

Os experimentos foram estruturados de forma incremental para avaliar o impacto individual de cada aprimoramento no *pipeline* de extração. O desenvolvimento seguiu uma progressão em quatro etapas: iniciou-se com abordagens determinísticas baseadas em Regex, avançando para a incorporação de técnicas de normalização textual e suporte a formatos *defanged*. Em seguida, houve a transição da fonte de dados (de PDF para HTML) e a integração com modelos NER. Por fim, implementou-se uma validação mista, explorando a complementaridade entre a validação estrutural do Regex e a análise contextual do NER.

Para mensurar a eficácia de cada etapa, os resultados processados foram comparados a um gabarito de referência contendo os IoCs esperados. O desempenho foi avaliado com base nas métricas de *Precision*, *Recall* e *F1-score*, além do monitoramento do tempo de execução de cada abordagem.

O gabarito foi elaborado manualmente a partir da anotação dos IoCs identificados no relatório analisado, servindo como base de comparação para validar a capacidade de extração das abordagens empregadas. Dessa forma, foi possível verificar a quantidade de indicadores corretamente identificados.

4.3.1. Etapa 1: Baseline com Expressões Regulares Simples

O primeiro experimento consistiu em uma abordagem inicial baseada em expressões regulares aplicadas sobre o texto extraído do relatório PDF. Nesse estágio, os padrões Regex ainda não apresentavam robustez suficiente para diferenciar corretamente os diferentes tipos de IoCs. Como consequência, diversos termos incompatíveis com o formato de `HASH` foram incorretamente classificados como `HASH`, incluindo exemplos como "3cx", "d090", "e4d3" e "event-id". Isso ocorreu pois o padrão utilizado aceitava sequências hexadecimais sem restrições adequadas de tamanho e contexto.

Além disso, os padrões implementados não foram capazes de identificar adequadamente outros tipos de IoCs, como `URL`, `endereço IP` e `PORTA`. Os resultados desse teste evidenciaram a necessidade de aprimorar os padrões Regex e melhorar o pré-processamento textual.

4.3.2. Etapa 2: Expansão de Regras e Normalização Textual

No segundo experimento, foram introduzidas melhorias nos padrões Regex com o objetivo de aumentar a capacidade de detecção de IoCs presentes em formatos frequentemente utilizados em relatórios de CTI. Inicialmente, foram adicionados padrões para reconhecimento de `endereço IP` em formato *defanged*, como "10[.]0[.]0[.]1", técnica amplamente utilizada para impedir a ativação acidental de indicadores maliciosos.

Também foram adicionadas regras para reconhecimento de `URLs` sem os prefixos "http://" e "https://", além do suporte a formatos *defanged* utilizando "hxxp" e "hxxps". Paralelamente, foi implementada uma etapa de normalização textual, substituindo múltiplas quebras de linha e espaços consecutivos por um único espaço. Essa modificação teve como objetivo reduzir fragmentações ocasionadas pela extração textual do PDF, melhorando a integridade dos IoCs presentes no texto.

Além disso, foi incorporado um filtro adicional para evitar que nomes de arquivos ou fragmentos textuais fossem incorretamente reconhecidos como `URL`. Apesar das melhorias implementadas, os resultados mostraram que o Regex ainda apresentava limitações significativas, sendo capaz de identificar principalmente `URL`, enquanto outros tipos de IoCs continuavam com baixa taxa de detecção.

4.3.3. Etapa 3: Transição para HTML e Integração com NER

Nos experimentos seguintes, optou-se por alterar a fonte de dados utilizada no *pipeline*. Em vez da extração textual direta do PDF, passou-se a utilizar a versão HTML do relatório. Essa modificação foi motivada pelas limitações observadas na extração do PDF, que frequentemente introduzia fragmentações, perdas de contexto e separações indevidas em IoCs.

Com a utilização do HTML, observou-se uma melhoria significativa na integridade textual dos dados processados. Nesse cenário, o Regex passou a identificar hashes completas corretamente, ao contrário dos testes anteriores, nos quais apenas fragmentos eram reconhecidos. Entretanto, ainda ocorreram falsos positivos, como a classificação do

arquivo `18.12.416.msi` como `endereço IP`.

Paralelamente, foi integrado ao *pipeline* um modelo NER. O modelo foi capaz de identificar diversos IoCs, porém ainda apresentou classificações inadequadas, incluindo "3CXDesktopApp" identificado como `HASH`, e "|" identificado como `PORTA`. Esses resultados indicaram que, embora o modelo NER tivesse potencial para capturar padrões contextuais, ele ainda necessitava de maior refinamento e treinamento adicional.

4.3.4. Etapa 4: Abordagem Híbrida e Validação Estrutural

No quarto experimento, foram implementadas melhorias adicionais nos padrões Regex visando reduzir falsos positivos e aumentar a confiabilidade dos indicadores extraídos. Para `IPs`, passou-se a validar explicitamente o intervalo permitido para cada octeto, restringindo os valores ao intervalo de 0 a 255.

No caso das URLs, foram adicionadas restrições para impedir que domínios começassem ou terminassem com hífens, reduzindo classificações incorretas. Para `HASH`, foram definidos tamanhos exatos compatíveis com os algoritmos presentes nos relatórios analisados: MD5: 32 caracteres; SHA-1: 40 caracteres; SHA-256: 64 caracteres.

Adicionalmente, utilizou-se *negative lookahead* para descartar hashes inválidas compostas inteiramente por zeros. Por fim, foi introduzida uma estratégia de validação combinada entre o modelo NER e o Regex. Nesse processo, ambos os métodos foram executados simultaneamente.

A estratégia híbrida adotada buscou explorar a complementaridade entre Regex e NER. Enquanto as Expressões Regulares foram utilizadas para validação estrutural de indicadores com formatos rígidos, como `HASH` e `endereço IP`, o modelo NER foi empregado para capturar padrões contextuais e indicadores menos estruturados ou altamente ofuscados. Dessa forma, os resultados provenientes do NER não eram descartados automaticamente quando não confirmados pelo Regex, mas analisados conforme o tipo de entidade e seu contexto textual.

5. Resultados e discussão

Os resultados apresentados na Tabela 2, indicam que a abordagem híbrida apresentou melhor desempenho na extração de IoCs estruturados, especialmente `HASH`, `IP` e `URL`. Os experimentos demonstraram que a integração entre validação estrutural e análise contextual contribuiu significativamente para a redução de falsos positivos e melhoria do *F1-score*.

Tabela 2. Comparativo de resultados quantitativos das etapas de extração

Etapa	Fonte	Método Principal	Precisão	Recall	F1-score	Tempo (s)
Baseline	PDF	Regex Simples	0.00	0.00	0.00	0.11
Expansão de Regras	PDF	Regex Refinado	0.02	0.03	0.03	0.19
Integração NER	HTML	Regex + NER	0.58	1.00	0.71	0.38
Validação Híbrida	HTML	NER $\leftrightarrow$ Regex	0.67	1.00	0.78	0.36

Cabe destacar que a etapa de integração do modelo NER coincidiu com a adoção

da extração dos relatórios em formato HTML, substituindo a extração anterior baseada em PDF. Dessa forma, embora os resultados indiquem melhora no desempenho do *pipeline*, não é possível isolar quantitativamente a contribuição individual da mudança da fonte de dados e da incorporação do NER. Assim, os ganhos observados devem ser interpretados como resultado da combinação dessas modificações, constituindo uma limitação desta avaliação experimental.

A evolução incremental dos padrões Regex demonstrou que, quando cuidadosamente projetadas, a integração entre Regex e NER permitiu resultados promissores com baixo custo computacional e elevada precisão para esse tipo de tarefa. Além disso, mecanismos adicionais de validação estrutural contribuíram significativamente para a redução de falsos positivos ao longo dos experimentos, conforme evidenciado na Tabela 2.

Por outro lado, embora os modelos NER tenham apresentado potencial para identificação contextual de entidades, os experimentos evidenciaram limitações relevantes associadas à reduzida quantidade e diversidade do conjunto de treinamento. Modelos baseados em aprendizado de máquina demandam grandes volumes de dados anotados e balanceados para alcançar desempenho satisfatório, especialmente em domínios altamente especializados como CTI.

Dessa forma, os resultados sugerem que, para cenários envolvendo extração de IoCs estruturados em corpora limitados, abordagens híbridas podem representar uma solução mais eficiente, interpretável e de menor complexidade em comparação a modelos NER treinados com conjuntos reduzidos de dados.

6. Conclusão

Este trabalho apresentou uma abordagem híbrida para detecção de IoCs em textos não estruturados de CTI, integrando Regex e NER. Os resultados demonstraram que a combinação das duas técnicas superou as abordagens isoladas, alcançando um *F1-score* de 0.78 e reduzindo falsos positivos por meio da validação estrutural realizada pelo Regex. Além disso, observou-se que o Regex é mais eficiente para IoCs com sintaxe rígida, enquanto o NER complementa a solução ao fornecer análise contextual e identificar indicadores menos estruturados.

Como limitações, destacam-se a escassez de relatórios públicos de CTI, o desbalanceamento do conjunto de dados e os desafios de pré-processamento decorrentes da extração de informações de documentos em PDF. Além disso, a adoção simultânea da extração em HTML e da integração do NER não permitiu isolar quantitativamente a contribuição individual de cada uma dessas modificações para o desempenho observado.

Como trabalhos futuros, pretende-se ampliar o *corpus* com novas fontes de CTI, integrar a metodologia a fluxos automatizados de inteligência cibernética, investigar a recorrência e reutilização de IoCs em diferentes relatórios e explorar arquiteturas baseadas em *transformers* e mecanismos de correlação semântica entre IoCs, *malwares*, campanhas e grupos APT.

Por fim, acredita-se que a abordagem proposta possa servir como base para o desenvolvimento de ferramentas de apoio à inteligência cibernética, contribuindo para a automatização da extração de IoCs e para a redução do esforço manual na análise de relatórios de CTI.

Referências

Alam, M. T., Bhusal, D., Park, Y., and Rastogi, N. (2022). Cyner: A python library for cybersecurity named entity recognition.

Anishnama (2023). Understanding bidirectional lstm for sequential data processing.

Feng, X., He, S., Wei, X., Liu, R., Yue, H., and Wang, X. (2025). Prompt-bart: A named entity recognition model applied to cyber threat intelligence.

Filho, S. J., Gabriel, P., and Miani, R. (2023). Extração e análise de indicadores de comprometimento (iocs) em fóruns da dark web. In *Anais do XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 349–362, Porto Alegre, RS, Brasil. SBC.

Fujii, S., Kawaguchi, N., Shigemoto, T., and Yamauchi, T. (2023). Extracting and analyzing cybersecurity named entity and its relationship with noncontextual iocs from unstructured text of cti sources. *Journal of Information Processing*, 31:578–590.

Gao, P., Liu, X., Choi, E., Ma, S., Yang, X., and Song, D. (2024). Threatkg: An ai-powered system for automated open-source cyber threat intelligence gathering and management. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 1–12, New York, NY, USA. Association for Computing Machinery.

Jiang, Y., Hu, H., Li, Y., Li, F., Zhao, C., Chen, C., and Liu, Y. (2025). A zero-shot self-improving ner method for cyber threat intelligence via knowledge injection. *Cybersecurity*, 8.

Jo, H., Lee, Y., and Shin, S. (2022). Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text. *Computers & Security*, 120:102763.

Long, Z., Tan, L., Zhou, S., He, C., and Liu, X. (2019). Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Mouiche, I. and Saad, S. (2025a). Entity and relation extractions for threat intelligence knowledge graphs. *Computers & Security*, 148:104120.

Mouiche, I. and Saad, S. (2025b). Tijere: A novel threat intelligence joint extraction model based on analyst expert knowledge. *Knowledge-Based Systems*, 329:114346.

Nakayama, H. (2018). Get started with doccano.

Nogueira, D. (2025). N8n: O que É, como funciona e como instalar.

Team, M. D. S. R. (2025). From unstructured data to actionable intelligence: Using machine learning for threat intelligence.

Team, S. (2023). Deep learning for nlp - an overview.

Tho, N. D., Hieu, N. T., Tran, N. N. B., and Anh, N. P. (2023). Extracting multiple relations between entities from unstructured threat intelligence reports. *Journal of Science and Technology on Information security*, 3(20):5–13.