



**EEP**  
ESCOLA DE ENGENHARIA  
DE PIRACICABA



**FUMEP**  
Fundação Mantenedora da EEP

## **ENGENHARIA DE COMPUTAÇÃO**

Leonardo Munhoz Libardi

**Machine Learning na identificação de Links Maliciosos (*Phishing*)**

Piracicaba  
2024

Leonardo Munhoz Libardi

## **Machine Learning na identificação de Links Maliciosos (*Phishing*)**

Trabalho de Conclusão de Curso apresentado à  
Escola de Engenharia de Piracicaba como parte  
dos requisitos para obtenção do título de  
Bacharel em **Engenharia de Computação**.

Orientador: Prof. Clerivaldo José Roccia

Piracicaba  
2024

Leonardo Munhoz Libardi

## **Machine Learning na identificação de Links Maliciosos (*Phishing*)**

Trabalho de Conclusão de Curso apresentado à  
Escola de Engenharia de Piracicaba como parte  
dos requisitos para obtenção do título de  
Bacharel em Engenharia de Computação.

Piracicaba, 05 de novembro de 2024

### **Banca Examinadora:**

---

Nome e sobrenome do Professor Orientador – (Presidente)  
Titulação  
Instituição onde trabalha

---

Nome e sobrenome – (Membro)  
Titulação  
Instituição onde trabalha

---

Nome e sobrenome – (Membro)  
Titulação  
Instituição onde trabalha

*Dedico este trabalho aos meus pais Sra. Ana Paula Munhoz Libardi e Sr. Juliano Zangelmi Libardi, ao meu irmão Sr. Rafael Munhoz Libardi, à minha namorada Srta. Ana Flávia Ortiz de Paula, aos meus tios Sr. Mauricio Munhoz e Sr. Alexandre Zangelmi Libardi, as minhas tias, Sra. Viviane Munhoz Andrade e Sra. Vanessa Munhoz, as minhas avós, Sra. Ivete Ferrer Munhoz e Sra. Neide Zangelmi Libardi, aos meus falecidos avôs, Sr. Orlando Munhoz e Sr. Antônio Jair Libardi, ao meu sogro e minhas sogra, Sr. Ricardo Oliveira de Paula e Sra. Ana Fabíola Ortiz da Silva.*

## **AGRADECIMENTOS**

A realização deste trabalho é fruto de uma jornada marcada por aprendizado, desafios e apoio de pessoas fundamentais. Primeiramente, agradeço a Deus, pela força, inspiração e resiliência que me guiaram ao longo de todo o processo. Aos meus pais e familiares, por acreditarem em mim, nos momentos mais difíceis. À meu orientador/professor Clerivaldo Roccia pela paciência, dedicação e orientação valiosa durante a construção deste trabalho. Sua sabedoria e incentivo foram fundamentais para o desenvolvimento deste projeto. Aos amigos e colegas de curso, por compartilharem comigo esta caminhada, seja com palavras de apoio, troca de conhecimento ou simples momentos de descontração que aliviaram as tensões da jornada. A todos os professores e funcionários da EEP, que contribuíram direta ou indiretamente para minha formação acadêmica e pessoal.

Por fim, agradeço àqueles que, de forma anônima ou discreta, contribuíram para a concretização deste trabalho, seja com palavras, exemplos ou gestos de bondade. A todos, minha mais profunda gratidão.

## RESUMO

Este estudo investiga o emprego da Inteligência Artificial (IA) e *Machine Learning* (ML) na análise de dados com o propósito de identificar situações de ataques *phishing*. O objetivo principal é desenvolver um sistema de IA capaz de detectar padrões suspeitos e responder de forma eficaz a possíveis ataques virtuais de *phishing*. Para alcançar esse objetivo, foi adotada uma abordagem que combina revisão da literatura especializada com a implementação prática de algoritmos de IA em um ambiente de teste simulado. A pesquisa revela que a integração da IA na análise de dados é uma estratégia promissora para fortalecer a segurança cibernética. Os resultados obtidos demonstram a eficácia do sistema desenvolvido na detecção precoce de ameaças cibernéticas, permitindo uma resposta rápida e assertiva para mitigar potenciais danos, sendo os modelos de Regressão Linear e Linear SVM os modelos destaques, com uma métrica de 92% na qualidade do algoritmo (*F1 Score*) de cada um. A análise detalhada das métricas de desempenho do sistema revela uma taxa significativa de detecção de ameaças, com uma baixa taxa de falsos positivos, o que confirma a viabilidade e eficácia da abordagem proposta. Como conclusão, este estudo evidencia a capacidade da IA de analisar grandes volumes de dados em tempo real e identificar comportamentos anômalos, possibilitando uma resposta proativa diante de possíveis ataques. Além disso, ressalta-se a necessidade contínua de pesquisa e desenvolvimento de técnicas avançadas de IA para enfrentar os desafios em constante evolução no cenário da segurança cibernética. Em síntese, este estudo contribui significativamente para o avanço do conhecimento na área de segurança cibernética, oferecendo uma solução prática e aplicável para fortalecer a proteção digital contra ameaças virtuais, mais especificamente o *phishing*. Ao integrar a IA na análise de dados, torna-se possível não apenas identificar e neutralizar estes ataques, mas também antecipar e prevenir futuras ameaças, garantindo assim um ambiente online mais seguro e confiável para indivíduos e organizações.

**Palavras-chave:** Inteligência Artificial, Análise de Dados, Segurança Cibernética, Ameaças Cibernéticas, Detecção de Padrões, *Phishing*.

## **ABSTRACT**

This study investigates the use of Artificial Intelligence (AI) and Machine Learning (ML) in data analysis to identify phishing attack situations. The main objective is to develop an AI system capable of detecting suspicious patterns and responding effectively to potential phishing attacks. To achieve this goal, an approach was adopted that combines a review of specialized literature with the practical implementation of AI algorithms in a simulated test environment. The research reveals that the integration of AI in data analysis is a promising strategy to strengthen cybersecurity. The results obtained demonstrate the effectiveness of the developed system in the early detection of cyber threats, allowing a fast and assertive response to mitigate potential damage, with the Linear Regression and Linear SVM models being the standout models, with a metric of 92% in the quality of the algorithm (F1 Score) for each one. A detailed analysis of the system's performance statistics reveals a significant threat detection rate, with a low false positive rate, which confirms the prediction and effectiveness of the proposed approach. In conclusion, this study highlights the ability of AI to analyze large volumes of data in real time and identify anomalous behaviors, enabling a proactive response to potential attacks. Furthermore, it highlights the continued need for research and development of advanced AI techniques to address the ever-evolving challenges in the cybersecurity landscape. In summary, this study contributes significantly to the advancement of knowledge in the area of cybersecurity, offering a practical and applicable solution to strengthen digital protection against virtual threats, more specifically phishing. By integrating AI into data analysis, it becomes possible not only to identify and neutralize attacks, but also to anticipate and prevent future threats, thus ensuring a safer and more reliable online environment for individuals and organizations.

**Keywords:** Artificial Intelligence, Data Analysis, Cyber Security, Cyber Threats, Pattern Detection, Phishing.

## LISTA DE ILUSTRAÇÕES

FIGURA 1: Leitura do dataset.....	20
FIGURA 2: Gráfico do conteúdo do dataset.....	21
FIGURA 3: Stemming vs Lemmatization.....	22
FIGURA 4: Integração NLP Python.....	22
FIGURA 5: Resultado Integração Lemmatization e Stemmingção NLP Python.....	23
FIGURA 6: Linha de código do train_test_split.....	23
FIGURA 7: Linhas de código dos modelos e função fit().....	23
FIGURA 8: Equação Acurácia.....	24
FIGURA 9: Equação F1 Score.....	24
FIGURA 10: Equação Precisão.....	25
FIGURA 11: Equação Recall.....	25
FIGURA 12: Cálculo das métricas.....	26
FIGURA 13: Exemplo da Plot da confusion matrix.....	26
FIGURA 14: Gráfico de acurácia.....	28
FIGURA 15: Gráfico da qualidade do algoritmo.....	28
FIGURA 16: Matrizes de Confusão.....	29



## **LISTA DE TABELAS**

**TABELA 1: Tabela de resultados**

## LISTA DE ABREVIATURAS E SIGLAS

**FUMEP:** Fundação Municipal de Ensino de Piracicaba

**EEP:** Escola de Engenharia de Piracicaba

**ML:** Machine Learning

**IA:** Inteligência Artificial

**NLP:** Natural Language Processing

**Linear SVM:** Linear Support Vector Machine

## Sumário

1.	INTRODUÇÃO .....	12
2.	REFERENCIAL CIENTÍFICO E TECNOLÓGICO .....	13
3	MATERIAS E MÉTODOS.....	15
4.	DESENVOLVIMENTO.....	17
5.	CONSIDERAÇÕES FINAIS.....	29

## 1. INTRODUÇÃO

Na era digital em que vivemos, a segurança cibernética emergiu como um dos principais desafios enfrentados por organizações e indivíduos. Com o aumento exponencial de dados gerados e transações realizadas online, a necessidade de proteger sistemas e redes contra ataques cibernéticos torna-se cada vez mais importante. Nesse contexto, a Inteligência Artificial (IA) surge como uma ferramenta promissora para a detecção e prevenção de ameaças virtuais (STALLINGS, 2011).

A presente pesquisa se insere no campo da segurança cibernética, mais especificamente na área de análise de dados para o reconhecimento de ataques cibernéticos. Com base em uma revisão da literatura atualizada, constata-se que os ataques *phishing* têm se tornado cada vez mais sofisticados e frequentes, representando uma ameaça significativa para organizações de todos os setores e para os usuários individuais da internet (KASPERSKY LAB, 2018).

Os ataques *phishing* apresentam diversas formas e podem resultar em consequências devastadoras, como roubo de dados sensíveis, interrupção de serviços essenciais e danos financeiros. Diante desse cenário, torna-se essencial desenvolver técnicas avançadas de detecção e prevenção desses tipos de ataques, capazes de identificar padrões suspeitos e agir proativamente para mitigar potenciais danos.

O objetivo principal deste trabalho é desenvolver e implementar um sistema de IA para análise de dados, visando reconhecer e responder a ataques *phishing* de maneira eficaz. Ao final desta pesquisa, espera-se fornecer uma contribuição significativa para a área de segurança cibernética, apresentando uma solução viável e eficiente para proteger sistemas e redes contra ameaças virtuais.

Ao abordar esses aspectos, este estudo busca não apenas apresentar um panorama atualizado do problema dos ataques cibernéticos, mas também propor uma abordagem inovadora e eficaz para lidar com essa questão premente. A partir dessa introdução, o presente trabalho se desenvolverá, explorando em detalhes os fundamentos teóricos, metodológicos e práticos relacionados ao tema proposto.

## **2. REFERENCIAL CIENTÍFICO E TECNOLÓGICO**

A crescente digitalização dos processos e a expansão das redes de informação têm exposto sistemas e dados a uma variedade de ameaças cibernéticas cada vez mais sofisticadas. Nesse contexto, a necessidade de desenvolver e implementar métodos eficazes para a detecção e mitigação de ataques cibernéticos tornou-se primordial. A Inteligência Artificial (IA) e a análise de dados têm emergido como ferramentas poderosas na defesa contra possíveis ameaças, oferecendo novas possibilidades para a segurança cibernética.

### **2.1. Segurança Cibernética: Conceitos e Desafios**

A segurança cibernética refere-se às práticas e tecnologias utilizadas para proteger sistemas, redes e dados de ataques digitais. Segundo *Stallings* (2011), a segurança cibernética envolve a proteção de informações e sistemas de informações contra acessos não autorizados, ataques, destruição ou modificação. Com o aumento da complexidade e frequência dos ataques, as abordagens tradicionais de segurança, baseadas em assinaturas e regras fixas, mostraram-se insuficientes (SYMANTEC, 2019). Um dos métodos de ataque mais comuns é o *phishing*, uma forma de ataque em que os cibercriminosos enganam as vítimas para que revelem informações sensíveis, como senhas ou detalhes bancários, disfarçando-se como entidades confiáveis (JAKOBSSON; MYERS, 2006). O *phishing* representa um desafio contínuo na segurança cibernética, especialmente com a evolução de técnicas de ataque que utilizam engenharia social e personalização para aumentar a taxa de sucesso (KASPERSKY LAB, 2018).

### **2.2. Análise de Dados na Segurança Cibernética**

A análise de dados envolve a inspeção, limpeza e modelagem de dados com o objetivo de descobrir informações úteis e apoiar a tomada de decisões. Na segurança cibernética, a análise de dados é utilizada para identificar padrões e anomalias que possam indicar atividades maliciosas. A detecção de anomalias é um campo essencial na segurança cibernética, pois permite identificar comportamentos incomuns que

podem ser sinais de ataques (CHANDOLA, V.; BANERJEE, A.; KUMAR, V., 2009). No caso de *phishing*, a análise de dados pode ajudar a identificar e bloquear sites fraudulentos, com base em padrões comportamentais ou características típicas das comunicações de *phishing*, como domínios suspeitos, solicitações de informações sensíveis e erros gramaticais.

### **2.3. Inteligência Artificial e Aprendizado de Máquina**

A IA, especialmente o aprendizado de máquina (ML), tem revolucionado a forma como a análise de dados é conduzida na segurança cibernética. O aprendizado de máquina envolve a criação de algoritmos que podem aprender a partir de dados e fazer previsões ou tomar decisões sem serem explicitamente programados para isso (GOODFELLOW; BENGIO; COURVILLE, 2016). Técnicas de ML, como redes neurais, florestas aleatórias e máquinas de vetores de suporte, têm sido aplicadas com sucesso na detecção de anomalias e ameaças (SOMMER; PAXSON, 2010). No combate ao *phishing*, algoritmos de ML são frequentemente usados para identificar e bloquear tentativas de *phishing*, treinando modelos com grandes conjuntos de dados de exemplos de *phishing* e e-mails legítimos, permitindo assim uma detecção mais eficaz e automatizada. Atualmente, técnicas de Deep Learning têm mostrado grande potencial na análise de grandes volumes de dados em tempo real (KASPERSKY LAB, 2018). Especificamente no caso de *phishing*, a IA pode ser utilizada para analisar grandes quantidades de comunicações digitais em busca de padrões que indiquem fraudes, oferecendo defesas em tempo real contra tentativas de engano.

### 3 MATERIAS E MÉTODOS

Os materiais e ferramentas utilizadas nesta pesquisa foram:

Plataforma de Dados: Kaggle, utilizado para a obtenção dos conjuntos de dados.

Linguagem de Programação: Python, versão 3.12.4

Ambiente de Desenvolvimento: Visual Studio Code (Versão Atual).

Bibliotecas Python:

1. *Pandas*: Para a manipulação e análise de dados.
2. *Numpy*: Para operações com *arrays* e matrizes.
3. *Seaborn*: Para visualização de dados.
4. *Matplotlib*: Para visualização de dados, utilizado em conjunto com o *seaborn*.
5. *Scikit-Learn*: Para implementação dos modelos de ML, métricas, e pré-processamento de dados.
6. *NLTK*: Integração das técnicas de Lematização e Derivação para melhor precisão dos modelos de ML.
7. *Collections* (Counter): Coleção não ordenada onde os elementos são armazenados como chaves em um dicionário e sua contagem como valor *dict*.
8. *SMOTE*: Biblioteca utilizada para equilibrar o dataset com dados sintéticos.
9. *Time*: Contagem de tempo, para análise de desempenho dos ML.

Métodos utilizados:

1. *train\_test\_split*: Para a divisão dos dados em conjuntos de treinamento e teste.
2. *SMOTE*: Para o balanceamento do dataset com dados sintéticos.
3. *RegexTokenizer* e *SnowBallStemmer*: Integração do método *NLP*.
4. *Linear Regression*: Para a construção e avaliação do modelo de ML.
5. *Logistic Regression*: Para a construção e avaliação do modelo de ML.
6. *Linear SVC*: Para a construção e avaliação do modelo de ML.
7. *Bernoulli Naive Bayes*: Para a construção e avaliação do modelo de ML.
8. *Decision Tree Classifier*: Para a construção e avaliação do modelo de ML.

Os resultados dos modelos foram analisados e comparados utilizando métricas de desempenho como acurácia, precisão, recall e F1-score. A análise foi realizada com o auxílio da biblioteca *scikit-learn*, utilizando a função de *confusion\_matrix* e as métricas acima.



## 4. DESENVOLVIMENTO

### 4.1. Introdução ao *phishing* e a Aplicação de Machine Learning:

A rápida expansão do uso de tecnologias digitais trouxe consigo uma crescente exposição a ciberataques. Entre as formas mais comuns desses ataques está o *phishing*, no qual hackers se disfarçam de entidades confiáveis para enganar usuários e obter informações sensíveis, como senhas, números de cartão de crédito ou detalhes de contas bancárias (JAKOBSSON; MYERS, 2006). Esses ataques têm evoluído em sofisticação, utilizando técnicas avançadas de engenharia social e personalização de mensagens, tornando-se cada vez mais difíceis de detectar. Relatórios recentes apontam que o *phishing* é uma das principais ameaças à segurança cibernética, com atacantes explorando vulnerabilidades em e-mails, redes sociais e até mesmo em aplicativos de mensagens (KASPERSKY LAB, 2018).

Em um cenário onde grande parte das interações e transações ocorre em ambientes digitais, as consequências de ataques de *phishing* podem ser devastadoras tanto para indivíduos quanto para organizações, resultando em perdas financeiras, violação de dados e danos à reputação. Para enfrentar essa ameaça crescente, é fundamental desenvolver métodos mais eficazes de detecção. As abordagens tradicionais, como listas negras e regras estáticas, têm se mostrado insuficientes para lidar com a agilidade dos ataques, especialmente em um ambiente de ameaças em constante evolução (SYMANTEC, 2019).

Nesse contexto, o uso de Machine Learning (ML) tem se mostrado promissor como uma solução inovadora para detectar ataques de *phishing*. Algoritmos de aprendizado supervisionado e não supervisionado permitem a análise automática de grandes volumes de dados, identificando padrões de comportamento suspeitos com maior precisão e rapidez (GOODFELLOW; BENGIO; COURVILLE, 2016). Técnicas de ML oferecem uma vantagem significativa, pois conseguem aprender com dados históricos e ajustar-se a novas variantes de ataques, o que reduz a necessidade de intervenção manual e melhora a capacidade de resposta a novas ameaças.

Este trabalho explora a aplicação de diferentes algoritmos de ML, como Regressão Logística, Regressão Linear, *Linear SVC*, *Bernoulli Naive Bayes* e *Decision Tree Classifier*, na detecção de tentativas de *phishing* em ambientes digitais. Além de investigar a eficácia desses métodos, o estudo busca discutir as limitações sobre o desempenho no cenário simulado utilizado para as análises.

## **4.2. Modelos de Classificação:**

### **4.2.1. Regressão Logística:**

A Regressão Logística é amplamente utilizada para a classificação binária, sendo ideal para problemas onde o objetivo é prever a ocorrência de uma classe ou outra, como identificar se uma comunicação é *phishing* ou não. Esse modelo é baseado na probabilidade e ajuda a prever com precisão a chance de uma atividade maliciosa, com foco em variáveis categóricas (ZHANG et al., 2021). Além disso, é conhecida por sua simplicidade e interpretabilidade, o que facilita a explicação dos fatores que mais influenciam na detecção de *phishing* (GOODFELLOW; BENGIO; COURVILLE, 2016).

### **4.2.2. Regressão Linear:**

Embora tradicionalmente utilizada para prever variáveis contínuas, a Regressão Linear também contribui na análise de segurança cibernética ao ajudar a identificar tendências e relações lineares entre variáveis que possam indicar ataques de *phishing*. Esse modelo pode ser útil em análises exploratórias para entender fatores que influenciam a presença de ataques em determinados contextos (MONTGOMERY; PECK; VINING, 2021).

### **4.2.3. Linear Support Vector Classifier (Linear SVC):**

O *Linear Support Vector Classifier* (*Linear SVC*), uma variante do SVM, é uma técnica robusta para classificação de alta dimensionalidade, uma característica importante para identificar padrões complexos em dados de *phishing* (BURGES, 2000). Com capacidade de generalização eficiente, o SVC é eficaz ao separar

amostras legítimas de maliciosas, ajustando-se bem às necessidades da segurança cibernética e aos dados de alto volume (SCHÖLKOPF; SMOLA, 2002).

#### **4.2.4. Bernoulli Naive Bayes:**

O Bernoulli *Naive Bayes* é amplamente utilizado em dados binários, como a presença ou ausência de palavras-chave em e-mails suspeitos. Este modelo é eficiente para a detecção de *phishing*, onde variáveis podem ser tratadas como presentes ou ausentes, como URLs ou frases comuns em comunicações fraudulentas. Devido à sua rapidez e baixa complexidade computacional, é adequado para processamento em tempo real de grandes volumes de dados (MURPHY, 2012).

#### **4.2.5. Decision Tree Classifier:**

O *Decision Tree Classifier* é uma técnica que cria uma árvore de decisões baseada em critérios específicos, permitindo identificar os atributos mais relevantes para a classificação de *phishing*. Além de altamente interpretável, esse modelo é vantajoso para visualização e compreensão das características mais discriminantes para a detecção de *phishing*, o que facilita ajustes no sistema e aprimora a precisão (MITCHELL, 2006).

### **4.3. Metodologia:**

#### **4.3.1. Conjunto de Dados:**

Primeiramente, foi realizado a análise e limpeza dos dados que o *dataset* utilizado possui, utilizando a biblioteca Pandas.

FIGURA 1: Leitura do *dataset*

```

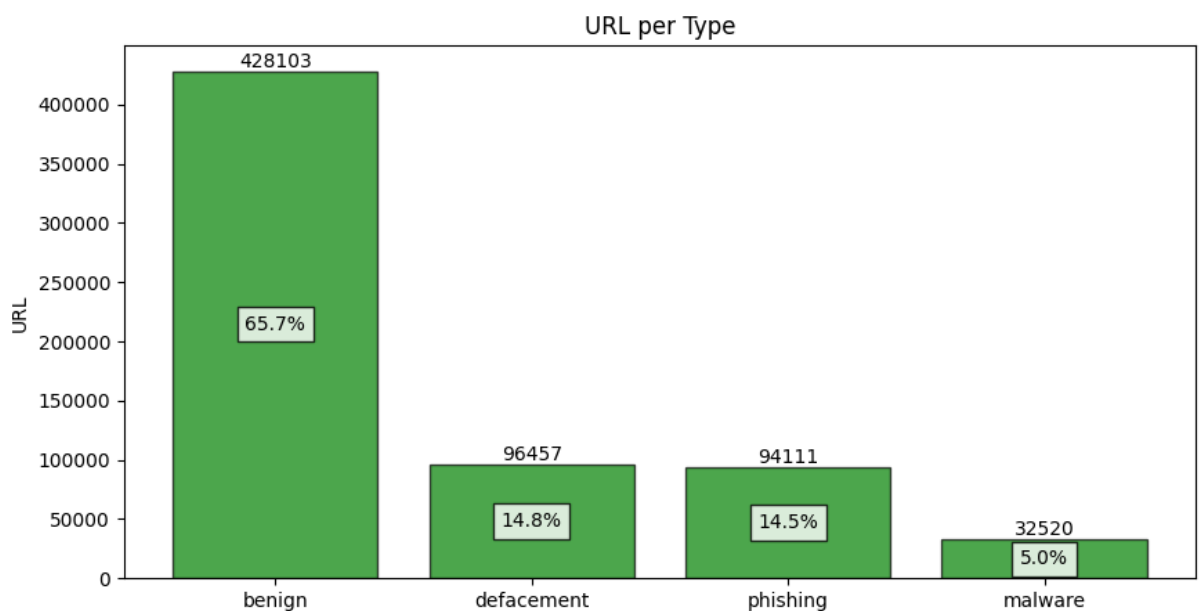
1 file_path = 'malicious_phish.csv'
2 df = pd.read_csv(file_path)
3 df.fillna(df.mode().iloc[0], inplace=True)
4 df.info()

```

FONTE: Figura do Autor

Abaixo, contém um gráfico que disponibiliza o conteúdo do conjunto de dados utilizado para o treinamento. <https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset>

FIGURA 2: Gráfico do conteúdo do *dataset*.



FONTE: Figura do Autor.

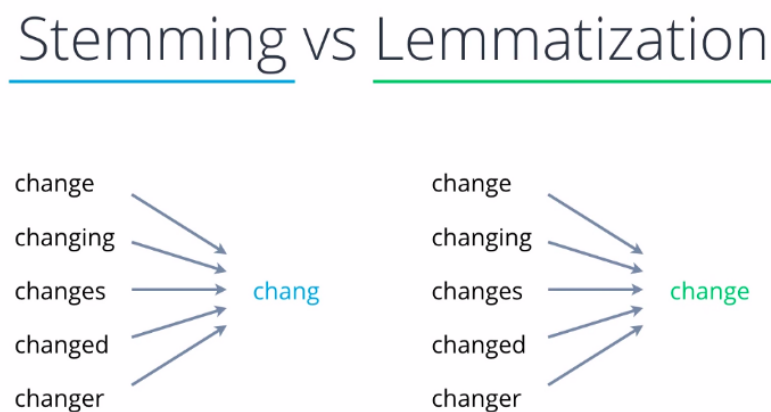
#### 4.3.2. Modelagem dos Dados:

Após realizado toda a preparação dos dados, foi iniciado a integração das técnicas de *Lemmatization* e *Stemming* (NLP). Foi escolhido esse tipo de técnica pois permite que os computadores compreendam, gerem e manipulem a linguagem humana. O

processamento de linguagem natural tem a capacidade de interrogar os dados com texto ou voz de linguagem natural (Oracle, 2021).

No *Stemming* é analisado cada *string* individualmente e reduzida à sua raiz, promovendo a indexação das palavras ou, como é chamado na técnica. Já o *Lemmatization*, é uma técnica mais apurada de vetorização que resulta em termos mais precisos, e é usada principalmente para modelar tópicos textuais, eliminando plurais e utilizando sinônimos. Pode-se seguir a imagem abaixo como exemplo:

FIGURA 3: *Stemming vs Lemmatization*



FONTE: Medium, 2022. Disponível em: <https://nirajbhoi.medium.com/stemming-vs-lemmatization-in-nlp-efc280d4e845>. Acesso em: novembro/2024

Integrando as técnicas acima no código do projeto:

FIGURA 4: Integração NLP Python

```
1 #Lemmatization
2 tokenizer = RegexpTokenizer(r'[A-Za-z]+')
3 df['text_tokenized'] = df.url.map(lambda t: tokenizer.tokenize(t))
4
5 #Stemming
6 stemmer = SnowballStemmer("english")
7 df['text_stemmed'] = df['text_tokenized'].map(lambda l: [stemmer.stem(word) for word in l])
8 df['text_sent'] = df['text_stemmed'].map(lambda l: ' '.join(l))
```

Fonte: Figura do Autor

Resultado exemplo:

FIGURA 5: Resultado Integração Lemmatization e Stemming

	url	type	text_tokenized	text_stemmed	text_sent
0	br-icloud.com.br	phishing	[br, icloud, com, br]	[br, icloud, com, br]	br icloud com br

FONTE: Figura do Autor.

#### 4.3.3. Criando Classes de Teste e Treino

As classes do modelo foram divididas em subconjuntos de treinamento e teste usando a função `train_test_split`. Este procedimento garante que o modelo seja treinado em uma parte dos dados e avaliado em uma parte independente, garantindo uma avaliação imparcial da performance.

FIGURA 6: Linha de código do `train_test_split`

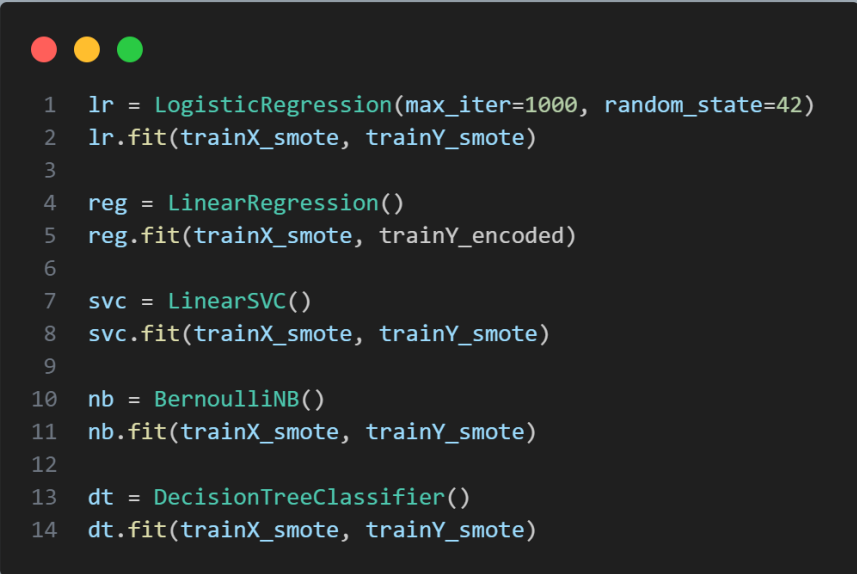
```
1 trainX, testX, trainY, testY = train_test_split(feat, df.type, test_size=0.3, random_state=42)
```

FONTE: Figura do Autor.

#### 4.3.4. Treinamento dos Modelos:

Para o treinamento, utilizou-se a função `LogisticRegression()`, `LinearRegression()`, `LinearSVC()`, `BernoulliNB()` e `DecisionTreeClassifier()`, todos da biblioteca Scikit-learn. Os treinamentos foram realizados em dados rotulados, e a precisão do modelo foi avaliada no conjunto de teste. Além disso, foi utilizado a função `fit()` para ajustar o modelo.

FIGURA 7: Linhas de código dos modelos e função `fit()`



```

1 lr = LogisticRegression(max_iter=1000, random_state=42)
2 lr.fit(trainX_smote, trainY_smote)
3
4 reg = LinearRegression()
5 reg.fit(trainX_smote, trainY_encoded)
6
7 svc = LinearSVC()
8 svc.fit(trainX_smote, trainY_smote)
9
10 nb = BernoulliNB()
11 nb.fit(trainX_smote, trainY_smote)
12
13 dt = DecisionTreeClassifier()
14 dt.fit(trainX_smote, trainY_smote)

```

FONTE: Figura do Autor.

#### 4.3.5. Avaliação de Desempenho:

Todos os modelos foram avaliados com base nas métricas de Acurácia, *F1 Score*, Precisão, *Recall* e *Confusion Matrix*. Utiliza-se a seguinte definição para as métricas acima:

- Acurácia: Quantidade de acertos dividido pelo total da amostra.

FIGURA 8: Equação Acurácia

$$Acurácia = \frac{Verdadeiros\ Positivos\ (TP) + Verdadeiros\ Negativos\ (VN)}{Total}$$

FONTE: Medium, 2020. Disponível em: <https://medium.com/@mateuspdua/machine-learning-métricas-de-avaliação-acurácia-precisão-e-recall-d44c72307959>. Acesso em: novembro/2024

- *F1 Score*: Une precisão e recall com o objetivo de fornecer um número único que determine a qualidade geral do modelo.

FIGURA 9: Equação *F1 Score*.

$$F1 = \frac{2 * precisão * recall}{precisão + recall}$$

FONTE: Medium, 2020. Disponível em: <https://medium.com/@mateuspdua/machine-learning-métricas-de-avaliação-acurácia-precisão-e-recall-d44c72307959>. Acesso em: novembro/2024

- **Precisão:** De todos os dados classificados como positivos, quantos são realmente positivos.

FIGURA 10: Equação Precisão

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos (TP)}}{\text{Verdadeiros Positivos (TP)} + \text{Falsos Positivos (FP)}}$$

FONTE: Medium, 2020. Disponível em: <https://medium.com/@mateuspdua/machine-learning-métricas-de-avaliação-accurácia-precisão-e-recall-d44c72307959>. Acesso em: novembro/2024

- **Recall:** Mostra a porcentagem de dados classificados como positivos comparando com a quantidade real de positivos da amostra.

FIGURA 11: Equação Recall

$$\text{Recall} = \frac{\text{Verdadeiros Positivos (TP)}}{\text{Verdadeiros Positivos (TP)} + \text{Falsos Negativos (FN)}}$$

FONTE: Medium, 2020. Disponível em: <https://medium.com/@mateuspdua/machine-learning-métricas-de-avaliação-accurácia-precisão-e-recall-d44c72307959>. Acesso em: novembro/2024

- **Confusion Matrix:** Tabela que mostra o número de acertos das previsões em relação aos números reais.

Todos esses cálculos foram realizados pelas funções, que a biblioteca do Scikit-Learn disponibiliza, nas seguintes linhas de código:



FIGURA 12: Cálculo das métricas

```
1 accuracy_lr = lr.score(testX, testY)
2 accuracy_reg = np.mean(y_pred_reg_labels.ravel() == testY.values)
3 accuracy_LSVC = svc.score(testX, testY)
4 accuracy_nbb = nb.score(testX, testY)
5 accuracy_dt = dt.score(testX, testY)
6
7 recall_lr = recall_score(testY, y_pred_lr, average='macro')
8 recall_reg = recall_score(testY, y_pred_reg_labels, average='macro')
9 recall_lsvc = recall_score(testY, y_pred_svc, average='macro')
10 recall_nbb = recall_score(testY, y_pred_nb, average='macro')
11 recall_dt = recall_score(testY, y_pred_dt, average='macro')
12
13 f1_lr = f1_score(testY, y_pred_lr, average='macro')
14 f1_reg = f1_score(testY, y_pred_reg_labels, average='macro')
15 f1_lsvc = f1_score(testY, y_pred_svc, average='macro')
16 f1_nbb = f1_score(testY, y_pred_nb, average='macro')
17 f1_dt = f1_score(testY, y_pred_dt, average='macro')
18
19 precision_lr = precision_score(testY, y_pred_lr, average='macro')
20 precision_reg = precision_score(testY, y_pred_reg_labels, average='macro')
21 precision_lsvc = precision_score(testY, y_pred_svc, average='macro')
22 precision_nbb = precision_score(testY, y_pred_nb, average='macro')
23 precision_dt = precision_score(testY, y_pred_dt, average='macro')
24
```

FONTE: Figura do Autor.

FIGURA 13: Exemplo do Plot da *Confusion Matrix*.

```
1 CM_LR = confusion_matrix(testY, y_pred_lr)
2 sns.heatmap(CM_LR, annot=True, fmt=".0f", cmap="Blues")
3 plt.title('Matriz de Confusão - Regressão Logística')
4 plt.xlabel('Previsão')
5 plt.ylabel('Real')
6 plt.show()
```

FONTE: Figura do Autor.

#### 4.4. Resultados:

Após as métricas serem calculadas e definidas, foram obtidos os seguintes resultados:

TABELA 1: Tabela de resultados

Modelo	Acurácia (%)	F1 Score (%)	Precision Score (%)	Recall (%)	Tempo de Execução (s)
Regressão Logística	0.92	0.91	0.90	0.94	66,90
Regressão Linear	0.84	0.85	0.84	0.90	546,85
Linear SVM	0.91	0.91	0.89	0.93	787,51
<i>Bernoulli Naive Bayes</i>	0.84	0.85	0.83	0.90	2,51
<i>Decision Tree Classifier</i>	0.88	0.88	0.87	0.93	602,80

FONTE: Tabela do Autor.

➤ **Regressão Logística:**

Este modelo obteve a maior acurácia (92%) e um bom equilíbrio entre precisão (90%) e recall (94%). Ele apresentou um ótimo desempenho para classificar URLs com rapidez relativamente alta (66,9 segundos), sendo uma excelente escolha quando se trata de equilíbrio na eficiência e precisão.

➤ **Regressão Linear:**

Embora tenha sido obtido resultados bons, com acurácia de 84% e F1 Score de 85%, o tempo de execução foi significativamente alto (546,85 segundos). Isso indica que, para grandes conjuntos de dados, a aplicação desse método pode ser menos prática.

➤ **Linear SVM:**

Oferece uma acurácia quase equivalente à regressão logística (91%), mas com um custo computacional muito maior (787,51 segundos). Apesar disso, é uma escolha viável quando os recursos computacionais são robustos.

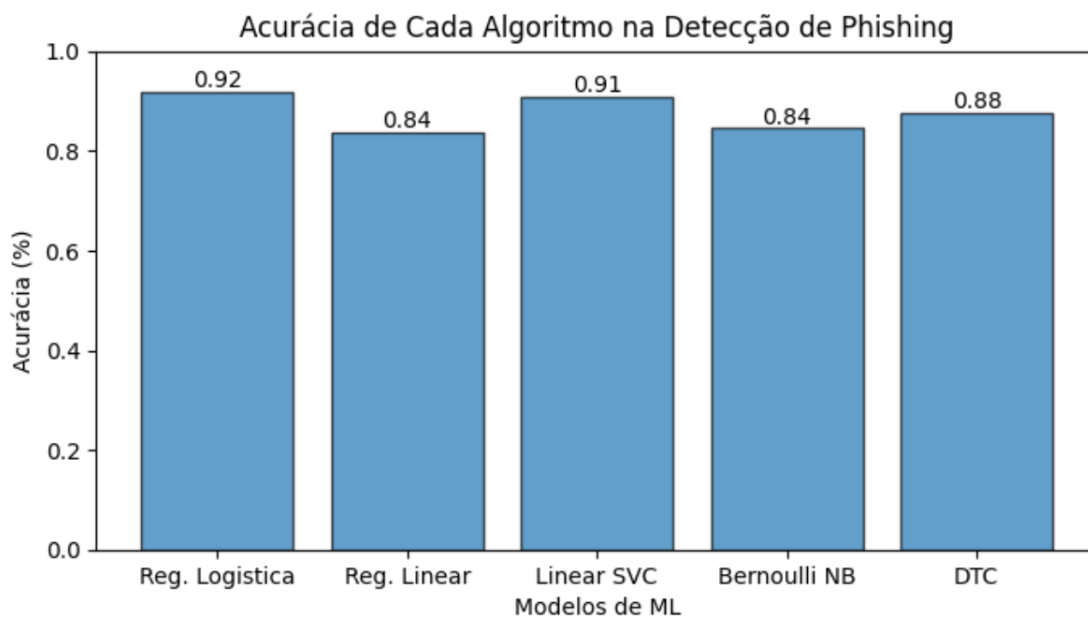
➤ ***Bernoulli Naive Bayes:***

Este modelo é o mais rápido de todos (2,51 segundos), com resultados modestos em termos de acurácia (84%) e precisão (83%). É ideal para cenários onde a velocidade é mais importante que a precisão absoluta.

➤ *Decision Tree Classifier:*

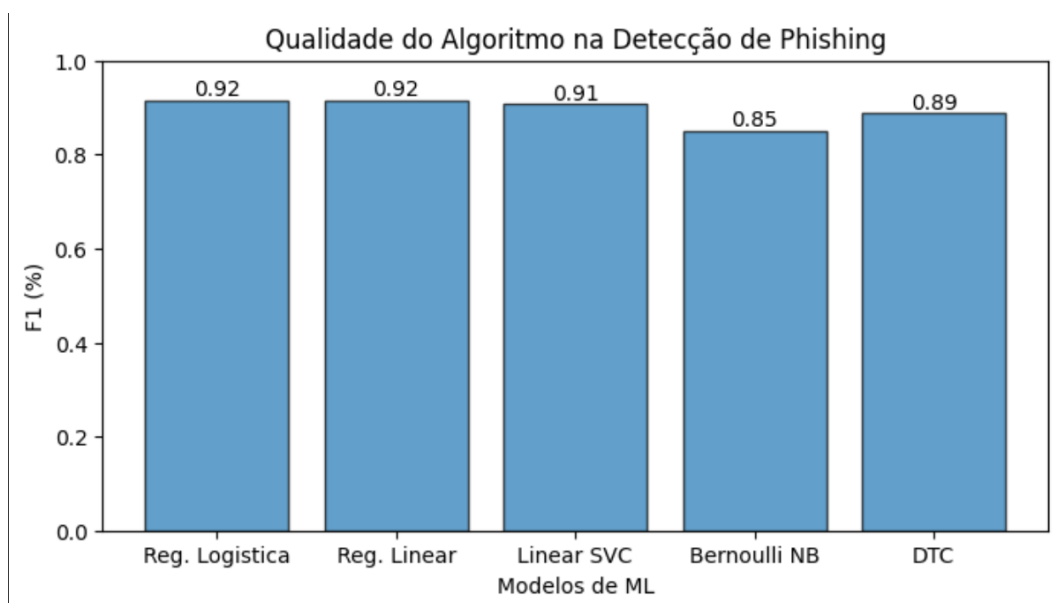
Obteve desempenho intermediário, com acurácia de 88% e tempo de execução alto (602,80 segundos). É uma boa opção quando interpretabilidade é um fator importante.

FIGURA 14: Gráfico de acurácia



FONTE: Figura do Autor.

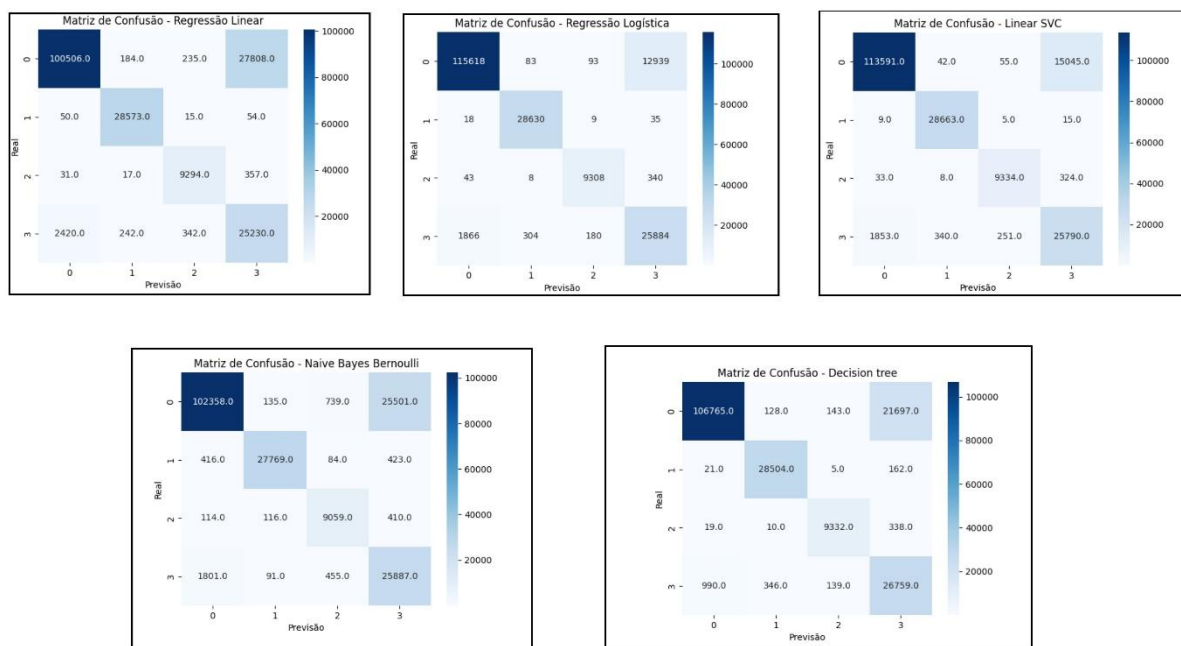
FIGURA 15: Gráfico da qualidade do algoritmo



FONTE: Figura do Autor.

Além desses dados, também foi obtido a Matriz de confusão de todos os modelos, apresentando resultados satisfatórios, mas foi apresentado uma pequena anomalia que pode ser visualizado pelos gráficos, segue abaixo:

FIGURA 16: Matrizes de Confusão



FONTE: Figura do Autor.

Esta pequena anomalia citada acima, pode ser visualizada em todas as matrizes de confusão, onde a previsão da classe de número três fez previsões relativas pela classe de número zero, apresentando um número elevado de erros nessa região da matriz previsão. No entanto, ainda sim apresentam números agradáveis de acerto para as previsões das outras classes.

Além desses pontos, também é válido trazer informações mais detalhadas sobre as diferenças de acurácia e precisão, tempo de execução e do desempenho do Recall. Abordando o tema de diferenças de acurácia e precisão, modelos como a Regressão Logística e o *Linear SVC* se destacam nesse ponto, pois são mais adequados para problemas binários com dados lineares ou quase lineares (ALPAYDIN, 2020). Por outro lado, o *Bernoulli Naive Bayes* teve menor precisão, possivelmente devido à sua suposição simplista de independência entre as

características, que nem sempre é válida para os dados de URLs (TAN; STEINBACH; KUMAR, 2018). Se tratando de tempo de execução, pode-se dizer que é bem notável as diferenças entre os modelos, porém, se destacam *Bernoulli Naive Bayes* e a Regressão Logística pois são muito mais rápidos uma vez que suas implementações são menos complexas (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Já o Linear SVC e o *Decision Tree Classifier* exigem maior poder computacional devido ao número de cálculos realizados durante o ajuste do modelo (SCI-KIT LEARN DEVELOPERS, 2024). Sobre desempenho do *Recall*, a Regressão Logística (94%) novamente foi o mais alto, indicando que ela é particularmente eficaz para identificar URLs phishing, reduzindo falsos negativos (ALPAYDIN, 2020). Já os modelos *Decision Tree Classifier* e Linear SVC também tiveram bons valores de recall (93%), pois naturalmente se destacam em identificar corretamente as classes problemáticas (MULLER; GUIDO, 2016).

## 5. CONSIDERAÇÕES FINAIS

### 5.1. Discussão sobre os Resultados

Os resultados obtidos demonstraram que todos os modelos analisados apresentaram desempenho satisfatório na detecção de ataques de *phishing*, cada qual com características específicas que os tornam mais adequados a diferentes cenários. Diferenças significativas foram observadas nas métricas de acurácia, precisão, recall e tempo de execução, o que reflete a diversidade de abordagens e a adequação dos modelos para diferentes condições operacionais.

Modelos como a Regressão Logística e o Linear SVC destacaram-se pela elevada acurácia e pela capacidade de identificar padrões de forma consistente em conjuntos de dados complexos. Esses modelos demonstraram desempenho especialmente eficaz em cenários onde a separação entre classes era bem definida. O Linear SVC, em particular, mostrou maior robustez em situações com dados de alta

dimensionalidade, confirmando sua eficiência em lidar com características complexas e variadas. Por outro lado, o *Bernoulli Naive Bayes* apresentou vantagens claras em termos de velocidade de processamento e simplicidade computacional. Esse modelo foi especialmente eficiente ao trabalhar com variáveis binárias, como a presença ou ausência de palavras-chave ou links suspeitos em e-mails. Sua capacidade de processar rapidamente grandes volumes de dados torna-o ideal para aplicações em tempo real, embora seu desempenho em acurácia tenha sido inferior aos modelos baseados em vetores de suporte.

Esses resultados evidenciam que não há uma solução única que seja ideal para todos os contextos. A escolha do modelo mais adequado dependerá das necessidades específicas da aplicação, como a priorização da velocidade em sistemas que operam em tempo real ou a necessidade de alta precisão em ambientes onde erros podem ser críticos.

## **5.2. Dificuldades Encontradas**

Entre os principais desafios enfrentados, destaca-se a necessidade de uma maior quantidade e diversidade de dados rotulados para o treinamento eficaz de modelos supervisionados. A obtenção de dados representativos é crucial para que os algoritmos sejam capazes de identificar padrões complexos e generalizar adequadamente para novos cenários. Porém, a rotulagem manual de grandes volumes de dados é um processo trabalhoso, caro e suscetível a erros, o que limita a disponibilidade de bases de dados de qualidade.

Além disso, ataques mais sofisticados, como o *Spear Phishing*, continuam a representar um obstáculo significativo. Esse tipo de ataque, caracterizado pela personalização de mensagens para enganar indivíduos ou organizações específicas, explora vulnerabilidades humanas e contextuais, tornando sua detecção mais desafiadora. Esses ataques frequentemente ultrapassam as capacidades de modelos tradicionais, exigindo o desenvolvimento de algoritmos mais robustos e adaptáveis, capazes de identificar nuances sutis em padrões comportamentais ou contextuais. Esses desafios destacam a importância de pesquisas contínuas e do desenvolvimento de tecnologias que combinem precisão, escalabilidade e flexibilidade.

## **5.3. Trabalhos Futuros**

Como contribuição, este trabalho apresenta uma abordagem prática e eficiente para o fortalecimento da segurança cibernética, utilizando técnicas de IA para identificar padrões maliciosos em tempo real. Contudo, enfatiza-se a importância de pesquisas futuras que explorem o uso de técnicas avançadas, como redes neurais profundas e algoritmos de ensemble, além da criação de bases de dados mais amplas e representativas.

## REFERÊNCIAS

- ALPAYDIN, Ethem. *Introduction to Machine Learning*. 4th ed. Cambridge: MIT Press, 2020.
- BUCZAK, A. L.; GUVEN, E. A survey of data mining and machine learning methods for cybersecurity intrusion detection. *IEEE Communications Surveys & Tutorials*, v. 18, n. 2, p. 1153-1176, 2016.
- BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, v. 2, p. 121-167, 2000.
- CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Computing Surveys*, v. 41, n. 3, p. 1-58, 2009.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer, 2009.
- JAKOBSSON, M.; MYERS, S. *Phishing and countermeasures: Understanding the increasing problem of electronic identity theft*. New York: Wiley, 2006.
- KAGGLE. Malicious URLs dataset. Disponível em: [www.kaggle.com/datasets/sid321axn/malicious-urls-dataset](https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset). Acesso em 15 ago. 2024.
- KASPERSKY LAB. Spam and phishing in 2018. Disponível em: <https://www.kaspersky.com>. Acesso em: 12 nov. 2024.
- MITCHELL, T. *Machine Learning*. New York: McGraw Hill, 2006.
- MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. *Introduction to Linear Regression Analysis*. 6. ed. Hoboken: John Wiley & Sons, 2021.
- MULLER, Andreas C.; GUIDO, Sarah. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. 1st ed. Sebastopol: O'Reilly Media, 2016.
- MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press, 2012.
- ORACLE. What is natural language processing?. Disponível em: <https://www.oracle.com/br/artificial-intelligence/what-is-natural-language-processing/>. Acesso em: 19 nov. 2024.
- SCHÖLKOPF, B.; SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge: MIT Press, 2002.



SCI-KIT LEARN DEVELOPERS. *Scikit-learn: Machine Learning in Python*. Disponível em: <https://scikit-learn.org>. Acesso em: 19 nov. 2024.

SOMMER, R.; PAXSON, V. Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy, p. 305-316, 2010.

STALLINGS, W. *Criptografia e segurança de redes: princípios e práticas*. 4. ed. São Paulo: Pearson Prentice Hall, 2011.

SYMANTEC. Internet security threat report. Disponível em: <https://docs.broadcom.com/doc/istr-24-executive-summary-en>. Acesso em: 12 nov. 2024.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. *Introduction to Data Mining*. 2nd ed. Harlow: Pearson, 2018.

ZHANG, H. The optimality of Naive Bayes. In: Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference. Miami Beach: AAAI Press, 2004.