DISSERTAÇÃO DE MESTRADO PROFISSIONAL

FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT

Paulo Magno de Melo Rodrigues Alves

Programa de Pós-Graduação Profissional em Engenharia Elétrica
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT

PAULO MAGNO DE MELO RODRIGUES ALVES

ORIENTADOR: VINÍCIUS PEREIRA GONÇALVES, Ph.D
COORIENTADOR: GERALDO PEREIRA ROCHA FILHO, Ph.D

DISSERTAÇÃO DE MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPEE.MP.053
BRASÍLIA/DF, JUNHO/2023

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT

Paulo Magno de Melo Rodrigues Alves

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. Vinícius Pereira Gonçalves, Ph.D, FT/UnB _____
Orientador

Prof. Fábio Lúcio Lopes de Mendonça, Ph.D, FT/UnB _____
Examinador Interno

Prof.José Rodrigues Torres Neto, Ph.D, Universidade Federal do Piauí (UFPI) _____
Examinador Externo

FICHA CATALOGRÁFICA

ALVES, PAULO MAGNO DE MELO RODRIGUES
FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT [Distrito Federal] 2023.
xvi, 61 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2023).
Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Processamento de Linguagem Natural
2. Inteligência Cibernética
3. Táticas, Técnicas e Procedimentos
4. Machine Learning
I. ENE/FT/UnB
II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ALVES, P. M. M. R. (2023). *FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT*. Dissertação de Mestrado Profissional, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 61 p.

CESSÃO DE DIREITOS

AUTOR: Paulo Magno de Melo Rodrigues Alves
TÍTULO: FRAMEWORK PARA CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT.
GRAU: Mestre em Engenharia Elétrica ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, a Universidade de Brasília tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

Paulo Magno de Melo Rodrigues Alves
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico este trabalho à minha família, que forneceu o tempo e a motivação para entregar essa pequena contribuição para um mundo mais seguro.

AGRADECIMENTOS

Agradeço inicialmente à minha querida esposa, Penina, cujo apoio, companheirismo e compreensão foram fundamentais para me motivar e permitir chegar à conclusão dessa jornada.

Aos professores Vinicius Pereira Gonçalves, orientador, e Geraldo Rocha Pereira Filho, coorientador, pela lucidez e paciência na condução da pesquisa e pelas dicas e correções ao longo do trabalho.

Ao professor Robson de Oliveira Albuquerque, por ajudar a iluminar novos caminhos em momento de dificuldade.

Aos colegas do PPEE/UnB, pelo companheirismo nas atividades e pelas amizades formadas ou reforçadas ao longo da jornada acadêmica.

Por fim, agradeço a Deus pelo dom da vida, pela inquietude intelectual e pela força para superar os desafios e barreiras enfrentadas durante esse trabalho.

RESUMO

Informações relativas às Táticas Técnicas e Procedimentos (TTP) observados em um ataque são importantes para os profissionais de segurança cibernética. Contudo, elas são costumeiramente disseminadas na forma de textos não estruturados, dificultando a acesso e, portanto, o trabalho dos ciberanalistas. Esse trabalho apresenta um *framework* para o enfrentamento desse problema por meio do BERT (*Bidirectional Encoder Representations from Transformers*), modelo de NLP derivado da Arquitetura de Tranformadas. Assim, foram utilizadas 11 variantes BERT, estado da arte no campo de NLP, para classificar sentenças de acordo com o *framework* MITRE ATT&CK para TTP. O *dataset* utilizado inicialmente foi a base de sentenças do instituto MITRE, sendo uma parte usada no treinamento e outra na avaliação dos modelos. Posteriormente foi realizada validação em um conjunto de sentenças manualmente anotadas extraído de relatórios de CTI (*Cyber Threat Intelligence*) públicos. Investigou-se também os efeitos de alguns hiperparâmeros escolhidos no treinamento de ajuste fino dos modelos. O objetivo foi identificar o modelo e a combinação de hiperparâmetros que melhor se adequariam à tarefa de classificação proposta. Como resultado, verificou-se que os melhores modelos apresentaram acurácia de 0,8264 e 0,7875 nos dois conjuntos de dados utilizados, demonstrando a viabilidade e o potencial do uso dos modelos BERT nessa complexa tarefa do domínio cibernético. Por fim, realiza-se análise qualitativa de algumas das sentenças erroneamente classificadas pelo framework, de modo a compreender melhor porque o modelo erra e obter *insights* que potencialmente ajudem a melhorar a performance.

ABSTRACT

Information upon Tactics, Techniques and Procedures (TTP) observed in an attack are important to cybersecurity defenders. However, they are mostly disseminated through unstructured text, hindering access and the job of ciberanalysts. This work presents a framework for tackling this problem by using BERT (Bidirectional Encoder Representations from Transformers), a model derived from the Transformers Architecture. We use 11 variants of BERT, a state-of-the-art approach in Natural Language Processing, to classify sentences according to MITRE ATT&CK framework for TTP. The dataset used is MITRE's database of sentences (examples) and part of it is used in training and part in the models evaluation. Validation is also done against a set of manually annotated sentences extracted from public CTI reports. The effect of some chosen hyperparameters on the fine-tuning of the models are also investigated. The purpose is to identify the best model and the finest combination of hyperparameters for the proposed classification task. As a result, we observed that the best models presented an accuracy of 82.64% and 78.75% on the two datasets tested, demonstrating the feasibility and potential of the application of BERT models in the complex task of TTP classification. At last, we analyze some of the sentences misclassified by the framework to better understand why the models are missing and thus gather insights about possibilites to further improve performance.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Os ataques cibernéticos têm aumentado não apenas em volume, mas também em complexidade. Ataques por *malwares* em geral e *ransonwares* tiveram um crescimento de 358% e 435%, respectivamente, apenas no ano de 2020 [9]. Os últimos anos têm presenciado alguns dos ataques mais sofisticados e críticos já ocorridos. Esse cenário decorre, em parte, da dependência cada vez maior que as organizações têm de redes de computadores e outras tecnologias [10]. A crescente dependência digital da mundo contemporâneo alterou formas de funcionamento da sociedade e, à medida que o mundo se torna mais digital, as ameaças seguem o mesmo caminho [11]. Nesse contexto de dependência digital e sistemas tecnológicos cada vez mais complexos, as ameaças cibernéticas por vezes parecem estar superando a capacidade da sociedade lidar com elas com de modo eficaz [9]. A Figura 1.1 abaixo mostra a tendência de crescimento de ataques cibernéticos significativos [1]:

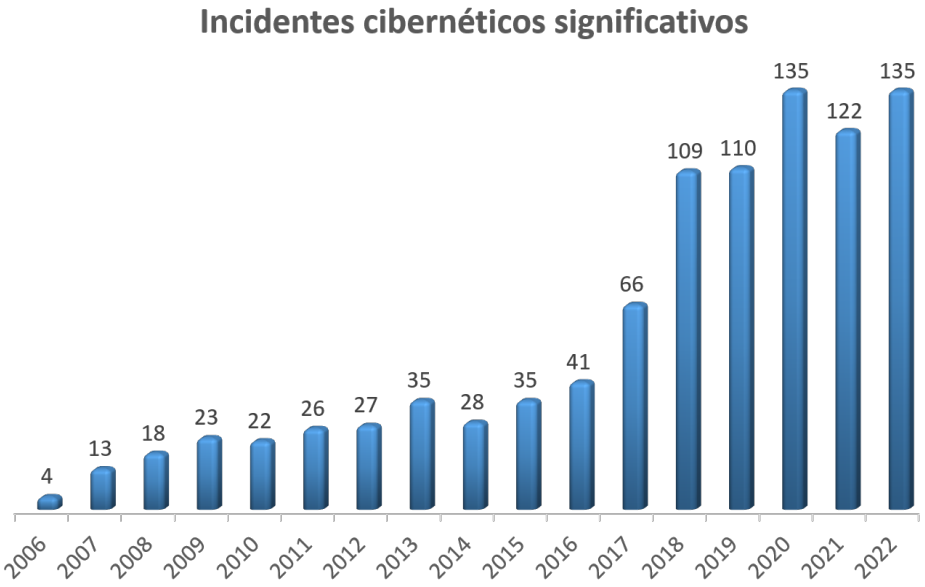


Figura 1.1: Número de incidentes cibernéticos significativos por ano. Extraído de [1]

Defender redes de computadores nesse cenário é uma tarefa desafiadora. Os defensores necessitam de informação acionável, precisa e oportuna, uma vez que novas ameaças precisam ser mitigadas com rapidez e eficiência. Informações técnicas sobre ameaças cibernéticas normalmente podem ser encontradas em relatórios de Inteligência de Ameaças Cibernéticas (CTI, na sigla em inglês para *Cyber Threat Intelligence*). Informações de CTI permitem que o profissional de segurança cibernética tenha maior agilidade e tome decisões mais informadas acerca da prevenção, resposta a incidentes ou recuperação de sistemas após ataques cibernéticos [12]. Esses relatórios difundem informações essenciais sobre o *modus operandi* dos atacantes, particularmente Indicadores de Comprometimento (IOC, *Indicators of Compromise*) e Táticas, Técnicas e Procedimentos (TTP).

Indicadores de Comprometimento apresentam informações sobre dados brutos específicos (IPs, *hashes*, domínios etc). Esses dados trazem informações principalmente sobre ferramentas e infraestruturas utiliza-

das no ataque. Muitos feeds de inteligência de ameaças focam nos IOCs e muitas soluções tradicionais de segurança são baseadas em IOCs. Contudo, tais dados não são suficientes para uma proteção adequada, pois carecem de contexto informacional adequado para melhor descrever o padrão de ataque [13, 14]. O conhecido modelo da Pirâmide da Dor [15], ilustrado na Figura 1.2 considera que os IOCs constituem dados simples e triviais, os quais fornecem inteligência de menor valor.

Figura 1.2: Pirâmide da Dor em segurança cibernética

TTPs, por sua vez, descrevem não *o que* é utilizado em um ataque (ferramentas e infraestrutura, por exemplo), mas *como* esse ataque é realizado (as táticas, técnicas e procedimentos empregados que formam a sigla TTP). Esse tipo de conhecimento permite descrever o ataque de forma mais pormenorizada e fornece uma compreensão holística do ataque essencial para tornar mais eficaz a tomada de decisão dos profissionais de segurança. Dessa forma, TTPs encontram-se no topo da Pirâmide [15], consistindo nas informações mais valiosas para os profissionais de segurança cibernética.

O atual cenário de disponibilização e popularização de ferramentas de ataque, tem levado a uma proliferação das ameaças cibernéticas. O advento, no meio digital, do crime como serviço (*crime as a service* também acarretou a profusão de novos atores maliciosos. O forte incentivo financeiro advindo tanto dos crimes cibernéticos como da atuação de Estados nacionais ensejou uma maior sofisticação desses grupos maliciosos, com campanhas de ataques cibernéticos mais complexas e refinadas, denominadas de Ameaças Persistentes Avançadas (APT, sigla em inglês para *Advanced Persistent Threats*). APTs podem ser definidos como ataques direcionados e sofisticados visando acesso aos sistemas de uma organização para exfiltração de dados [16, 17]. Muitas vezes os grupos envolvidos nesses ataques possuem suporte de um Estado (nação), que os empregam principalmente para fins de ciberespionagem. Nos últimos anos, houve considerável aumento no número de APTs [16].

Ameaças Persistentes Avançadas, muitas vezes, possuem a capacidade de modificar seus próprios IOCs [18, 14, 16]. Dessa forma, informações sobre TTPs são cada vez mais relevantes, pois descrevem o comportamento do atacante, sendo, portanto, menos voláteis. A capacidade de proteger a rede e prevenir contra TTPs aumenta consideravelmente a barreira de custo do ataque, visto que, uma vez que o defensor esteja preparado para repelir determinadas ações maliciosas, o atacante precisará aprender novos padrões

de ataque.

Considerando a evolução das ações maliciosas, a organização MITRE elaborou o *framework* ATT&CK, uma base de conhecimento de comportamento dos atacantes que descreve as TTPs conhecidas [18]. A atual versão do ATT&CK (divulgada em abril de 2022) inclui 14 táticas, 191 técnicas e 386 subtécnicas (The MITRE Corporation, 2022). Organizado na forma de uma matriz, esse *framework* busca criar uma padrão para descrever o comportamento e as ações de uma agente cibernético malicioso. Contudo, como as TTPs são normalmente divulgadas em relatórios de CTI na forma de textos não estruturados, classificar esses textos em centenas de técnicas e subtécnicas permanece sendo uma tarefa desafiadora.

A multiplicidade de fontes de CTI produz sobrecarga de informações e torna quase impraticável ao analista extrair manualmente TTPs dessa massa de relatórios [19, 20, 21]. Buscando solucionar esse problema, pesquisadores de segurança cibernética tem recorrido cada vez mais a técnicas de Processamento de Linguagem Natural (NLP, da sigla em inglês para *Natural Language Processing*) e Recuperação de Informação (IR, da sigla em inglês para *Information Retrieval*). Percebeu-se também que a automação é essencial e muitos estudos recentes tem combinado métodos de inteligência artificial com NLP [16, 22, 23, 24].

O campo do Processamento de Linguagem Natural obteve grandes avanços com a incorporação de técnicas de aprendizado de máquina. O trabalho de Conneau *et al.* [25] sobre representação universal de sentenças mostrou que métodos de transferência de aprendizado possuem aplicabilidade em tarefas NLP. Vaswani *et al.* [26] propuseram a arquitetura de Transformadas, um modelo baseado em mecanismo de auto-atenção para representar entradas e saídas.

A arquitetura de Transformadas empregada conjuntamente à técnica de Transferência de Aprendizado permitiu o surgimento dos grandes modelos de linguagem (LLM, da sigla em inglês para *Large Language Models*). Esses modelos podem ser pré-treinados em um volume descomunal de dados e depois disponibilizados para aplicação em tarefas de NLP mais simples (*downstream tasks*), numa etapa denominada "ajuste fino". Proporcionam, assim, grande economia de tempo e recursos, pois permitem utilizar redes neurais de grandes dimensões sem a necessidade de treinar toda a rede para cada uso.

Um dos principais LLM surgido foi o BERT (*Bidirectional Encoder Representations from Transformers*), proposto por Devlin *et al.* [7], que inova com um mecanismo de auto-atenção que não exige a leitura da sentença de forma sequencial, mas recebe em sua entrada a sentença completa, com todos os tokens processados simultaneamente. Esse modelo alcançou desempenho que o alçou a condição de estado da arte para diversas tarefas NLP. Prottasha *et al.* [27] confirmaram essa condição ao testar diferentes modelos de representação (Word2Vec, GloVe, FastText e BERT) e demonstrar que o BERT, com o ajuste fino adequado supera os demais modelos em diversas tarefas de NLP.

O domínio da segurança cibernética, contudo, apresenta uma variedade de dificuldades específicas para o Processamento de Linguagem Natural, como, por exemplo, a necessidade de compreensão de termos técnicos em constante evolução e mutação. Não obstante a evolução no NLP com a aplicação de modelos de aprendizado de máquina, a segurança cibernética ainda não parece ter se beneficiado completamente desses avanços [28]. O *framework* demonstrado neste trabalho inova ao modelar o BERT para aplicação no problema de classificação de texto específico para mapeamento de TTPs junto a um framework estruturado (MITRE ATT&CK) e suas principais contribuições são:

a) adoção de *framework* no estado da arte da arquitetura de Transformadas em NLP (BERT) utilizando 11 modelos diferentes para classificar sentenças nas 253 TTPs técnicas e subtécnicas de ataques cibernéticos mais comuns tabuladas na matrix ATT&CK do MITRE;

b) condução de uma varredura de diferentes combinações de hiperparâmetros selecionados de ajuste fino para avaliação da correlação dos parâmetros com a performance e aprimoramento de desempenho;

c) identificação da melhor configuração dos hiperparâmetros escolhidos e o melhor modelo BERT para classificação de TTPs presentes em textos não estruturados de relatórios de ameaças cibernéticas.

1.1 PROBLEMA DE PESQUISA

O problema de pesquisa que este trabalho se propôs a explorar é a classificação de sentenças de texto não estruturado do domínio cibernético reportando táticas, técnicas ou procedimentos em classes de TTPs definidas em um *framework* padronizado (MITRE ATT&CK). Trata-se de um problema complexo pois existem centenas de classes (TTPs) possíveis para o classificador e a tarefa requer um nível de Entendimento de Linguagem Natural (*Natural Language Understanding*, subdomínio de NLP) por parte de um sistema computacional.

1.2 OBJETIVOS

A hipótese que se pretende investigar é que o uso de um grande modelo de linguagem (LLM) permite alcançar bons resultados para o problema proposto. O LLM escolhido foi o BERT, pois alcança resultados no estado da arte no subdomínio de NLP relativo ao Entendimento de Linguagem Natural [27]. Foram testadas 11 variantes de BERT na tarefa de classificação das TTPs. O objetivo do trabalho é propor uma abordagem para testar e identificar o modelo BERT de melhor desempenho no problema especificado.

Foi realizada uma varredura em dois parâmetros BERT escolhidos (taxa de aprendizado e tamanho do lote). O objetivo específico foi buscar valores valores para esses parâmetros que aprimorem o desempenho do *framework*. Conduziu-se também uma análise qualitativa dos erros de classificação com o objetivo específico de compreender as razões dos erros de modo a permitir possibilidades de melhoria em trabalhos futuros.

1.3 JUSTIFICATIVA

A atividade de inteligência tem como matéria-prima dados e informações e oferece conhecimento como produto do seu trabalho, resultado de seus processos metodológicos próprios [29]. Com o grande volume de informações disponíveis na Era da Informação (*Big Data*), apropriar-se de novas tecnologias e ferramentas produzidas com o intuito de facilitar o tratamento de informações, como a inteligência artificial, é essencial

à constituição de unidades de inteligência eficientes nos paradigmas da sociedade moderna.

Além disso, o ambiente de atuação dos serviços de inteligência do mundo inteiro tem cada vez mais migrado para o meio digital. A Inteligência de Ameaças Cibernéticas produz informações essenciais para segurança da informação das organizações e, dessa forma, constitui um ramo de crescente importância na atividade de inteligência profissional.

A Administração Pública Federal (APF) nem sempre consegue acompanhar a evolução tecnológica com investimentos em recursos humanos e sistemas para fazer frente aos avanços das ameaças cibernéticas. É necessário dotar a APF de mecanismos de automação que tornem mais eficiente o trabalho dos servidores responsáveis pela segurança da informação.

Nesse contexto, pesquisar a aplicabilidade de frameworks de aprendizado de máquina em NLP no contexto de segurança cibernética, representa contribuição relevante à proteção dos ativos informacionais do Estado. Ademais, a proposta desse trabalho é aderente tanto à atividade de inteligência (função precípua da Agência Brasileira de Inteligência, órgão patrocinador desse Mestrado Profissional) quanto à temática de segurança cibernética (tema central do programa acadêmico).

1.4 PUBLICAÇÕES

Os resultados da pesquisa do presente trabalho deram origem a duas publicações em eventos científicos. O artigo "MODELO DE CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT"foi publicado nos Anais da 9ª Conferência Ibero-Americana de Computação Aplicada [30], tendo recebido da organização do evento Menção Honrosa como um dos três melhores artigos. A continuação da pesquisa levou ainda à publicação do artigo “Leveraging BERT’s Power to Classify TTP from Unstructured Text” em *Proceedings of 7th Workshop on Communication Networks and Power Systems 2022* (WCNPS 2022 - Qualis B3) [31].

As publicações podem ser referenciadas da seguinte forma:

- ALVES, Paulo Magno de Melo Rodrigues; GONÇALVES, Vinícius Pereira; FILHO, Geraldo Pereira Rocha. MODELO DE CLASSIFICAÇÃO DE TTP BASEADO EM TRANSFORMADAS BERT. In: Atas das Conferências Ibero-Americanas Computação Aplicada 2022 e WWW/Internet 2022. Lisboa, Portugal: International Association for Development of the Information Society, 2022. p. 51–58.
- ALVES, Paulo Magno de Melo Rodrigues; GONÇALVES, Vinícius Pereira; FILHO, Geraldo Pereira Rocha. Leveraging BERT’s Power to Classify TTP from Unstructured Text. In: 2022 Workshop on Communication Networks and Power Systems (WCNPS 2022). Fortaleza, Brasil: Institute of Electrical and Electronics Engineers (IEEE), 2022.

1.5 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está estruturado como explanado a seguir. O capítulo 2 apresenta a fundamentação teórica dos conceitos empregados na pesquisa. O capítulo 3, Trabalhos Relacionados, promove uma revisão bibliográfica da literatura relacionada. O capítulo 4 apresenta o *framework* implementado, explicitando como foi feita a preparação dos dados e os modelos e configurações utilizados no experimento. O capítulo 5 expõe e aborda a discussão sobre os resultados alcançados com a abordagem proposta e a varredura de hiperparâmetros, além de expor análise qualitativa dos erros de classificação observados *framework*. Ao final, o capítulo 6 apresenta as conclusões do experimento, as limitações encontradas e discute possibilidades de desenvolvimentos futuros convergentes com a linha de pesquisa desse trabalho.

2 REFERENCIAL TEÓRICO

Esse capítulo traz uma breve explanação de alguns conhecimentos teóricos relevantes para uma melhor compreensão do trabalho. Encontra-se dividido em 3 seções, em conformidade com os principais domínios do conhecimento empregados no trabalho: Inteligência de Ameaças Cibernéticas, Processamento de Linguagem Natural e Aprendizado de Máquina.

A seção de Inteligência de Ameaças Cibernéticas contextualiza o conceito de ameaças cibernéticas e traz conceitos de inteligência clássica em uma subseção. Em seguida, há duas seções tratando da definição de CTI (*Cyber Threat Intelligence*) e explicando os diferentes tipos de CTI.

A seção seguinte trata de Processamento de Linguagem Natural (NLP). São abordados alguns conceitos fundamentais como os principais subdomínios e as etapas analíticas tradicionalmente empregadas em NLP. Menciona algumas tarefas comuns na área e traz um breve histórico que permite compreender a evolução de NLP e como se chegou nos paradigmas atuais.

Por fim, a seção de Aprendizado de Máquina apresenta o paradigma mais atual de NLP. Há uma explicação das Redes Neurais Artificiais (RNA) que mostra como o neurônio biológico e o cérebro humano inspiraram essa tecnologia. Além disso, aborda também a técnica de Transferência de Aprendizado, a Arquitetura de Transformadas e o framework BERT, um grande modelo linguístico utilizado nesse trabalho de pesquisa.

2.1 INTELIGÊNCIA DE AMEAÇAS CIBERNÉTICAS

A análise léxica da expressão Inteligência de Ameaças Cibernéticas leva à imediata inferência semântica de tratar-se de um ramo da inteligência, à semelhança dos ramos militar, policial ou corporativa (entre outros). Com efeito, CTI deriva seus princípios da inteligência clássica e seu entendimento depende tanto da compreensão dessa atividade como dos conceitos relativos à ameaças cibernéticas. A definição de ameaças cibernéticas, contudo, não é consensual, variando conforme os objetivos e as motivações do autor [32, 33]. Sob o aspecto conceitual, as principais diferenças ocorrem em razão do enfoque quanto ao entendimento de ameaça como um conjunto de ações ou circunstâncias ou como um agente ou ator malicioso.

Na primeira vertente, o governo americano, por meio do NIST (National Institute of Standards and Technology), define ameaças cibernéticas como “qualquer circunstância ou evento com o potential para impactar operações, ativos ou indivíduos de uma organização por meio de sistemas da informação via acesso não autorizado, destruição, divulgação ou modificação de informação e/ou negação de serviço” (tradução nossa) [34].

A OTAN, todavia, define a ameaça como “um grupo ou indivíduo que tem tanto a capacidade como a intenção de provocar dano” [35]. Semelhante entendimento tem o governo canadense, que define como um “ator malicioso que, usando a internet, aproveita-se de vulnerabilidade em um produto para o propósito de

explorar uma rede e a informação nela contida" (tradução nossa) [36].

A Estratégia Cibernética Nacional do Reino Unido, por sua vez, procura abarcar ambas as vertentes conceituais ao definir ciberameaças como "qualquer coisa capaz de comprometer a segurança ou causar dano a sistemas de informação e dispositivos conectados à internet, e os dados e os serviços neles, primariamente por meios cibernéticos" (tradução nossa) [37].

Neste trabalho adotou-se a concepção mais abrangente de que a ameaça consiste tanto no agente malicioso capaz e motivado a provocar danos aos sistemas de informação como nas suas ações maliciosas.

2.1.1 Inteligência

A compreensão do termo inteligência passa pelo entendimento de sua peculiaridade semântica. Em sua obra clássica "*Strategic Intelligence for American World Policy*", Sherman Kent expôs a particular polissemia do termo por meio de uma acepção trina: inteligência pode referir-se a um produto, uma atividade ou uma organização [29].

Oliveira consigna que a acepção trina produz ambiguidades [38]. Na frase "Graças à inteligência, foi possível neutralizar a ação do inimigo", o termo "inteligência" é passível de interpretação segundo três significados distintos. O leitor pode questionar: graças ao produto de inteligêcia elaborado, graças à atividade de inteligência ou graças ao órgão de inteligência foi possível neutralizar a ação do inimigo? A resposta dependerá do contexto. Os profissionais de inteligência estão habituados à definição trina de Sherman, presente nas principais doutrinas de inteligência pelo mundo [38, 29, 39].

No Brasil, a Agência Brasileira de Inteligência (Abin) define a atividade como "o exercício de ações especializadas para obtenção e análise de dados, produção de conhecimentos e proteção de conhecimentos para o país" [40]. Na sentença anterior é possível observar as três acepções. A Abin é a organização de inteligência, o exercício de ações especializadas refere-se à atividade e a produção de conhecimentos indica o produto dessa atividade.

O conceito da Abin também revela outro aspecto doutrinário importante relacionado à inteligência ao diferenciar dados e conhecimentos. O Decreto nº 4.376, de 13 de setembro de 2002 [41], que dispõe sobre a organização e o funcionamento do Sistema Brasileiro de Inteligência (Sisbin), menciona também o termo "informações", definindo inteligência como "a atividade de obtenção e análise de dados e informações e de produção e difusão de conhecimentos, dentro e fora do território nacional, relativos a fatos e situações de imediata ou potencial influência sobre o processo decisório, a ação governamental, a salvaguarda e a segurança da sociedade e do Estado".

Resta manifesto, portanto, que, no ambiente de inteligência, dados, informações e conhecimentos possuem significados distintos. Como o domínio de CTI toma emprestados esses conceitos, convém, diferenciá-los adequadamente. Dados são observações do mundo, registros de eventos. Não permitem, sozinhos, a compreensão de uma situação [39, 42]. À guisa de exemplo, pode-se dizer que a expressão "50 tanques" constitui meramente um dado. Indica apenas uma quantidade relacionada a um objeto. O dado, por si só, não esclarece tratar-se de tanques militares ou de água, por exemplo, pois carece de contexto.

A informação, por sua vez, consiste no resultado de um primeiro processamento dos dados, que são

organizados e contextualizados [39, 33]. Peter Drucker afirma que “informações são dados dotados de relevância e propósito” [43]. Quando o dado da frase anterior é tratado conjuntamente a outros, a sentença pode tornar-se “há 50 tanques militares no país A”. Agora é possível perceber um contexto militar e geográfico para o dado. No entanto, essa informação, ainda que possua relevância e propósito, ainda não é suficiente para o assessoramento de tomada de decisão em nível estratégico.

O conhecimento consiste no resultado do processamento, interpretação e análise das informações com a finalidade de auxiliar a tomada de decisão [33]. No exemplo anterior, um exemplo de conhecimento seria a frase“há 50 tanques militares no país A, direcionando-se de forma hostil à nossa fronteira”. Nesse caso, há combinação de informações, interpretação e análise que subsidiam o decisor a tomar uma ação. Por essa razão, no ambiente de inteligência, costuma-se dizer que o conhecimento precisa ser acionável.

Na doutrina brasileira, o termo conhecimento é utilizado como sinônimo de inteligência em sua acepção de produto, como pode ser visto na definição supracitada da Abin. Para chegar ao estágio de conhecimento ou inteligência, o dado precisa ser tratado por metodologia especializada. O processo basilar de tratamento de dados nesse domínio é denominado Ciclo de Inteligência.

2.1.1.1 Ciclo de Inteligência

O Ciclo de inteligência consiste na representação do processo metodológico que trata e responde às demandas informacionais do decisor no âmbito da inteligência [39, 44]. Possui algumas variações, mas comumente conta com as seguintes fases: direção e preparação; coleta; processamento; análise; e difusão. A Figura 2.1 mostra as fases e o encadeamento delas no Ciclo:

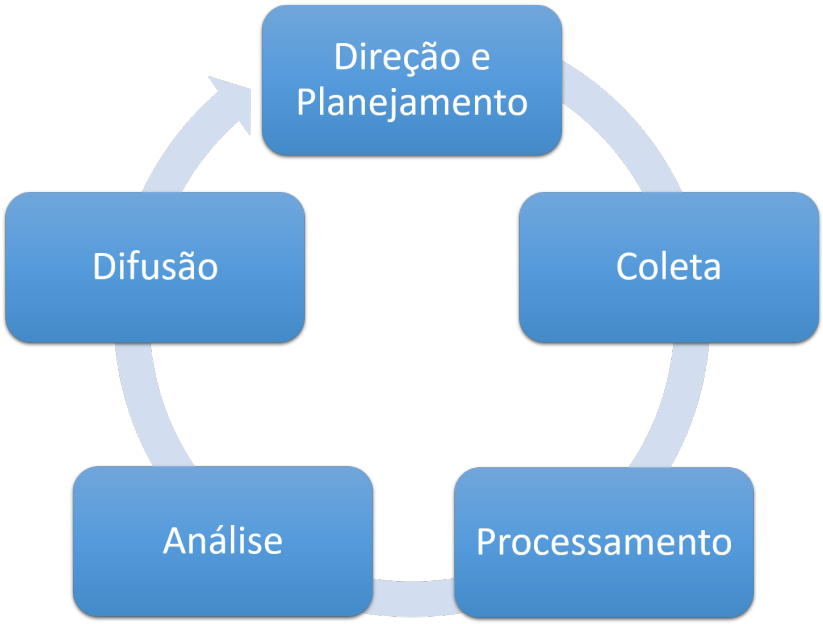


Figura 2.1: Ciclo clássico de inteligência.

De forma sintética, as fases podem ser descritas como a seguir [39, 44]:

- **Direção e Planejamento**: consiste no recebimento da demanda e o planejamento de inteligência de

como atender à necessidade informacional do cliente, detalhando os as fontes e os recursos a serem empregados.

- **Coleta**: também chamada de fase de Reunião, consiste na coleta de dados brutos. As fontes, segundo a CIA (*Central Intelligence Agency* [44], agência governamental americana), podem ser a inteligência de sinais (SIGINT), de imagens (IMINT), humana (HUMINT), de medidas e assinaturas (MASINT), de fontes abertas (OSINT) e geoespacial (GEOINT).
- **Processamento**: consiste na organização dos dados e envolve a conversão de quantidades massivas de dados em uma forma utilizável pelos analistas.
- **Análise**: converte os dados e informações recebidos em um produto acabado de inteligência por meio da integração e avaliação.
- **Difusão**: disseminação do produto acabado da inteligência aos tomadores de decisão.

Importante observar que, como se trata de um ciclo, há, após a fase de Difusão um retorno à fase de Planejamento e Direção. Esse retorno consiste em um *feedback* fornecido pelos tomadores de decisão sobre o produto recebido. Esse *feedback* representa uma avaliação de resultados que irá realimentar o ciclo para ajustes. Em algumas versões do ciclo, particularmente na inteligência comercial, o *feedback* é entendido como uma fase específica, a da Revisão (ou Avaliação) [45].

O fluxo informacional em CTI também se orientará pelos processos concebidos para a inteligência tradicional. Dessa forma, compreendidos a ideia de ameaça cibernética e alguns conceitos fundamentais de inteligência, pode-se adentrar a definição de Inteligência de Ameaças Cibernéticas.

2.1.2 Definição de CTI

A expressão Inteligência de Ameaças Cibernéticas exprime um conceito que não apresenta definição consensual [32]. Diferentes organizações apresentam definições distintas de acordo com seus propósitos e conveniência. Empresas que produzem soluções de segurança, por exemplo, buscam encaixar seus produtos dentro de um conceito de CTI que lhes seja mais comercialmente vantajoso [32, 46].

Conti *et al.* [47] afirmam que a Inteligência de Ameaças se refere ao "conjunto de dados coletados, analisados e aplicados concernentes à ameaças de segurança, atores, *exploits*, *malware*, vulnerabilidades e indicadores de comprometimento" (tradução nossa). O Instituto SANS, por sua vez, entende CTI como "informações analisadas sobre a intenção, a capacidade e as oportunidades de que se beneficiam adversários que têm redes de computador como alvos" (tradução nossa) [48]. O Bank of England, banco central inglês, define como "informação sobre ameaças ou atores de ameaças que provê compreensão suficiente para a mitigação de um evento danoso" (tradução nossa) [33].

Também é importante compreender que, quando autores ou organizações mencionam CTI, podem estar se referindo a diferentes tipos de dados.

2.1.3 Tipos de CTI

A Inteligência de Ameaças Cibernéticas normalmente é classificada em três níveis, de acordo com o tipo de informação que provê [46, 33]:

- **Tático**: inteligência mais técnica baseada em indicadores para defender contra adversários ou proativamente "caçá-los" na rede (*threat hunting*.
- **Operacional**: inteligência focada nas motivações, capacidades e comportamentos do adversário.
- **Estratégico**: inteligência sobre os riscos e implicaçoes que as ameaças representam para o negócio, visando a tomada de decisão de alto nível e o direcionamento de investimentos em segurança.

A tabela 2.1 apresenta algumas características dos níveis de CTI:

Tabela 2.1: Características dos níveis de CTI. Adaptado de [8]

Tipo	**Vida útil**	**Fonte**	**Dados fornecidos (output)**
Tático	Curta (meses)	Eventos de segurança, exemplares de malware, emails de phishing, infraestrutura de ataque	Dados atômicos e estruturados para leitura por máquina a exemplo de IOCs como IPs, domínios, hashes e assinaturas de malware.
Operacional	Média (meses a ano)	Análise de famílias de malwares, grupos de ameaças e comportamento humano	Pequenos textos em tópicos descrevendo técnicas de persistência e comunicações, perfis de atacantes e vítimas, descrições de TTPs, padrões e metodologias utilizadas.
Estratégico	Longa (anos)	Grandes campanhas, intrusões com múltiplas vítimas	Textos corridos sobre vitimologia, tendências, mapeamento de intrusões e campanhas e correlação com eventos econômicos e geopolíticos.

O nível estratégico apresenta certo grau de subjetividade e mostra-se importante para o decisor de alto nível na gestão corporativa. Os níveis tático e operacional são utilizados pelos profissionais de segurança. O nível tático normalmente é suprido por diversos *feeds* de CTI que fornecem IOCs e outros dados semelhantes. No nível operacional, a descrição precisa de comportamentos dos atacantes mostra-se essencial para uma adequada tomada de decisão de segurança. Esse tipo de informação, para ser útil, precisa ser referenciada em uma base de conhecimentos sistematizada desses comportamentos adversários. A matriz MITRE ATT&CK tem tornado-se referência na comunidade de segurança cibernética para a padronização de ações de agentes adversos [13, 20].

2.1.4 MITRE ATT&CK

O MITRE é uma organização americana sem fins lucrativos fundada em 1958 com a visão de buscar soluções para problemas complexos que ameaçem a segurança nacional. Com um portfólio de centenas de patentes, já promoveu avanços em áreas como tecnologia radar, segurança cibernética, GPS, segurança de tráfego aéreo, telecomunicações (5G/6G), pesquisa oncológica, veículos autônomos, inteligência artificial e biologia sintética [2].

No campo da segurança cibernética, o MITRE lançou diversas ferramentas e *frameworks* para auxiliar profissionais da área [2]. Em 2013, a instituição começou a desenvolver um processo para modelar o comportamento de um adversário pós comprometimento da rede. Essa modelagem foi lançada em 2015, sob a forma da matriz ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*). Trata-se de uma base de conhecimento dos comportamentos de agentes cibernéticos adversos que tem se tornado padrão na descrição de TTPs [18]. Essa matriz ajudou a lançar o conceito de *Threat-Informed Defense*, uma estratégia de cibersegurança que prioriza a inteligência de ameaças e utiliza esse entendimento para prevenir, detectar, mitigar ataques e tomar decisões de segurança cibernética complexas [49].

A matriz ATT&CK é baseada em um conjunto de técnicas e subtécnicas que representam ações que o adversário pode realizar para atingir um objetivo [50]. Esses objetivos são representados pelas táticas. A Figura 2.2 mostra a matriz, na qual temos na primeira linha as táticas. Abaixo de cada tática temos uma coluna que elenca as técnicas. As subtécnicas ficam ocultas por uma questão de facilidade de visualização.

Figura 2.2: Matriz MITRE ATT&CK. Adaptado de [2].

As táticas traduzem os objetivos do atacante, isto é, a razão para realizar uma ação. Ao analisar o comportamento do adversário, respondem "o porquê" de tais ações serem executadas [50]. O atacante

pode, por exemplo, estar buscando acesso a credenciais. Esse objetivo encontra-se representado na tática *Credential Access*, na oitava coluna da matriz mostrada na Figura 2.2.

Cada tática apresenta inúmeras técnicas e respondem a perguntas relativas ao "como" um adversário alcança seus objetivos ou "o que" ele ganha ao executar a ação [50]. No caso ilustrado anteriormente, a tática *Credential Access* possui 17 técnicas identificadas. Um atacante poderia, por exemplo, fazer um *dump* de credenciais do sistema operacional para obter credenciais úteis dentro de uma rede (técnica *OS Credential Dumping)*.

As subtécnicas, por sua vez, discriminam de forma mais específica como o comportamento descrito pela técnica é utilizado pra atingir o objetivo tático. No exemplo anterior, a técnica de *OS Credential Dumping* pode ser executada de oito diferentes formas como acesso à memória LSASS, ao *Security Account Manager*, ao /etc/passwd, ao /etc/shadow etc. A Figura 2.3 mostra o exemplo da técnica expandida pelas subtécnicas.

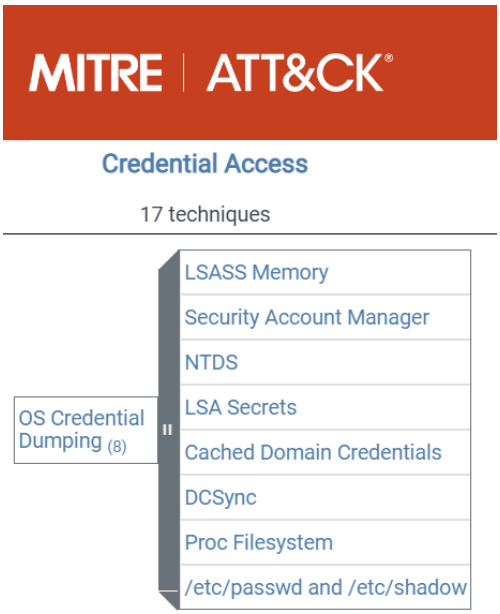


Figura 2.3: Exemplo de tática, técnica e subtécnica na matriz MITRE ATT&CK. Adaptado de [2].

A base de conhecimentos do MITRE apresenta também alguns exemplos de procedimentos (*example procedures*), que consistem em frases exemplificativas sobre como as técnicas e subtécnicas aparecem em relatórios de CTI. Para a situação mostrada anteriormente, considerando a tática *Credential Access*, a técnica *OS Credential Dumping* e a subtécnica *Security Account Manager*, há diversas frases ilustrativas como, por exemplo "*Blue Mockingbird has used Mimikatz to retrieve credentials from LSASS memory*".

A matriz MITRE ATT&CK provê uma organização metodológica das TTPs conhecidas empregadas por atacantes. Diferentemente dos IOCs, que são dados de fácil leitura por máquinas, as TTPs costumam ser divulgadas como texto não estruturado em relatórios de CTI. Processar essas informações com eficácia requer o emprego de técnicas de Processamento de Linguagem Natural aplicadas aos textos de relatórios de CTI.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural consiste em uma área de pesquisa que explora como computadores podem compreender e manipular a linguagem natural [51, 52, 53, 54]. O epíteto "natural" na expressão visa distinguir a linguagem cotidiana, utilizada por humanos, de outras formas, como as notações matemáticas ou linguagens de programação, as quais apresentam vocabulários e regras mais restritos [55, 53]. NLP surge da combinação de métodos de outras duas áreas do conhecimento, linguística e inteligência artificial, e delas deriva muitos de seus princípios e conceitos [54, 52].

NLP pode ser subdividida em dois subdomínios principais: o Entendimento da Linguagem Natural (NLU, da sigla em inglês para *Natural Language Understanding*) e a Geração de Linguagem Natural (NLG, da sigla em inglês para *Natural Language Generation*) [54, 52, 51]. As pesquisas voltadas para NLU visam possibilitar que as máquinas interpretem e analisem a linguagem humana. Já os trabalhos em NLG buscam fazer com que máquinas produzam textos coerentes e inteligíveis.

Processar linguagem natural consiste em tarefa computacionalmente complexa e, dessa forma, os estudos na área costumam decompor o processo em cinco etapas analíticas [51, 56]: léxica, sintática, semântica, do discuro e pragmática. A análise léxica, também conhecida como análise morfológica, refere-se ao estudo do texto no nível das palavras individuais. Realiza-se a busca por morfemas (menor unidade linguística com significado gramatical). Nessa etapa são empregados os métodos de tokenização (separação das sentenças e palavras em tokens), lematização ou stemização (redução a um radical) e rotulação dos componentes do discurso (*parts of speech tagging*, classificação das classes gramaticais do termos, como verbo, pronome, substantivo etc).

A análise sintática examina a estrutura de uma sentença e determina as relações sintáticas entre as palavras, rotulando-as em suas funções (sujeito, verbo, adjunto, objeto etc). A análise semântica verifica uma sentença palavra por palavra e como um todo em busca de extrair significado das sentenças. A análise do discurso procura sentido nas relações lógicas entre as sentenças. Por vezes, o significado de uma frase no texto depende da frase anterior. Um pronome, por exemplo, pode ter seu referente em uma frase antecessora. Já a análise pragmática pressupõe que a máquina tenha conhecimento não apenas do texto fornecido, mas de outras informações externas, que caracterizam um contexto. Nessa etapa ocorrem, por exemplo, desambiguações de expressões polissêmicas.

O Processamento de Linguagem Natural é utilizado para resolver computacionalmente algumas tarefas linguísticas, com aplicabilidades diversas no cotidiano humano. Algumas das tarefas de NLP mais comuns são:

- **classificação de textos**: categorização de textos dentro de classes pré-definidas.
- **modelagem de tópicos**: busca do tema ou assunto principal de um texto
- **análise de sentimento**: busca extrair aspectos subjetivos do texto (emoção, humor, opinião) e é bastante utilizado para extrair tendências de opinião pública (particularmente por meio de mídias sociais).
- **sumarização**: elaboração de sínteses de textos longos.

- **tradução automática**: tradução de textos em um idioma a partir de sua versão em outro idioma.
- **resposta a perguntas**: elaboração de respostas textuais coerentes a uma dada pergunta.
- **geração de linguagem natural (NLG, da sigla em inglês para *Natural Language Generation*)**: estruturação de textos em linguagem humana de forma coerente e inteligível.

A tradução automática e a geração de linguagem natural foram as duas tarefas que originalmente impulsionaram as pesquisas no domínio de NLP.

2.2.1 Breve Histórico

O Processamento de Linguagem Natural como campo de estudo tem origens que remontam ao final da década de 1940, com pesquisas de tradução automática. Em 1947, o matemático americano Warren Weaver escreve uma carta ao também matemático Norbert Wiener comentando que os esforços de identificação de padrões linguísticos para quebra de códigos criptográficos durante a guerra poderiam ser usados em tempos de paz para auxiliar instituições internacionais no problema da tradução [57]. O texto foi transformado em memorando em 1949 e publicado.

Em 1950, o matemático britânico Alan Turing publicou seu artigo seminal *Computing Machinery and Intelligence* [58], no qual propôs o que passou a ser conhecido como teste de Turing. A proposta consistia em uma versão do "jogo da imitação", em que uma máquina tentaria emular, por meio da linguagem, o comportamento humano. Durante as décadas seguintes, o teste de Turing passou a ser considerado um paradigma para o alcance da inteligência artificial. Em 1954, o experimento de Georgetown, uma colaboração da universidade homônima e a IBM, mostrou a tradução automática de 49 frases selecionadas do idioma russo para a língua inglesa [59, 60]. O experimento apresentava vocabulário limitado a 250 palavras e apenas seis regras gramaticais, porém gerou enorme entusiasmo na comunidade científica do período. Os cientistas participantes previam que entre três e cinco anos a tradução automática seria um problema resolvido.

A partir da segunda metade da década de 1950, Noam Chomsky publica alguns artigos introduzindo os conceitos de gramática gerativa transformacional e gramática universal. O linguista americano propôs a existência de princípios estruturais universais (inatos ao ser humano) para a construção de sentenças. O trabalho de Chomsky, por um lado, influencia pesquisadores a buscar definir manualmente regras gramaticais para decomposição, processamento e análise dos textos; e, por outro lado, possibilita dimensionar a dificuldade do problema da tradução automática [61].

Outros campos de NLP também começaram a ser pesquisados, como a subárea de geração de linguagem natural (NLG, da sigla em inglês pra *Natural Language Generation*). Um marco inicial desse domínio foi o software ELIZA, desenvolvido no Massachusets Institute of Technology (MIT) pelo cientista alemão Joseph Weizenbaum e lançado em 1966. Considerado o primeiro chatbot (robô de conversação com humanos), utilizava regras de busca por padrões (palavras chave) e substituições que emulavam uma conversa com um psicoterapeuta [53]. Aplicava a abordagem reflexiva da escola Rogeriana de psicoterapia, na qual diversas perguntas eram rebatidas para o paciente. Dessa forma, quando ELIZA não encontrava uma resposta razoável por meio das palavras chave, a pergunta era devolvida reflexivamente ao interlocutor.

As abordagens baseadas em regras, contudo, obtiveram sucesso muito limitado devido à complexidade e à variabilidade da linguagem humana. Em 1964 a Fundação Nacional da Ciência americana criou um comitê, o Automatic Language Processing Advisory Committee (ALPAC), para para avaliar o progresso da tradução automática. O relatório ALPAC de 1966 trouxe severas críticas às pesquisas da área, concluindo que a tradução automática era mais lenta, menos precisa e duas vezes mais custosa que a tradução humana [62]. Afirmou ainda que seria mais vantajoso investir em auxílios ao tradutor humano, como dicionários automáticos. O relatório implicou a drástica redução de recursos nos Estados Unidos para pesquisas em NLP, particularmente no campo do tradução automática.

Não obstante o período de escassez nos Estados Unidos, pesquisas continuaram em outras partes do mundo. A Universidade de Grenoble, na França, havia criado, em 1961, o Centre d'Études pour la Traduction Automatique (CETA). Voltado à pesquisa de tradução automática entre francês e russo, o projeto seguia a abordagem *interlingua*, isto é, a criação de uma representação semântica neutra (independente das línguas usadas) como uma espécie de idioma neutro intermediário entre as línguas de origem e destino da tradução. Entre 1967 e 1971, foi usado para traduzir 400.000 palavras de textos russos de matemática e física para o idioma francês [60].

No Canadá, em 1965 foi lançado o grupo TAUM (*Traduction Automatique à l'Université de Montréal*), com o objetivo de pesquisar tradução automática de textos entre os idiomas inglês e francês no país bilíngue [60]. Na União Soviética, a Teoria Sentido-Texto (*Meaning-Text Theory*, proposta em 1965, e os demais trabalhos de Igor MelĆhuk, fundamentam pesquisas linguísticas baseadas na abordagem de *interlingua*. MelĆhuk emigrou para o Canadá em 1976, mas suas ideias, como as funções lexicais universais, continuaram a influenciar a pesquisa não apenas na Rússia, mas também em outras partes do mundo.

Avanços tanto conceituais quanto na operacionalização das ferramentas levaram a alguns avanços na década de 1970. No campo teórico, William Woods introduziu o conceito de Redes de Transição Aumentada (*Augmented Transition Networks*) para a representação de linguagem natural. A ideia consiste no uso recursivo de de máquinas de estado finitas para concluir o estado de uma frase e realizar eventuais desambiguações semânticas necessárias [63]. Nas campo das pesquisas aplicadas, percebe-se que é possível obter bons resultados de tradução em domínios específicos do conhecimento quando se introduzem restrições (vocabulário e regras gramaticais) à entrada de texto.

Nesse período, cresce a demanda por sistemas de tradução automática. O sistema SYSTRAN foi implementado em 1970 na Força Aérea dos Estados Unidos, para tradução entre os idiomas russo e inglês e, em 1976, na Comunidade Europeia (atualmente União Europeia), para traduções entre diferentes idiomas [59]. Também em 1970, na França, surgiu o TITUS, sistema de tradução multilíngue com entrada controlada [59]. Em 1972, a Universidade Chinesa de Hong Kong lançou o CULT para tradução de textos matemáticos do chinês para o inglês, no entanto sua saída ainda necessitava de trabalho humano extensivo de pós-edição [59].

Com a ascensão do Japão e da Alemanha a posições de protagonimo econômico global, surgem, no início da década de 80 diversos sistemas de tradução automática envolvendo os idiomas japonês e inglês (PENSEE, MELTRAN, AS-TRANSAC, HICATS, ATLAS etc) e alemão e inglês (LOGOS e METAL) [59]. Contudo, a maioria das modelos utilizados até então eram baseados na abordagem simbólica, isto é, regras pré-escritas que deveriam ser aplicadas pelos modelos [63].

As décadas de 1980 e 1990 presenciam o início de uma revolução no campo de NLP com a aplicação de abordagens estatísticas e o uso crescente de algoritmos de aprendizado de máquina. Algoritmos como árvores de decisão conseguiam deduzir melhor um resultado ótimo para tarefas de NLP e modelos probabilísticos suportavam a decisão provendo avaliação de confiança.

Após a virada do milênio percebe-se a diversificação e o crescimento do uso de algoritmos de aprendizado de máquina, como Naive Bayes, *Support Vector Machines* (SVM) e Redes Neurais Artificiais, para tarefas de NLP. O surgimento de ferramentas e bibliotecas de NLP, como o CoreNLP e o Stanza (ambas da Universidade de Stanford, nas linguagens Java e Python, respectivamente), o NLTK (*Natural Language Toolkit*) e o Spacy contribuíram para tornar mais acessível o desenvolvimento de aplicações de NLP.

A evolução de NLP seguiu com o uso de novas técnicas e métodos de inteligência artificial, como aprendizado profundo (*deep learning*), transferência de aprendizado (*transfer learning*) e transformadas [63]. Verificou-se maior ênfase em abordagens de aprendizado de máquina supervisionado e não supervisionado, bem como em técnicas de pré-treinamento e ajuste fino (*finetuning*) em grandes quantidades de dados. As pesquisas convergiram fortemente para a aplicação de inteligência artificial à solução de tarefas de NLP [52].

Grandes modelos de linguagem baseados em transformadas revolucionam a solução de tarefas de NLP. O BERT (*Bidirectional Encoder Representations from Transformers*), lançado em 2018, atinge o estado da arte em diversas tarefas, principalmente aquelas relacionadas ao subdomínio de NLU (*Natural Language Understanding*). O GPT (*Generative Pre-Training*, avançou o estado da arte em NLG (*Natural Language Generation*), com seu chatbot (chatGPT) alcançando popularidade mundial, inclusive fora dos ambientes estritamente técnicos.

Esses desenvolvimentos têm levado ao rápido crescimento de aplicações de NLP em áreas como assistentes virtuais, análise de sentimentos em redes sociais, tradução automática em tempo real, respostas de perguntas, sumarização e classificação de textos. As aplicações de NLP encontram-se cada vez mais incorporadas em diversos setores da sociedade, incluindo medicina, finanças, atendimento ao cliente, marketing e análise de dados.

2.3 APRENDIZADO DE MÁQUINA

Na era do grande volume de dados, o chamado *Big Data*, dados são produzidos e consumidos constantemente. O *Big Data* implicou a necessidade de um paradigma complementar na computação de dados. Anteriormente, para resolver um problema computacional, utilizáva-se um algoritmo, isto é, em um conjunto de instruções para resolver um problema ou executar determinada tarefa. Em problemas envolvendo dados, algoritmos transformam dados de entrada em uma saída. Contudo, em diversos problemas modernos, a crescente complexidade dos problemas a serem resolvidos computacionalmente e o grande volume de dados implicou a necessidade de buscar soluções mais sofisticadas, com maior nível de automação [64].

Na era do *Big Data*, diversos problemas lidam com grandes quantidades de dados. Esse volume torna inviável o estabelecimento de instruções codificadas previamente para tratar todas as possíveis conclusões que podem ser obtidas dos dados. Dessa forma, é necessário que a máquina extraia dinamicamente o

algoritmo, que é aprendido e ajustado à medida que mais dados entram e mais saídas são produzidas.

O Aprendizado de Máquina (AM) consiste na programação de computadores para otimizar um determinado critério de performance utilizando dados exemplificativos ou experiências passadas [65]. O que se deseja é que a máquina aprenda um algoritmo dinâmico que forneça soluções razoáveis ao problema apresentado.

O AM representa um novo paradigma na programação de computadores [3]. Tradicionalmente, computadores são programados por humanos por meio de regras escritas (o programa), que transformam os dados de entrada em respostas. Na Aprendizagem de Máquina, a máquina recebe os dados de entrada e algumas respostas e desses dois conjuntos deriva as regras (que poderão ser aplicadas posteriormente a outros dados de entrada). A Figura 2.4 ilustra essa mudança de paradigma:

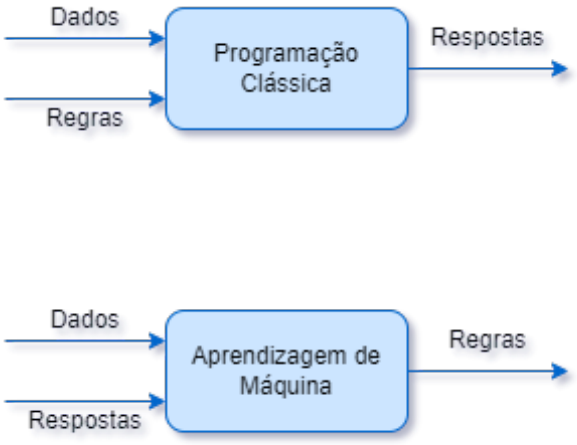


Figura 2.4: Mudança de paradigma na programação de computadores. Adaptado de [3]

Considere, como exemplo, o problema da predição do comportamento de consumidores. Trata-se de um problema para o qual há forte incentivo econômico para a busca de soluções razoáveis. Contudo, o comportamento do consumidor muda de acordo com o tempo e com outras variáveis individuais [65]. Um algoritmo estático não é capaz de acompanhar a multitude de variáveis e suas mudanças. A Aprendizagem de Máquina proporciona sistemas que aprendem dinamicamente, fornecendo soluções razoáveis ao problema ainda que suas variáveis de entrada se alterem, uma vez que aprende com as mudanças. Essa adaptabilidade é essencial em NLP, dada a complexidade dos sistemas linguísticos humanos e a imprevisibilidade das entradas.

Aprendizado de máquina e Processamento de Linguagem Natural são dois domínios do conhecimento que se complementam e beneficiam-se mutuamente. Feigenbaum afirmou que "o problema da aquisição de conhecimento é o gargalo crítico da inteligência artificial" [66]. Se, por um lado, NLP precisa da aprendizagem de máquina para processar a complexidade da linguagem, a qual não pode ser captada por meros sistemas de regras fixas; por outro, NLP pode resolver o problema do gargalo do conhecimento em IA ao adquirir conhecimento na enorme base textual da humanidade.

Entre os principais avanços em Aprendizado de Máquina que permitiram significativa evolução em NLP estão o desenvolvimento de Redes Neurais Artificiais multicamadas e a Transferência de Aprendizado.

2.3.1 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são modelos de inteligência artificial inspirados no funcionamento do cérebro humano [6, 3]. Dessa forma, a compreensão do funcionamento do cérebro humano permite o entendimento de princípios subjacentes às RNA.

O neurônio é a unidade básica de formação do cérebro biológico [5, 4]. Trata-se de uma célula de aspecto pouco usual e constituída de três partes principais: o corpo celular, pequenas ramificações denominadas dendritos e uma ramificação mais extensa chamada de axônio [6, 4]. O axônio inicia-se no corpo celular e em sua outra extremidade encontram-se terminações nervosas que transmitem impulsos aos dendritos de outros neurônios por meio de uma região chamada sinapse, na qual circulam substâncias neurotransmissoras. A Figura 2.5 ilustra as partes componentes de um neurônio biológico:

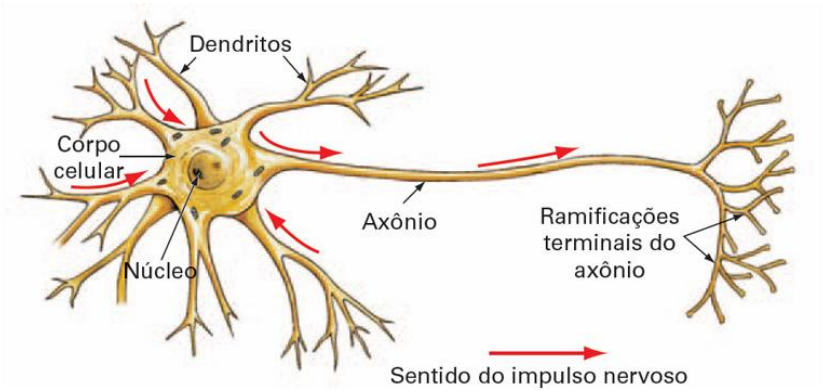


Figura 2.5: Neurônio biológico [4].

O cérebro humano é formado por bilhões de neurônios e trilhões de conexões entre eles. Essa rede neural altamente interconectada torna-o em um dispositivo extremamente eficiente. Em 1943, o neurofisiologista Warren McCulloch e o matemático Walter Pitts propuseram uma simplificação matemática do funcionamento dos neurônios no cérebro dos animais [6, 4]. Essa proposta ficou conhecida como o primeiro modelo de neurônio artificial. Funcionava pela ativação do neurônio quando mais do que um determinado número de entradas era ativado. A despeito da simplicidade, os autores demonstraram que qualquer proposição lógica poderia ser computada por uma rede desses neurônios.

Em 1957, Frank Rosenblatt propôs o Perceptron, uma arquitetura ligeiramente diferente para um neurônio artificial [6]. Na arquitetura Perceptron, o neurônio funciona com uma unidade de limiar lógica (TLU, da sigla em inglês para *threshold logic unit*), também conhecida como unidade de limiar linear. A Figura 2.6 ilustra o funcionamento de um neurônio Perceptron.

Para um dado neurônio k as entradas x são associadas a pesos w. A TLU computa a soma ponderada v_k dos pesos das entradas [6, 5]:

$$v_k = \sum_{j=1}^{m} w_{kj}\, x_j \tag{2.1}$$

A saída y_k é determinada pela aplicação de uma função de ativação φ ao resultado da soma computada e a uma entrada de viés, a qual é fixa e sempre igual a 1 [6, 5]:

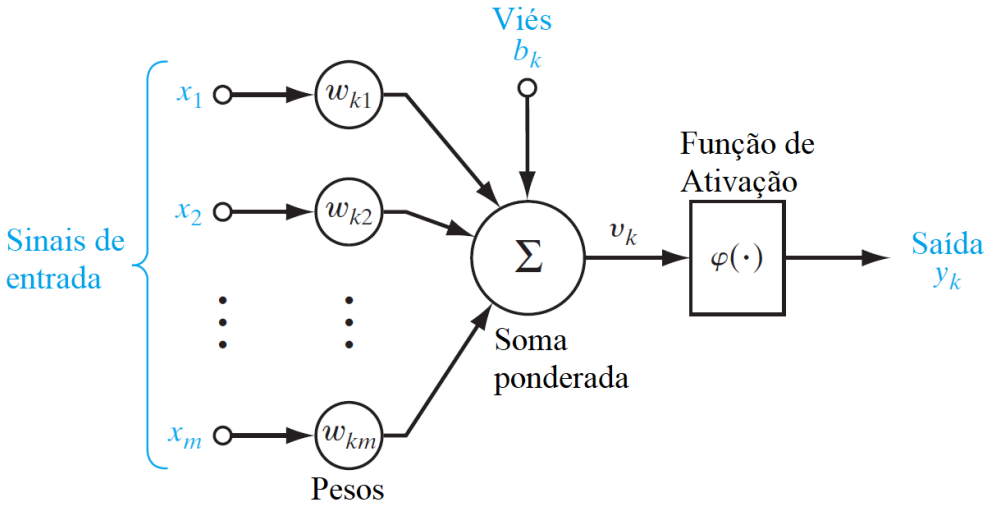


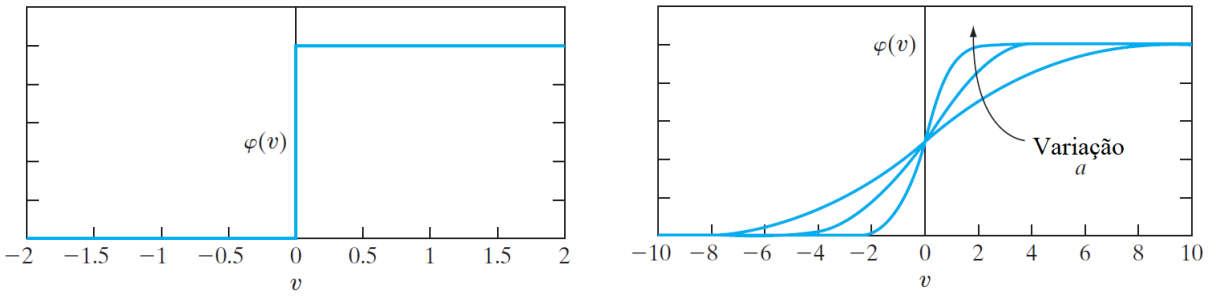
Figura 2.6: Neurônio na arquitetura Perceptron. Adaptado de [5].

$$y_k = \varphi\left(v_k\right) \tag{2.2}$$

Há diversas funções de ativação, sendo as mais comuns a função degrau (*heaviside* ou *step*) e as funções *sigmoid* [5]. Para a função degrau, tem-se:

$$\varphi\left(v\right) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \tag{2.3}$$

Um neurônio Perceptron propõe-se a realizar uma classificação em duas classes (binária) baseada nas entradas. Essa classificação implica a definição de um hiperplano separando duas regiões de decisão [5]. Neurônios baseados na função degrau são conhecidos como modelos McCulloch-Pitts, em homenagem aos pioneiros. A saída retorna valor 1 sempre que o valor for não negativo e 0 nos outros casos. Trata-se de uma abordagem "tudo ou nada", característica da curva de uma função degrau, como pode ser visto na Figura 2.7a.



(a) Função de Ativação Degrau (Heaviside)

(b) Funções de Ativação *Sigmoid*

Figura 2.7: Funções de Ativação mais comuns. Adaptado de [5]

Funções *sigmoid* têm gráfico na forma de "S", como observado na Figura 2.7b, e são a forma de ativação mais comum em redes neurais [5]. São funções de crescimento contínuo e apresentam um equilíbrio

entre o comportamento linear e não linear. Um exemplo de função *sigmoid* é a função logística, definida como

$$\varphi(v) = \left(\frac{1}{1 + e^{-av}} \right) \tag{2.4}$$

onde a representa o parâmetro de inclinação da função e quanto maior o valor de a, menor a inclinação da curva (no limite, quando a tende ao infinito, a função *sigmoid* iguala-se à função degrau). Enquanto a função degrau assume apenas os valores 0 ou 1, uma *sigmoid* pode assumir qualquer valor entre 0 e 1, apresentando, portanto, um comportamento mais suave. Outra distinção importante é que a função degrau não apresenta derivada no ponto 0, enquanto as funções *sigmoid* são diferenciáveis em toda a faixa de valores [5]. Essa última característica seria essencial para o desenvolvimento de algoritmos de correção de erros baseados em gradiente descendentes.

A combinação de vários neurônios artificiais origina o que se convencionou chamar de Redes Neurais Artificiais. O encadeamento de camadas de neurônios originou a chamada Perceptron Multicamadas (MLP, sigla em inglês para *Multi-Layer Perceptron*). As primeiras redes Perceptron eram compostas de três camadas: entrada, camada intermediária (oculta) e saída, conforme ilustrado na Figura 2.8.

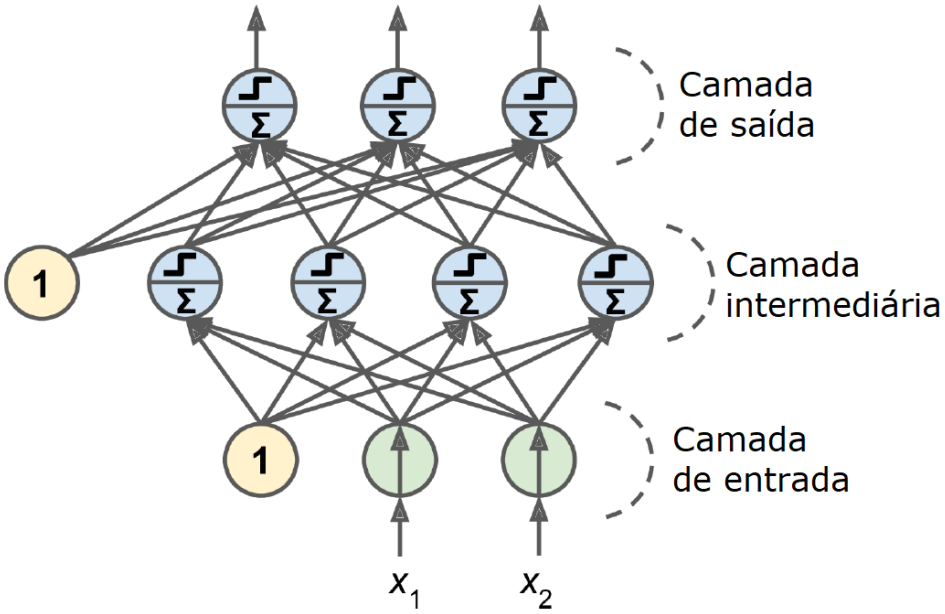


Figura 2.8: Perceptron Multicamadas. Adaptado de [6].

Modelos como o ilustrado na Figura 2.8 estabeleceram o paradigma multicamadas inicial, conhecido atualmente como aprendizado raso (*shallow learning*). No entanto, pesquisadores logo perceberam que a utilização de diversas camadas poderia, teoricamente, resolver problemas mais complexos. O empilhamento de inúmeras camadas intermediárias ocultas deu início ao paradigma mais moderno conhecido como aprendizado profundo (*deep learning*) [6]. Todavia, pesquisadores tiveram, durante anos, dificuldades em treinar com sucesso redes neurais multicamadas como as MLPs [6].

Em 1962, Rosenblatt introduziu o conceito teórico de uma "correção pela propagação reversa de erros". Em 1970, a tese de mestrado de Seppo Linnainmaa formulou, inclusive programaticamente (na linguagem

FORTRAN), a ideia de Rosenblatt. Em 1986, David Rumelhart, Geoffrey Hinton e Ronald Williams publicaram um artigo seminal no qual desenvolvem o algoritmo *Backpropagation* [6]. Utilizando-se de uma técnica eficiente para computação automática dos gradientes descendentes em somente dois passos, o algoritmo descobre como cada peso ou termo de viés deve ser ajustado para reduzir o erro.

O algoritmo *Backpropagation* foi um marco nas pesquisas de redes neurais pois, ao facilitar a convergência, possibilitou o treinamento de redes cada vez mais profundas (com mais camadas intermediárias) para resolução de problemas mais sofisticados. Os avanços em *deep learning* suscitaram o desenvolvimento de novas técnicas que exploram a arquitetura multicamadas, como a Transferência de Aprendizado.

2.3.2 Transferência de Aprendizado

A Transferência de Aprendizado (*Transfer Learning*) é uma técnica de aprendizagem de máquina que consiste na reutilização de um modelo pré-treinado como ponto de partida para um modelo a ser aplicado em um novo problema, com aproveitamento do conhecimento adquirido no pré-treinamento para aperfeiçoar generalizações na resolução do novo problema [67].

Os seres humanos utilizam métodos de transferência de aprendizado naturalmente. Um indivíduo que tenha aprendido a andar de bicicleta, por exemplo, ao tentar aprender a pilotar uma moto, certamente aproveitará conceitos de seu aprendizado na bicicleta (equilíbrio em duas rodas, movimentação do guidão etc), a despeito de tratar-se de duas tarefas claramente distintas.

As principais vantagens da técnica de transferência de aprendizado consistem em economia de tempo de treinamento dos modelos, melhoria da performance na segunda tarefa e possibilidade de prescindir de grandes quantidades de dados na segunda tarefa [68]. Trata-se de método muito utilizado nos domínios de visão computacional e processamento de linguagem natural.

Nas redes neurais multicamadas, as primeiras camadas (mais próximas da entrada) aprendem características mais gerais enquanto as últimas (mais próximas da saída) aprendem os aspectos mais específicos [69]. Usando a visão computacional como exemplo, ao alimentar a rede com um banco de dados como o ImageNet em uma tarefa de classificação de animais, as primeiras camadas podem distinguir animais de veículos, árvores, edifícios etc. As camadas intermediárias podem distinguir mamíferos de aves, peixes e insetos. As últimas camadas poderiam ser empregadas para distinguir diferentes tipos de mamíferos.

Sob a ótica da arquitetura das redes neurais, a Transferência de Aprendizado consiste em aproveitar o aprendizado (pesos das conexões) das primeiras camadas e retreinar as últimas para uma tarefa específica. No exemplo anterior, a camada de saída ou algumas poucas camadas poderiam ser retreinadas para identificar cães em imagens genéricas, por exemplo. Praticamente todo o aprendizado anterior seria aproveitado, não sendo necessário treinar toda a rede a partir do zero. O aproveitamento dos pesos de diversas camadas adquiridos no treinamento prossibilita um ganho considerável em termos de economia de recursos computacionais.

Esse conceito de aproveitamento do aprendizado anterior constitui a base dos Grandes Modelos de Linguagem (LLM, sigla em inglês para *Large Language Models*). Esses modelos são treinados em uma quantidade extraordinária de textos e disponibilizados posteriormente para ajuste fino em tarefas alterna-

tivas (*downstream tasks*). O desenvolvimento dos LLMs deve-se em grande medida ao surgimento da Arquitetura de Transformadas, que permitiu combinar a técnica de Transferência de Aprendizado e o mecanismo de auto-atenção em uma estrutura de Rede Neural Artificial.

2.3.3 Arquitetura de Transformadas e o Mecanismo de Atenção

No *paper* que deu origem à Arquitetura de Transformadas, em 2017, entitulado "*Attention is all you need*" ("Atenção é tudo o que você precisa", tradução nossa) [26], os autores trazem o conceito de auto-atenção. Trata-se de mecanismo que emula uma capacidade cognitiva humana, assim como no caso das Redes Neurais Artificiais. Os seres humanos possuem uma capacidade inata de prestar atenção no que é relevante e ignorar o restante, isto é, filtrar o ruído. Ignoramos tudo o que não representa uma ameaça ou requeira uma reação imediata de nós [70].

De forma semelhante, percebeu-se que os modelos de aprendizado de máquina precisariam atentar apenas a componentes realmente relevantes e não desperdiçar recursos computacionais com o restante. Suponha-se, à guisa de exemplo, que se deseje realizar a tradução automática da frase "*We like this article*" do idioma inglês para o português. Para traduzir o verbo "*like*", é importante dar atenção ao termo antecessor ("*we*"), pois em português, a forma que o verbo tomará dependerá de seu sujeito (conjugação verbal). O mesmo termo "*like*" pode ser traduzido de diferentes formas:

- I **like** -> Eu **gosto**
- We **like** -> Nós **gostamos**

No processo de tradução da frase em questão, ao traduzir o termo "*like*", desde que seja dada a devida atenção ao termo antecessor ("*we*"), o restante da frase pode ser ignorado. Continuando a percorrer a sentença, encontra-se o termo "*this*". Para a tradução dessa palavra, é importante observar o termo sucessor ("*article*"). Isso porque, em português, o termo "*this*" pode ser traduzido para diferentes gêneros, de acordo com o substantivo com o qual se vincula:

- **this** article -> **desse** artigo
- **this** research -> **dessa** pesquisa

Assim, a correta tradução da frase em inglês "*We like this article*" para sua correspondente em português "Nós gostamos desse artigo" depende de atenção aos detalhes corretos. O mesmo conceito aplica-se a diferentes tarefas de NLP. O significado de uma palavra é fortemente afetado por seu contexto [71]. A arquitetura de Transformadas foi projetada para aplicar o conceito de atenção em NLP. A Figura 2.9 apresenta um diagrama simplificado dessa arquitetura ilustrado com a frase exemplo anterior.

Originalmente pensada para a tarefa de tradução automática, essa arquitetura é constituída de dois componentes principais: os blocos Encoder e Decoder. Cada um desses blocos é composto de camadas. As camadas do Encoder possuem duas subcamadas: autoatenção e *feed foward*. As sentenças de entrada são separadas em *tokens* (palavras ou partes de palavras) e cada *token* é expresso na forma de um vetor. Essa

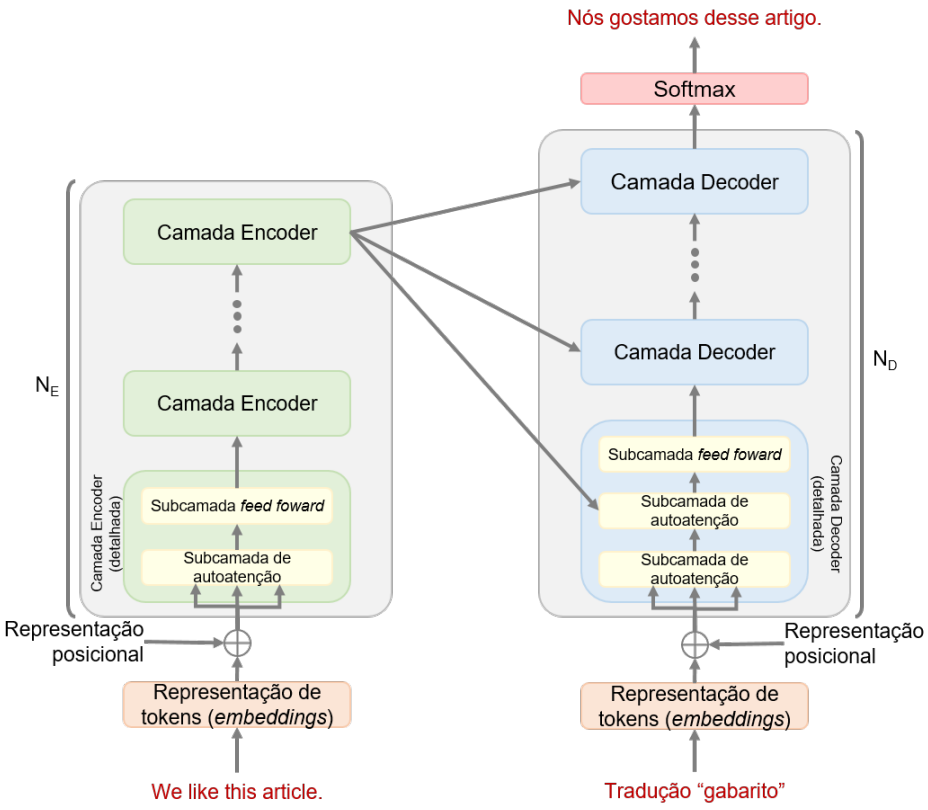


Figura 2.9: Arquitetura de Transformadas - diagrama ilustrativo simplificado.

sequência constitui a representação (*embedding*) de *tokens* da entrada, a qual é somada a uma representação posicional (indica a posição do *token* na frase) antes de ser enviada ao bloco Encoder. Para fins de explicação e exemplificação da arquitetura, considera-se que cada palavra corresponde a um token.

A subcamada de autoatenção recebe essa sequência de palavras e produz um vetor de atenção que estima a importância de cada termo na frase para uma determinada palavra. Para tanto, recebe e processa todas as palavras simultaneamente. Para minimizar a predominância da importância da própria palavra para ela mesma, esse processo é feito em diversas "cabeças de atenção" (*attention heads*) simultaneamente (inicializadas com pesos aleatórios distintos). Os vetores de atenção são repassados e uma rede neural do tipo *feed foward* os transforma em uma nova sequência no formato aceitável pela subcamada de atenção da próxima camada. O processo é repetido N_{E} vezes, sendo esta variável o número de camadas do bloco Encoder.

Os vetores resultantes desse processo sequencial, na última camada de encoder, são repassados a todas as camadas de decoders. O bloco Decoder, diferente do Encoder, trabalha a sentença sequencialmente, palavra por palavra. Assim, para prever a próxima palavra, terá conhecimento apenas das palavras já previstas anteriormente na frase. Em cada camada Decoder há uma subcamada de autoatenção inicial. No treinamento para tradução, ela receberá a sentença correta no idioma alvo. Essa sentença será deslocada à direita (*shifted right*) pois, como o Decoder trabalha sequencialmente, não faria sentido conhecer de antemão a palavra que se deseja traduzir naquele passo.

Há ainda uma subcamada de autoatenção intermediária que recebe os vetores do bloco Encoder. Re-

cebe também, da subcamada de autoatenção inicial, o "gabarito" dos termos anteriores ao que se está trabalhando (na tradução do quarto termo, por exemplo, recebe o "gabarito" até o terceiro termo da subcamada anterior). Os resultados em cada camada são passados por uma rede *feed foward*, que prepara os vetores resultantes para o próximo passo. À semelhança do que ocorre no bloco Encoder, o processo é repetido N_{D} vezes, equivalente ao número de camadas do bloco Decoder. Após o processamento pelo bloco Decoder, uma camada softmax é utilizada para determinar as probabilidades do próximo termo a ser colocado na frase em construção.

Conforme mencionado anteriormente, a Arquitetura de Transformadas foi pensada para a tarefa de tradução automática, de modo que seus componentes possuem função específica nesse processo. No entanto, os blocos Encoder e Decoder podem ser utilizados separadamente. Assim, a Arquitetura de Transformadas deu origem a três tipos de LLM, de acordo com os componentes utilizados:

- Somente Encoder: o bloco Encoder recebe uma sentença na entrada e produz como saída uma representação vetorial que capta as relações e dependências semânticas entre os termos utilizados. Dessa forma, é melhor em tarefas NLU, isto é, que requeiram compreensão de linguagem natural como classificação de sentenças e extração de entidades do texto (NER, sigla em inglês para *Named Entity Recognition*). Os principais representantes dessa categoria são o BERT e suas variantes.
- Somente Decoder: o bloco Decoder recebe textos na entrada e produz sequências textuais na saída. Como esses modelos utilizam apenas o conhecimento pregresso do que já foi processado para predizer o próximo termo, são também conhecidos como modelos autorregressivos. Esse tipo é otimizado para tarefas que envolvam a geração de texto, como chatbots, por exemplo. Os principais representantes dessa categoria são GPT e XLNet.
- Encoder-Decoder: utilizam os dois blocos da Arquitetura de transformadas. Recebem textos na entrada e utilizam a representação produzida pelo Encoder para gerar, no Decoder, saídas de texto. São otimizados para geração de texto que requeiram uma entrada de texto específica, como nas tarefas de tradução e sumarização. Os principais representantes dessa categoria são BART e T5.

O problema de pesquisa abordado nesse trabalho encontra-se no campo de NLU e envolve classificação de sentenças. Dessa forma, a investigação deste trabalho concentrou-se nos modelos que utilizam somente o bloco Encoder, ou seja, BERT e suas variantes.

2.3.4 BERT

Em 2018, o Google lançou e tornou *open source* o BERT (*Bidirectional Encoder Representations from Transformers*). Trata-se de um *framework* de aprendizagem de máquina para processamento de linguagem natural baseado em transformadas.

A codificação dos dados de entrada de BERT consiste em uma combinação de três representações (*embeddings*) para cada fragmento de texto ou palavra (*token*):

- **representação de tokens**: separação em subpalavras e acréscimo de *tokens* especiais ([CLS] no início das sentenças e [SEP] ao final);

- **representação de segmentos**: cada *token* recebe um indicador que marca se pertence à sentença A ou à sentença B da entrada. Para tarefas com entrada de sentença única, essa representação torna-se igual para todos os *tokens*; e
- **representação de posição**: sinaliza a posição de cada *token* no texto de entrada.

A figura 2.10 mostra a representação de duas frases em inglês utilizadas no artigo original de Devlin *et al.* [7]. Para cada *token*, sua representação será constituída da combinação das representações de *token*, de segmento e de posição.

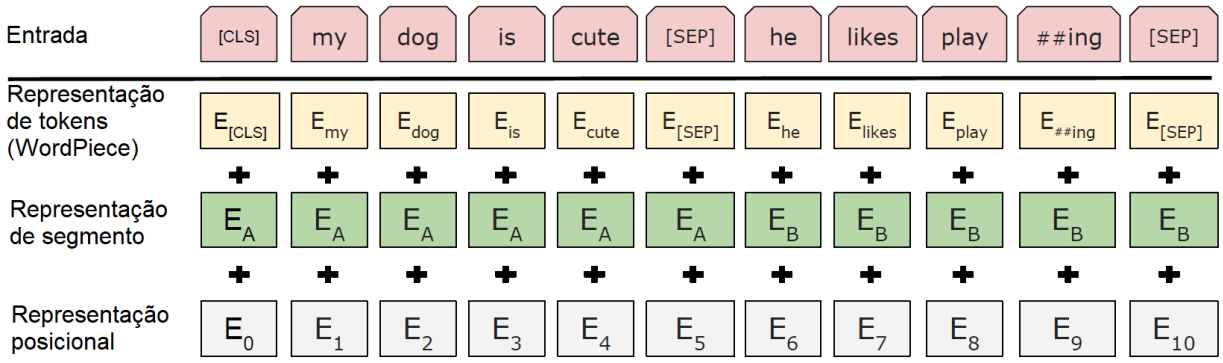


Figura 2.10: Entrada do BERT. Adaptado de [7]

BERT utiliza o algoritmo WordPiece como parte de seu processo de "tokenização". O WordPiece é um algoritmo de segmentação de palavras que divide uma palavra em subpalavras menores, permitindo que o modelo aprenda a lidar com palavras novas e desconhecidas [72]. O processo de "tokenização" do BERT pode ser sumarizado da seguinte maneira:

- O texto de entrada é dividido em frases e cada frase é dividida em *tokens* usando o algoritmo WordPiece.
- Um *token* especial [CLS] é adicionado no início de cada frase.
- *Tokens* especiais [SEP] são adicionados entre as frases para separá-las umas das outras.
- Os *tokens* são convertidos em IDs numéricos utilizando um vocabulário pré-definido.

O algoritmo WordPiece divide as palavras em subpalavras com base na frequência do corpus. Uma das principais vantagens do WordPiece é que ele permite que o modelo lide com palavras que não estão no vocabulário. Por exemplo, a palavra "aviãozinho" passa a ser segmentada nos *tokens* [avi], [##ão] e [##zinho] em que "##" indica *tokens* vinculados ao anterior em uma mesma palavra. Caso o modelo nunca tenha visto a palavra "aviãozinho", ele buscará sentido e construirá uma representação a partir dos *tokens* de cada subpalavra e das possíveis combinações entre elas. Isso permite que o modelo lide com palavras novas ou fora do vocabulário, descrevendo-as com subpalavras que ele já conhece.

Em resumo, o BERT utiliza o algoritmo WordPiece durante o processo de "tokenização" para dividir as palavras em subpalavras e construir uma representação numérica dos textos de entrada. O tamanho

do vocabulário pode ser significativamente reduzido usando o WordPiece, facilitando o treinamento do modelo e economizando recursos computacionais. O vocabulário do WordPiece em inglês possui apenas 30.522 palavras [73].

Uma das inovações do framework BERT é que seu pré-treinamento é realizado com base em duas tarefas distintas: Modelo de Linguagem Mascarada (MLM, sigla em inglês para *Masked Language Model*) e Predição da Próxima Sentença (NSP, sigla em inglês para *Next Sentence Prediction*). No pré-treinamento dos modelos, MLM e NSP são treinados conjuntamente com o objetivo de minimizar a função de perda combinada das duas tarefas [7, 74].

2.3.4.1 Masked Language Model

No MLM, BERT deve predizer 15% dos tokens do texto de entrada, os quais são escolhidos aleatoriamente. Nessa amostra de tokens que devem ser preditos, 80% são substituídos pelo token [MASK], 10% com uma palavra aleatória e 10% utilizam a palavra original (a divisão parece ter sido empiricamente estabelecida pelos autores, sem maiores explicações no artigo a respeito das porcentagens). O objetivo da tarefa é predizer o vocabulário original da sentença. BERT resolve a tarefa MLM emulando a capacidade cognitiva humana. Quando falta uma palavra em uma sentença, os seres humanos tentam preencher essa lacuna com algum termo que faça sentido com as palavras anteriores e posteriores ao termo ausente. Suponha-se, por exemplo, que em uma comunicação digital, um ser humano ouça a frase a seguir, cujo espaço representado por [_____] significa uma falha de sinal que não permitiu escutar a palavra:

A mulher entrou na loja e comprou um [___] de sapatos.

O cérebro humano analisa tanto as palavras anteriores como as posteriores e conclui que o termo mais lógico seria “par”. Caso apenas os termos antecedentes fossem analisados (“A mulher entrou na loja e comprou um”), várias possibilidades preencheriam adequadamente o espaço vazio: brinquedo, carro, computador, armário etc. Caso houvesse análise apenas dos termos posteriores (“de sapatos”), haveria outras possibilidades razoáveis: caixa, andar (verbo), par, loja etc. Contudo, quando são analisadas as palavras anteriores e posteriores simultaneamente, percebe-se que o termo mais provável para o preenchimento lógico da sentença seria “par”. O cérebro humano faz esse processamento instintivamente.

BERT emula essa capacidade, analisando os *tokens* de entrada anteriores e posteriores, daí a expressão bidirecional do acrônimo. No entanto, como os *tokens* são processados simultaneamente - e não em um sentido e no outro -, seria mais preciso afirmar que BERT é não direcional [75, 74]. A Figura 2.11 exemplifica o pré-treinamento pela tarefa MLM com a frase em inglês “*How are you doing today*” com a máscara no termo “*you*”:

Convém observar que o modelo BERT padrão (Base Uncased) confere 98,53% de probabilidade de o termo mascarado ser “*you*”. Contudo, quando acrescenta-se um ponto de interrogação ao final da frase (“How are [MASK] doing today?”), a probabilidade sobe para 99,36%. BERT “entende” o contexto da frase e o ponto de interrogação acrescenta informações importantes que fortalecem a predição inicial.

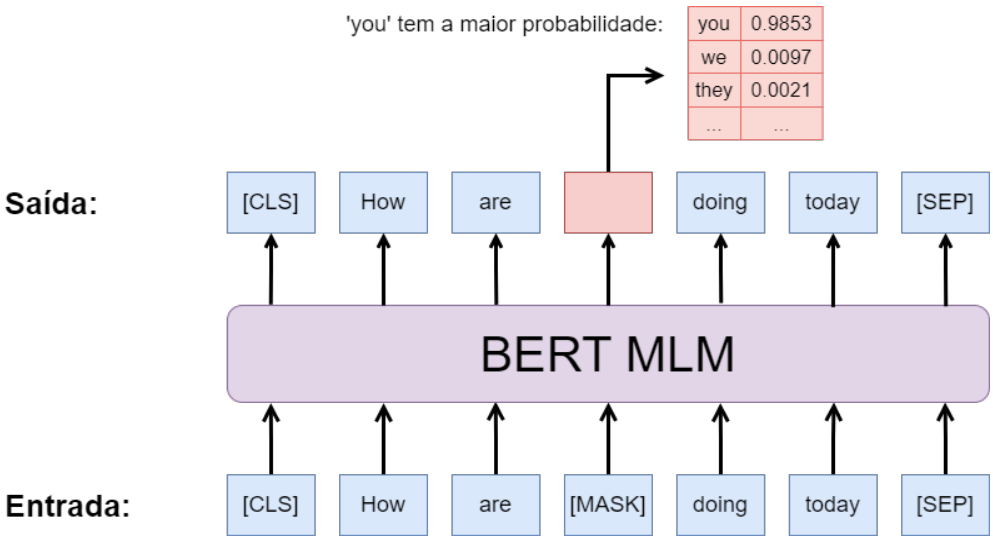


Figura 2.11: BERT Masked Language Model. Adaptado de [7]

2.3.4.2 Next Sentence Prediction

No treinamento de Predição de Próxima Sentença (NSP), a entrada é composta por um par de sentenças e a tarefa consiste em determinar se a segunda sentença é uma continuação da primeira. Nesse treinamento, em 50% dos casos, a segunda sentença no par é a sucedânea da primeira e, nos outros 50%, uma sentença aleatória é alocada na segunda posição do par.

Na tarefa de NSP, o modelo recebe duas sequências e deve prever se a segunda sequência é uma continuação da primeira. O objetivo é ajudar o modelo a entender a relação entre duas frases e como elas se conectam. Dessa forma, BERT consegue ter uma compreensão melhor do contexto das sentenças.

Os tópicos mencionados nesse capítulo representam os conceitos teóricos essenciais empregados no *framework* apresentado neste trabalho. Os domínios do conhecimento de Inteligência de Ameaças Cibernéticas (CTI), Processamento de Linguagem Natural (NLP) e Aprendizado de Máquina (AM) e suas inter-relações constituem os pilares da fundamentação teórica necessária para a compreensão desta pesquisa. CTI representa a área temática diretamente beneficiada pelo trabalho, visto tratar-se de *framework* voltado para auxiliar o trabalho de profissionais desse setor. NLP representa a área conceitual mais diretamente relacionada à essa pesquisa, dada a utilização de um LLM. A AM fornece, por meio das redes neurais, o arcabouço ferramental utilizado para a implementação do *framework*.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma síntese da revisão bibliográfica realizada na literatura científica da área. Foram enfatizados os trabalhos que buscam lidar com o mesmo problema de pesquisa. Há uma breve descrição das abordagens utilizadas e da limitações observadas. Ao final, há uma tabela comparativa que sintetiza as diferenças entre as pesquisas mencionadas e o *framework* implementado neste trabalho.

Técnicas de Processamento de Linguagem Natural e de Recuperação de Informações têm sido amplamente empregadas em campos como pesquisa médica, finanças, comércio e recursos humanos [76]. No domínio da segurança cibernética, grande parte da pesquisa relacionada à NLP concentra-se na extração de IOCs de textos não estruturados [20, 77], mineração de dados relativos a ataques em mídias sociais [78, 79] ou coleta de outros dados relacionados de segurança cibernética [19, 23, 80]. TTPs fornecem aos profissionais de segurança um necessário contexto do comportamento do ataque que não pode ser extraído apenas de IOCs e outros dados. Contudo, TTPs são normalmente encontradas em textos não estruturados, dificultando sua pesquisa.

Uma das maiores dificuldades no uso de NLP no domínio cibernético é a pouca disponibilidade de bases de dados consistentes e anotadas [81]. Conjuntos de dados relacionados a TTPs são ainda mais raros e essa escassez dificulta o avanço de pesquisas em classificação de TTPs [13, 82]. A despeito da importância, ainda há pouca pesquisa voltada ao problema da extração de TTPs de textos não estruturados [24, 83].

Niakanlahiji *et al.* [80] propuseram o SECCMiner para extrair informações de relatórios de APTs úteis aos profissionais de segurança. O SECCMiner trabalha em nível de documentos extraindo o texto de arquivos no formato PDF. Em seguida, utiliza expressões regulares para separar as sentenças e emprega a técnica de *part-of-speech tagging* para rotular os termos. Busca frases nominais e utiliza o método estatístico TF-IDF (*Term Frequency-Inverse Document Frequency*) para dar pesos aos termos. Compara-se então com um grupo de frases nominais pré-estabelecido com base na matriz ATT&CK e na experiência dos autores. A pesquisa busca relacionar essas frases nominais com determinados APTs. Essa pesquisa tem a desvantagem de não utilizar métodos de aprendizado de máquina, mais adaptáveis, mas apenas métodos de recuperação de informação e estatísticos. Além disso, os autores não se propõem a classificar segundo um framework de TTPs consolidado, somente utilizando o ATT&CK como base para uma pequena ontologia própria.

Ghazi *et al.* [23] apresentam uma abordagem combinando algumas técnicas clássicas de NLP e machine learning com o objetivo de extrair informações sobre o comportamento dos atacantes de textos de CTI e classificá-las segundo o padrão STIX (*Structured Threat Information Expression*). Os autores realizaram uma anotação manual de textos retirados de blogs de segurança cibernética e relatórios de CTI e a utilizam para treinamento de aprendizado de máquina. A proposta era treinar um classificador NER (*Named Entity Recognition*) com labels anotados do domínio específico de segurança cibernética. Utilizam o CRF Classifier da biblioteca Stanford NLP para o treinamento e obtém precisão de 58% e recall de 54%. A limitação desse trabalho deve-se ao fato de que a técnica clássica NER, embora útil para extrair algumas informações de textos, não permite, sozinha, que o sistema aprenda o suficiente dos documentos para rea-

lizar uma classificação efetiva. Atualmente, os grandes modelos de linguagem (LLM) realizam essa tarefa de modo mais efetivo.

Legoy *et al.* [20] experimentaram com alguns métodos de representação de texto combinados com diferentes classificadores. Sua pesquisa define como melhor combinação, dentre os métodos pesquisados, a representação de texto por meio da combinação de *bag of words* com o método TF-IDF em conjunto com o classificador Linear Support Vector (LinearSVC). Adicionando pós-processamento à essa combinação, os autores propuseram o rcATT (*Report Classification by ATT&CK Tactics and Techniques*), uma ferramenta para extrair TTPs de relatórios de inteligência de ameaças cibernéticas. Entre as principais limitações desse trabalho, encontra-se o fato de que não foram empregados os *Large Language Models* (LLMs), paradigma mais atual de aprendizado de máquina para a solução de problemas de classificação envolvendo NLP. Ademais, o trabalho foi feito apenas para táticas e técnicas, não incluindo a posterior atualização do *framework* ATT&CK que adicionou o conceito de subtécnicas.

Husari *et al.* [21] propuseram o TTPDrill para identificação e classificação de TTPs em textos não estruturados. Essa abordagem utiliza o método de representação TF-IDF em conjunto com uma versão modificada do algoritmo de recuperação de informações BM25. O TTPDrill realiza um *scrapping* no site da empresa de segurança Symantec e assinala textos relevantes para CTI. Em seguida, busca identificar ações relacionadas a ameaças cibernéticas e faz um mapeamento dessas ações para a matriz ATT&CK e o padrão STiX Attack. O TTPDrill trabalha no nível de sentenças (e não com textos inteiros ou documentos), recebe fragmentos de textos relacionados a CTI do site da Symantec e procura por combinações do tipo Sujeito-Verbo-Objeto (SVO). Para cada combinação SVO encontrada, busca correspondência dentro de uma ontologia predefinida (pelos autores) de ações conhecidas, denominadas "sentenças base".

Os autores do TTPDRill afirmam ter alcançado resultados de precisão de 84% e *recall* de 82%. Trabalhos posteriores, no entando, constestaram esse números, alegando que a abordagem proposta obtém resultados consideravelmente inferiores [13, 84]. A ontologia pré-definida de sentenças base constitui uma das desvantagens dessa abordagem, pois limita o escopo do que poderia ser considerado ações maliciosas cibernéticas. Além disso, o modelo é baseado em regras fixas e emprega técnicas antigas de recuperação de informação, não utilizando conceitos mais recentes e consolidados em NLP, como o emprego de aprendizado de máquina.

Os mesmos autores posteriormente propuseram o ActionMiner [85], modelo que busca por pares Verbo-Objeto que representem ações de ameaça. Essa modelagem faz uso dos conceitos de entropia e informação mútua da Teoria da Informação combinados com algumas técnicas fundamentais de NLP (*dependency parsing, part-of-speech tagging*). O Action Miner, como o nome indica, busca ações maliciosas em textos de segurança cibernética. Para tanto utiliza as técnicas e conceitos mencionados para selecionar pares VO (Verbo-Objeto). Entre as desvantagens dessa abordagem estão os fatos de não tirar proveito de técnicas de aprendizado de máquina ou grandes modelos linguísticos e de não classificar segundo um padrão de TTPs aceito na comunidade de segurança cibernética.

Uma abordagem diferente para o problema de extração de TTPs de textos não estruturados é proposta por Satvat *et al.* [86]. Os autores apresentam a ferramenta EXTRACTOR, a qual entrega duas saídas principais: sumariza o comportamento do atacante em um conjunto de frases curtas e apresenta esse comportamento na forma de grafos de proveniência. Utilizam BERT para classificar frases extraídas de textos

de segurança cibernética em proveitosas ou não. Em seguida, utilizam uma rede neural do tipo BiLSTM (*Bidirectional Long Short-Term Memory*) para remover verbosidade e produzir sumários concisos. Os textos mais enxutos facilitam a identificação de sujeitos, verbos e objetos, permitindo a aplicação da técnica SRL (*Semantic Role Labeling*, uma evolução da técnica *dependency parsing*) que identifica relações entre os termos. Essas relações são então utilizadas para montar a representação visual na forma de grafos. Essa abordagem possui a desvantagem de não classificar segundo um framework consolidado, como o MITRE ATT&CK, que facilita a compreensão pelo analista de segurança. Também subutiliza o potencial de BERT, empregando-o em uma tarefa bastante simples.

A pesquisa de Ayoade *et al.* [84], por sua vez, trabalhou no nível de documento (processa todo o documento e não sentença por sentença) e empregou métodos de correção de viés, de propagação de confiança e de estimativa de importância de pesos para fazer predições de táticas e técnicas presentes em relatórios de CTI. Os métodos KMM, KLIEP e arulSIF foram aplicados para estimar a importância dos pesos, passando esses dados para um classificador SVM (*Support Vector Machine*). Diferentemente da pesquisa de Ayoade *et al.*, a abordagem deste trabalho de pesquisa lida com redes neurais de grandes modelos linguísticos (LLM) pré-treinados para melhor entendimento de linguagem natural e menos suscetíveis a variações nos dados de entrada.

You *et al.* [13] propuseram o framework TIM (*Threat Intelligence Mining*), desenvolvendo a ferramenta TCENet (*Threat Context Enhanced Network*. Essa ferramenta efetua a raspagem de dados em websites de cinco empresas distintas para angariar relatórios de ameaças. A solução trabalha no nível de sentenças e faz sua análise agrupando conjuntos de três sentenças de modo a buscar mais contexto. Também busca combinar IOCs (IPs, nomes de domínio, hashes de arquivo, chaves de registro etc.) mostrados nos relatórios com as técnicas da matriz do MITRE, de modo a enriquecer o conhecimento das TTPs com mais dados contextuais. Utiliza SentenceBERT para realizara representação (*embeddings*) das sentenças e uma BiLSTM para classificação, além de uma rede neural convolucional para extração de features para enriquecimento dos dados. Esse estudo apresenta uma limitação importante ao restringir o escopo da pesquisa às cinco técnicas mais populares e uma tática da matriz ATT&CK. Além disso, não explora todo o potencial dos grandes modelos de linguagem pré-treinados ao utilizar apenas para fins de representação das sentenças.

Outro trabalho relevante nesse campo é o TRAM (*Threat Report ATT&CK Mapper*), feito pela instituição MITRE [87]. Essa ferramenta aplica Regressão Logística na tarefa de predição de técnicas relacionadas a cada sentença, utilizando as sentenças exemplos da base do MITRE para treinamento do algoritmo. Uma desvantagem desse trabalho consiste na necessidade de revisão manual de cada classificação proposta. Além disso, o método de Regressão Logística é pouco eficiente se comparado aos classificadores dos LLM.

O presente trabalho propõe-se a abordar algumas das lacunas de pesquisa supramencionadas. O diferencial da abordagem implementada baseia-se em um conjunto de aspectos, a saber: mapeamento para framework de TTPs consolidado (ATT&CK, no caso desta pesquisa), emprego de aprendizado de máquina, classificação para número significativo de técnicas (na casa de centenas, no mínimo) e uso de grandes modelos de linguagem (LLM) na tarefa de classificação. A Tabela 3.1 descreve os trabalhos correlatos segundo esses aspectos:

Tabela 3.1: Comparativo entre pesquisas correlatas

Trabalho	**Framework consolidado**	**Aprendizado de máquina**	**Abrangência de técnicas**	**Uso de LLM**
Niakanlahiji *et al.* (2018) [80]	Não	Não	Não	Não
Ghazi *et al.* (2018) [23]	Sim	Sim	Sim	Não
Legoy *et al.* (2019) [20]	Sim	Sim	Sim	Não
Husari *et al.* (2017) [21]	Sim	Não	Não	Não
Husari *et al.* (2018) [85]	Não	Não	Não	Não
Satvat *et al.* (2021) [86]	Não	Sim	Sim	Sim
Ayoade *et al.* (2018) [84]	Sim	Sim	Sim	Não
You *et al.* (2022) [13]	Sim	Sim	Não	Sim
TRAM (2019) [87]	Sim	Sim	Sim	Não
Presente trabalho [30, 31]	**Sim**	**Sim**	**Sim**	**Sim**

Conforme explicitado nos textos descritivos anteriores relacionados aos trabalhos correlatos e sintetizados na Tabela 3.1, o presente trabalho é o único a reunir todos os elementos acima. Convém ainda observar que mesmo os trabalhos que fazem algum uso de LLM [13, 86], não exploram sua potencialidade completa, utilizando-os apenas para fins de representação (*embedding*) das sentenças e não para tarefas de classificação. Dessa forma, até onde investivado, a pesquisa aqui apresentada consiste no único trabalho a aplicar um LLM (BERT) ao problema de classificação de TTPs a partir de texto não estruturado fazendo correspondência com um *framework* consolidado.

4 *FRAMEWORK* PARA CLASSIFICAÇÃO DE TTP UTILIZANDO BERT

Esse capítulo descreve a estratégia delineada para enfrentar o problema de pesquisa definido, isto é, identificar TTPs a partir de sentenças e classificar essas frases de acordo com a matriz ATT&CK. Diferentemente dos trabalhos correlatos expostos no Capítulo 3, foi empregada aqui uma abordagem distinta e inovadora: aplicação de um grande modelo de linguagem (LLM), o BERT, diretamente à tarefa de classificação (e não apenas para fins de representação das sentenças). A Figura 4.1 apresenta uma visão geral do *framework* modelado com o objetivo de avaliar o desempenho de BERT no problema de pesquisa.

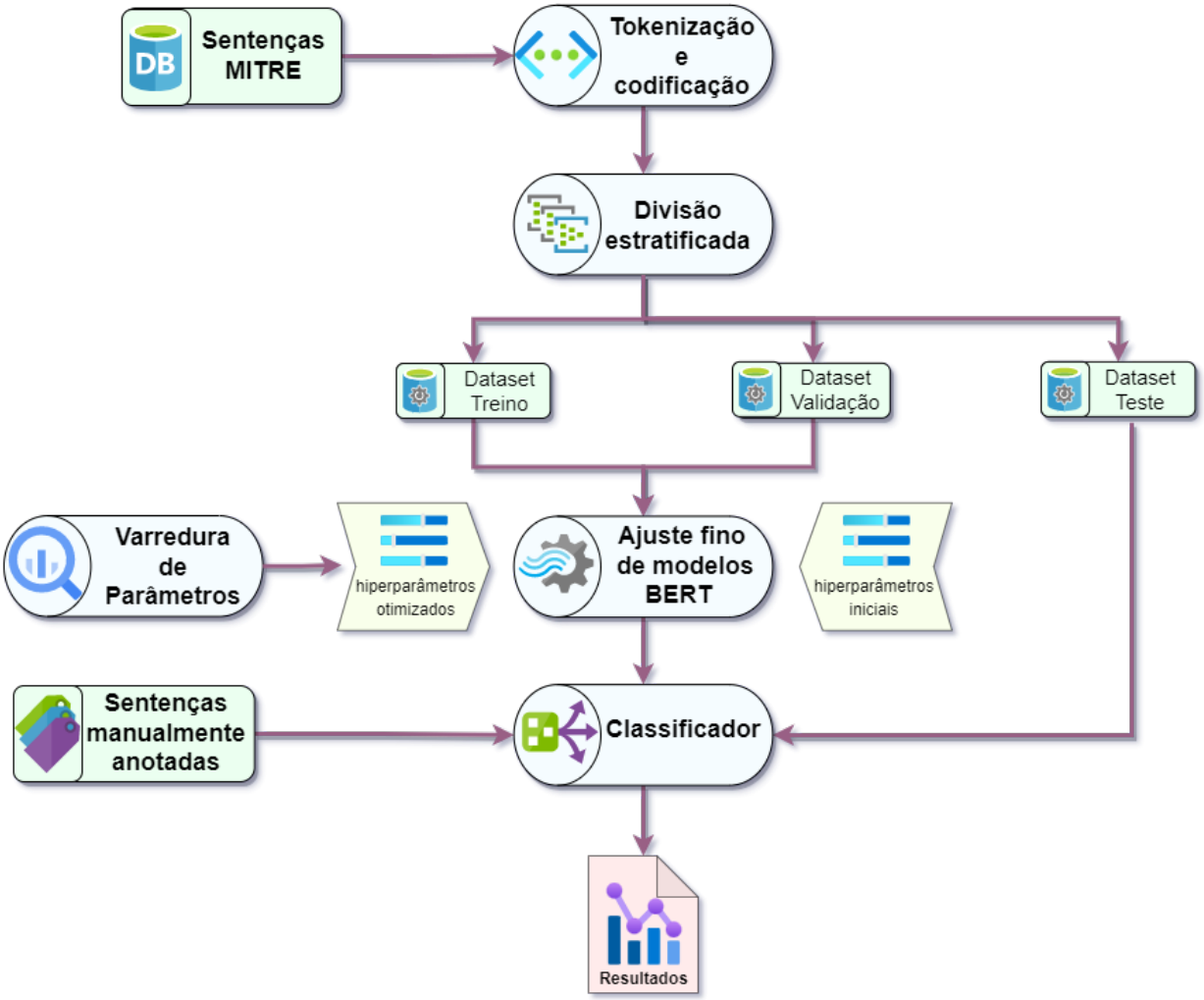


Figura 4.1: Visão geral do *framework* para classificação de TTPs utilizando BERT.

A primeira etapa consiste na preparação dos dados, realizando a separação em tokens e codificação das sentenças segundo o formato BERT. A seguir, aplica-se um procedimento de estratificação para dividir o conjunto de sentenças em subconjuntos de treinamento, validação e teste. A estratificação significa que todas as técnicas estarão representadas em cada subconjunto. Isso evita que uma divisão aleatória distribua

todas as sentenças exemplificativas de uma técnica para o subconjunto teste, por exemplo. Nesse caso hipotético, não haveria sentenças dessa técnica no subconjunto de treinamento e o modelo não aprenderia sobre ela.

Procede-se, então, o ajuste fino de 11 modelos BERT usando os subconjuntos de treinamento e validação, empregando hiperparâmetros iniciais retirados da literatura científica disponível. Posteriormente, realiza-se uma varredura de valores em dois hiperparâmetros escolhidos de acordo com a literatura em busca de valores que potencializem o desempenho. Por fim, utilizou-se tanto os hiperparâmetros iniciais como os otimizados para executar a tarefa de classificação no *dataset* de teste e um conjunto de sentenças manualmente anotadas e os resultados foram analisados.

As subseções a seguir apresentam com maior detalhamento os dados utilizados e descrevem as etapas supramencionadas.

4.1 DATASET MITRE

Desde o lançamento do framework ATT&CK, em 2015, o MITRE mantém uma base de conhecimento anotada manualmente de informações extraídas de relatórios de CTI [50]. Esse repositório, entre outros dados, conta com 10360 sentenças ilustrativas de TTPs. A Tabela 4.1 mostra alguns exemplos:

Tabela 4.1: Exemplo de sentenças exemplo da base do MITRE com as correspondentes técnicas ou subtécnicas.

Sentença	**ID da técnica**	**Nome da técnica**
The NETWIRE payload has been injected into benign Microsoft executables via process hollowing.	T1055.012	Process Injection: Process Hollowing
Sykipot contains keylogging functionality to steal passwords.	T1056.001	Input Capture: Keylogging
Mosquito deletes files using DeleteFileW API call.	T1070.004	Indicator Removal on Host: File Deletion
RTM has initiated connections to external domains using HTTPS.	T1071.001	Application Layer Protocol: Web Protocols
Dragonfly has compromised user credentials and used valid accounts for operations.	T1078	Valid Accounts
Patchwork payloads download additional files from the C2 server.	T1105	Ingress Tool Transfer
PlugX has a module to create, delete, or modify Registry keys.	T1112	Modify Registry
Sandworm Team's CredRaptor tool can collect saved passwords from various internet browsers.	T1555.003	Credentials from Password Stores: Credentials from Web Browsers

Continua na próxima página

Tabela 4.1 – *Continuação da página anterior*

Sentença	**ID da técnica**	**Nome da técnica**
TA551 has sent spearphishing attachments with password protected ZIP files.	T1566.001	Phishing: Spearphishing Attachment
APT33 has sent spearphishing emails containing links to .hta files.	T1566.002	Phishing: Spearphishing Link

Entre as 576 técnicas e subtécnicas, 466 possuem pelo menos uma sentença ilustrativa. Isso decorre do fato de tratar-se de um repositório alimentado manualmente pelo MITRE [18, 50]. A instituição retira seus exemplos das divulgações de campanhas cibernéticas em diversos meios. Assim, algumas técnicas mais comumente empregadas possuem centenas de exemplos ao passo que algumas, de utilização mais rara, ainda não tiveram sentenças exemplificativas inseridas pelo MITRE em sua base de dados. Uma característica relevante desse repositório de exemplos é que todos são elaborados a partir de casos reais e associam uma campanha cibernética a uma técnica ou subtécnica. As frases seguem um padrão morfológico SVO (Sujeito-Verbo-Objeto) [18, 50].

A Tabela 4.2 mostra as técnicas ou subtécnicas com mais exemplos na base do MITRE:

Tabela 4.2: Técnicas ou subtécnicas mais comuns no repositório de sentenças do MITRE.

Técnica:subtécnica	**ID**	**Nº de exemplos**
Ingress Tool Transfer	T1105	371
System Information Discovery	T1082	311
Obfuscated Files or Information	T1027	303
Application Layer Protocol: Web Protocols	T1071.001	283
Command and Scripting Interpreter: Windows Command Shell	T1059.003	277
File and Directory Discovery	T1083	259
Process Discovery	T1057	225
Indicator Removal on Host: File Deletion	T1070.004	216
Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.001	209
System Network Configuration Discovery	T1016	204

A técnica com maior número de exemplos é *Ingress Tool Transfer* (T1105), que não possui subtécnicas. A descrição dessa técnica no *framework* ATT&CK diz que "adversários podem transferir ferramentas ou outros arquivos de um sistema externo para o ambiente comprometido" [88]. Com o aumento do nível de complexidade dos ataques, tornaram-se cada vez mais comuns as campanhas em múltiplos estágios, que inserem no ambiente alvo um arquivo simples como vetor de infecção inicial (*dropper*), que busca outros módulos. Essa busca é retratada justamente pela técnica T1105.

A técnica *Obfuscated Files or Information* (T1027) possui seis subtécnicas. Contudo, a maior parte das sentenças exemplo está registrada apenas com a técnica. Em muitas análises de campanhas cibernéticas,

a obfucação é mencionada sem muitos detalhes, nem sempre permitindo especificar em qual subtécnicas seria possível encaixar a ação descrita na análise.

Já a técnica *Indicator Removal on Host* (T1070) é descrita da seguinte forma: "adversários podem apagar ou modificar artefatos gerados no sistema hospedeiro para remover evidências de sua presença ou prejudicar a defesa" [88]. Em outras palavras, trata-se de apagar rastros. Nesse caso, há uma predominância de uma subtécnica (*Indicator Removal on Host: File Deletion* - T1070.004) que apresenta 216 exemplos enquanto a segunda subtécnica mais comum possui apenas 40 sentenças. Isso indica que, nas referências do MITRE, o método mais comumente utilizado de apagar rastros é por meio da remoção de arquivos.

Outras subtécnicas, contudo, não apresentam exemplos na base do MITRE. *Boot or Logon Initialization Scripts: Login Hook* (T1037.002) constitui uma subtécnica de persistência muito específica, que é conhecida, mas não possui sentenças ilustrativas. Isso ocorre porque, como o MITRE elabora as sentenças a partir de casos reais, para que uma técnica ou subtécnica tenha exemplos, é necessária uma referência associando-a a uma campanha cibernética.

Há também algumas técnicas e subtécnicas que, por serem muito específicas, possuem um ou alguns escassos exemplos. Em razão da necessidade de dados para treinamento dos modelos de aprendizagem de máquina, o escopo deste trabalho foi limitado às técnicas e subtécnicas que apresentam pelo menos 5 exemplos. Sob esse critério, foram aproveitadas 9909 sentenças exemplo (95.6% do total da base) e trabalhou-se com as 253 técnicas mais comuns, isto é, aquelas que possuem ao menos 5 sentenças exemplificativas.

No repositório do MITRE, cada sentença é categorizada em uma única técnica ou subtécnica. Dada essa especificidade, o problema de classificação de TTPs foi modelado neste trabalho usando a abordagem multiclasse. Em um problema do tipo multiclasse cada amostra recebe um único rótulo. Assim, neste caso concreto, cada uma das 253 técnicas ou subtécnicas constituirá uma classe.

4.2 PREPARAÇÃO DOS DADOS

Uma peculiaridade desse conjunto de dados é o significativo desbalanceamento entre as classes, problema comum em bases de dados textuais [89]. Na base ora utilizada, a maior classe apresenta 371 exemplos enquanto as menores, pelas restrições experimentais impostas, possuem cinco sentenças. Contudo, pesquisas mostram que BERT lida bem com bases desbalanceadas e estratégias de extensão de dados (*data augmentation*) não impactam significativamente a performance [81, 90, 91, 92].

Para utilizar o modelo BERT, as sentenças precisam ser individualmente "tokenizadas" (separadas em *tokens* que representam palavras ou parte de palavras, além dos *tokens* de controle) e codificadas (os *tokens* devem ter representações numéricas, pois o modelo, na verdade, enxerga conjuntos de matrizes numéricas representando o texto). À guisa de exemplo, considere-se a sentença "*BlackTech has used DLL side loading by giving DLLs hardcoded names and placing them in searched directories*". Trata-se de frase ilustrativa, na base do MITRE, da subtécnica *DLL Side-Loading* da técnica *Hijack Execution Flow* [88]. A Figura 4.2 mostra como fica a sentença exemplo anterior ao ser "tokenizada" de acordo com o algoritmo WordPiece em preparação para ingestão por modelo BERT:

[CLS] Black ##T ##ech has used DL ##L side loading by giving DL ##L ##s hard ##code ##d names and placing them in searched director ##ies . [SEP] [PAD]

Figura 4.2: Exemplo de sentença “tokenizada”.

O exemplo acima foi truncado nos primeiros 100 *tokens* para fins ilustrativos, visto que todos os *tokens* restantes são do tipo [PAD]. Observe-se que, no exemplo, há três *tokens* especiais: [CLS], [SEP] e [PAD]. O [CLS] consiste em um *token* de classificação e sua representação final (última camada oculta da rede neural de BERT) contém informação agregada de todos os *tokens*. No caso da tarefa de pré-treinamento de Predição de Próxima Sentença (NSP), por exemplo, retorna a probabilidade de o par de sentenças serem contínuas ou não (*isNext* ou *notNext*). o *token* [CLS] está presente no início de cada sentença ou conjunto de sentenças.

O *token* especial [SEP] representa um separador e está presente ao final de cada sentença isolada ou entre sentenças nas tarefas que envolvam mais de uma delas. Já o *token* [PAD] consiste apenas em um preenchimento nulo até o limite estabelecido para o número de *tokens*, pois BERT necessita receber sentenças de mesmo comprimento. Na etapa de codificação, foi utilizada a biblioteca PyTorch para criar tensores que alimentaram a rede neural. Dessa forma, é atribuído o código 101 para o *token* [CLS], 102 para o [SEP] e 0 (zero) para o [PAD] e cada outro *token* distinto recebe uma numeração, conforme formato especificado no modelo BERT. Dessa forma, no exemplo anterior, a codificação produz o tensor PyTorch mostrado na Figura 4.3:

[101, 2117, 1942, 11252, 1144, 1215, 26624, 2162, 1334, 10745, 1118, 2368, 26624, 2162, 1116, 1662, 13775, 1181, 2666, 1105, 6544, 1172, 1107, 8703, 1900, 1905, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Figura 4.3: Tensor PyTorch da sentença exemplo.

O conjunto de dados foi dividido em datasets de treinamento, validação e teste na proporção 60:20:20, obtendo 5945, 1982 e 1982 sentenças nos respectivos datasets. Foi adotada uma estratégia de amostragem estratificada para garantir que cada dataset possuísse exemplos de todas as técnicas. A estratificação implicou que mesmo as menores classes (cinco sentenças, por limitação definida para o experimento) estariam representadas nos três datasets (na proporção 3:1:1).

Para fins de validação, após realizar treinamento dos modelos BERT nas sentenças exemplo do repositório MITRE, os modelos foram utilizados para realizar predições em um conjunto de 80 sentenças extraídas e anotadas manualmente de relatórios de CTI públicos (dataset de inferência). Essas sentenças

foram retiradas de 18 documentos públicos, originários de 16 distintas instituições, de modo a propiciar variedade de linguagem e estilos de escrita.

4.3 MODELOS E CONFIGURAÇÕES

Inicialmente foi estabelecida uma linha de base por meio da utilização de um modelo simplificado que combina a representação TF-IDF (*Term Frequency-Inverse Document Frequency*) com um classificador por Regressão Logística. O TF-IDF consiste na aplicação de medidas estatísticas ao modelo elementar de representação de sentenças em NLP, o *bag of words* ("sacola de palavras"). O *bag of words* consiste em uma estrutura simples que guarda cada uma das palavras únicas de um texto e o número de ocorrências.

No TF-IDF, a primeira medida estatística (TF, sigla em inglês para *Term Frequency*) consiste na "frequência dos termos". Significa que cada termo recebe um peso de acordo com sua frequência no texto. Essa medida consiste na divisão do número de ocorrências de uma palavra pelo total de palavras. Contudo, algumas palavras são tão comuns que aparecerão frequentemente em praticamente todos os documentos. Suponha-se uma análise de 100 artigos em português sobre ataques cibernéticos. A preposição "de" e os artigos "a" e "o", por exemplo, provavelmente aparecerão com alta frequência em todos os artigos. Já a palavra "Stuxnet", aparecerá em apenas alguns artigos.

A segunda medida (IDF, sigla em inglês para *Inverse Document Frequency*) trabalha com o inverso da frequência nos documentos. Isso significa que terão maiores pesos os termos que aparecerem em menos arquivos. Essa medida privilegia palavras únicas de determinados documentos, isto é, que não aparecem regularmente em todos os textos, e elimina palavras comuns que pouco acrescentam ao significado. Dessa forma, no conjunto hipotético de 100 documentos sobre ataques cibernéticos em português, a preposição "de" e os artigos "a" e "o" seriam descartados e o termo "Stuxnet" receberia peso maior. O TF-IDF combina essas duas medidas (multiplicando-as) para gerar pesos para os termos. Foi muito utilizado em tarefas de classificação, extração de palavras-chave, recuperação de informações e sumarização [93, 94].

A regressão logística consiste em um método básico de modelagem probabilística que busca predizer a relação entre variáveis quando a variável dependente é categórica, isto é, dada por valores discretos pré-definidos. Trata-se de modelo largamente utilizado em diversos campos como economia, finanças, ciências sociais e ciências de dados [95]. Foram utilizadas, como *baseline*, a representação das sentenças por meio do TF-IDF e a predição de classes por meio de Regressão Linear, tendo em conta que esses dois modelos são considerados basilares nessas duas tarefas.

Posteriormente, foram processados os dados utilizando onze versões do *Bidirectional Encoder Representations from Transformers*. Os modelos BERT já contemplam tanto a representação das sentenças quanto tarefas preditivas [7]. As diferentes versões escolhidas foram: BERT Base Cased, BERT Base Uncased, BERT Large Cased, BERT Large Uncased, RoBERTa Base, RoBERTa Large, DistilRoBERTa, DistilBERT Uncased, DistilBERT Cased, SecBERT e SecRoBERTa.

BERT Base Cased e BERT Base Uncased são os dois modelos básicos disponibilizados. São diferenciados entre si pelo fato de o primeiro levar em conta caracteres maiúsculos (*cased*) e o segundo normalizar todos os caracteres em letras minúsculas (*uncased*). Ambos possuem 12 camadas de *encoders*, 768 subca-

madas ocultas, 12 cabeças de atenção (*attention heads*) e 110 milhões de parâmetros.

BERT Large Cased e BERT Large Uncased são modelos ampliados do BERT. Contam com 24 camadas de *encoders*, 1024 subcamadas ocultas, 16 cabeças de atenção e 340 milhões de parâmetros. Os modelos Base levam quatro dias para realizar o pré-treinamento em quatro TPUs (*Tensor Processor Units*). Os modelos Large, para realizarem o mesmo pré-treinamento em quatro dias, necessitam de 16 TPUs [70].

Os modelos RoBERTa (Base e Large) são uma variação do BERT cuja principal distinção consiste na utilização de uma abordagem de mascaramento dinâmico na tarefa de pré-treinamento *Masked Language Modeling*. RoBERTa Base utiliza a mesma arquitetura das versões BERT Base, possuindo 12 camadas de *encoders*, 768 subcamadas ocultas, 12 cabeças de atenção e 125 milhões de parâmetros. A diferença de parâmetros pode ser explicada por algumas abordagens diferentes na tokenização e pré-treinamento.

Os modelos DistilBERT (Case e Uncased) são versões mais enxutas de BERT. Embora contem com 40% menos parâmetros que os modelos Base, rodam cerca de 60% mais rápido que os modelos Base e, não obstante, atingem cerca de 95% de sua performance. As versões distilBERT possuem 6 camadas de *encoders*, 768 subcamadas ocultas, 12 cabeças de atenção e cerca de 65 milhões de parâmetros.

Os dois últimos modelos, SecBERT e SecRoBERTa, são variantes de BERT Base e RoBERTa Base, respectivamente, cujo treinamento foi realizado em textos de segurança cibernética e apresentam vocabulário próprio. As seguintes fontes foram utilizadas no treinamento desses modelos:

- *APTnotes*
- *Stucco-Data: Cyber security data sources*
- *CASIE: Extracting Cybersecurity Event Information from Text*
- *SemEval-2018 Task 8: Semantic Extraction from CybersecUrity REports using Natural Language Processing (SecureNLP).*

Para estabelecer a parametrização inicial, recorreu-se a recomendações de pesquisas anteriores [7, 96] e experimentação. Devlin *et al.* [7] sugerem algumas especificações que devem funcionar com boa performance quando aplicadas a diferentes tarefas: tamanho de lotes de 16 ou 32; taxa de aprendizagem de 5e-5, 3e-5 ou 2e-5; treinamento (ajuste fino) por até quatro épocas. Em artigo específico sobre ajuste fino, Sun *et al.* [96] propuseram que tamanhos de lote de 24, taxa de aprendizado de 2e-5, comprimento máximo de sentença de 128 e 4 épocas de treinamento constituem hiperparâmetros razoáveis para obter performance adequada para a maioria das tarefas. Jeawak *et al.* [97] realizaram ajuste fino para classficação em domínios específicos utilizando quatro épocas, taxa de aprendizado de 2e-5, tamanho de lote de 16 e comprimento máximo de 256.

Analisando as sentenças da base do MITRE, percebe-se que, após os processos de tokenização e codificação, a maior sentença apresenta 136 tokens não nulos (antes do preenchimento com o token nulo [PAD]). Dessa forma, o parâmetro *max_length* foi definido como 256, de modo a ter margem confortável para sentenças ainda mais longas nos relatórios de CTI. Cada uma das frases examinadas foi completada com caracteres nulos (*padding*) de acordo com esse limite. Considerando o *max_length* generoso e o alto número de classes (253 *labels*), optou-se inicialmente por um tamanho de lote conservador de 16, para

garantir que os passos de treinamento caberiam na memória GPU disponível. O grande número de classes também implica uma convergência mais lenta, de modo que estabeleceu-se empiricamente o valor de 30 épocas de treinamento para verificar se a função de perda mostrava convergência. Por fim, foi utilizada a taxa de aprendizagem de 2e-5 sugerida em todos os estudos supramencionados.

Também foi conduzida uma análise de parametrização buscando potenciais melhorias de desempenho e examinando o efeito das alterações de hiperparâmetros na performance do modelo. Devido ao alto custo computacional da tarefa, foram selecionados os hiperparâmetros de taxa de aprendizado (utilizando as configurações 1e-4, 5e-5, 2e-5 e 1e-5) e tamanho do lote (com as configurações 8, 16, 24 e 32) e aplicaram-se todas as combinações entre esses hiperparâmetros por 10 épocas para cada par de valores. A taxa de aprendizado é, possivelmente, o hiperparâmetro mais importante para realizar ajuste fino [98, 99, 100]. O efeito do tamanho de lote na acurácia ainda não parece ser completamente compreendido, com diferentes estudos chegando a conclusões distintas [101, 102, 103]. O modelo escolhido para a utilização na varredura de parâmetros foi BERT Base Uncased, de tamanho mediano entre as diferentes versões. Após esse procedimento, o melhor ajuste identificado foi aplicado aos modelos BERT para examinar o efeito no desempenho.

4.4 MÉTRICAS UTILIZADAS

Optou-se por utilizar a métrica da acurácia para avaliação dos modelos. Em problemas do tipo multi-classe, as médias micro de precisão, recall e F-measure igualam-se à acurácia. As médias macro, por sua vez, são bastante afetadas pelo desbalanceamento de classes. Como na base MITRE não há uma dominância de classes (a maior classe representa apenas 3,58% do total) e o problema de classificação investigado nessa pesquisa não apresenta preferência ou precedência entre as classes, a métrica da acurácia proporciona boa compreensão do desempenho global. A acurácia é definida conforme a fórmula 4.1 :

$$Acurácia = \frac{VP + VN}{VP + FN + VN + FP} \tag{4.1}$$

na qual VP significa Verdadeiro Positivo; VN, Verdadeiro Negativo; FN, Falso Negativo; e FP, Falso Positivo.

Na varredura de hiperparâmetros, foi utilizado ainda o coeficiente de correlação de Pearson (correlação linear) para verificar a relação entre a acurácia e os hiperparâmetros investigados. O coeficiente r de correlação de Pearson é dado pela Fórmula 4.2:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.2}$$

onde x e y são amostras das duas variáveis cuja relação se averigua, $\bar{x}$ e $\bar{y}$ são os valores médios das variáveis e n é o número de amostras.

As métricas acima fecham a descrição da metodologia e permitem avaliar os resultados dos passos anteriores. A escolha do dataset, a preparação dos dados, os modelos escolhidos e as configurações aplicadas

foram avaliados segundo a métrica da acurácia e a correlação de Pearson permitiu observar uma relação entre os hiperparâmetros utilizados e a acurácia.

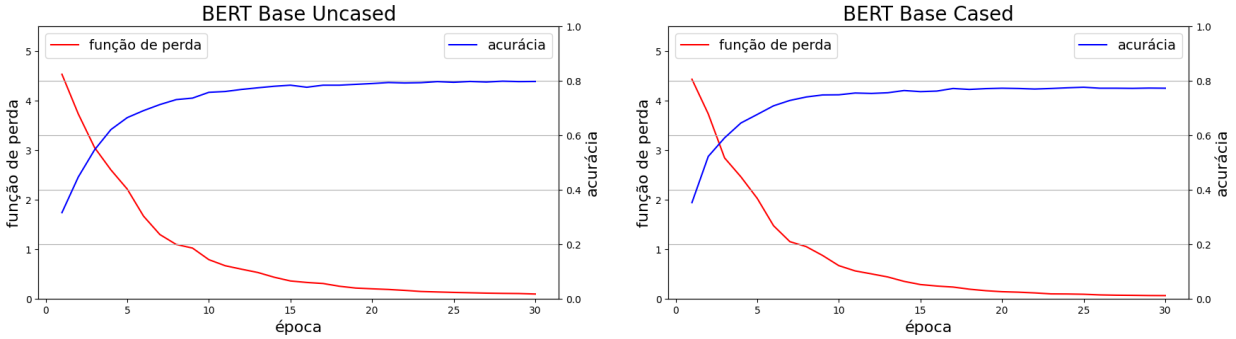
5 RESULTADOS E DISCUSSÃO

Esse capítulo descreve os resultados obtidos no experimento realizado. São mostrados os gráficos de treinamento dos 11 modelos BERT utilizados e os valores de acurácia obtidos são tabelados e comentados. Na primeira subseção Análise de Parâmetros mostra-se a varredura paramétrica realizada e os resultados de correlação entre parâmetros e acurácia. Por fim, realiza-se, na última seção, análise qualitativa dos erros de classificação.

O modelo TF-IDF/Regressão Logística, utilizado como linha de base, obteve uma acurácia de 0,6051 no dataset de teste e 0,4770 no dataset de inferência. É importante ressaltar que, embora esses valores sejam baixos, ainda assim são significativos. Convém pontuar que, para o problema em questão, há 253 classes. A chance de acerto aleatório de uma classificação com 253 classes balanceadas, por exemplo, seria de $1/253$, ou seja 0,00395 (bastante inferior, portanto aos valores obtidos). Dessa forma, o modelo escolhido como linha de base utilizando Regressão Logística para classificação mostra-se significativamente superior à classificação aleatória, tendo essa função sido utilizada para essa tarefa, por exemplo, no projeto TRAM (*Threat Report ATT&CK Mapper*), do MITRE [87].

Realizou-se então a etapa de ajuste fino dos onze modelos BERT, com os hiperparâmetros iniciais, sobre o *dataset* de treinamento. A Figura 5.1 mostra as curvas de acurácia e função de perda (*cross-entropy loss*) para os modelos treinados (ajuste fino) por 30 épocas. Observa-se que os eixos verticais das subfiguras da Figura 5.1 não apresentam unidade, pois a função de perda não possui unidade e a acurácia é medida entre 0 e 1 (ou sua percentagem correspondente). As curvas apresentam o comportamento esperado para o treinamento de aprendizagem de máquina, com a acurácia percorrendo uma curva ascendente e a função de perda decaindo. Isso significa que, para os hiperparâmetros iniciais propostos, todos os modelos convergiram, isto é, "aprenderam".

Observados os comportamentos adequados na etapa de treinamento ajuste fino com os hiperparâmetros iniciais, todos os modelos podem ser empregados na tarefa de predição de TTPs presentes nas sentenças nos *datasets* de teste e de inferência (sentenças manualmente anotadas). A Tabela 5.1 apresenta a acurácia obtida nos *datasets* de teste e de inferência para cada um dos modelos:



(a) (b)

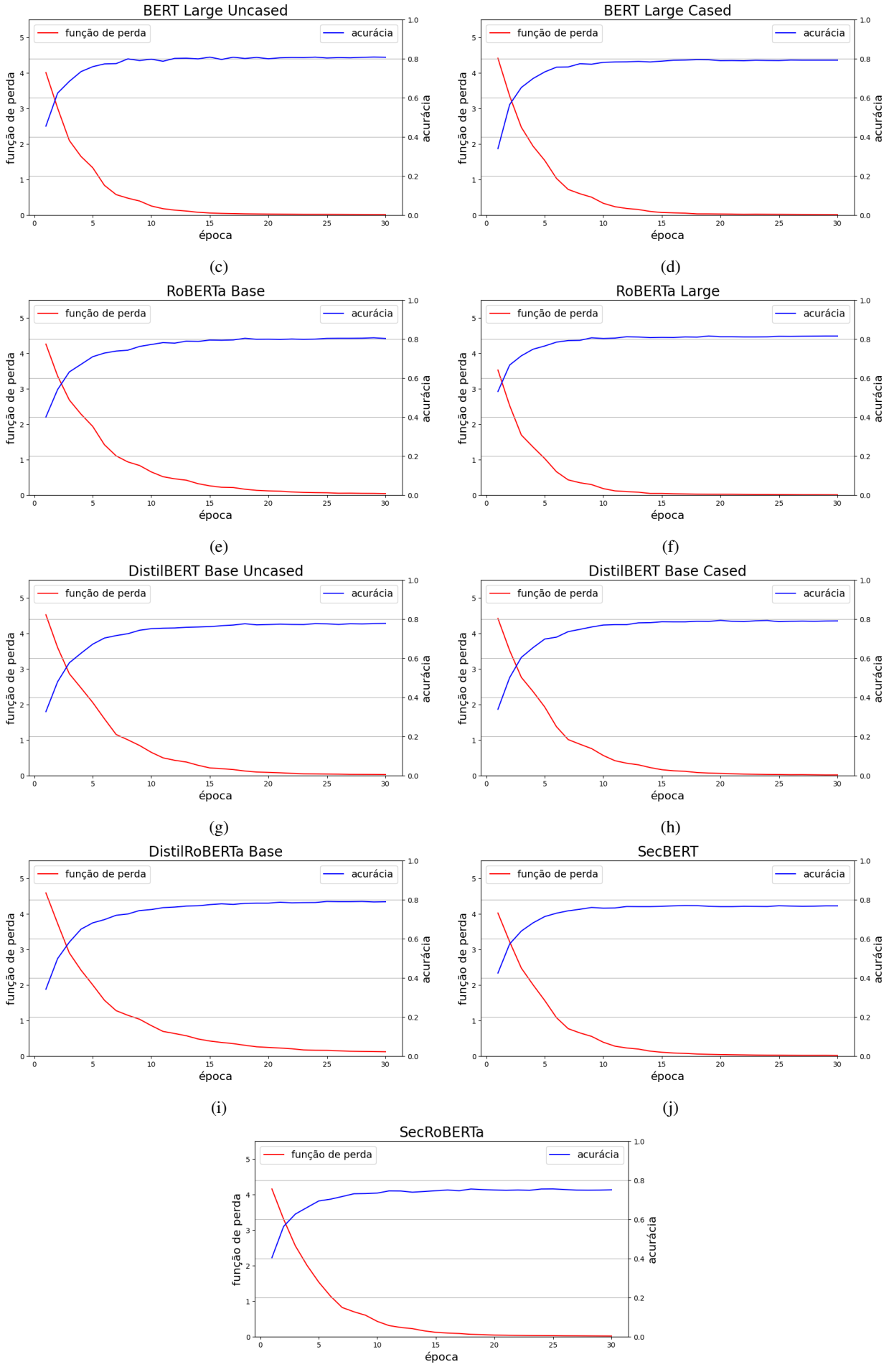


Figura 5.1: Acurácia e Função de Perda dos 11 modelos BERT testados.

Tabela 5.1: Acurácia dos modelos BERT na classificação de TTPs nos datasets de teste e de inferência utilizando os hiperparâmetros iniciais.

Modelos	**Dataset de teste**	**Dataset de inferência**
BERT Base Uncased	0,7719	0,6375
BERT Base Cased	0,7906	0,7125
BERT Large Uncased	0,8143	0,7250
BERT Large Cased	0,8032	**0,7875**
RoBERTa Base	0,7951	0,7000
RoBERTa Large	**0,8264**	0,7750
DistilRoBERTa Base	0,7931	0,6500
DistilBERT Base Uncased	0,7840	0,7125
DistilBERT Base Cased	0,7729	0,6750
SecBERT	0,7830	0,7000
SecRoBERTa	0,7633	0,7000

Os modelos que obtiveram o melhor desempenho foram RoBERTa Large e BERT Large Cased, com uma acurácia de 0,8264 e 0,7875 nos *datasets* de teste e inferência, respectivamente. Ambos são modelos grandes, com redes neurais de 24 camadas e pré-treinados com 355 milhões e 340 milhões de parâmetros, respectivamente. Na arquitetura de transformadas do BERT, o tamanho afeta o desempenho, ainda que não de forma drástica [7]. Dessa forma, o resultado mostra-se dentro do esperado, com os modelos maiores obtendo melhor performance.

Percebe-se, pela Tabela 5.1 que, assim como o modelo de linha de base (TF-IDF/Regressão Logística), os resultados das predições nos dados de inferência são piores que os obtidos para o *dataset* de teste. Avaliou-se que esse resultado se deve ao fato de que as sentenças retiradas de relatórios de CTI (inferência) são mais variadas e mais complexas que os exemplos da base do MITRE (teste). Além disso, as sentenças presentes no *dataset* de inferência foram elaboradas por 16 organizações diferentes, enquanto no *dataset* de teste todos os dados foram curados pelo mesmo ente (MITRE). Diferentes organizações e analistas apresentam padrões, convenções e estilos de escrita distintos, tornando os dados mais heterogêneos e dificultando a tarefa de classificação.

Constatou-se ainda que os modelos treinados em textos do domínio cibernético não apresentaram resultados superiores, contrariando as expectativas. Os modelos específicos da área cibernética encontrados, SecBERT e SecRoBERTa, foram treinados em textos diversos que incluem um repositório de relatórios de CTI. A obtenção de performance superior seria uma hipótese razoável nesse caso. Contudo, infelizmente os modelos não apresentam informações suficientes sobre o treinamento realizado (qual a tarefa NLP utilizada no treinamento, por exemplo) e suas condições, dificultando inferências sobre as razões do desempenho obtido.

Observa-se também que os modelos BERT apresentaram menor diferença entre o desempenho no teste e na inferência. A linha de base apresentou uma diferença de 12,8 pontos percentuais no desempenho, enquanto os modelos BERT com as configurações iniciais tiveram uma diferença média de 8,4 pontos percentuais. Esse dado representa um indicativo de que BERT teve menos dificuldades de adaptar-se às

inúmeras variações (comprimento das frases, estilo de escrita, vozes passiva e ativa) presentes nos textos extraídos de relatórios de CTI.

Os resultados do experimento também mostram a relevância da quantidade de amostras na capacidade de predição dos modelos. Utilizando o modelo de melhor acurácia, RoBERTa Large, verificou-se a acurácia individualizada obtida para as dez técnicas ou subtécnicas mais comuns no *dataset* de teste. A Tabela 5.2 mostra esses dados:

Tabela 5.2: Acurácia das técnicas ou subtécnicas com maior amostra de sentenças exemplificativas.

Técnica:subtécnica	**ID**	**Acurácia**
Ingress Tool Transfer	T1105	0,9420
System Information Discovery	T1082	0,9137
Obfuscated Files or Information	T1027	0,9231
Application Layer Protocol: Web Protocols	T1071.001	1,0000
Command and Scripting Interpreter: Windows Command Shell	T1059.003	0,8846
File and Directory Discovery	T1083	0,8421
Process Discovery	T1057	0,9773
Indicator Removal on Host: File Deletion	T1070.004	0,9111
Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.001	0,8667
System Network Configuration Discovery	T1016	0,8409

Os dados individualizados mostram que o modelo, para todas as dez técnicas com mais sentenças de amostra, obteve desempenho superior à acurácia total obtida no *dataset* de teste (82,64%). Das dez técnicas, seis obtiveram acurácia superior à 90%, tendo uma das técnicas (*Application Layer Protocol: Web Protocols*) alcançado a marca de 100%.

Os resultados obtidos no experimento demonstram que BERT alcança boa performance no problema de classificação de TTPs. Comparando o melhor modelo (RoBERTa Large) ao modelo de linha de base, percebe-se um incremento de 22,1 pontos percentuais na acurácia. A comparação entre os resultados aqui obtidos e outros trabalhos precedentes é dificultada pelo fato de que, a despeito do objetivo ser semelhante, a similaridade entre os trabalhos é limitada por diferentes premissas iniciais.

Entre os mais similares, o TCENet, mostrou classificação de TTPs com uma acurácia de 94,1%. No entanto, os testes do TCENet envolveram apenas as cinco técnicas (ou subtécnicas) mais populares e uma tática [13]. A pesquisa desenvolvida neste trabalho aplicou modelos BERT para classificação de 253 técnicas e subtécnicas. Husari et al [13] alegam terem obtido precisão de 84% e recall de 82%, no entanto o experimento foi realizado sob pressupostos significativamente diferentes. A abordagem daquele trabalho não empregava aprendizado de máquina e baseava-se em uma ontologia previamente construída que precisaria ser refeita manualmente a cada atualização do framework ATT&CK.

5.1 ANÁLISE DE HIPERPARÂMETROS

Buscando aprimorar o desempenho na etapa do ajuste fino, foi conduzida uma análise de parametrização na qual foram testadas 16 possíveis combinações de pares taxa de aprendizado/tamanho do lote. A Figura 5.2 abaixo apresenta os resultados:

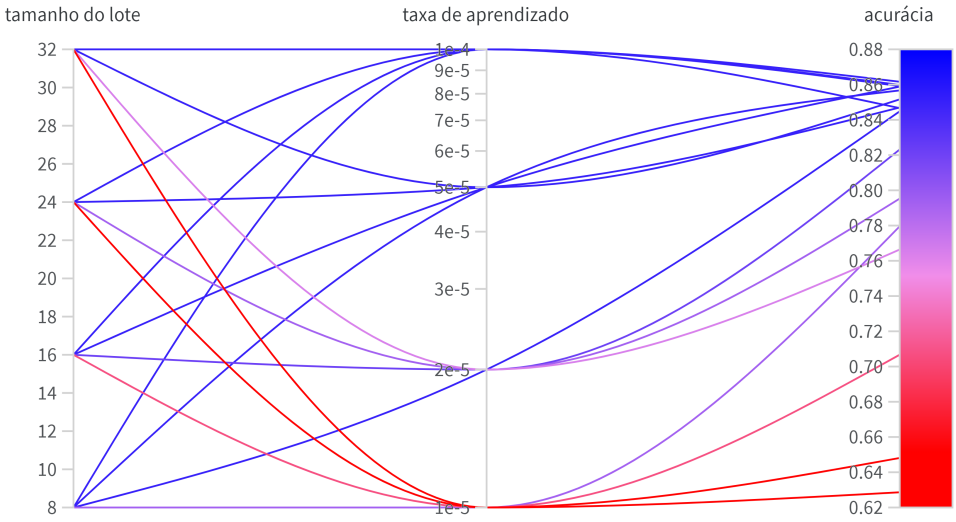


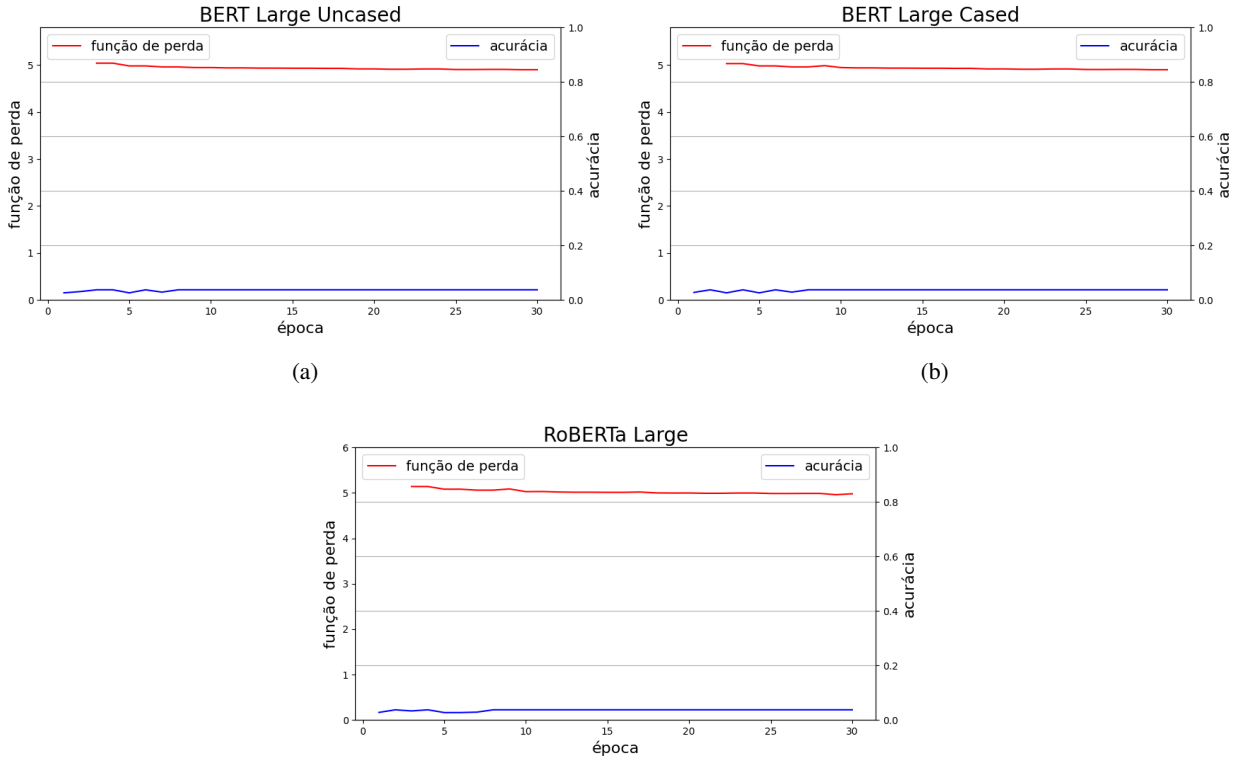
Figura 5.2: Análise de parametrização para taxa de aprendizado e tamanho do lote.

As variáveis taxa de aprendizado e acurácia apresentaram um coeficiente de correlação de Pearson positivo de 0,670. Esse dado informa que, nas condições do experimento, é provável que maiores taxas de aprendizado impliquem acurácia mais elevada. O tamanho do lote, contudo, não apresentou correlação significativa (-0,283) com a métrica escolhida. Esses resultados vão ao encontro do conceito de Goodfellow *et al.* [98] de que a taxa de aprendizado é o mais importante hiperparâmetro a ser ajustado em modelos de aprendizagem de máquina.

A melhor combinação observada entre os hiperparâmetros testados foi: taxa de aprendizado de 1e-4 e tamanho de lote 24. No entanto, ao aplicar essa taxa, todos os modelos do tipo “Large” incorreram no chamado esquecimento catastrófico. Essa situação consiste na incapacidade da rede neural de reter informações antigas quando apresentada a informações novas. Constitui problema comum na aprendizagem de máquina no campo de NLP, particularmente quando se utilizam taxas de aprendizado mais altas [96, 104].

Na ocorrência do fenômeno do esquecimento catastrófico, as curvas da função de perda e acurácia apresentam-se como duas retas paralelas. Não há convergência para os valores esperados. A Figura 5.3 mostra o gráfico dos modelos “Large” atingidos:

Aplicando a combinação de hiperparâmetros encontrada na varredura aos 11 modelos BERTs estudados, foram obtidos os resultados explicitados na Tabela 5.3:



(a) (b)

(c)

Figura 5.3: Esquecimento Catastrófico - acurácia e função de perda dos modelos “Large”

Tabela 5.3: Acurácia dos modelos BERT na classificação de TTPs nos datasets de teste e de inferência utilizando os hiperparâmetros otimizados. EC corresponde a situações de esquecimento catastrófico.

Modelos	**Dataset de teste**	**Dataset de inferência**
BERT Base Uncased	0,7996	0,7000
BERT Base Cased	0,7840	0,7250
BERT Large Uncased	EC	EC
BERT Large Cased	EC	EC
RoBERTa Base	0,8007	0,6875
RoBERTa Large	EC	EC
DistilRoBERTa Base	**0,8012**	0,7538
DistilBERT Base Uncased	0,7825	**0,7625**
DistilBERT Base Cased	0,7936	0,7125
SecBERT	0,7926	0,6750
SecRoBERTa	0,7845	0,7000

Percebe-se uma pequena tendência de melhoria de desempenho. Os modelos “destilados” (DistilBERT Cased e Uncased e DistiRoBERTa, modelos mais enxutos, com redes neurais de seis camadas e treinados com 65, 66 e 85 milhões de parâmetros respectivamente) apresentaram os maiores avanços. Contudo, nenhum dos modelos menores atingiu as marcas de desempenho dos modelos Large alcançados com os hiperparâmetros iniciais.

5.2 ANÁLISE QUALITATIVA DE ERROS DE CLASSIFICAÇÃO

Finalizado o experimento, foi realizada uma análise qualitativa dos erros de classificação visando comprender onde a abordagem proposta comete erros e discernir as razões por trás desses erros. Os resultados dessa análise permitem potencialmente recomendar ajustes em trabalhos futuros para aprimoramento de performance. Observando uma amostragem randomizada de erros de classificação, foram constatados ao menos quatro casos relevantes para análise:

- **Caso 1**: o *label* predito é mais preciso que o *label* anotado no dataset;
- **Caso 2**: tanto o *label* predito como o *label* anotado são corretos e consistem em subtécnicas de uma mesma técnica;
- **Caso 3**: tanto o *label* predito como o *label* anotado são corretos, mas pertencem a técnicas distintas;
- **Caso 4**: ambos os *labels* são corretos e há ainda pelo menos uma outra possibilidade de *label* correto.

A Tabela 5.4 abaixo mostra algumas situações exemplificativas de erros de classificação no dataset do MITRE:

Tabela 5.4: Exemplos de sentenças consideradas erroneamente classificadas. Inclui a classificação correta (*label* anotado na base do MITRE) e a classificação predita por BERT.

	Sentença	***Label* anotado**	***Label* predito**
1	MuddyWater has performed credential dumping with LaZagne.	OS Credential Dumping: Cached Domain Credentials (T1003.005)	OS Credential Dumping (T1003)
2	SHIPSHAPE achieves persistence by creating a shortcut in the Startup folder.	Boot or Logon Autostart Execution: Shortcut Modification (T1547.009)	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder (T1547.001)
3	MobileOrder has a command to upload information about all running processes to its C2 server.	Process Discovery (T1057)	Exfiltration Over C2 Channel (T1041)
4	JPIN can use the command-line utility cacls.exe to change file permissions.	Command and Scripting Interpreter: Windows Command Shell (T1059.003)	File and Directory Permissions Modification: Windows File and Directory Permissions Modification (T1222.001)
5	IcedID can inject itself into a suspended msiexec.exe process to send beacons to C2 while appearing as a normal msi application.	System Binary Proxy Execution: Msiexec (T1218.007)	Process Injection (T1055)

O primeiro caso é ilustrado pela sentença um da Tabela 5.4. Os *labels* são similares: o *framework* previu a técnica e a anotação manual do *dataset* marcou uma subtécnica dentro da mesma técnica. Possivelmente o analista humano anotou a subtécnica por conta de outras informações de contexto de algum

relatório de CTI. Contudo, a frase "*MuddyWater has performed credential dumping with LaZagne*" possibilita afirmar apenas que se trata da técnica *OS Credential Dumping* (T1003). Não permite, isoladamente, especificar a subtécnica *Cached Domain Credentials* (T1003.005) presente na anotação prévia. Dessa forma, a predição, foi mais precisa que a anotação prévia do dataset.

A sentença dois representa o segundo caso, no qual a anotação prévia e a predição apontaram para duas subtécnicas da mesma técnica. Na frase "*SHIPSHAPE achieves persistence by creating a shortcut in the Startup folder*" observamos uma mistura das duas subtécnicas, de modo que ambos os *labels* mostram-se adequados. Houve a utilização de atalho (T1547.009, *label* anotado) e também foi descrita a autoexecução por meio da pasta Startup (T1547.001, *label* predito).

O terceiro caso consiste na situação em que os *labels* anotados e previstos apontam para técnicas distintas, mas ambas podem ser inferidas das sentenças exemplo. Na frase "*MobileOrder has a command to upload information about all running processes to its C2 server*" identifica-se tanto a técnica *Process Discovery* (T1057, *label* anotado) como a técnica *Exfiltration Over C2 Channel* (T1041, *label* predito). Essas situações ocorrem porque, muitas vezes, mesmo sentenças curtas exprimem mais de uma técnica.

Semelhantemente, temos o quarto caso, em que os labels anotado e previsto apontam para subtécnicas de técnicas distintas. A frase "JPIN can use the command-line utility cacls.exe to change file permissions" expressa tanto o emprego do par 'técnica:subtécnica' *Command and Scripting Interpreter: Windows Command Shell* (T1059.003, *label* anotado) como de *File and Directory Permissions Modification: Windows File and Directory Permissions Modification* (T1222.001, *label* predito).

É possível até mesmo que as sentenças exemplo exprimam mais de duas técnicas. A quinta frase ilustra essa situação. Em "*IcedID can inject itself into a suspended msiexec.exe process to send beacons to C2 while appearing as a normal msi application*" temos a ocorrência de System Binary Proxy Execution: Msiexec (T1218.007, *label* anotado), bem como de Process Injection (T1055, *label* predito). Analisando a sentença, contudo, o trecho " *send beacons to C2*" permite inferir o uso da técnica Exfiltration Over C2 Channel (T1041).

Assim, algumas das predições consideradas incorretas, na realidade, apresentam *labels* adequados. Há potencial de aprimoramento da performance se essas predições pudessem ser consideradas. Da avaliação qualitativa dos erros, percebe-se que sentenças detalhando ações de *malwares* ou grupos maliciosos frequentemente envolvem múltiplas técnicas ou subtécnicas em uma mesma frase. As frases dois a cinco da Tabela 5.4, são ilustrativas dessa situação.

Essa observação é importante porque, embora a base de dados do MITRE apresente classificação unívoca (um *label* para cada sentença), algumas das sentenças aceitariam mais de um *label*. Dessa forma, os dados admitem uma classificação *multilabel*. Essa abordagem, traria, teoricamente, melhoria de performance, pois não penalizaria a acurácia do modelo em situações como as descritas nos casos dois a cinco.

Os gráficos e os dados tabelados nesse capítulo mostraram que, mesmo com os parâmetros iniciais, foi satisfeita a hipótese inicial de que LLMs permitiriam alcançar bons resultados no problema de classificação de TTPs a partir de textos (sentenças) não estruturados. A análise paramétrica mostrou que um dos hiperparâmetros investigados (taxa de aprendizado) apresenta correlação moderadamente positiva com

a acurácia, porém o ganho (quando ocorre), é marginal e incorre-se no risco do fenômeno do esquecimento catastrófico. Por fim, a análise qualitativa de amostragem dos erros de classificação possibilitou uma melhor compreensão dos resultados alcançados.

6 CONCLUSÕES

Uma segurança cibernética efetiva requer que os profissionais da área tenham à disposição recursos que facilitem a aquisição de informações essenciais ao seu trabalho. Na era do *Big Data*, o excesso de informações pode prejudicar a eficácia da proteção de ativos de rede. As informações advindas de fontes não estruturadas, particularmente, são uma fonte de preocupação para a segurança, pois seu processamento demanda muito dos recursos humanos. O auxílio da automação é essencial.

Nesse sentido, o trabalho desenvolvido nessa pesquisa contribui ao aplicar modelos BERT, estado da arte em NLU, ao problema de classificar TTPs segundo um *framework* consolidado (MITRE ATT&CK). Foi utilizado o repositório de sentenças rotuladas do MITRE com uma estratégia de amostragem estratificada e o *dataset* foi convertido em um formato adequado ao BERT. Após isso, 11 diferentes modelos BERT foram empregados, obtendo as melhores acurácias com o RoBERTa Large (*dataset* de teste) e BERT Large Cased (*dataset* de inferência). As acurácias obtidas, de 0,8264 e 0,7875, respectivamente, permitem confirmar a hipótese de que a aplicação de LLMs à tarefa de classificação do problema de pesquisa produz bons resultados.

Foi realizada uma "varredura" em dois parâmetros para verificar como suas diversas combinações afetavam a acurácia. Os efeitos da taxa de aprendizagem e tamanho do lote na acurácia foram investigados, buscando optimização. Constatou-se experimentalmente que o tamanho do lote não apresentou correlação com a acurácia nas condições pesquisadas. Verificou-se que a taxa de aprendizado pode produzir pequenas melhorias na acurácia, mas sob o risco do fenômeno do esquecimento catastrófico, como observado nos modelos maiores (Large) para a taxa mais alta.

A análise qualitativa de uma amostra dos erros, por sua vez, permitiu uma compreensão mais aprofundada das falhas de classificação dos modelos. Foram constatadas ao menos quatro situações recorrentes em que as classificações tidas como incorretas deveriam ser consideradas válidas. Em todas essas situações, a abordagem *multilabel* permitiria corrigir essas distorções.

6.1 LIMITAÇÕES

O presente trabalho encontrou e buscou contornar algumas limitações. A primeira limitação refere-se à dificuldade em encontrar *datasets* rotulados para análise de TTPs. A única base de dados consistente e de dimensão razoável foi o *dataset* de sentenças exemplificativas do MITRE. Esse repositório, bem-organizado e mantido, apresenta mais de dez mil sentenças e, mesmo assim, algumas técnicas não possuem exemplos. Além disso, possui a peculiaridade de todas as frases estarem estruturadas sintaticamente na forma SVO (Sujeito-Verbo-Objeto), o que pode ser útil em algumas pesquisas - particularmente no domínio de recuperação de informação (*infomation retrieval*) -, mas reduz a variabilidade de estilos de escrita fornecidos para o treinamento de sistemas de aprendizado de máquina, limitando-os a conhecer sentenças de formatos muito semelhantes.

O conceito de transferência de aprendizado permitiu a popularização do uso dos LLMs ao diminuir seu custo computacional a valores razoáveis para pesquisadores de diferentes níveis. Análises de sentimentos, por exemplo, são bastante beneficiadas pelo uso desses modelos. Contudo, diferente de um sistema de análise de sentimentos que possui número bastante limitado de classes, a presente pesquisa buscava realizar a classificação em 253 diferentes TTPs. O treinamento (ajuste fino) de modelos nessas condições é computacionalmente - e portanto, financeiramente - custoso. A plataforma colaborativa utilizada (Google Colab) apresenta limitações no poder computacional entregue, mesmo nas versões pagas. Esse fato implicou, por exemplo, o uso de um único modelo (BERT Base Uncased) para a varredura de parâmetros, pois consistia na parte mais custosa do experimento.

6.2 TRABALHOS FUTUROS

O BERT mostrou-se eficaz na classificação de sentenças sob uma abordagem multiclasse. Constatou-se, no entanto, que essa abordagem produz algumas pequenas distorções na classificação das sentenças. No futuro, é possível entender esse trabalho para uma modelagem *multilabel*, mitigando, por exemplo, o problema de sentenças longas com múltiplos TTPs descritos.

O emprego de maiores recursos computacionais pode também permitir uma investigação mais extensiva dos hiperparâmetros escolhidos, com a observação de uma gama mais ampla de combinações de valores. Além disso, os efeitos de outros hiperparâmetros (como decaimento de pesos ou taxa de *dropout*) podem ser verificados. Há um amplo campo de pesquisa a ser explorado relativo à utilização de NLP na segurança cibernética.

6.3 CONSIDERAÇÕES FINAIS

Além da contribuição técnica já destacada, é importante ressaltar o alinhamento da pesquisa com instrumentos normativos nacionais e recomendações internacionais. A Política Nacional de Segurança da Informação (PNSI) [105] elenca, entre seus objetivos, o fomento da pesquisa científica e da inovação na área de segurança da informação. A Estratégia Nacional de Segurança Cibernética (E-Ciber) [106] ressalta a importância de ferramentas de automação de segurança que utilizem inteligência artificial e aprendizado de máquina. A Política Nacional de Inteligência (PNI) [107] destaca os ataques cibernéticos como uma das principais ameaças à segurança nacional. Já a Estratégia Nacional de Inteligência (ENINT) [108] ressalta que essas ameaças produzem demanda por soluções capazes de ampliar o nível de segurança da informação.

No plano internacional, a Organização das Nações Unidas (ONU) [109] delineia como uma ameaça moderna o crescimento dos ataques cibernéticos em escala, severidade e complexidade. Além disso, reconhece a diferença de capacidade entre os países e propõe recomendações como o uso de padrões para compartilhamento de informações e o desenvolvimento de pesquisas acadêmicas que auxiliem a mitigar a ameaça crescente dos ataques cibernéticos.

O presente trabalho vai ao encontro dessas diretrizes e recomendações ao pesquisar forma inovadora de empregar inteligência artificial no auxílio à segurança cibernética. Demonstrou-se que a arquitetura BERT de transformadas constitui uma ferramenta útil e relevante para equacionar o problema de classificação de TTPs retirados de bases textuais. O emprego de ferramentas de NLP, como os LLMs, no campo da segurança cibernética ainda permanece pouco explorado. A segurança cibernética precisa apropriar-se desses instrumentos de forma a fornecer uma melhor proteção aos ativos informacionais no mundo moderno.

REFERÊNCIAS BIBLIOGRÁFICAS

1 CSIS. *Significant Cyber Incidents Since 2006*. [S.l.], 2023. 73 p. Último acesso em 21 jan 2023. Disponível em: <https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents>.

2 MITRE. *Our Story*. 2023. Último acesso em 30 abril 2022. Disponível em: <https://www.mitre.org/>.

3 CHOLLET, F. *Deep Learning with Python*. 2. ed. [S.l.]: Manning Publications, 2021.

4 ACADEMY, D. S. *Deep Learning Book*. 2023. Último acesso em 27 maio 2023. Disponível em: <https://www.deeplearningbook.com.br>.

5 HAYKIN, S. *Neural networks and learning machines*. 3. ed. [S.l.]: Pearson Prentice Hall, 2009.

6 GéRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2. ed. Califórnia, Estados Unidos: O'Reilly Media, 2019.

7 DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota, USA: [s.n.], 2019.

8 ANOMALI. *What is Threat Intelligence?* Último acesso em 27 maio 2023. Disponível em: <https://www.anomali.com/resources/what-is-threat-intelligence>.

9 FORUM, W. E. *The Global Risks Report 2022: 17th Edition*. [S.l.], 2022. Disponível em: <www3.weforum.org/docs/WEF_The_Global_Risks_Report_2022.pdf>.

10 BAHRAMI, P. N.; DEHGHANTANHA, A.; DARGAHI, T.; PARIZI, R. M.; CHOO, K.-K. R.; JAVADI, H. H. S. Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tac-tics, techniques, and procedures. *Journal of Informations Processing Systems*, v. 15, p. 885–889, agosto 2019.

11 BEJTLICH, R. Strategic defence in cyberspace: Beyond tools and tactics. In: GEERS, K. (Ed.). *Cyber War in Perspective: Russian Aggression against Ukraine*. Tallinn: NATO CCDCOE Publications, 2015. cap. 18.

12 LIBERATO, M. *SecBERT: Analyzing reports with BERT-like models*. Dissertação — University of Twente, Enschede, Netherlands, 12 2022.

13 YOU, Y.; JIANG, J.; JIANG, Z.; YANG, P.; LIU, B.; FENG, H.; WANG, X.; LI, N. Tim: threat context-enhanced ttp intelligence mining on unstructured threat data. *Cybersecurity*, v. 5, n. 3, February 2022.

14 ZHU, Z.; DUMITRAS, T. Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: SPRINGER (Ed.). *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. London, UK: IEEE, 2018. p. 458–472.

15 BIANCO, D. *The Pyramid of Pain*. 2013. Disponível em: <https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>.

16 QUINTERO-BONILLA, S.; REY, Á. M. del. A new proposal on the advanced persistent threat: A survey. *Applied Sciences*, v. 10, n. 11, junho 2020. Disponível em: <https://www.mdpi.com/2076-3417/10/11/3874>.

17 HEJASE, H. J.; FAYYAD-KAZAN, H. F.; MOUKADEM, I. Advanced persistent threats (apt): An awareness review. *Journal of Economics and Economic Education Research*, v. 21, n. 6, 2020.

18 STROM, B. E.; BATTAGLIA, J. A.; KEMMERER, M. S.; KUPERSANIN, W.; MILLER, D. P.; WAMPLER, C.; WHITLEY, S. M.; WOLF, R. D. *Finding Cyber Threats with ATT&CK-Based Analytics*. [S.l.], 2017. Disponível em: <https://www.mitre.org/publications/technical-papers/finding-cyber-threats-with-attck-based-analytics>.

19 RANADE, P.; PIPLAI, A.; JOSHI, A.; FININ, T. Cybert: Contextualized embeddings for the cybersecurity domain. In: IEEE (Ed.). *2021 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2021. p. 3334–3342.

20 LEGOY, V.; CASELLI, M.; SEIFERT, C.; PETER, A. *Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports*. Enschede, Netherlands, 2019.

21 HUSARI, G.; AL-SHAER, E.; AHMED, M.; CHU, B.; NIU, X. Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources. In: *ACSAC 2017: Proceedings of the 33rd Annual Computer Security Applications Conference*. New York, NY, USA: Association for Computing Machinery, 2017. (ACSAC '17), p. 103–115. ISBN 9781450353458.

22 HAREL, Y.; GAL, I. B.; ELOVICI, Y. Cyber security and the role of intelligent systems in addressing its challenges. In: *ACM Transactions on Intelligent Systems and Technology (TIST) - Special Issue: Cyber Security and Regular Papers*. New York, NY, USA: Association for Computing Machinery, 2017. v. 8, n. 4.

23 GHAZI, Y.; ANWAR, Z.; MUMTAZ, R.; SALEEM, S.; TAHIR, A. A supervised machine learning based approach for automatically extracting high-level threat intelligence from unstructured sources. In: *2018 International Conference on Frontiers of Information Technology (FIT)*. Islamabad, Pakistan: IEEE, 2018. p. 129–134.

24 RAHMAN, R.; MAHDAVI-HEZAVEH, R.; WILLIAMS, L. A literature review on mining cyberthreat intelligence from unstructured texts. In: *2020 International Conference on Data Mining Workshops (ICDMW)*. Sorrento, Italy: [s.n.], 2020.

25 CONNEAU, A.; KIELA, D.; SCHWENK, H.; BARRAULT, L.; BORDES, A. Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017. p. 670–680.

26 VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMES, A. N.; KAISER, L.; POLOSUKHIN, I. Attention is all you need. In: *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA: Curran Associates Inc., 2017. v. 30, p. 6000–6010.

27 PROTTASHA, N. J.; SAMI, A.; KOWSHER, M.; MURAD, S. A.; BAIRAGI, A. K.; MASUD, M.; BAZ, M. Transfer learning for sentiment analysis using bert based supervised fine-tuning. *Sensors*, v. 22, n. 11, May 2022.

28 INSTITUTE, P. *6th Cyber Resilient Organization Study*. [S.l.], 2021. Disponível em: <https://www.ibm.com/resources/guides/cyber-resilient-organization-study/>.

29 KENT, S. *Strategic Intelligence for American World Policy*. Princeton, NJ: Princeton University Press, 1949.

30 ALVES, P. M. d. M. R. A.; GONCALVES, V. P.; FILHO, G. P. R. Modelo de classificação de ttp baseado em transformadas bert. In: *Atas das Conferências Ibero-Americanas Computação Aplicada 2022 e WWW/Internet 2022*. Lisboa, Portugal: International Association for Development of the Information Society, 2022. p. 51–58.

31 ALVES, P. M. d. M. R. A.; GONCALVES, V. P.; FILHO, G. P. R. Leveraging bert's power to classify ttp from unstructured text. In: *2022 Workshop on Communication Networks and Power Systems (WCNPS 2022)*. Fortaleza, Brasil: Institute of Electrical and Electronics Engineers (IEEE), 2022.

32 ABU, M. S.; SELAMAT, S. R.; ARIFFIN, A.; YUSOF, R. Cyber threat intelligence – issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 10, n. 1, p. 371–379, 04 2018. ISSN 2502-4752.

33 ENGLAND, B. of. *CBEST Intelligence-Led Testing: Understanding Cyber Threat Intelligence Operations*. [S.l.], 2016.

34 DODSON, D.; MONTGOMERY, D.; POLK, T.; RANGANATHAN, M.; SOUPPAYA, M. *Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)*. [S.l.], 2021.

35 IANCU, N.; FORTUNA, A.; BARNA, C.; TEODOR, M. *Countering Hybrid Threats: Lessons Learned from Ukraine*. [S.l.]: IOS Press, 2016. v. 128. (NATO Science for Peace and Security Series - E: Human and Societal Dynamics, v. 128). ISBN: 978-1-61499-650-7 (Print); 978-1-61499-651-4 (Online).

36 CANADá. *Glossary*. Último acesso em 7 de Maio de 2022. Disponível em: <https://www.cyber.gc.ca/en/glossary>.

37 UNIDO, R. *National Cyber Strategy 2022: Pioneering a cyber future with the whole of the UK*. [S.l.], 2022. Último acesso em 7 de Maio de 2022. Disponível em: <https://www.gov.uk/government/publications/national-cyber-strategy-2022/national-cyber-security-strategy-2022>.

38 OLIVEIRA, H. F. M. d. Reflexões sobre o conceito de inteligência. *Revista Brasileira de Ciências Policiais*, v. 4, n. 2, p. 11–23, 2014. ISSN Eletrônico 2318-6917.

39 BRASIL. Doutrina nacional da atividade de inteligência: fundamentos doutrinários. Agência Brasileira de Inteligência, Brasília, DF, p. 105, 07 2016.

40 BRASIL. *Inteligência e Contrainteligência*. 2022. Último acesso em 13 de maio de 2023. Disponível em: <https://www.gov.br/abin/pt-br/assuntos/inteligencia-e-contrainteligencia>.

41 BRASIL. Decreto nº 4.376, de 13 de setembro de 2022. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2002. Dispõe sobre a organização e o funcionamento do Sistema Brasileiro de Inteligência, instituído pela Lei no 9.883, de 7 de dezembro de 1999, e dá outras providências. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/2002/d4376.htm>.

42 MINISTERIO, F. *O que são dados, informação e conhecimento? E inteligência ou sabedoria?* 2022. Último acesso em 13 de maio de 2023. Disponível em: <https://www.linkedin.com/pulse/o-que-s\%C3\%A3o-dados-informa\%C3\%A7\%C3\%A3o-e-conhecimento-ou-fernanda-ministerio/?originalSubdomain=pt>.

43 DRUCKER, P. F. The coming of the new organization. *Harvard Business Review*, n. 66, p. 45–53, 01 1988. Disponível em: <https://hbr.org/1988/01/the-coming-of-the-new-organization>.

44 UNIDOS, E. Work of a nation: the center of intelligence. Central Intelligence Agency, Washington, DC, Estados Unidos, p. 60, 09 2020.

45 BUBACH, R. *O ciclo da inteligência e os requisitos para a produção do conhecimento*. Dissertação (Mestrado) — Universidade de Vila Velha, Vila Velha, Espírito Santo, Brasil, 05 2019.

46 GREER, A. *CTI, CTI, CTI: Applying better terminology to threat intelligence objects*. [S.l.], 2020.

47 CONTI, M.; DARGAHI, T.; DEHGHANTANHA, A. Cyber threat intelligence: Challenges and opportunities. In: DEHGHANTANHA, A.; CONTI, M.; DARGAHI, T. (Ed.). *Cyber Threat Intelligence*. [S.l.]: Springer International Publishing AG, 2018, (Advances in Information Security, v. 70). p. 334.

48 BROWN, R.; STIRPARO, P. *SANS 2022 Cyber Threat Intelligence Survey*. [S.l.], 2022.

49 ADINEH, R. *Threat Informed Defense*. 2023. Último acesso em 07 maio 2023. Disponível em: <https://www.linkedin.com/pulse/threat-informed-defense-reza-adineh/>.

50 STROM, B. E.; APPLEBAUM, A.; MILLER, D. P.; NICKELS, K. C.; PENNINGTON, A. G.; THOMAS, C. B. *MITRE ATT&CK: Design and Philosophy*. [S.l.], 2020. Disponível em: <https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf>.

51 DALE, R. Classical approaches to natural language processing. In: INDURKHYA, N.; DAMERAU, F. (Ed.). *HANDBOOK OF NATURAL LANGUAGE PROCESSING*. 2nd. ed. [S.l.]: Chapman Hall/CRC, 2010, (Machine Learning Pattern Recognition). cap. 1, p. 676.

52 EISENSTEIN, J. *Natural Language Processing*. Cambridge, Massachusetts, Estados Unidos: The MIT Press, 2018. 587 p.

53 JOSEPH, S.; HLOMANI, H.; LETSHOLO, K.; KANIWA, F.; SEDIMO, K. Natural language processing: A review. *International Journal of Research in Engineering Applied Sciences*, v. 6, n. 3, p. 207–216, 3 2016.

54 KHURANA, D.; KOLI, A.; KHATTER, K.; SINGH, S. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, n. 82, p. 3713–3744, 07 2022.

55 BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. 1. ed. Califórnia, Estados Unidos: O'Reilly Media, 2009.

56 PRASAD, S. *The 5 phases in Natural Language Processing*. 2021. Último acesso em 07 maio 2023. Disponível em: <https://shashank-prasad11.medium.com/the-5-phases-in-natural-language-processing-c67a72ec742f>.

57 HUTCHINS, J. The history of machine translation in a nutshell. 2014.

58 TURING, A. M. Computing machinery and intelligence. *Mind*, LIX, n. 236, p. 433–460, outubro 1950.

59 HUTCHINS, J. Machine translation: A brief history. In: KOERNER, E. F. K.; ASHER, R. E. (Ed.). *Concise history of the language sciences: from the Sumerians to the cognitivists*. Oxford: Pergamon Press, 1996. p. 431–445.

60 SLOCUM, J. A survey of machine translation: its history, current status, and future prospects. *Computational Linguistics*, v. 11, n. 1, p. 1–17, 01 1985.

61 CHAPMAN, W. Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, n. 18, p. 544–551, 09 2011.

62 HUTCHINS, J. Machine translation over fifty years. *Histoire, Epistemologie, Langage*, XXII, n. 1, p. 07–31, 2001.

63 JOHRI, P.; KHATRI, S. K.; AL-TAANI, A. T.; SABHARWAL, M.; SUVANOV, S.; KUMAR, A. Natural language processing: History, evolution, application, and futurewor. In: *Proceedings of 3rd International Conference on Computing Informatics and Networks*. [S.l.]: Springer Nature Singapore, 2021.

64 FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. 2nd. ed. Rio de Janeiro: LTC, 2021.

65 ALPAYDIN, E. *Introduction to Machine Learning*. 3rd. ed. Cambridge, Massachusetts, Estados Unidos: The MIT Press, 2014.

66 HOEKSTRA, R. The knowledge reengineering bottleneck. *Semantic Web – Interoperability, Usability, Applicability*, IOS Press, v. 1, n. 1, p. 1–5, 2010.

67 BRONWLEE, J. *A Gentle Introduction to Transfer Learning for Deep Learning*. 2017. Último acesso em 27 maio 2023. Disponível em: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>.

68 SARKAR, D. *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*. 2018. Último acesso em 27 maio 2023. Disponível em: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>.

69 BAHETI, P. *A Newbie-Friendly Guide to Transfer Learning*. 2021. Último acesso em 27 maio 2023. Disponível em: <https://www.v7labs.com/blog/transfer-learning-guide>.

70 MULLER, B. *BERT 101 - State Of The Art NLP Model Explained*. 2022. Último acesso em 14 maio 2023. Disponível em: <https://huggingface.co/blog/bert-101>.

71 HUGGINGFACE. *How do Transformers work?* Último acesso em 14 maio 2023. Disponível em: <https://huggingface.co/learn/nlp-course/chapter1/4>.

72 WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016. Disponível em: <https://www.researchgate.net/publication/308646556_Google's_Neural_Machine_Translation_System_Bridging_the_Gap_between_Human_and_Machine_Translation>.

73 NAYAK, A.; TIMMAPATHINI, H. P.; PONNALAGU, K.; VENKOPARAO, V. Domain adaptation challenges of bert in tokenization and sub-word representations of out-of-vocabulary words. In: *Proceedings of the First Workshop on Insights from Negative Results in NLP*. [S.l.]: Association for Computational Linguistics, 2020. p. 1–5. ISBN 978-1-952148-66-8.

74 HOREV, R. *BERT Explained: State of the art language model for NLP*. 2018. Último acesso em 27 maio 2023. Disponível em: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>.

75 KHALID, S. *BERT Explained: A Complete Guide with Theory and Tutorial*. 2019. Último acesso em 27 maio 2023. Disponível em: <https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c>.

76 HAJJAR, A. J. *Top 30 NLP Use Cases: Comprehensive Guide for 2022*. 2021. Last accessed 27 August 2022. Disponível em: <https://research.aimultiple.com/nlp-use-cases/>.

77 LIAO, X.; YUAN, K.; WANG, X.; LI, Z.; XING, L.; BEYAH, R. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In: *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Vienna, Austria: Association for Computing Machinery, 2016.

78 VADAPALLI, S. R.; HSIEH, G.; NAUER, K. S. Twitterosint: Automated cybersecurity threat intelligence collection and analysis using twitter data. In: *Proceedings of the International Conference on Security and Management (SAM)*. Athens, Greece: [s.n.], 2018.

79 DIONíSIO, N.; ALVES, F.; FERREIRA, P.; BESSANI, A. Cyberthreat detection from twitter using deep neural networks. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest, Hungary: [s.n.], 2019.

80 NIAKANLAHIJI, A.; WEI, J.; CHU, B.-T. A natural language processing based trend analysis of advanced persistent threat techniques. In: *2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: [s.n.], 2018. p. 2995–3000.

81 TIKHOMIROV, M.; LOUKACHEVITCH, N.; SIROTINA, A.; DOBROV, B. Using bert and augmentation in named entity recognition for cybersecurity domain. In: *Natural Language Processing and Information Systems. NLDB 2020. Lecture Notes in Computer Science()*. Saarbrücken, Germany: Springer, Cham, 2020. v. 12089, p. 16–24.

82 RIERA, T. S.; HIGUERA, J.-R. B.; HIGUERA, J. B.; HERRAIZ, J.-J. M.; MONTALVO, J.-A. S. A new multi-label dataset for web attacks capec classification using machine learning techniques. *Computers & Security*, v. 120, June 2022.

83 SAUERWEIN, C.; PFOHL, A. Towards automated classification of attackers' ttps by combining nlp with ml techniques. 2018. Disponível em: <https://arxiv.org/abs/2207.08478>.

84 AYOADE, G.; CHANDRA, S.; KHAN, L.; HAMLEN, K.; THURAISINGHAM, B. Automated threat report classification over multi-source data. In: IEEE (Ed.). *IEEE 4th International Conference on Collaboration and Internet Computing*. Philadelphia, PA, USA: IEEE, 2018.

85 HUSARI, G.; NIU, X.; CHU, B.; AL-SHAER, E. Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. Miami, FL, USA: IEEE Press, 2018.

86 SATVAT, K.; GJOMEMO, R.; VENKATAKRISHNAN, V. N. Extractor: Extracting attack behavior from threat reports. In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. Vienna, Austria: IEEE, 2021. p. 598–615.

87 YODER SARAH; LASKY, J. *Automating Mapping to ATT&CK: The Threat Report ATT&CK Mapper (TRAM) Tool*. 2019. Disponível em: <https://medium.com/mitre-attack/automating-mapping-to-attack-tram-1bb1b44bda76>.

88 MITRE. *MITRE ATT&CK*. Último acesso em 4 junho 2023. Disponível em: <https://attack.mitre.org/>.

89 PADURARIU, C.; BREABAN, M. E. Dealing with data imbalance in text classification. In: *Procedia Computer Science. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019*. [S.l.: s.n.], 2019. v. 159, p. 736–745.

90 MADABUSHI, H. T.; KOCHKINA, E.; CASTELLE, M. Cost-sensitive bert for generalisable sentence classification with imbalanced data. In: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 125–134.

91 IIKURA, R.; OKADA, M.; MORI, N. Improving bert with focal loss for paragrah segmentation of novels. In: *Distributed Computeing and Artificial Intelligence, 17th International Conference (DCAI 0220). Advances in Intelligent Systems and Computing*. L´Aquila, Italy: Springer, Cham, 2020. v. 1237, p. 21–30.

92 OAK, R.; DU, M.; YAN, D.; TAKAWALE, H.; AMIT, I. Malware detection on highly imbalanced data through sequence modeling. In: *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. London, United Kingdom: Association for Computing Machinery, 2019. p. 37–48.

93 CHRISTIAN, H.; AGUS, M. P.; SUHARTONO, D. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech - Computer, Mathematics and Engineering Applications*, v. 7, n. 4, p. 286–294, 12 2016.

94 QAISER, S.; ALI, R. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, v. 181, n. 1, p. 25–29, 07 2018.

95 EDGAR, T.; MANZ, D. *Research Methods for Cyber Security*. [S.l.]: Syngress - Elsevier, 2017.

96 SUN, C.; QIU, X.; XU, Y.; HUANG, X. How to fine-tune bert for text classification? In: *Chinese Computational Linguistics: 18th China National Conference, CCL 2019*. Kunming, China: Springer International Publishing, 2019. p. 194–206.

97 JEAWAK, S.; ESPINOSA-ANKE, L.; SCHOCKAERT, S. Cardiff university at semeval-2020 task 6: Fine-tuning bert for domain-specific definition classification. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona, Spain: [s.n.], 2020.

98 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition*. Illustrated. [S.l.]: The MIT Press, 2016.

99 JEPKOECH, J.; MUGO, D. M.; KENDUIYWO, B. K.; TOO, E. C. The effect of adaptive learning rate on the accuracy of neural networks. *International Journal of Advanced Computer Science and Applications*, p. 736–751, 2021.

100 BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. In: MONTAVON, G. (Ed.). *Neural Networks: Tricks of the Trade*. Second. Berlin, Germany: Springer Berlin Heidelberg, 2012. p. 437–478.

101 ALDIN, N. B.; ALDIN, S. S. A. B. Accuracy comparison of different batch size for a supervised machine learning task with image classification. In: *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*. [S.l.: s.n.], 2022. p. 316–319.

102 KESKAR, N. S.; NOCEDAL, J.; TANG, P. T. P.; MUDIGERE, D.; SMELYANSKIY, M. On large-batch training for deep learning: Generalization gap and sharp minima. In: *5th International Conference on Learning Representations, ICLR 2017*. Toulon, France: [s.n.], 2017.

103 HE, F.; LIU, T.; TAO, D. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In: *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2019. v. 32, p. 1143–1152.

104 KAUSHIK, P.; GAIN, A.; KORTYLEWSKI, A.; YUILLE, A. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. In: *5th Workshop on Meta-Learning at NeurIPS 2021*. Virtual: [s.n.], 2021.

105 BRASIL. Decreto nº 9.637, de 26 de dezembro de 2018. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2018. Institui a Política Nacional de Segurança da Informação, dispõe sobre

a governança da segurança da informação, e altera o Decreto nº 2.295, de 4 de agosto de 1997, que regulamenta o disposto no art. 24, caput , inciso IX, da Lei nº 8.666, de 21 de junho de 1993, e dispõe sobre a dispensa de licitação nos casos que possam comprometer a segurança nacional. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/decreto/D9637.htm>.

106 BRASIL. Decreto nº 10.222, de 5 de fevereiro de 2020. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2020. Aprova a Estratégia Nacional de Segurança Cibernética. Disponível em: <https://www.in.gov.br/en/web/dou/-/decreto-n-10.222-de-5-de-fevereiro-de-2020-241828419>.

107 BRASIL. Decreto nº 8.793, de 29 de junho de 2016. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2016. Fixa a Política Nacional de Inteligência. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/decreto/D8793.htm>.

108 BRASIL. Decreto de 15 de dezembro de 2017. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2017. Aprova a Estratégia Nacional de Inteligência. Disponível em: <https://www.gov.br/abin/pt-br/centrais-de-conteudo/publicacoes/ENINT.pdf>.

109 UNIDAS, O. das N. *Report of the Group of Governmental Experts on Advancing Responsible State Behaviour in Cyberspace in the Context of International Security*. [S.l.], 2021. Último acesso em 4 junho 2023. Disponível em: <https://digitallibrary.un.org/record/3991743?ln=en>.