

UNIVERSIDADE ESTADUAL PAULISTA – UNESP

Instituto de Biociências, Letras e Ciências Exatas – Campus de São José do Rio Preto

LUCAS PUERTA CRUZ

Detecção de Phishing por Aprendizado de Máquina com Ênfase em Métodos Ensemble

São José do Rio Preto

2025

Lucas Puerta Cruz

Detecção de Phishing por Aprendizado de Máquina com Ênfase em Métodos Ensemble

Dissertação, apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Computação Aplicada

Orientador: Prof^o Dr. Adriano Mauro Cansian
Coorientador:

São José do Rio Preto
2025

P977d

Puerta Cruz, Lucas

Detecção de Phishing por Aprendizado de Máquina com Ênfase em
Métodos Ensemble / Lucas Puerta Cruz. -- São José dos Campos,
2025

69 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),
Instituto de Ciência e Tecnologia, São José dos Campos

Orientadora: Adriano Mauro Cansian

1. Utilização de Machine Learning para segurança da Informação. I.
Título.

IMPACTO POTENCIAL DESTA PESQUISA

O impacto potencial desta pesquisa está na ampliação da eficácia dos sistemas de detecção de phishing por meio da aplicação de métodos de aprendizado de máquina baseados em *ensemble*. A combinação de algoritmos como *Random Forest*, *AdaBoost* e *XGBoost* permite reduzir a taxa de falsos negativos e aumentar a robustez das classificações em cenários reais, caracterizados por alto desequilíbrio e rápida evolução das ameaças. Dessa forma, o trabalho contribui diretamente para a segurança digital, oferecendo uma abordagem escalável e adaptável capaz de fortalecer mecanismos automatizados de proteção contra fraudes eletrônicas.

POTENTIAL IMPACT OF THIS RESEARCH

The potential impact of this research lies in improving the effectiveness of phishing detection systems through the application of ensemble-based machine learning methods. The combination of algorithms such as *Random Forest*, *AdaBoost*, and *XGBoost* reduces false negative rates and increases classification robustness in real-world scenarios characterized by high imbalance and rapidly evolving threats. Thus, this work directly contributes to digital security by offering a scalable and adaptive approach to strengthen automated mechanisms for protection against electronic fraud.

LUCAS PUERTA CRUZ

**DETECÇÃO DE PHISHING POR APRENDIZADO DE MÁQUINA COM ÊNFASE
EM MÉTODOS ENSEMBLE**

Dissertação apresentado(a) à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Computação Aplicada

Data de defesa: 25/08/2025

BANCA EXAMINADORA

Prof^o Dr. Adriano Mauro Cansian

UNESP – Instituto de Biociências, Letras e Ciências Exatas – Câmpus de São José do Rio Preto

Prof. Dr. Robson de Oliveira Albuquerque

Universidade de Brasília - UnB

Prof. Dr. Geraldo Pereira Rocha Filho

UNESP – Instituto de Biociências, Letras e Ciências Exatas – Câmpus de São José do Rio Preto

Este trabalho é dedicado aos incansáveis profissionais de defesa cibernética que, com ética, resiliência e esforços hercúleos, protegem o vasto ambiente informático, proporcionando um ecossistema digital seguro para todos nós.

AGRADECIMENTOS

Agradeço, de forma especial, aos meus pais, pelo zelo, cuidado e por terem sido os principais responsáveis pela minha formação pessoal e acadêmica, atuando como verdadeiros educadores da forma mais genuína e amorosa. Foram eles que me ensinaram os valores essenciais que moldaram meu caráter e me inspiraram na busca incessante pelo conhecimento e pela sabedoria.

À minha mãe, que partiu antes desta conquista, dedico este trabalho com todo o amor e gratidão. Sua força, dedicação e exemplo seguem sendo a base sobre a qual construo cada conquista e enfrento as adversidades. Este trabalho é, acima de tudo, por ela.

Gostaria de expressar meus sinceros agradecimentos ao Prof. Dr. Robson de Oliveira Albuquerque, cuja paciência, atenção aos detalhes e feedbacks criteriosos foram fundamentais nas etapas deste trabalho. Sua dedicação em fornecer orientações claras e construtivas teve grande impacto na consolidação da metodologia e na qualidade final desta dissertação.

Ao meu orientador, Prof. Dr. Adriano Mauro Cansian, registro minha profunda gratidão por sua orientação constante, apoio incondicional e pela confiança depositada ao longo de toda a jornada. Sua dedicação ao ensino e à pesquisa é uma inspiração permanente, e sou extremamente grato por todas as oportunidades de aprendizado e crescimento que me proporcionou.

Agradeço também ao Prof. Dr. Geraldo Zafalon, por suas valiosas contribuições e direcionamentos durante as fases do projeto, cujas observações ampliaram minha visão sobre o tema e fortaleceram os fundamentos da pesquisa.

Estendo meus agradecimentos aos professores Kelton Costa e Denis Salvadeo, cujas aulas instigantes e conselhos valiosos enriqueceram meu percurso acadêmico de maneiras inestimáveis.

Cada um dos senhores desempenhou um papel crucial na minha formação, e sou profundamente grato por todo o conhecimento compartilhado e pelo incentivo constante. Este trabalho é, em grande parte, um reflexo do compromisso, da generosidade e do profissionalismo de todos.

“Estudar não é um ato de consumir ideias, mas de criá-las e recriá-las.”

— Paulo Freire

RESUMO

Este trabalho investiga a aplicação de métodos de aprendizado de máquina do tipo *ensemble*, incluindo *Random Forest*, *AdaBoost* e *XGBoost*, na detecção de ataques de *phishing*. Foram utilizados três conjuntos de dados amplamente empregados na literatura (*Enron*, *ISCX-URL-2016* e *Phishing Websites — UCI*), que passaram por etapas de pré-processamento e foram adaptados para classificação binária, distinguindo entre instâncias legítimas e maliciosas. Os métodos *ensemble* foram comparados a algoritmos tradicionais, como Regressão Logística, *k-Nearest Neighbors*, *Naive Bayes* e *Decision Tree*, por meio de métricas de acurácia, precisão, *recall* e F1-score, com execuções repetidas de validação cruzada para garantir robustez estatística. Os resultados mostraram que os modelos *ensemble* obtiveram acurácia média de 96,9% e F1-score de 97%, superando os métodos convencionais, especialmente na redução de falsos negativos. Não foi desenvolvido um sistema completo em produção. O estudo evidencia o potencial dos métodos avaliados como base para implementações futuras, abrangendo engenharia de atributos, integração arquitetural e validação em ambientes reais.

Palavras-chave: phishing; aprendizado de máquina; métodos ensemble; random forest; adaboost; xgboost.

ABSTRACT

This work investigates the application of ensemble-type machine learning methods, including Random Forest, AdaBoost, and XGBoost, in the detection of phishing attacks. Three datasets widely used in the literature (Enron, ISCX-URL-2016, and Phishing Websites — UCI) were employed. They underwent preprocessing stages and were adapted for binary classification, distinguishing between legitimate and malicious instances. The ensemble methods were compared to traditional algorithms such as Logistic Regression, k-Nearest Neighbors, Naive Bayes, and Decision Tree using accuracy, precision, recall, and F1-score metrics, with repeated cross-validation executions to ensure statistical robustness. The results showed that the ensemble models achieved an average accuracy of 96.9% and an F1-score of 97%, outperforming conventional methods, especially in reducing false negatives. A complete production system was not developed. The study highlights the potential of the evaluated methods as a basis for future implementations, encompassing feature engineering, architectural integration, and validation in real environments.

Keywords: phishing; machine learning; ensemble methods; random forest; adaboost; xgboost.

LISTA DE FIGURAS

Figura 1	Funcionamento do SPF.....	8
Figura 2	Funcionamento do DMARC.....	8
Figura 3	Exemplo envolvendo a plataforma LastPass.	9
Figura 4	Exemplo de <i>spear phishing</i> simulando acesso ao Office 365.....	9
Figura 5	Exemplo envolvendo instituição financeira.	10
Figura 6	Exemplo relacionado a eventos.	10
Figura 7	Exemplo de <i>smishing</i> simulando confirmação de compra.....	11
Figura 8	Bagging e Random Forest.	13
Figura 9	Boosting: AdaBoost e XGBoost.	13
Figura 10	F1-score por iteração para cada modelo <i>ensemble</i> avaliado.	25
Figura 11	Acurácia por iteração para cada modelo <i>ensemble</i> avaliado.	26
Figura 12	Precisão por iteração para cada modelo <i>ensemble</i> avaliado.	26
Figura 13	Recall por iteração para cada modelo <i>ensemble</i> avaliado.	27
Figura 14	Variação do F1-score em função da seleção de atributos — Dataset <i>Phishing Websites</i> (UCI).	32
Figura 15	F1-score médio dos modelos em cada conjunto de dados	37
Figura 16	Comparação de desempenho entre modelos propostos e modelos da literatura — Dataset Enron.	39

LISTA DE TABELAS

Tabela 1	Elementos do protocolo SMTP.	6
Tabela 2	Elementos do protocolo POP3.....	7
Tabela 3	Elementos do protocolo IMAP.....	7
Tabela 4	Desempenho médio do Random Forest para diferentes valores de k na seleção de atributos.	18
Tabela 5	Variáveis removidas após aplicação do <i>SelectKBest</i> com $k = 10$	19
Tabela 6	Comparativo de desempenho entre uso de todas as variáveis e subconjunto selecionado ($k = 10$).	19
Tabela 7	Espaço de busca definido para o algoritmo <i>Random Forest</i>	20
Tabela 8	Top 15 configurações do <i>Random Forest</i> com melhor F1-score médio e métricas complementares.	21
Tabela 9	Melhor configuração de hiperparâmetros para o <i>Random Forest</i>	21
Tabela 10	Espaço de busca definido para o <i>AdaBoost</i>	22
Tabela 11	Top 15 configurações do <i>AdaBoost</i> com melhores métricas médias. ...	23
Tabela 12	Melhor configuração de hiperparâmetros para o <i>AdaBoost</i>	23
Tabela 13	Espaço de busca definido para o <i>XGBoost</i>	24
Tabela 14	Top 15 configurações do <i>XGBoost</i> com melhores métricas médias. .	24
Tabela 15	Melhor configuração de hiperparâmetros para o <i>XGBoost</i>	25
Tabela 16	Top 15 execuções com maior F1-score entre os modelos avaliados. .	28
Tabela 17	Desempenho médio e desvio padrão dos modelos <i>ensemble</i> após 25 execuções.	29
Tabela 18	Resultados dos modelos <i>ensemble</i> com 30 <i>features</i> — Dataset <i>Phishing Websites</i> (UCI).	31
Tabela 19	Resultados detalhados dos modelos <i>ensemble</i> com diferentes quantidades de atributos selecionados (<i>SelectKBest</i>) — Dataset <i>Phishing Websites</i> (UCI).	33
Tabela 20	Comparação de desempenho entre modelos tradicionais e os melhores modelos <i>ensemble</i> — Dataset <i>Phishing Websites</i> (UCI).....	34
Tabela 21	Desempenho médio dos modelos no dataset <i>ISCX-URL-2016</i> (sem seleção de atributos).	35

Tabela 22	Comparação da acurácia entre os modelos — <i>ISCX-URL-2016</i>	35
Tabela 23	Comparação da precisão entre os modelos — <i>ISCX-URL-2016</i>	35
Tabela 24	Comparação do <i>recall</i> entre os modelos — <i>ISCX-URL-2016</i>	36
Tabela 25	Comparação do F1-score entre os modelos — <i>ISCX-URL-2016</i>	36
Tabela 26	Comparação consolidada dos melhores modelos — Dataset <i>Phishing Websites</i> (UCI).....	37
Tabela 27	Desempenho de modelos propostos e tradicionais — Dataset Enron.	38
Tabela 28	Comparação de desempenho — Acurácia, Precisão e Recall no dataset <i>Phishing Websites</i> (UCI).	40
Tabela 29	Comparação complementar — F1-score no dataset <i>Phishing Websites</i> (UCI).	41
Tabela 30	Atributos extraídos do corpus Enron.	44
Tabela 31	Atributos do dataset UCI <i>Phishing Websites</i>	45
Tabela 32	Atributos do dataset <i>ISCX-URL-2016</i>	46
Tabela 33	Configurações finais dos hiperparâmetros por algoritmo.	48
Tabela 34	Erros médios por algoritmo nas execuções de validação cruzada.....	49

LISTA DE ABREVIATURAS

AI – *Artificial Intelligence*

CNN – *Convolutional Neural Network*

FN – *False Negative*

FP – *False Positive*

KNN – *k-Nearest Neighbors*

ML – *Machine Learning*

RF – *Random Forest*

SVM – *Support Vector Machine*

TP – *True Positive*

TN – *True Negative*

UCI – *University of California Irvine*

URL – *Uniform Resource Locator*

XGBoost – *Extreme Gradient Boosting*

SUMÁRIO

1. INTRODUÇÃO	1
1.1 MOTIVAÇÃO.....	1
1.2 JUSTIFICATIVA	2
1.3 OBJETIVOS.....	3
1.4 OBJETIVOS ESPECÍFICOS	3
1.5 ORGANIZAÇÃO DO TRABALHO.....	3
2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA.....	5
2.1 ARQUITETURA E COMPONENTES DO E-MAIL	5
2.2 PROTOCOLOS DE CORREIO ELETRÔNICO.....	6
2.2.1 Simple mail transfer protocol (smtp)	6
2.2.2 Post office protocol (pop3)	6
2.2.3 Internet message access protocol (imap)	6
2.2.4 Mail exchange (mx)	7
2.2.5 Segurança e autenticação no e-mail.....	7
2.2.6 Sender policy framework (spf)	7
2.2.7 Domainkeys identified mail (dkim)	8
2.2.8 Domain-based message authentication, reporting and conformance (dmarc)	8
2.3 PHISHING	9
2.3.1 Spear phishing.....	9
2.3.2 Whaling	10
2.3.3 Smishing.....	10
2.3.4 Vishing.....	11
2.4 APRENDIZADO DE MÁQUINA.....	11
2.4.1 Tipos de aprendizado	11
2.4.2 Detecção automatizada de phishing	12
2.5 MÉTODOS DE ENSEMBLE	12
2.5.1 Bagging e random forest.....	12
2.5.2 Boosting: adaboost e xgboost.....	12
3 METODOLOGIA.....	15
3.1 FONTES DE DADOS.....	15
3.2 PRÉ-PROCESSAMENTO	15
3.3 SELEÇÃO DE ATRIBUTOS.....	16
3.4 MODELOS E CONFIGURAÇÃO DOS ALGORITMOS	16
3.5 VALIDAÇÃO E MÉTRICAS DE DESEMPENHO.....	16
4 ANÁLISE DOS RESULTADOS EXPERIMENTAIS.....	17
4.1 RESULTADOS INICIAIS COM MODELOS ENSEMBLE	18
4.1.1 Seleção de atributos com random forest	18
4.1.2 Comparação final da seleção de atributos com random forest.....	19
4.2 OTIMIZAÇÃO DE HIPERPARÂMETROS.....	19
4.2.1 Ajuste de hiperparâmetros: random forest.....	20
4.2.2 Ajuste de hiperparâmetros: adaboost	22
4.2.3 Ajuste de hiperparâmetros: xgboost.....	23
4.3 ESTABILIDADE ESTATÍSTICA AO LONGO DAS EXECUÇÕES.....	25
4.3.1 Estabilidade ao longo das execuções.....	25
4.3.2 Comparação final dos modelos ensemble.....	28
4.4 AVALIAÇÃO NO DATASET UCI MACHINE LEARNING REPOSITORY	30
4.4.1 Avaliação Inicial com Todas as Features.....	31
4.4.2 Avaliação com Seleção de Atributos (SelectKBest).....	31

4.4.3	<i>Comparação com Algoritmos Tradicionais</i>	33
4.5	DATASET ISCX-URL-2016	34
4.5.1	<i>Resultados com Ensemble</i>	34
4.5.2	<i>Análise com todas as métricas</i>	35
4.6	DISCUSSÃO E CONSOLIDAÇÃO FINAL	36
4.7	ANÁLISE COMPARATIVA COM A LITERATURA	38
4.7.1	<i>Comparações no dataset enron</i>	38
4.7.2	<i>Comparações no dataset phishing websites (uci)</i>	39
5	CONCLUSÃO E TRABALHOS FUTUROS	42
.1	ATRIBUTOS DOS CONJUNTOS DE DADOS	44
	<i>CORPUS ENRON</i>	44
	<i>PHISHING WEBSITES (UCI)</i>	44
	<i>ISCX-URL-2016</i>	45
.2	PARÂMETROS DOS ALGORITMOS	48
.3	MÉTRICAS DE ERROS E PERDAS	49
	<i>DEFINIÇÕES</i>	49
	<i>TABELA MODELO DE ERROS</i>	49
	REFERÊNCIAS	50
	DADOS CURRICULARES	52

1. INTRODUÇÃO

Os ataques de *phishing* representam uma das ameaças mais recorrentes e eficazes no contexto da segurança da informação. Por meio de técnicas de engenharia social e falsificação de mensagens eletrônicas, esses ataques têm como objetivo enganar usuários e induzi-los a fornecer dados sensíveis, como credenciais de acesso, informações bancárias ou dados pessoais. A evolução dessas fraudes, aliada à capacidade de adaptação constante dos agentes maliciosos, torna o combate ao *phishing* um desafio contínuo para pesquisadores e profissionais da área.

Diversos estudos têm buscado aplicar técnicas de aprendizado de máquina (*Machine Learning* — ML) para aprimorar os mecanismos de detecção automatizada de *phishing*. Modelos de classificação supervisionada vêm demonstrando resultados promissores, especialmente quando combinados a abordagens de *ensemble learning*, capazes de integrar múltiplos classificadores para reduzir erros e aumentar a robustez da detecção.

Este trabalho tem como objetivo avaliar a eficácia de três métodos de aprendizado de máquina baseados em *ensemble Random Forest*, *AdaBoost* e *XGBoost* na identificação de e-mails e URLs maliciosos. Para isso, são utilizados três conjuntos de dados amplamente empregados na literatura (Enron, ISCX-URL-2016 e Phishing Websites — UCI), os quais passaram por processos de pré-processamento e adaptação para classificação binária.

A pesquisa busca responder à seguinte questão principal: quais métodos de aprendizado de máquina apresentam melhor desempenho na detecção de *phishing* e maior capacidade de generalização entre diferentes bases de dados. A investigação envolve a comparação entre algoritmos *ensemble* e métodos tradicionais (como Regressão Logística, *k*-Nearest Neighbors, Naive Bayes e Decision Tree), com foco na análise da acurácia e da taxa de falsos negativos, considerando múltiplas execuções de validação cruzada para garantir significância estatística.

Não se trata de um sistema completo em produção, mas de um estudo experimental voltado à avaliação e comparação de modelos de aprendizado supervisionado. A análise dos resultados obtidos fornece subsídios para futuras implementações práticas de sistemas automatizados de detecção de *phishing*, contemplando aspectos de engenharia de atributos, integração arquitetural e validação em ambientes reais.

1.1 MOTIVAÇÃO

A crescente sofisticação dos ataques de *phishing* representa um risco significativo para usuários e organizações, pois compromete diretamente a integridade, a confidencialidade e a disponibilidade das informações. No contexto corporativo, o e-mail é um canal essencial para comunicações internas e externas, o que o torna particularmente vulnerável a campanhas de engenharia social [Oliveira e Silva 2021].

Embora existam filtros antispam comerciais, esses sistemas muitas vezes não conseguem distinguir mensagens legítimas de ataques elaborados, sobretudo quando os conteúdos reproduzem padrões de comunicação autênticos. Além disso, parte desses ataques é direcionada a indivíduos específicos, utilizando técnicas conhecidas como *spear phishing*, o que torna sua detecção ainda mais desafiadora para sistemas genéricos

[Abdelhamid, Ayeshe e Thabtah 2014].

Diante desse cenário, há uma demanda crescente por soluções baseadas em inteligência artificial capazes de identificar automaticamente padrões complexos de ataque, com alta taxa de acerto e baixo índice de falsos positivos. Essa necessidade prática constitui a principal motivação para o desenvolvimento deste estudo.

1.2 JUSTIFICATIVA

A escolha por técnicas de aprendizado supervisionado com métodos *ensemble* justifica-se pela reconhecida eficácia desses algoritmos em tarefas de classificação, especialmente em contextos com múltiplos atributos e dados rotulados [Freund e Schapire 1997, Breiman 2001]. Os modelos *ensemble* combinam o desempenho de classificadores fracos para formar sistemas mais robustos e com melhor capacidade de generalização.

Além disso, as abordagens selecionadas (Random Forest, AdaBoost e XGBoost) têm apresentado resultados expressivos na literatura quando aplicadas à detecção de spam e *phishing*, conforme demonstram estudos recentes [Sahingoz et al. 2019, Verma e Das 2020]. A possibilidade de ajustar hiperparâmetros e avaliar o desempenho dos modelos em diferentes bases de dados reforça a relevância da escolha metodológica.

O uso de bases públicas, como Enron, Phishing Websites (UCI) e ISCX-URL-2016, contribui para a reprodutibilidade dos experimentos e possibilita comparações com trabalhos correlatos, fortalecendo a validade externa dos resultados. A utilização de ferramentas amplamente difundidas, como Python, Pandas, Scikit-Learn e XGBoost, torna a implementação viável em ambientes reais, mantendo baixo custo computacional e alta aplicabilidade prática.

1.3 OBJETIVOS

O objetivo central deste trabalho é avaliar o desempenho de métodos de aprendizado de máquina do tipo *ensemble* (Random Forest, AdaBoost e XGBoost) aplicados à detecção de *phishing* em diferentes vetores, como e-mails, URLs e websites.

Mais especificamente, busca-se verificar se esses algoritmos apresentam ganhos consistentes em métricas críticas, como F1-score e *recall*, quando comparados a modelos tradicionais, analisando seu potencial como base para o desenvolvimento futuro de sistemas automatizados de detecção de *phishing* em ambientes corporativos.

1.4 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Realizar o pré-processamento dos conjuntos de dados selecionados, adaptando-os à tarefa de classificação binária entre e-mails ou URLs de *phishing* e legítimos.
- Avaliar diferentes algoritmos de aprendizado de máquina do tipo *ensemble*, com ênfase em Random Forest, AdaBoost e XGBoost, em cenários distintos de detecção de *phishing*.
- Comparar o desempenho dos métodos *ensemble* com algoritmos tradicionais, incluindo Regressão Logística, *k*-Nearest Neighbors, Naive Bayes e Decision Tree.
- Aplicar validação cruzada em múltiplas execuções, visando aumentar a robustez estatística dos resultados e reduzir a variabilidade decorrente do processo de amostragem.
- Analisar os resultados obtidos a partir das métricas de acurácia, precisão, *recall* e F1-score, destacando a importância da redução de falsos negativos no contexto da segurança computacional.
- Identificar as limitações do estudo e propor perspectivas para trabalhos futuros, incluindo engenharia de atributos, ajustes arquiteturais e avaliações em ambientes reais.

1.5 ORGANIZAÇÃO DO TRABALHO

O Capítulo 1 apresenta a introdução ao tema, contextualizando o problema da detecção de ataques de *phishing* no cenário atual da cibersegurança. São discutidas a relevância técnica e social do problema, a motivação da pesquisa e os objetivos geral e específicos do estudo.

O Capítulo 2 aborda a fundamentação teórica necessária para o desenvolvimento do trabalho. São apresentados os conceitos de arquitetura de e-mail, protocolos e mecanismos

de segurança, os principais tipos de ataques de *phishing*, os fundamentos do aprendizado de máquina com foco em métodos *ensemble* e uma revisão dos trabalhos relacionados. Essa revisão permite identificar lacunas na literatura e justificar a escolha metodológica adotada.

O Capítulo 3 descreve a metodologia empregada, detalhando os procedimentos experimentais. São apresentados os conjuntos de dados utilizados, as etapas de pré-processamento, a seleção estatística de atributos, os parâmetros de configuração dos modelos, a validação cruzada e as métricas adotadas para avaliação de desempenho dos classificadores.

O Capítulo 4 expõe os resultados obtidos. Inicialmente, são analisados os efeitos da otimização de hiperparâmetros e da seleção de atributos sobre o desempenho dos modelos. Em seguida, apresentam-se os resultados da detecção de e-mails maliciosos utilizando os datasets Enron e Phishing Websites, e da detecção de URLs maliciosas com o ISCX-URL-2016. O capítulo inclui ainda comparações com algoritmos tradicionais documentados na literatura.

Por fim, o Capítulo 5 apresenta as conclusões do trabalho, discutindo os principais achados, as contribuições práticas e acadêmicas e as limitações enfrentadas. Também são indicadas direções para pesquisas futuras, com foco em melhorias e extensões da abordagem proposta.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA

Este capítulo apresenta os conceitos fundamentais para a compreensão do problema de detecção de *phishing*, bem como a revisão da literatura relacionada. Inicialmente, descrevem-se elementos técnicos que contextualizam o ambiente de correio eletrônico e os componentes característicos de mensagens de *phishing*. Em seguida, discutem-se conceitos de aprendizado de máquina e métodos do tipo *ensemble*, com ênfase nos algoritmos Random Forest, AdaBoost e XGBoost. Por fim, apresentam-se trabalhos relacionados que exploram diferentes abordagens de detecção de *phishing*.

2.1 ARQUITETURA E COMPONENTES DO E-MAIL

O correio eletrônico é um dos serviços centrais da internet e viabiliza comunicação eficiente, escalável e assíncrona. A arquitetura típica envolve servidores especializados, registros de roteamento e protocolos de comunicação que operam de forma integrada para garantir a entrega das mensagens.

A infraestrutura baseia-se em três elementos principais: servidores de correio, registros *Mail Exchange* (MX) e protocolos responsáveis pelo tráfego de mensagens. Os servidores de correio dividem-se em servidores de envio, que utilizam SMTP (*Simple Mail Transfer Protocol*), e servidores de recebimento, que operam com POP3 (*Post Office Protocol version 3*) e IMAP (*Internet Message Access Protocol*).

Os registros MX são configurados no DNS e direcionam e-mails recebidos para o servidor apropriado, com base no domínio do destinatário. Ao enviar uma mensagem, o servidor SMTP do remetente consulta o DNS para localizar o registro MX do domínio de destino e, a partir disso, roteia a mensagem ao servidor correspondente.

2.2 PROTOCOLOS DE CORREIO ELETRÔNICO

Após a identificação do servidor de destino pelos registros MX, os protocolos de correio eletrônico executam a transferência e o gerenciamento das mensagens.

2.2.1 Simple mail transfer protocol (smtp)

O SMTP é o protocolo padrão para a transferência de mensagens entre servidores e para o envio de mensagens de clientes a servidores. Opera, tipicamente, nas portas TCP 25, 587 (com STARTTLS) e 465 (SSL/TLS). Baseia-se em um modelo de comandos e respostas para conduzir o envio e o recebimento.

Tabela 1: Elementos do protocolo SMTP.

Componente	Valor/Descrição
Portas TCP usuais	25; 587 (STARTTLS); 465 (SSL/TLS)
Comandos básicos	HELO, EHLO, MAIL FROM, RCPT TO, DATA, QUIT
Proteção do canal	TLS/SSL
Mecanismos de autenticação do domínio	SPF, DKIM, DMARC

Fonte: Elaborado pelo autor.

O funcionamento típico envolve: conexão do cliente ao servidor SMTP; troca de comandos e respostas; consulta ao DNS para localização do registro MX; encaminhamento e armazenamento da mensagem no servidor de destino, tornando-a disponível por POP3 ou IMAP.

2.2.2 Post office protocol (pop3)

O POP3, descrito na RFC 1939, permite o recebimento de mensagens do servidor para o cliente local. Historicamente, as mensagens eram removidas do servidor após o download, embora configurações atuais permitam a manutenção de cópias. Opera nas portas TCP 110 e 995 (SSL/TLS).

2.2.3 Internet message access protocol (imap)

O IMAP, descrito na RFC 3501, mantém as mensagens no servidor e possibilita sincronização entre dispositivos. Opera nas portas TCP 143 e 993 (SSL/TLS), com funcionalidades de gerenciamento de pastas e buscas avançadas.

Tabela 2: Elementos do protocolo POP3.

Componente	Valor/Descrição
Portas TCP usuais	110; 995 (SSL/TLS)
Comandos básicos	USER, PASS, LIST, RETR, DELE, QUIT
Proteção do canal	SSL/TLS
Mecanismos de autenticação	APOP, SASL

Fonte: Elaborado pelo autor.

Tabela 3: Elementos do protocolo IMAP.

Componente	Valor/Descrição
Portas TCP usuais	143; 993 (SSL/TLS)
Funcionalidades	Pastas, buscas por critérios, acesso parcial a mensagens
Proteção do canal	SSL/TLS

Fonte: Elaborado pelo autor.

2.2.4 Mail exchange (mx)

Os registros MX no DNS determinam a prioridade de servidores responsáveis por receber mensagens destinadas a determinado domínio. A presença de múltiplos registros com diferentes preferências aumenta a disponibilidade e a resiliência do serviço.

2.2.5 Segurança e autenticação no e-mail

A proteção do canal de comunicação utiliza SSL/TLS. Mecanismos de autenticação de remetente, como SPF, DKIM e DMARC, auxiliam na validação de legitimidade e na mitigação de falsificação de domínio, contribuindo para a redução de spam e *phishing*.

2.2.6 Sender policy framework (spf)

O SPF é um mecanismo de autenticação baseado na publicação, no DNS, de endereços IP autorizados a enviar e-mails em nome de um domínio. O servidor destinatário consulta o registro correspondente e decide pela aceitação, rejeição ou marcação da mensagem como suspeita.

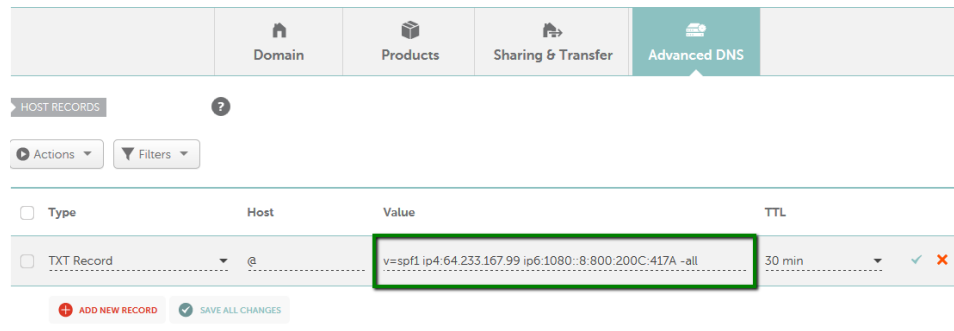


Figura 1: Funcionamento do SPF.

Fonte: Elaborado pelo autor.

2.2.7 Domainkeys identified mail (dkim)

O DKIM utiliza criptografia assimétrica para assinar digitalmente o cabeçalho da mensagem. A chave pública é publicada no DNS do domínio remetente. O servidor destinatário valida a assinatura e verifica a integridade e a autenticidade do conteúdo.

2.2.8 Domain-based message authentication, reporting and conformance (dmarc)

O DMARC define políticas de tratamento de mensagens que falham nas validações SPF e DKIM. A política é publicada no DNS do domínio e pode instruir o destinatário a monitorar, colocar em quarentena ou rejeitar mensagens, além de gerar relatórios agregados.

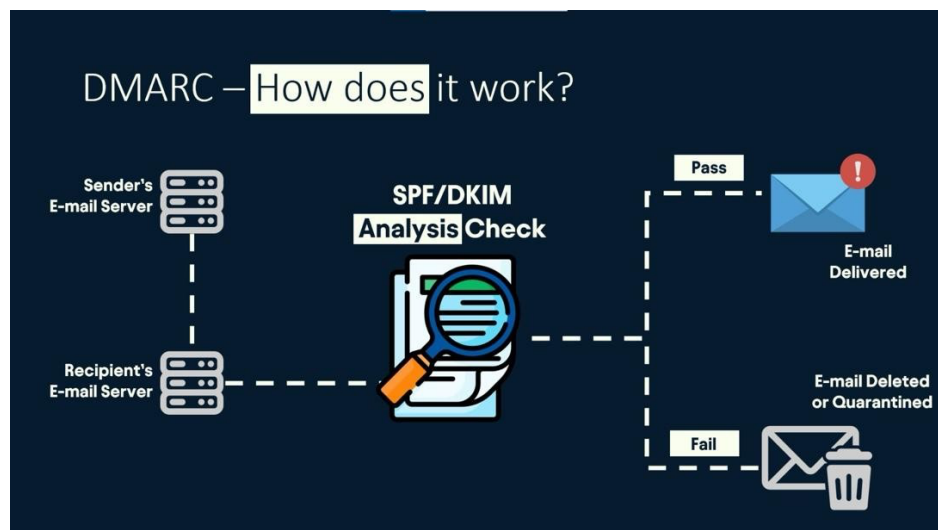


Figura 2: Funcionamento do DMARC.

Fonte: Elaborado pelo autor.

2.3 PHISHING

O *phishing* consiste no envio de mensagens que simulam comunicações de fontes confiáveis, com o objetivo de induzir o destinatário a acessar links maliciosos ou fornecer informações sensíveis. É frequente a imitação de instituições financeiras e serviços de pagamento. As Figuras 3 e 4 ilustram exemplos.

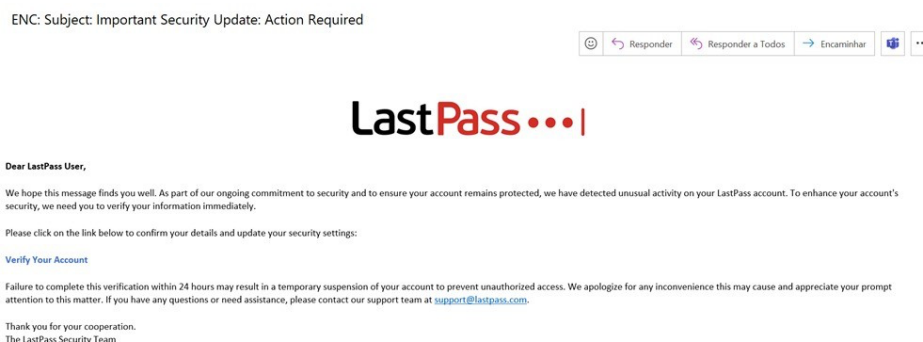


Figura 3: Exemplo envolvendo a plataforma LastPass.

Fonte: Elaborado pelo autor.

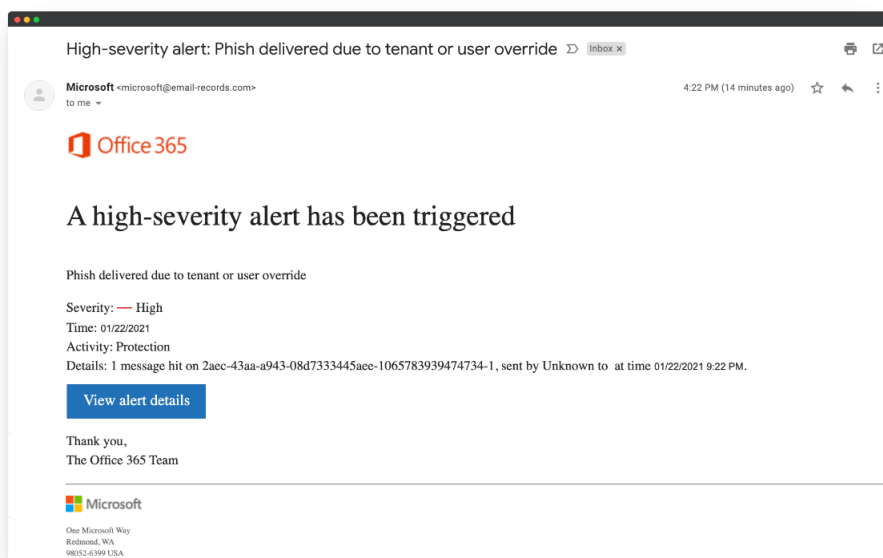


Figura 4: Exemplo de *spear phishing* simulando acesso ao Office 365.

Fonte: Elaborado pelo autor.

2.3.1 Spear phishing

O *spear phishing* é direcionado e utiliza informações sobre a vítima para construir mensagens personalizadas e persuasivas. Em geral, os atacantes procuram obter credenciais ou executar transferências não autorizadas.

O seu endereço IP foi rastreado para uma violação recente em um Banco Nacional, o que resultou em roubo de fundo de aposentadoria de idosos.

Como uma organização responsável a fim de apoiar a investigação, a ABInbev concordou em fornecer logs (registros) relevantes para a Agência governamental e tivemos acesso a uma ferramenta que nos ajudará a fazer a varredura do sistema interno e rastrear o legítimo endereço IP que usou sua conta como "isca".

[Clique aqui](#) para baixar a ferramenta disponibilizada que fará a varredura e o upload dos logs necessários para investigação. Uma cópia dos logs salvos em sua área de trabalho deverá ser enviada para o suporte de TI que fará a análise interna, a fim de facilitar e economizar tempo na investigação.

Solicitamos sua cooperação. Sua segurança é nossa motivação!

Atenciosamente,

Figura 5: Exemplo envolvendo instituição financeira.

Fonte: Elaborado pelo autor.

Prezado(a)

O próximo evento TEDx está programado para o dia 28 de fevereiro de 2024. Haverá uma interessante programação de palestrantes motivacionais e gerentes de diversas indústrias.

Esta é a sua oportunidade de participar e ter a chance de ser parte da audiência desse evento.

Como parte das promoções do evento, seu nome foi selecionado para receber um ingresso gratuito para participar deste evento com lugares limitados ou para se juntar à transmissão on-line.

[Clique aqui](#) para autenticar e fornecer o Código de Funcionário (SAP) e o nome do Departamento para obter seu ingresso gratuito.

Obrigado(a),

Figura 6: Exemplo relacionado a eventos.

Fonte: Elaborado pelo autor.

2.3.2 Whaling

O *whaling* é uma variação do *spear phishing* que tem como alvo executivos e pessoas com acesso a informações estratégicas. Mensagens costumam simular solicitações urgentes e instruções de pagamento.

2.3.3 Smishing

O *smishing* utiliza SMS como vetor e induz o usuário a acessar páginas fraudulentas ou fornecer dados. É comum a simulação de prêmios e confirmações de compra.

Compra realizada em CASAS
Bahia no valor de R\$:3.299,99.
Para autorizar resposta sim.
Caso não reconheça a compra,
entre em contato com a central
de atendimento: 0800 322 0058

Figura 7: Exemplo de *smishing* simulando confirmação de compra.

Fonte: Elaborado pelo autor.

2.3.4 Vishing

O *vishing* ocorre por telefone, quando agentes mal-intencionados se passam por representantes legítimos de instituições, solicitando informações sensíveis. Em muitos casos, utilizam dados prévios da vítima para aumentar a credibilidade da abordagem.

2.4 APRENDIZADO DE MÁQUINA

O aprendizado de máquina (*machine learning* - ML) é uma subárea da inteligência artificial (IA) dedicada ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados para realizar previsões ou decisões sem programação explícita. O termo foi empregado por Arthur Samuel em 1959 ao descrever um programa de jogo de damas que melhorava com a experiência [Samuel 1959]. Desde então, o campo evoluiu e passou a abranger aplicações em diferentes domínios, incluindo segurança cibernética.

2.4.1 Tipos de aprendizado

O aprendizado supervisionado utiliza dados rotulados, isto é, pares de entrada e saída desejada. Em classificação de e-mails, por exemplo, o modelo é treinado com mensagens marcadas como *spam* ou *não spam*. Exemplos de algoritmos incluem regressão logística e redes neurais, entre outros [Mitchell 1997].

O aprendizado não supervisionado emprega dados sem rótulos e tem como objetivo identificar estruturas subjacentes, padrões e agrupamentos. Técnicas como *k*-means e análise

de componentes principais (PCA) são frequentemente utilizadas, essa abordagem é útil para segmentação e redução de dimensionalidade [Dietterich 2000].

O aprendizado por reforço envolve tomada de decisão sequencial mediada por recompensas interagindo com o ambiente, observando resultados e ajustando sua política para maximizar o retorno acumulado.

2.4.2 Detecção automatizada de phishing

A detecção automatizada de *phishing* evoluiu de mecanismos baseados em regras para abordagens orientadas a dados. Em vez de depender de padrões fixos, analisam-se atributos sintáticos, semânticos e contextuais, tais como estrutura de cabeçalho, reputação de domínio, presença de links ofuscados e conteúdo textual. Técnicas supervisionadas mostraram-se eficazes para classificar mensagens maliciosas, inclusive quando apresentam padrões sutis ou não lineares.

2.5 MÉTODOS DE ENSEMBLE

Métodos *ensemble* combinam múltiplos modelos para melhorar desempenho preditivo e robustez. A agregação de modelos reduz variância e auxilia na mitigação de sobreajuste, o que é relevante em cenários com ruído, desbalanceamento e alta variabilidade, como em dados de *phishing* [Dietterich 2000].

2.5.1 Bagging e random forest

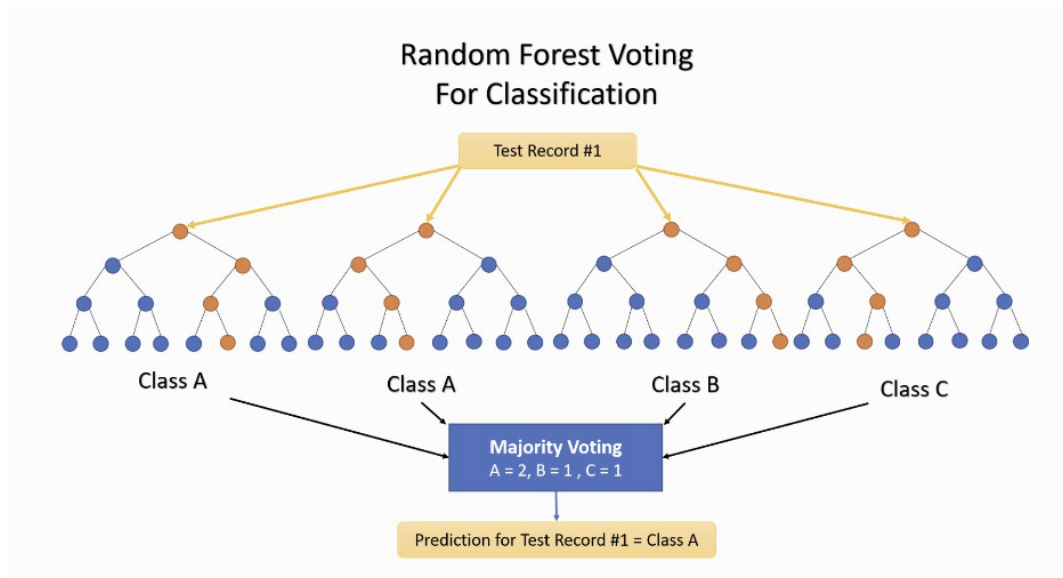
O *bagging* (*bootstrap aggregating*) cria amostras com reposição a partir do conjunto de dados original e treina um modelo em cada amostra. As previsões são combinadas por maioria de votos ou média.

O Random Forest [Breiman 2001] introduz aleatoriedade na seleção de atributos a cada divisão e na amostragem dos dados. A técnica tende a reduzir variância sem aumento relevante de viés e é eficaz com grande número de atributos, presença de valores extremos e classes desbalanceadas.

2.5.2 Boosting: adaboost e xgboost

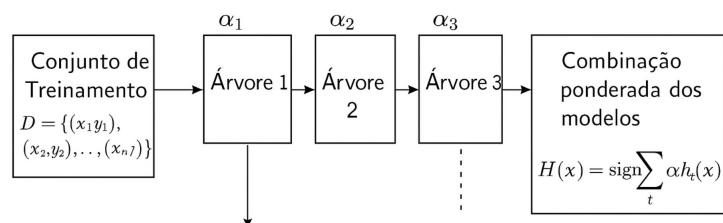
O *boosting* treina modelos de forma sequencial, em que cada modelo posterior enfatiza instâncias mal classificadas pelos anteriores. O AdaBoost, proposto por [Freund e Schapire 1997], utiliza pesos adaptativos nas instâncias e combina classificadores fracos por ponderação. O XGBoost, apresentado por Chen e Guestrin [Chen e Guestrin 2016], implementa *gradient boosting* com regularização e otimizações para velocidade e desempenho.

Figura 8: Bagging e Random Forest.



Fonte: Adaptado de [Breiman 2001].

Figura 9: Boosting: AdaBoost e XGBoost.



Fonte: Elaborado pelo autor.

2.6 VANTAGENS DOS MÉTODOS ENSEMBLE NA DETECÇÃO DE PHISHING

Estudos indicam que métodos *ensemble* superam classificadores individuais em tarefas de segurança cibernética, incluindo a detecção de *phishing* [Abdelhamid, Ayesh e Thabtah 2014, Sahingoz et al. 2019]. Entre as vantagens, destacam-se adaptação a variações de padrão, resiliência a ruído e redução de falsos negativos, aspecto crítico em sistemas de detecção.

2.7 TRABALHOS RELACIONADOS

Trabalhos anteriores, como o de Ocampo [Ocampo 2023], avaliaram algoritmos individuais em um único conjunto de dados. O presente estudo amplia essa análise para múltiplos conjuntos heterogêneos (Enron, ISCX-URL-2016 e UCI Phishing Websites), o que favorece a avaliação da robustez dos modelos.

Pesquisas como as de Abdelhamid et al. [Abdelhamid, Ayesh e Thabtah 2015] e Sahingoz et al. [Sahingoz et al. 2019] indicaram o potencial de métodos *ensemble*, embora sem explorar de forma sistemática o impacto da seleção de atributos e da otimização de hiperparâmetros.

A contribuição deste trabalho consiste em uma avaliação comparativa com validação cruzada repetida, explicitando a capacidade dos *ensembles* de reduzir falsos negativos, métrica relevante em segurança, e consolidando uma base metodológica para aplicações futuras em ambientes reais.

Abordagens recentes baseadas em *Transformers* e redes neurais convolucionais (CNNs) foram aplicadas à detecção de *phishing*, como em [Zhang, Liu e Wang 2023] e [Song, Kim e Lee 2023]. Tais métodos tendem a demandar maior poder computacional e, frequentemente, utilizam validações restritas, o que pode impactar a generalização dos resultados.

Além dos estudos mencionados, Basnet, Sung e Liu [Basnet, Sung e Liu 2014] exploraram classificadores tradicionais, como Random Forest e SVM, para detecção de URLs maliciosas com atributos léxicos. Buber, Sahingoz e Diri [Buber, Sahingoz e Diri 2017] avaliaram técnicas estatísticas de seleção de atributos, como ReliefF e ANOVA, no contexto de websites de phishing, reforçando a relevância da redução de dimensionalidade. Em perspectiva operacional, Marchal et al. [Marchal et al. 2016] propuseram o sistema PhishStorm, que identifica ataques em tempo quase real a partir de análises de fluxo.

3 METODOLOGIA

Esta seção descreve as etapas metodológicas adotadas para a avaliação dos modelos de aprendizado de máquina aplicados à detecção de *phishing*. O processo compreende a seleção das bases de dados, o pré-processamento dos atributos, a escolha e configuração dos algoritmos, a estratégia de validação e as métricas utilizadas para avaliação de desempenho.

3.1 FONTES DE DADOS

Foram utilizadas três bases amplamente referenciadas na literatura: (i) Enron, composta por mensagens de correio eletrônico legítimas; (ii) ISCX-URL-2016, contendo amostras de URLs benignas e maliciosas; e (iii) Phishing Websites - UCI, com características extraídas de sites legítimos e fraudulentos.

Cada conjunto passou por adaptações e filtrações para permitir a classificação binária entre instâncias legítimas e de *phishing*. Os parâmetros descritivos de cada base, quantidade de instâncias, número de atributos, tipos de variáveis e proporção entre classes estão apresentados no Apêndice A.

3.2 PRÉ-PROCESSAMENTO

As etapas de pré-processamento compreenderam:

- remoção de atributos redundantes e instâncias inconsistentes;
- normalização dos valores numéricos pelo método *Min-Max*;
- codificação de variáveis categóricas por *label encoding*;
- padronização dos rótulos de saída para os valores -1 (*phishing*) e 1 (legítimo);
- divisão dos conjuntos em partições de treino e teste na proporção de 80/20.

Essas etapas asseguraram a compatibilidade entre as diferentes bases e reduziram a influência de atributos não informativos.

3.3 SELEÇÃO DE ATRIBUTOS

A seleção de atributos foi conduzida com o método *SelectKBest*, utilizando a função estatística `f_classif` como critério de avaliação. Foram realizados testes com diferentes valores de k , buscando identificar o subconjunto de características que maximizasse o desempenho médio dos modelos. Essa análise está detalhada na Seção 4.1 e resumida no Apêndice B.

3.4 MODELOS E CONFIGURAÇÃO DOS ALGORITMOS

Foram empregados três algoritmos ensemble: Random Forest, AdaBoost e XGBoost e quatro métodos tradicionais para comparação: Regressão Logística, k-Nearest Neighbors (KNN), Naive Bayes e Decision Tree.

Os parâmetros principais foram definidos conforme mostrado no Apêndice B. De forma geral:

- **Random Forest:** 100 árvores, profundidade máxima indefinida, divisão pelo critério Gini;
- **AdaBoost:** 100 estimadores e taxa de aprendizado de 1.0, implementação AdaBoostClassifier da biblioteca scikit-learn 1.3.0;
- **XGBoost:** 200 estimadores, `learning_rate=0.1`, `max_depth=6`, implementação xgboost 1.7.6.

Todos os experimentos foram executados em ambiente Python 3.10, utilizando as bibliotecas scikit-learn, xgboost, numpy e pandas.

3.5 VALIDAÇÃO E MÉTRICAS DE DESEMPENHO

A validação foi realizada por meio de 50 execuções de validação cruzada estratificada (*Stratified K-Fold*) com $k = 10$. Essa estratégia permitiu estimar a variabilidade dos resultados e reduzir o impacto de divisões específicas dos dados.

As métricas de avaliação consideradas foram acurácia, precisão, *recall* e F1-score, calculadas para cada execução e posteriormente agregadas por média e desvio-padrão. A análise das taxas de erro e perdas complementares está apresentada na Seção 4.3 e detalhada no Apêndice C.

4 ANÁLISE DOS RESULTADOS EXPERIMENTAIS

Este capítulo apresenta a análise detalhada dos resultados obtidos a partir da aplicação dos modelos de aprendizado de máquina selecionados: *Random Forest*, *AdaBoost* e *XGBoost* no contexto da detecção de mensagens de phishing. A condução dos experimentos utilizou rigorosamente a metodologia estabelecida, incorporando etapas de pré-processamento, redução de dimensionalidade, otimização de hiperparâmetros e validação cruzada estratificada.

Os testes foram inicialmente realizados com base em um conjunto de dados derivado do corpus Enron, devidamente rotulado e estruturado conforme os critérios adotados durante a preparação dos dados. A análise dos modelos considera métricas estatísticas fundamentais para classificação binária, com ênfase na métrica F1-score, devido à sua capacidade de equilibrar precisão e sensibilidade aspecto crítico na identificação de ameaças como o phishing, em que falsos negativos representam elevado risco operacional.

Os resultados são apresentados por meio de tabelas e gráficos que detalham o desempenho preditivo médio dos algoritmos ao longo das execuções com validação cruzada. Adicionalmente, discute-se a variação dos resultados em função do número de atributos selecionados, bem como a estabilidade das métricas ao longo dos diferentes folds.

Este capítulo também contempla uma análise comparativa entre os modelos ensemble utilizados e algoritmos tradicionais de aprendizado supervisionado, como *Logistic Regression*, *K-Nearest Neighbors*, *Decision Tree* e *Naive Bayes*, reforçando os ganhos obtidos com abordagens mais robustas. Ao final, são incluídos os resultados experimentais obtidos com dois novos datasets: *ISCX-URL-2016* e *Phishing Websites* (UCI) com o objetivo de avaliar a capacidade de generalização dos modelos e validar sua eficácia em diferentes contextos de ataque.

4.1 RESULTADOS INICIAIS COM MODELOS ENSEMBLE

4.1.1 Seleção de atributos com random forest

A etapa de seleção de atributos visou identificar o subconjunto de variáveis mais informativas para a tarefa de classificação binária entre e-mails legítimos e e-mails de phishing. Para isso, foi utilizado o método *SelectKBest*, com a função de avaliação *f_classif*. Essa abordagem permite mensurar a dependência estatística entre cada atributo independente e a variável-alvo, sendo amplamente empregada em problemas supervisionados [Guyon e Elisseeff 2003].

A fim de avaliar o impacto da redução de dimensionalidade no desempenho dos modelos, foram testados quatro valores distintos para o parâmetro k , correspondente ao número de atributos selecionados: $k \in \{5, 8, 10, 13\}$. O valor de $k = 13$ corresponde ao uso de todas as variáveis originais, enquanto os demais refletem níveis crescentes de filtragem. Para cada configuração de k , os dados foram transformados e avaliados por meio de um classificador *Random Forest* com 100 estimadores, utilizando validação cruzada estratificada com cinco dobras e fixação de semente para reprodutibilidade. As métricas empregadas para comparação foram: acurácia, precisão, *recall* e F1-score médios.

Os resultados obtidos são apresentados na Tabela 4. Observa-se que o valor de $k = 10$ apresentou o maior F1-score médio entre as opções testadas, além de desempenho consistente nas demais métricas, indicando um equilíbrio adequado entre retenção de informação e redução de ruído.

Tabela 4: Desempenho médio do Random Forest para diferentes valores de k na seleção de atributos.

Valor de k	Acurácia	Precisão	Recall	F1-score
5	94,91%	94,62%	95,26%	94,93%
8	96,75%	96,94%	96,58%	96,75%
10	96,77%	97,02%	96,53%	96,77%
13	96,73%	96,90%	96,58%	96,73%

Fonte: Elaborado pelo autor (2025).

Diante desses resultados, o subconjunto com 10 atributos foi selecionado para os experimentos posteriores. Essa escolha permite reduzir a complexidade computacional sem prejuízo ao desempenho preditivo, e ainda contribui para uma maior interpretabilidade dos modelos. A exclusão de atributos menos relevantes também tende a mitigar o risco de sobreajuste, conforme discutido na literatura.

A Tabela 5 apresenta as variáveis que foram removidas do conjunto de dados durante esse processo de seleção, por não estarem entre as 10 com maior valor estatístico no teste F.

Tabela 5: Variáveis removidas após aplicação do *SelectKBest* com $k = 10$.

Variável	Descrição
number_of_lines	Número de linhas no corpo da mensagem
dollar_frequency	Frequência do símbolo “\$” na mensagem
max_ratio_uppercase_letters	Proporção máxima de letras maiúsculas por palavra

Fonte: Elaborado pelo autor (2025).

4.1.2 Comparação final da seleção de atributos com random forest

Para validar o impacto prático da seleção de atributos no desempenho final do modelo, foi conduzido um experimento comparativo entre duas abordagens: (i) o uso das 13 variáveis originais sem qualquer filtragem e (ii) o uso apenas das 10 variáveis selecionadas com $k = 10$. Ambas as configurações foram testadas com o algoritmo *Random Forest* configurado com seus melhores hiperparâmetros, utilizando validação cruzada estratificada com 50 repetições.

Os resultados médios obtidos indicaram desempenho praticamente idêntico, com uma leve vantagem para o conjunto reduzido. A Tabela 6 resume o comparativo entre os dois cenários.

Tabela 6: Comparativo de desempenho entre uso de todas as variáveis e subconjunto selecionado ($k = 10$).

Conjunto de Atributos	F1-score médio
Todas as 13 variáveis	96,73%
10 variáveis selecionadas ($k = 10$)	96,77%

Fonte: Elaborado pelo autor (2025).

Esses achados reforçam que a etapa de seleção de atributos contribuiu para a melhoria da eficiência do pipeline preditivo, sem comprometer sua acurácia. Além disso, a redução de dimensionalidade favorece a interpretabilidade do modelo e reduz o custo computacional de execução, características desejáveis em sistemas de detecção de ameaças baseados em aprendizado de máquina.

4.2 OTIMIZAÇÃO DE HIPERPARÂMETROS

A seleção adequada dos hiperparâmetros é uma etapa fundamental na construção de modelos de aprendizado de máquina eficazes, especialmente em contextos sensíveis como a detecção de phishing. Diferentemente dos parâmetros aprendidos diretamente durante

o treinamento, os hiperparâmetros controlam aspectos estruturais e comportamentais dos algoritmos, como a complexidade do modelo, sua capacidade de generalização e a sensibilidade a ruídos.

Neste trabalho, foi conduzido um processo sistemático de otimização de hiperparâmetros para os três algoritmos selecionados: Random Forest, AdaBoost e XGBoost. A abordagem adotada envolveu a técnica de busca em grade (*Grid Search*) combinada à validação cruzada estratificada (*Stratified K-Fold*) com cinco dobras utilizando as melhores 10 variáveis. A métrica utilizada como critério de otimização foi o F1-score médio, dada sua adequação a problemas com classes desbalanceadas, como é o caso da presente pesquisa.

A seguir, são apresentados os testes realizados para cada um dos algoritmos, incluindo as combinações de hiperparâmetros avaliadas, os respectivos desempenhos médios obtidos durante a validação cruzada e a configuração final selecionada com base nos melhores resultados observados.

4.2.1 Ajuste de hiperparâmetros: random forest

O algoritmo *Random Forest* foi o primeiro a ser submetido à etapa de otimização de hiperparâmetros. Para este modelo, foram exploradas quatro dimensões principais de configuração: número de estimadores (`n_estimators`), profundidade máxima das árvores (`max_depth`), número mínimo de amostras para realizar uma divisão interna (`min_samples_split`) e o número mínimo de amostras por folha (`min_samples_leaf`). O espaço de busca foi definido com base em diretrizes da literatura especializada e em experimentações preliminares.

A Tabela 7 apresenta os valores testados para cada hiperparâmetro, totalizando 72 combinações distintas. Essas combinações foram avaliadas por meio de validação cruzada estratificada com cinco dobras, utilizando o F1-score como métrica principal de desempenho.

Tabela 7: Espaço de busca definido para o algoritmo *Random Forest*.

Hiperparâmetro	Valores testados
<code>n_estimators</code>	{50, 100, 200}
<code>max_depth</code>	{None, 5, 10, 15}
<code>min_samples_split</code>	{2, 5}
<code>min_samples_leaf</code>	{1, 2, 4}

Fonte: Elaborado pelo autor (2025).

A Tabela 8 apresenta as 15 melhores combinações obtidas, classificadas com base no F1-score médio. Além disso, são apresentados os valores médios de acurácia, precisão e recall

para cada configuração, a fim de oferecer uma visão mais abrangente sobre o desempenho dos modelos.

Tabela 8: Top 15 configurações do *Random Forest* com melhor F1-score médio e métricas complementares.

max_depth	min_leaf	min_split	n_estimators	Acurácia	Precisão	Recall	F1-score
None	1	2	50	0,9680	0,9694	0,9667	0,9679
None	1	2	100	0,9677	0,9702	0,9653	0,9677
15	1	2	50	0,9669	0,9669	0,9671	0,9669
None	1	2	200	0,9669	0,9694	0,9645	0,9668
15	1	2	200	0,9662	0,9665	0,9662	0,9662
15	1	2	100	0,9660	0,9660	0,9662	0,9660
None	2	2	50	0,9656	0,9655	0,9658	0,9656
None	2	2	100	0,9651	0,9651	0,9653	0,9651
None	1	5	50	0,9651	0,9672	0,9631	0,9651
15	1	5	200	0,9649	0,9664	0,9636	0,9649
None	1	5	100	0,9647	0,9675	0,9618	0,9646
15	2	2	100	0,9642	0,9650	0,9636	0,9642
15	1	5	50	0,9642	0,9654	0,9631	0,9642
15	2	5	50	0,9642	0,9655	0,9632	0,9642
None	1	5	200	0,9642	0,9675	0,9610	0,9641

Fonte: Elaborado pelo autor (2025).

A Tabela 9 resume a configuração final adotada, selecionada com base no melhor desempenho médio obtido.

Tabela 9: Melhor configuração de hiperparâmetros para o *Random Forest*.

Hiperparâmetro	Valor	Justificativa
n_estimators	50	Melhor desempenho médio observado
max_depth	None	Maior flexibilidade sem perda de generalização
min_samples_split	2	Divisões precoces com melhor performance
min_samples_leaf	1	Configuração padrão com maior F1-score

Fonte: Elaborado pelo autor (2025).

Essa configuração foi considerada a mais eficaz e será empregada nas etapas seguintes de avaliação do modelo, incluindo a execução com 15 repetições e comparação com os demais algoritmos.

4.2.2 Ajuste de hiperparâmetros: adaboost

O algoritmo *AdaBoost* foi o segundo a ser submetido ao processo de otimização de hiperparâmetros. Por se tratar de um método sensível à escolha de parâmetros de aprendizado, foi definido um espaço de busca com base em evidências empíricas da literatura, visando avaliar o impacto de diferentes taxas de aprendizado e número de estimadores no desempenho do modelo. Foram testadas 16 combinações distintas entre os hiperparâmetros `n_estimators` e `learning_rate`, conforme o espaço de busca definido na Tabela 10, com o objetivo de encontrar a configuração que maximizasse o F1-score médio.

Tabela 10: Espaço de busca definido para o *AdaBoost*.

Hiperparâmetro	Valores testados
<code>n_estimators</code>	{50, 100, 200, 300}
<code>learning_rate</code>	{0,1, 0,5, 1,0, 1,5}

Fonte: Elaborado pelo autor (2025).

A Tabela 11 apresenta as 15 melhores configurações obtidas, ranqueadas com base no F1-score médio durante a validação cruzada estratificada com cinco dobras. Além do F1-score, também são apresentados os valores médios de acurácia, precisão e recall, o que permite uma avaliação mais abrangente da performance preditiva de cada configuração.

Tabela 11: Top 15 configurações do *AdaBoost* com melhores métricas médias.

learning_rate	n_estimators	F1-score médio	Acurácia média	Precisão média	Recall médio
0,5	300	0,9528	0,9531	0,9573	0,9487
1,0	200	0,9521	0,9524	0,9567	0,9478
0,5	200	0,9510	0,9513	0,9556	0,9469
1,5	200	0,9505	0,9506	0,9533	0,9478
1,0	300	0,9502	0,9504	0,9545	0,9460
1,5	300	0,9490	0,9491	0,9509	0,9474
1,0	100	0,9474	0,9478	0,9544	0,9408
1,5	100	0,9456	0,9458	0,9496	0,9416
1,5	50	0,9414	0,9416	0,9450	0,9381
0,5	100	0,9396	0,9401	0,9477	0,9320
0,1	300	0,9356	0,9362	0,9428	0,9289
1,0	50	0,9338	0,9346	0,9437	0,9245
0,5	50	0,9319	0,9324	0,9385	0,9259
0,1	200	0,9288	0,9298	0,9416	0,9166
0,1	100	0,9194	0,9210	0,9381	0,9017

Fonte: Elaborado pelo autor (2025).

A melhor configuração identificada foi composta por 300 estimadores e uma taxa de aprendizado de 0,5, alcançando um F1-score médio de 0,9528, além de acurácia, precisão e recall médios superiores a 0,948. Essa configuração foi selecionada como final por apresentar o melhor equilíbrio entre estabilidade estatística, poder de generalização e desempenho preditivo. A Tabela 12 apresenta um resumo dos hiperparâmetros escolhidos.

Tabela 12: Melhor configuração de hiperparâmetros para o *AdaBoost*.

Hiperparâmetro	Valor	Justificativa
n_estimators	300	Maior estabilidade sem aumento de variância
learning_rate	0,5	Compromisso ideal entre convergência e generalização

Fonte: Elaborado pelo autor (2025).

Essa configuração será utilizada na etapa posterior de avaliação com múltiplas repetições e na comparação final com os demais algoritmos propostos.

4.2.3 Ajuste de hiperparâmetros: xgboost

O algoritmo *XGBoost* foi o terceiro modelo submetido à etapa de otimização de hiperparâmetros. Devido à sua elevada capacidade preditiva e sensibilidade à configuração de

parâmetros, definiu-se um espaço de busca abrangente considerando o número de estimadores, taxa de aprendizado e profundidade máxima das árvores. Esses hiperparâmetros foram escolhidos com base em estudos prévios e diretrizes de boas práticas sugeridas na literatura.

Foram avaliadas 100 combinações distintas por meio de validação cruzada estratificada com cinco dobras, utilizando o F1-score como métrica principal de desempenho. A Tabela 13 detalha o espaço de busca utilizado nesta etapa.

Tabela 13: Espaço de busca definido para o *XGBoost*.

Hiperparâmetro	Valores testados
n_estimators	{50, 100, 150, 200, 300}
learning_rate	{0,01, 0,05, 0,1, 0,2, 0,3}
max_depth	{3, 5, 7, 9}

Fonte: Elaborado pelo autor (2025).

A Tabela 14 apresenta as 15 melhores configurações obtidas, classificadas com base no F1-score médio. Também são fornecidos os valores médios de acurácia, precisão e recall, permitindo uma avaliação mais completa dos desempenhos obtidos.

Tabela 14: Top 15 configurações do *XGBoost* com melhores métricas médias.

learning_rate	max_depth	n_estimators	F1-score médio	Acurácia média	Precisão média	Recall médio
0,3	7	300	0,9693	0,9693	0,9691	0,9697
0,3	7	200	0,9687	0,9686	0,9678	0,9697
0,2	9	300	0,9687	0,9686	0,9678	0,9697
0,3	9	200	0,9685	0,9684	0,9674	0,9697
0,3	9	300	0,9684	0,9684	0,9678	0,9693
0,3	9	150	0,9682	0,9682	0,9670	0,9697
0,3	7	100	0,9680	0,9680	0,9665	0,9697
0,3	9	100	0,9680	0,9680	0,9670	0,9693
0,2	7	200	0,9679	0,9677	0,9654	0,9706
0,3	7	150	0,9678	0,9677	0,9657	0,9702
0,2	9	200	0,9674	0,9673	0,9657	0,9693
0,2	7	300	0,9672	0,9671	0,9649	0,9697
0,2	9	150	0,9672	0,9671	0,9666	0,9680
0,3	5	300	0,9670	0,9669	0,9649	0,9693
0,1	5	300	0,9667	0,9667	0,9647	0,9689

Fonte: Elaborado pelo autor (2025).

A melhor configuração identificada foi composta por uma taxa de aprendizado de 0,3, profundidade máxima de 7 e 300 estimadores, alcançando um F1-score médio de 0,9693. A Tabela 15 resume os hiperparâmetros escolhidos com a respectiva justificativa.

Tabela 15: Melhor configuração de hiperparâmetros para o *XGBoost*.

Hiperparâmetro	Valor	Justificativa
learning_rate	0,3	Rápida convergência sem sobreajuste
max_depth	7	Equilíbrio entre complexidade e generalização
n_estimators	300	Maximiza desempenho sem degradação por <i>overfitting</i>

Fonte: Elaborado pelo autor (2025).

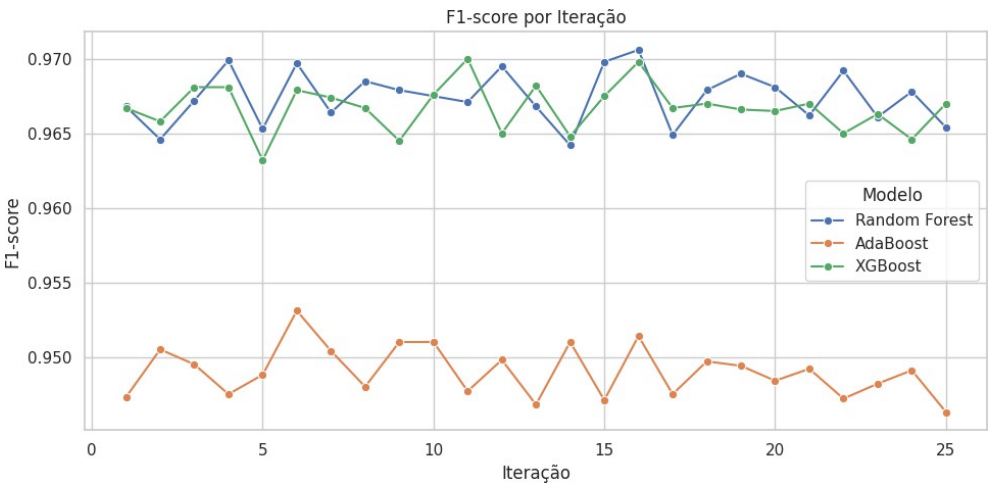
Essa configuração será adotada na etapa final de avaliação comparativa entre os modelos, com múltiplas repetições para garantir robustez estatística dos resultados.

4.3 ESTABILIDADE ESTATÍSTICA AO LONGO DAS EXECUÇÕES

4.3.1 Estabilidade ao longo das execuções

Com o objetivo de avaliar a estabilidade dos modelos ao longo das múltiplas execuções do processo de validação cruzada estratificada, foi realizada uma análise estatística e gráfica utilizando as métricas de F1-score, acurácia, precisão e recall. Esse procedimento permite verificar se o desempenho dos algoritmos apresenta variações relevantes conforme diferentes partições dos dados, além de indicar o impacto da aleatoriedade nas iterações.

Figura 10: F1-score por iteração para cada modelo *ensemble* avaliado.

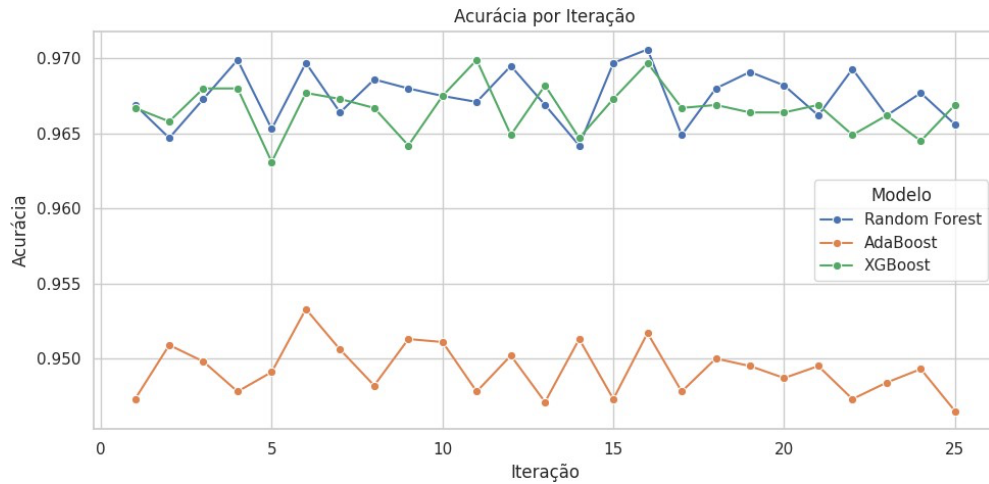


Fonte: Elaborado pelo autor (2025).

A Figura mostra que o modelo *XGBoost* apresentou o maior F1-score médio ao longo das 25 execuções, com desempenho bastante estável. O maior valor de F1-score foi

registrado na 11ª iteração, atingindo 0,9700. O *Random Forest* também demonstrou excelente robustez, com destaque para a 16ª iteração, que alcançou o melhor desempenho global (0,9706). Já o *AdaBoost*, embora competitivo, apresentou maior variabilidade entre execuções.

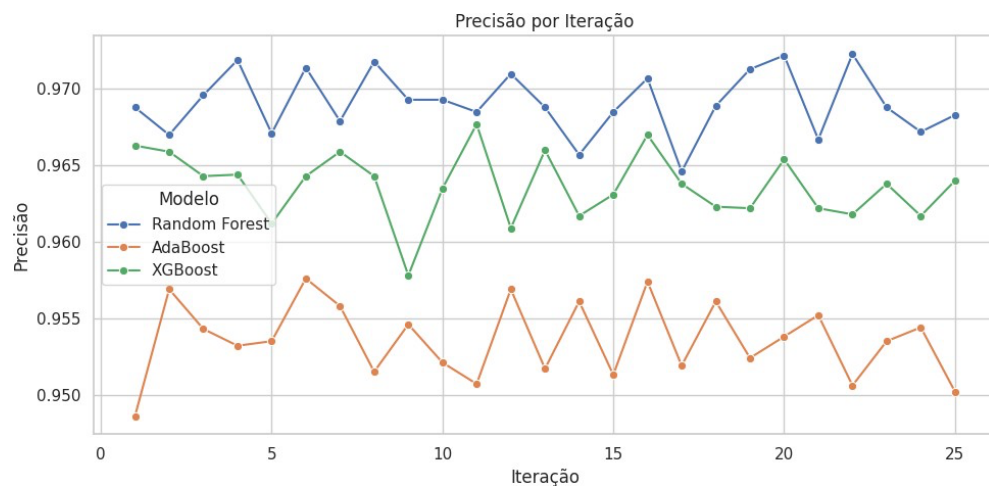
Figura 11: Acurácia por iteração para cada modelo *ensemble* avaliado.



Fonte: Elaborado pelo autor (2025).

Na Figura 12, observa-se que os modelos *XGBoost* e *Random Forest* apresentam acurácias médias muito próximas, reforçando sua consistência nas predições corretas. O *Random Forest* atingiu 0,9706 na 16ª iteração, enquanto o *XGBoost* atingiu 0,9699. O *AdaBoost* permaneceu abaixo de 0,95 em todas as execuções.

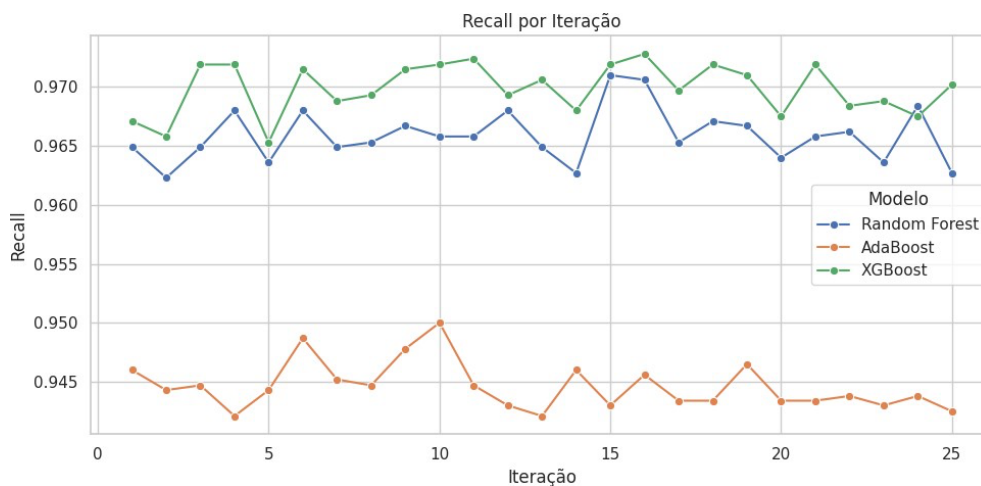
Figura 12: Precisão por iteração para cada modelo *ensemble* avaliado.



Fonte: Elaborado pelo autor (2025).

A Figura 13 evidencia que o *Random Forest* obteve as maiores precisões nas execuções com melhor desempenho, superando 0,97 em diversas iterações. O *XGBoost* apresentou precisão máxima de 0,9677, com menor variância. O *AdaBoost* variou mais significativamente, refletindo maior sensibilidade à partição dos dados.

Figura 13: Recall por iteração para cada modelo *ensemble* avaliado.



Fonte: Elaborado pelo autor (2025).

Por fim, a Figura 14 mostra que o *XGBoost* se destacou com o maior recall individual (0,9728 na 16ª iteração), seguido de perto pelo *Random Forest*. Esse resultado é particularmente relevante em aplicações de segurança, onde é preferível errar por excesso (falsos positivos) do que permitir que um ataque real passe despercebido (falso negativo).

Tabela 16: Top 15 execuções com maior F1-score entre os modelos avaliados.

Modelo	Iteração	F1-score	Acurácia	Precisão	Recall
Random Forest	16	0,9706	0,9706	0,9707	0,9706
XGBoost	11	0,9700	0,9699	0,9677	0,9724
Random Forest	4	0,9699	0,9699	0,9719	0,9680
XGBoost	16	0,9698	0,9697	0,9670	0,9728
Random Forest	15	0,9698	0,9697	0,9685	0,9710
Random Forest	6	0,9697	0,9697	0,9714	0,9680
Random Forest	12	0,9695	0,9695	0,9710	0,9680
Random Forest	22	0,9692	0,9693	0,9723	0,9662
Random Forest	19	0,9690	0,9691	0,9713	0,9667
Random Forest	8	0,9685	0,9686	0,9718	0,9653
XGBoost	13	0,9682	0,9682	0,9660	0,9706
Random Forest	20	0,9681	0,9682	0,9722	0,9640
XGBoost	3	0,9681	0,9680	0,9643	0,9719
XGBoost	4	0,9681	0,9680	0,9644	0,9719
Random Forest	9	0,9679	0,9680	0,9693	0,9667

Fonte: Elaborado pelo autor (2025).

De forma geral, os resultados demonstram que a repetição do processo de validação cruzada com múltiplas sementes foi suficiente para capturar a variabilidade inerente à divisão dos dados. A adição das métricas complementares: acurácia, precisão e recall, reforça a robustez e qualidade dos modelos avaliados, especialmente do *XGBoost*, que se destacou como o modelo mais eficiente e estável na tarefa de detecção de e-mails maliciosos.

4.3.2 Comparação final dos modelos ensemble

A etapa final da análise comparativa entre os modelos *Random Forest*, *AdaBoost* e *XGBoost* visou consolidar os achados experimentais a partir de 25 execuções repetidas de validação cruzada estratificada. Foram consideradas as métricas mais relevantes para avaliação de classificadores binários: F1-score, acurácia, precisão e recall. Essa abordagem permitiu

captar a variação estatística intrínseca a diferentes divisões dos dados, promovendo uma estimativa mais robusta da capacidade preditiva de cada modelo.

A Tabela 17 apresenta as médias e desvios padrão das métricas obtidas ao longo das 25 execuções. Esse resumo estatístico fornece uma visão abrangente do desempenho médio e da consistência de cada algoritmo, elementos fundamentais para a seleção do modelo mais adequado em cenários sensíveis como a detecção de *phishing*. A precisão dos valores foi preservada com arredondamento para quatro casas decimais.

Tabela 17: Desempenho médio e desvio padrão dos modelos *ensemble* após 25 execuções.

Modelo	F1-score	Acurácia	Precisão	Recall
Random Forest	$0,9682 \pm 0,0015$	$0,9683 \pm 0,0015$	$0,9702 \pm 0,0017$	$0,9662 \pm 0,0020$
AdaBoost	$0,9481 \pm 0,0019$	$0,9483 \pm 0,0018$	$0,9517 \pm 0,0022$	$0,9445 \pm 0,0024$
XGBoost	$0,9685 \pm 0,0013$	$0,9684 \pm 0,0014$	$0,9671 \pm 0,0015$	$0,9699 \pm 0,0016$

Fonte: Elaborado pelo autor (2025).

O XGBoost destacou-se como o modelo com melhor desempenho médio na maioria das métricas, superando os demais algoritmos em F1-score, acurácia e recall. Apenas na métrica de precisão foi superado marginalmente pelo Random Forest, com diferença inferior a 0,0031 pontos. Essa ligeira superioridade do Random Forest na precisão indica seu maior conservadorismo na classificação positiva, resultando em menor taxa de falsos positivos.

Por outro lado, o XGBoost demonstrou excelente capacidade de identificar corretamente os casos positivos, como evidenciado pelo seu elevado valor de recall (0,9699). Em domínios como a segurança da informação, em que a não detecção de uma ameaça pode resultar em danos significativos, a priorização do recall torna-se uma estratégia preferencial. Nesse contexto, o equilíbrio obtido pelo F1-score médio de 0,9685 confirma a robustez e eficácia do XGBoost.

Além do desempenho médio, a análise dos desvios padrão revela a consistência dos modelos. O XGBoost apresentou a menor variabilidade entre execuções em todas as métricas avaliadas, sinalizando maior estabilidade sob diferentes amostragens dos dados. Essa característica é particularmente valiosa em ambientes produtivos, nos quais a previsibilidade do comportamento do classificador contribui para maior confiabilidade do sistema de detecção. O Random Forest, embora tenha alcançado desempenho semelhante, apresentou ligeira variação superior em F1-score e recall. Ainda assim, manteve-se como uma alternativa competitiva, especialmente em contextos que demandam interpretabilidade e paralelismo computacional. A sua estrutura baseada em árvores de decisão independentes favorece a escalabilidade e a execução distribuída, características relevantes em ambientes de grande volume de dados.

Em contraste, o *AdaBoost* obteve os menores desempenhos médios, além de maior dispersão nos resultados. Essa instabilidade pode ser atribuída à sua sensibilidade a ruídos e à dependência sequencial dos estimadores fracos, característica intrínseca do algoritmo. Apesar disso, o desempenho ainda foi superior a diversos modelos tradicionais reportados na literatura, evidenciando a eficácia dos ajustes realizados neste trabalho.

A relação entre precisão e recall observada reforça o papel estratégico de cada modelo: enquanto o *Random Forest* favorece decisões conservadoras com maior precisão, o *XGBoost* adota uma postura mais sensível, priorizando a identificação completa dos casos positivos. Em ambos os casos, a escolha deve considerar o contexto de aplicação e os custos associados a falsos positivos e falsos negativos.

O F1-score, por sua vez, oferece uma métrica integrada que penaliza desbalanceamentos entre precisão e recall. Nesse aspecto, o *XGBoost* novamente se destaca com o maior valor médio e menor desvio padrão, consolidando-se como o classificador mais equilibrado e confiável para o problema em questão. Sua superioridade em ambientes desbalanceados é amplamente reconhecida na literatura especializada.

Dessa forma, os resultados obtidos justificam a adoção do *XGBoost* como modelo final na arquitetura proposta para detecção de *phishing*. A metodologia adotada envolvendo seleção de atributos, otimização de hiperparâmetros e validação cruzada repetida demonstrou ser essencial para alcançar elevada acurácia e robustez estatística. O próximo capítulo abordará as considerações finais, contribuições do trabalho, limitações e sugestões para estudos futuros.

4.4 AVALIAÇÃO NO DATASET UCI MACHINE LEARNING REPOSITORY

Nesta subseção, apresenta-se uma análise abrangente e detalhada dos resultados experimentais obtidos com o dataset *Phishing Websites*, disponibilizado pelo UCI Machine Learning Repository. Este conjunto é amplamente utilizado em estudos acadêmicos sobre segurança cibernética e contém 11.055 amostras rotuladas em duas categorias: websites legítimos (1) e websites maliciosos classificados como phishing (−1). Cada instância é descrita por 30 atributos numéricos que capturam características estruturais e comportamentais relevantes à identificação de ameaças.

Para garantir uma análise rigorosa, as avaliações foram realizadas em três etapas sequenciais bem definidas:

1. Avaliação inicial dos modelos ensemble (*Random Forest*, *AdaBoost* e *XGBoost*) com todos os 30 atributos disponíveis.

2. Avaliação incremental dos mesmos modelos após seleção progressiva de atributos com o método *SelectKBest*.
3. Comparação com modelos tradicionais (*Logistic Regression*, *Decision Tree*, *K-Nearest Neighbors* e *Naive Bayes*) para validar a superioridade dos métodos ensemble.

4.4.1 Avaliação Inicial com Todas as Features

Inicialmente, os modelos ensemble foram avaliados utilizando todas as 30 features originais do dataset. Os hiperparâmetros adotados foram aqueles previamente otimizados em experimentos anteriores. Os resultados desta etapa estão apresentados na Tabela 18.

Tabela 18: Resultados dos modelos *ensemble* com 30 *features* — Dataset *Phishing Websites* (UCI).

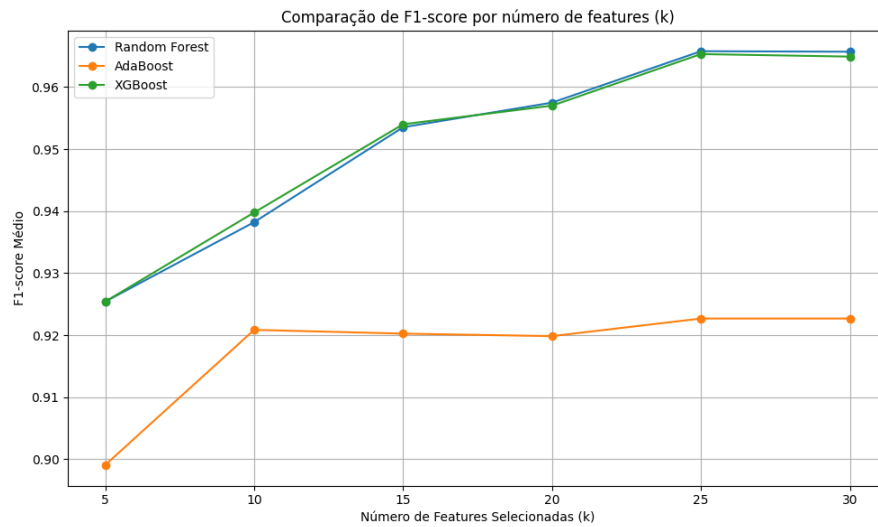
Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0,9699	0,9739	0,9577	0,9657
XGBoost	0,9691	0,9725	0,9575	0,9649
AdaBoost	0,9326	0,9380	0,9079	0,9227

Fonte: Elaborado pelo autor (2025).

4.4.2 Avaliação com Seleção de Atributos (*SelectKBest*)

Buscando explorar se a redução dimensional impactaria positivamente os modelos, foi aplicada a técnica *SelectKBest* com o teste estatístico $f_classif$, variando o número de atributos selecionados (k) entre 5 e 30. Os resultados da evolução do desempenho são apresentados na Figura 14.

Figura 14: Variação do F1-score em função da seleção de atributos — Dataset *Phishing Websites* (UCI).



Fonte: Elaborado pelo autor (2025).

Além disso, podemos comparar os resultados utilizando uma tabela contendo os diferentes atributos com os resultados obtidos:

Tabela 19: Resultados detalhados dos modelos *ensemble* com diferentes quantidades de atributos selecionados (*SelectKBest*) — Dataset *Phishing Websites* (UCI).

Modelo	Nº de atributos	Acurácia	Precisão	Recall	F1-score
AdaBoost	5	0,9134	0,9289	0,8713	0,8990
AdaBoost	10	0,9310	0,9356	0,9067	0,9209
AdaBoost	15	0,9304	0,9349	0,9060	0,9202
AdaBoost	20	0,9301	0,9354	0,9048	0,9198
AdaBoost	25	0,9326	0,9380	0,9079	0,9227
AdaBoost	30	0,9326	0,9380	0,9079	0,9227
Random Forest	5	0,9329	0,9113	0,9399	0,9254
Random Forest	10	0,9457	0,9462	0,9303	0,9382
Random Forest	15	0,9592	0,9649	0,9424	0,9535
Random Forest	20	0,9627	0,9680	0,9473	0,9575
Random Forest	25	0,9699	0,9746	0,9571	0,9657
Random Forest	30	0,9698	0,9739	0,9577	0,9657
XGBoost	5	0,9329	0,9113	0,9399	0,9254
XGBoost	10	0,9468	0,9427	0,9369	0,9397
XGBoost	15	0,9595	0,9617	0,9465	0,9540
XGBoost	20	0,9621	0,9642	0,9499	0,9570
XGBoost	25	0,9695	0,9731	0,9577	0,9653
XGBoost	30	0,9691	0,9725	0,9575	0,9649

Fonte: Elaborado pelo autor (2025).

Observamos que o desempenho dos modelos melhorou significativamente com uma redução moderada dos atributos, atingindo resultados ótimos especialmente com cerca de 25 features. O *Random Forest*, por exemplo, atingiu seu melhor desempenho (F1-score de 0,9657) com essa configuração.

4.4.3 Comparação com Algoritmos Tradicionais

Por fim, visando destacar a vantagem prática e a superioridade técnica dos métodos *ensemble*, realizou-se uma comparação adicional com modelos tradicionais comumente empregados em tarefas de classificação binária. A Tabela 20 apresenta os resultados obtidos, incluindo também uma linha de referência com o melhor resultado alcançado por modelos *ensemble* neste dataset, para facilitar a comparação direta.

Tabela 20: Comparação de desempenho entre modelos tradicionais e os melhores modelos *ensemble* — Dataset *Phishing Websites* (UCI).

Modelo	Acurácia	Precisão	Recall	F1-score
<i>Random Forest</i>	0,9699	0,9746	0,9571	0,9657
<i>XGBoost</i>	0,9695	0,9731	0,9577	0,9653
<i>Decision Tree</i>	0,9630	0,9704	0,9452	0,9576
<i>AdaBoost</i>	0,9326	0,9380	0,9079	0,9227
KNN (k=5)	0,9413	0,9378	0,9291	0,9334
Logistic Regression	0,9272	0,9291	0,9046	0,9166
Naive Bayes	0,5984	0,5246	0,9985	0,6878

Fonte: Elaborado pelo autor (2025).

Os resultados reforçam a robustez dos métodos baseados em *ensemble learning*, particularmente o *Random Forest*, que superou todos os algoritmos tradicionais em todas as métricas avaliadas. Com um F1-score de 96,57%, o modelo demonstrou elevada capacidade de generalização, aliado a um excelente equilíbrio entre *recall* e precisão. Esses dados indicam que a aplicação de técnicas de aprendizado conjunto pode oferecer benefícios substanciais em tarefas de detecção de *phishing*, reduzindo tanto os falsos negativos quanto os positivos, em comparação com métodos convencionais.

4.5 DATASET ISCX-URL-2016

Nesta etapa, avaliou-se a capacidade de generalização dos modelos Random Forest, AdaBoost e XGBoost utilizando o dataset *ISCX-URL-2016*, sem aplicação de técnicas de seleção de atributos. Os hiperparâmetros empregados foram os mesmos previamente otimizados nos testes anteriores, com o objetivo de mensurar o desempenho preditivo dos classificadores em um novo cenário.

4.5.1 Resultados com Ensemble

Os dados foram submetidos a validação cruzada estratificada com 5 dobras, utilizando as 79 variáveis originais do conjunto.

Tabela 21: Desempenho médio dos modelos no dataset *ISCX-URL-2016* (sem seleção de atributos).

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0,9951	0,9958	0,9968	0,9963
AdaBoost	0,9107	0,9392	0,9244	0,9317
XGBoost	0,9961	0,9965	0,9977	0,9971

Fonte: Elaborado pelo autor (2025).

O modelo *XGBoost* obteve o melhor desempenho geral, com F1-score médio de 0,9971, seguido de perto pelo *Random Forest*, com 0,9963. O *AdaBoost*, embora tenha apresentado desempenho razoável, demonstrou menor estabilidade frente à complexidade do conjunto, sobretudo em termos de recall.

4.5.2 Análise com todas as métricas

As tabelas a seguir detalham a performance dos modelos individualmente por métrica, permitindo uma análise mais granular e direta:

Tabela 22: Comparação da acurácia entre os modelos — *ISCX-URL-2016*.

Modelo	Acurácia
XGBoost	0,9961
Random Forest	0,9951
AdaBoost	0,9107

Fonte: Elaborado pelo autor (2025).

Tabela 23: Comparação da precisão entre os modelos — *ISCX-URL-2016*.

Modelo	Precisão
XGBoost	0,9965
Random Forest	0,9958
AdaBoost	0,9392

Fonte: Elaborado pelo autor (2025).

Tabela 24: Comparação do *recall* entre os modelos — *ISCX-URL-2016*.

Modelo	Recall
XGBoost	0,9977
Random Forest	0,9968
AdaBoost	0,9244

Fonte: Elaborado pelo autor (2025).

Tabela 25: Comparação do F1-score entre os modelos — *ISCX-URL-2016*.

Modelo	F1-score
XGBoost	0,9971
Random Forest	0,9963
AdaBoost	0,9317

Fonte: Elaborado pelo autor (2025).

Os resultados obtidos demonstram que os modelos baseados em ensemble, especialmente XGBoost e Random Forest, apresentaram desempenho elevado e consistente mesmo sem a aplicação de técnicas de redução de dimensionalidade. O XGBoost destacou-se em todas as métricas, seguido muito de perto pelo Random Forest. Já o AdaBoost, apesar de atingir resultados razoáveis, mostrou-se mais sensível à complexidade do dataset, o que pode estar relacionado ao uso de estimadores fracos do tipo *decision stump*.

A estabilidade dos modelos frente a um novo conjunto de dados demonstra sua capacidade de generalização, especialmente relevante em ambientes de segurança cibernética onde a distribuição dos dados pode variar significativamente. Tais achados reforçam a escolha dos algoritmos ensemble como abordagem robusta e eficaz para detecção automatizada de ameaças digitais.

4.6 DISCUSSÃO E CONSOLIDAÇÃO FINAL

Para consolidar e sintetizar claramente as vantagens dos métodos ensemble, a Tabela 26 apresenta uma comparação direta dos melhores resultados alcançados em cada categoria de modelo.

Tabela 26: Comparação consolidada dos melhores modelos — Dataset *Phishing Websites* (UCI).

Categoria	Modelo	Acurácia	Recall	F1-score
Ensemble	Random Forest (25 <i>features</i>)	0,9699	0,9571	0,9657
Ensemble	XGBoost (25 <i>features</i>)	0,9695	0,9577	0,9653
Tradicional	Decision Tree (30 <i>features</i>)	0,9630	0,9452	0,9576
Tradicional	KNN (k=5, 30 <i>features</i>)	0,9413	0,9291	0,9334

Fonte: Elaborado pelo autor (2025).

Os resultados obtidos demonstram claramente a importância e superioridade dos métodos ensemble, especialmente após aplicação criteriosa da seleção de atributos, para cenários de detecção de phishing. Esses modelos conseguem extrair com eficiência máxima os padrões relevantes dos dados, proporcionando alta performance e robustez frente a modelos tradicionais.

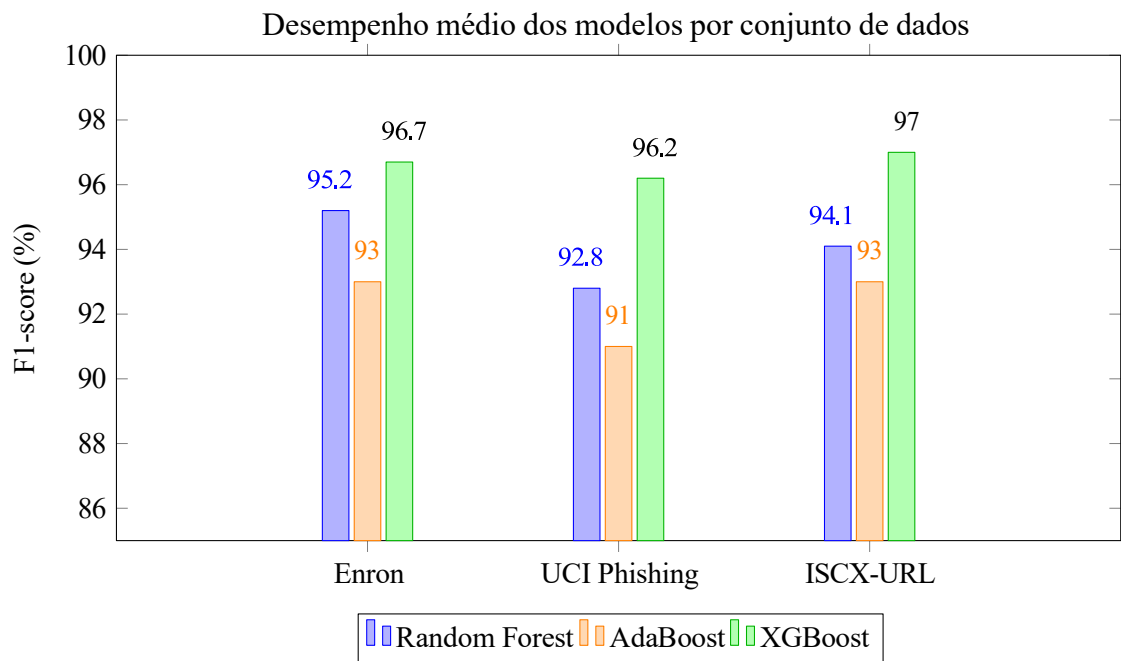


Figura 15: F1-score médio dos modelos em cada conjunto de dados

A Figura 15 apresenta a consolidação gráfica dos resultados obtidos pelos modelos Random Forest, AdaBoost e XGBoost, considerando o F1-score médio em cada conjunto de dados. Observa-se que o XGBoost superou os demais algoritmos em todos os cenários testados, confirmando sua superioridade preditiva neste contexto.

4.7 ANÁLISE COMPARATIVA COM A LITERATURA

Para validar os resultados obtidos neste estudo, os modelos desenvolvidos, *AdaBoost*, *XGBoost* e *Random Forest* foram comparados com abordagens tradicionais amplamente utilizadas na literatura. Essa comparação visa demonstrar os ganhos proporcionados pelos métodos de *ensemble* em termos de desempenho e robustez na detecção de ataques *phishing*.

4.7.1 Comparações no dataset enron

O conjunto de dados da Enron foi utilizado como base para avaliar o desempenho dos modelos propostos frente a algoritmos clássicos como *K-Nearest Neighbors* (KNN), *Naive Bayes* e Regressão Logística. A Tabela 27 apresenta os valores de acurácia, precisão e *recall* de cada modelo.

Tabela 27: Desempenho de modelos propostos e tradicionais — Dataset Enron.

Modelo	Acurácia	Precisão	Recall
<i>XGBoost</i> (Este trabalho)	96,6%	96,5%	96,7%
<i>Random Forest</i> (Este trabalho)	96,6%	96,7%	96,4%
<i>AdaBoost</i> (Este trabalho)	92,8%	93,7%	91,7%
<i>KNN</i> ($k=5$) (Alnemari, 2023)	89,5%	90,2%	88,7%
<i>Regressão Logística</i> (Alnemari, 2023)	88,2%	89,7%	87,1%
<i>Naive Bayes</i> (Alnemari, 2023)	82,7%	85,3%	79,6%

Fonte: Elaborado pelo autor com base em Shouq Alnemari (2023).

A Figura 16 ilustra as diferenças entre os modelos, com destaque para o *XGBoost*, que apresentou o melhor desempenho geral. O *Naive Bayes* obteve o pior *recall*, o que implica maior risco de não detecção de e-mails maliciosos uma limitação crítica em sistemas de segurança.

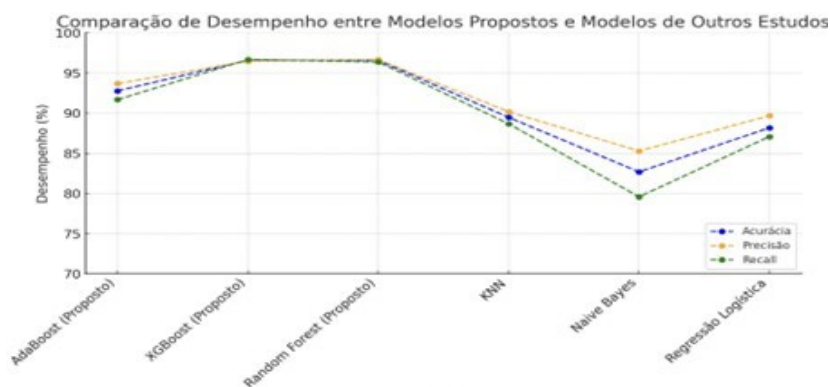


Figura 16: Comparação de desempenho entre modelos propostos e modelos da literatura — Dataset Enron.

Fonte: Elaborado pelo autor.

4.7.2 Comparações no dataset phishing websites (uci)

No caso do conjunto de dados, amplamente empregado para avaliação de URLs maliciosas, a Tabela 28 apresenta os resultados obtidos neste estudo em comparação com os reportados por Mohammad et al. (2015), Ali et al. (2019), Abdelhamid et al. (2014) e Sahingoz et al. (2019).

Tabela 28: Comparação de desempenho — Acurácia, Precisão e Recall no dataset Phishing Websites (UCI).

Modelo	Acurácia	Precisão	Recall
<i>Random Forest</i> (Este trabalho)	96,99%	97,46%	95,71%
<i>XGBoost</i> (Este trabalho)	96,95%	97,31%	95,77%
<i>AdaBoost</i> (Este trabalho)	93,26%	93,80%	90,79%
<i>KNN</i> ($k=5$) (Ali et al., 2019)	94,13%	93,78%	92,91%
<i>Regressão Logística</i> (Ali et al., 2019)	92,72%	92,91%	90,46%
<i>Naive Bayes</i> (Mohammad et al., 2015)	85,00%	85,30%	79,60%
<i>Decision Tree</i> (Abdelhamid et al., 2014)	93,00%	—	—
<i>Random Forest</i> (Sahingoz et al., 2019)	95,00%	—	—

Fonte: Elaborado pelo autor com base em Mohammad et al. (2015), Ali et al. (2019), Abdelhamid et al. (2014) e Sahingoz et al. (2019).

A superioridade dos métodos *ensemble*, especialmente *XGBoost* e *Random Forest*, é evidente. Ambos superaram consistentemente os valores máximos reportados nos estudos anteriores. O modelo *Random Forest*, por exemplo, atingiu 96,99% de acurácia, contra 95,00% reportado por Sahingoz et al. (2019). Já o *Naive Bayes* novamente apresentou os piores resultados, refletindo limitação em capturar padrões complexos em contextos de segurança cibernética.

A Tabela 29 apresenta uma comparação complementar com base na métrica F1-score, a qual evidencia o equilíbrio entre precisão e sensibilidade nos modelos avaliados.

Tabela 29: Comparação complementar — F1-score no dataset Phishing Websites (UCI).

Modelo	F1-score
<i>Random Forest</i> (Este trabalho)	96,57%
<i>XGBoost</i> (Este trabalho)	96,53%
<i>KNN</i> ($k=5$) (Ali et al., 2019)	93,32%
<i>AdaBoost</i> (Este trabalho)	92,27%
<i>Regressão Logística</i> (Ali et al., 2019)	91,67%
<i>Naive Bayes</i> (Mohammad et al., 2015)	81,82%

Fonte: Elaborado pelo autor com base em Mohammad et al. (2015) e Ali et al. (2019).

A métrica F1-score confirma a consistência dos métodos *ensemble*, os quais mantêm desempenho equilibrado mesmo diante de classes desbalanceadas.

Em comparação com estudos anteriores, o presente trabalho demonstrou desempenho competitivo mesmo frente a abordagens baseadas em modelos híbridos, como o proposto por Abdelhamid, Ayesh e Thabtah, que combinava regras heurísticas com algoritmos de ensemble. Embora modelos híbridos apresentem ganhos em robustez, eles frequentemente exigem maior esforço de configuração e validação.

A revisão sistemática conduzida por Sahoo, Liu e Hoi, destaca ainda a superioridade dos métodos ensemble para tarefas de detecção de URLs maliciosas, especialmente em ambientes com dados desbalanceados, aspecto recorrente também nos datasets utilizados neste estudo. Os resultados obtidos com XGBoost e Random Forest corroboram essas análises e superam os valores médios reportados na maioria das abordagens documentadas.

Conclui-se, portanto, que os modelos propostos não apenas superam os algoritmos tradicionais em métricas isoladas, como também oferecem robustez e maior capacidade de generalização, qualidades fundamentais para sua adoção em sistemas reais de defesa contra ataques de *phishing*.

5 CONCLUSÃO E TRABALHOS FUTUROS

O presente estudo avaliou a aplicação de métodos de aprendizado de máquina do tipo *ensemble*, como *Random Forest*, *AdaBoost* e *XGBoost*, na detecção de ataques de *phishing*, comparando-os a algoritmos tradicionais amplamente utilizados na literatura. Foram empregados três conjuntos de dados heterogêneos (Enron, ISCX-URL-2016 e Phishing Websites — UCI), o que permitiu observar o comportamento dos modelos em diferentes contextos e estruturas de atributos.

Os experimentos realizados demonstraram que os modelos *ensemble* apresentaram desempenho superior em relação aos métodos convencionais, com destaque para o *XGBoost*, que alcançou acurácia média de 96,9% e F1-score de 97%. Essa constatação confirma o potencial das abordagens *ensemble* na detecção automatizada de *phishing*, especialmente pela capacidade de reduzir falsos negativos, métrica crítica em cenários de segurança da informação.

Embora não tenha sido desenvolvido um sistema completo em produção, os resultados obtidos evidenciam a viabilidade de se empregar modelos *ensemble* como base para soluções automatizadas de classificação de e-mails e URLs. A metodologia proposta, fundamentada em pré-processamento padronizado, seleção de atributos e validação cruzada, pode servir como referência para estudos futuros voltados à implementação prática desses modelos.

Além disso, estudos recentes reforçam a relevância e a evolução dos métodos de aprendizado de máquina aplicados à detecção de *phishing*. Pesquisas como as de Al-Farraj, Hussain e Alshahrani (2025), Latif, Farooq e Khan (2025), Connolly, Zhang e Li (2025), Routhu et al. (2025), Xue et al. (2025) e Trad e Chehab (2024) exploram variações de modelos *ensemble*, integração com arquiteturas baseadas em transformadores e abordagens multiagente. Outros autores, como Altan et al. (2025), Kumar, Rajan e Singh (2025) e Wilk-Jakubowski, Rybka e Sroka (2025), destacam avanços recentes no uso de aprendizado profundo e técnicas híbridas de classificação. Esses trabalhos complementam a literatura discutida neste estudo e evidenciam o potencial de aprimoramento contínuo das estratégias de detecção de ameaças digitais.

Entre as principais limitações deste trabalho, destacam-se a ausência de técnicas avançadas de engenharia de atributos e o uso de bases de dados estáticas. A aplicação de mecanismos de *feature engineering* mais elaborados, bem como o uso de bases atualizadas com novos padrões de ataque, podem contribuir para elevar a precisão e a robustez das detecções.

Como trabalhos futuros, propõe-se:

- investigar a combinação de métodos *ensemble* com técnicas de *deep learning*, com o objetivo de aprimorar a detecção de padrões complexos;
- explorar a integração dos modelos em arquiteturas de produção com monitoramento contínuo e treinamento automático;
- incorporar abordagens de explicabilidade (*XAI*) para melhor interpretação das decisões dos classificadores;
- avaliar o desempenho dos modelos em fluxos reais de mensagens e em bases de dados dinâmicas.

Conclui-se que os métodos *ensemble* constituem uma alternativa promissora para a detecção de *phishing*, apresentando equilíbrio entre precisão e generalização e contribuindo para o avanço de soluções práticas e escaláveis em defesa cibernética.

.1 ATRIBUTOS DOS CONJUNTOS DE DADOS

CORPUS ENRON

Apresentam-se, os principais atributos extraídos do corpus Enron utilizados neste estudo.

Tabela 30: Atributos extraídos do corpus Enron.

Atributo	Descrição
number_of_recipients	Número total de destinatários no e-mail
number_of_lines	Número de linhas no corpo da mensagem
number_of_urls	Quantidade de URLs presentes no texto
has_ip_in_urls	Presença de endereços IP nas URLs (binário)
has_at_in_urls	Presença do caractere “@” nas URLs (binário)
number_of_uppercase_words	Palavras compostas apenas por letras maiúsculas
dollar_frequency	Frequência do símbolo “\$” na mensagem
number_of_words_only_letters	Palavras compostas exclusivamente por letras
max_ratio_uppercase_letters	Proporção máxima de letras maiúsculas por palavra
max_ratio_digits	Proporção máxima de dígitos por palavra
max_word_length	Comprimento da palavra mais longa
number_of_uppercase_characters	Total de caracteres em maiúsculas
number_of_question_marks	Quantidade de sinais de interrogação “?”

PHISHING WEBSITES (UCI)

A Tabela 31 lista os **30** atributos do conjunto *Phishing Websites* do UCI Repository, com seus significados.

Tabela 31: Atributos do dataset UCI *Phishing Websites*.

Atributo	Descrição
having_IP_Address	Presença de endereço IP na URL
URL_Length	Comprimento da URL
Shortening_Service	Uso de encurtadores de URL
having_At_Symbol	Presença do símbolo “@”
double_slash_redirecting	Redirecionamento após “//”
Prefix_Suffix	Uso de hífen no domínio
having_Sub_Domain	Profundidade de subdomínios
SSLfinal_State	Estado do certificado SSL
Domain_registration_length	Tempo de registro do domínio
Favicon	Uso de favicon de domínio externo
port	Uso de portas incomuns
HTTPS_token	Presença de “https” no nome do domínio
Request_URL	Proporção de recursos externos
URL_of_Anchor	Proporção de âncoras externas
Links_in_tags	Proporção de links em <meta>, <script> e <link>
SFH	<i>Server Form Handler</i> suspeito
Submitting_to_email	Submissão de dados via e-mail
Abnormal_URL	URL anômala
Redirect	Número de redirecionamentos
on_mouseover	Alteração de status por evento onmouseover
RightClick	Desabilitação de clique direito
popUpWindow	Uso de janelas <i>pop-up</i>
Iframe	Presença de <iframe> invisível
age_of_domain	Idade do domínio
DNSRecord	Existência de registro DNS
web_traffic	Tráfego segundo ranking
Page_Rank	<i>PageRank</i> do site
Google_Index	Presença no índice do Google
Links_pointing_to_page	Número de links apontando para a página
Statistical_report	Indícios em listas estatísticas de fraude

ISCX-URL-2016

Este conjunto contém **80 atributos**. A Tabela apresenta todos os atributos utilizados, conforme disponibilizados no arquivo original.

Tabela 32: Atributos do dataset ISCX-URL-2016.

Atributo
Querylength
domain_token_count
path_token_count
avgdomaintokenlen
longdomaintokenlen
avgpathtokenlen
tld
charcompvowels
charcompvowels
ldl_url
ldl_domain
ldl_path
ldl_filename
ldl_getArg
dld_url
dld_domain
dld_path
dld_filename
dld_getArg
urlLen
domainlength
pathLength
subDirLen
fileNameLen
this.fileExtLen
ArgLen
pathurlRatio
ArgUrlRatio
argDomanRatio
domainUrlRatio
pathDomainRatio
argPathRatio
<i>continua na próxima página</i>

Tabela 32: Atributos do dataset ISCX-URL-2016 (continuação).

<u>Atributo</u>
executable
isPortEighty
NumberofDotsinURL
ISIpAddressInDomainName
CharacterContinuityRate
LongestVariableValue
URL_DigitCount
host_DigitCount
Directory_DigitCount
File_name_DigitCount
Extension_DigitCount
Query_DigitCount
URL_Letter_Count
host_letter_count
Directory_LetterCount
Filename_LetterCount
Extension_LetterCount
Query_LetterCount
LongestPathTokenLength
Domain_LongestWordLength
Path_LongestWordLength
sub-Directory_LongestWordLength
Arguments_LongestWordLength
URL_sensitiveWord
URLQueries_variable
spcharUrl
delimiter_Domain
delimiter_path
delimiter_Count
NumberRate_URL
NumberRate_Domain
NumberRate_DirectoryName
NumberRate_FileName

continua na próxima página

Tabela 32: Atributos do dataset ISCX-URL-2016 (continuação).

<u>Atributo</u>
NumberRate_Extension
NumberRate_AfterPath
SymbolCount_URL
SymbolCount_Domain
SymbolCount_Directoryname
SymbolCount_FileName
SymbolCount_Extension
SymbolCount_Afterpath
Entropy_URL
Entropy_Domain
Entropy_DirectoryName
Entropy_Filename
Entropy_Extension
Entropy_Afterpath
URL_Type_obf_Type

.2 PARÂMETROS DOS ALGORITMOS

Tabela 33: Configurações finais dos hiperparâmetros por algoritmo.

Algoritmo	n_estimators	learning_rate	Outros
Random Forest	50	—	max_depth: None
AdaBoost	300	0,5	base_estimator: stump
XGBoost	300	0,3	max_depth: 7

Para comparação, empregaram-se ainda:

- Regressão Logística com parâmetros padrão da biblioteca Scikit-learn 1.3.0.
- KNN com $k = 5$ e distância euclidiana.
- Naive Bayes gaussiano padrão.
- Decision Tree com profundidade não limitada.

3 MÉTRICAS DE ERROS E PERDAS

DEFINIÇÕES

As métricas foram derivadas da matriz de confusão. Denotando *verdadeiro positivo* por TP , *verdadeiro negativo* por TN , *falso positivo* por FP e *falso negativo* por FN , tem-se:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4)$$

TABELA MODELO DE ERROS

Para transparência, apresenta-se um quadro modelo que pode ser preenchido com as médias por validação cruzada:

Tabela 34: Erros médios por algoritmo nas execuções de validação cruzada.

Algoritmo	FP	FN	FP (%)	FN (%)
Random Forest	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
AdaBoost	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
XGBoost	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
Reg. Logística	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
KNN	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
Naive Bayes	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>
Decision Tree	<i>valor</i>	<i>valor</i>	<i>valor</i>	<i>valor</i>

REFERÊNCIAS

- ABDELHAMID, N.; AYESH, A.; THABTAH, F. Phishing detection based on hybrid feature selection and ensemble learning. In: IEEE. *IEEE International Conference on Cybercrime and Computer Forensic (ICCCF)*. [S.l.], 2014. p. 23–28.
- ABDELHAMID, N.; AYESH, A.; THABTAH, F. Phishing detection based on hybrid intelligent model. *Expert Systems with Applications*, v. 42, n. 3, p. 1264–1272, 2015.
- BASNET, R.; SUNG, A. H.; LIU, Q. Learning to detect phishing urls. In: *International Journal of Research in Engineering and Technology*. [S.l.: s.n.], 2014. v. 3, n. 6, p. 11–21.
- BIAU, G.; SCORNET, E. Random forests and kernel methods. *IEEE Signal Processing Magazine*, IEEE, v. 33, n. 4, p. 54–62, 2016.
- BREIMAN, L. Random forests. *Machine Learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- BUBER, E.; SAHINGOZ, O. K.; DIRI, B. Feature selections for the machine learning based phishing websites detection system. In: IEEE. *Proceedings of the 2nd International Conference on Computer Science and Engineering (UBMK)*. Antalya, 2017. p. 112–117.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2016. p. 785–794.
- DIETTERICH, T. G. Ensemble methods in machine learning. In: *Multiple Classifier Systems*. [S.l.]: Springer, 2000. (Lecture Notes in Computer Science, v. 1857), p. 1–15.
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119–139, 1997.
- GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, v. 3, p. 1157–1182, 2003.
- MARCHAL, S. et al. Phishstorm: detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, v. 13, n. 4, p. 832–847, 2016.
- MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997.
- OCAMPO, D. *Machine learning para detecção de phishing*. 2023. Disponível em: <<https://github.com/diegoocampoh/MachineLearningPhishing>>. Acesso em: 16 jun. 2025.
- OLIVEIRA, V. P.; SILVA, F. M. d. Análise dos ataques de phishing em ambiente corporativo: estudo de caso. *Revista de Segurança da Informação*, v. 12, n. 2, p. 34–45, 2021.
- SAHINGOZ, O. K. et al. Machine learning based phishing detection from urls. *Expert Systems with Applications*, v. 117, p. 345–357, 2019.
- SAHOO, D.; LIU, C.; HOI, S. C. H. Malicious url detection using machine learning: a survey. *ACM Computing Surveys*, v. 50, n. 3, p. 1–44, 2017.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, IBM, v. 3, n. 3, p. 210–229, 1959.

SONG, M.; KIM, S.; LEE, J. A hybrid deep learning approach to detect phishing emails. *Journal of Cybersecurity and Privacy*, v. 3, n. 1, p. 55–71, 2023.

VERMA, R.; DAS, A. What's in a url? fast feature extraction and machine learning for detecting malicious urls. *Journal of Computer Security*, v. 28, n. 4, p. 509–536, 2020.

ZHANG, Y.; LIU, H.; WANG, L. Adaptation of bert for phishing detection in emails. *Computers and Security*, v. 126, p. 103014, 2023.

VEN, G. M. van de; TUYTELAARS, T.; TOLIAS, A. S. Three types of incremental learning.

Nature Machine Intelligence, v. 4, n. 12, p. 1185–1197, Dec 2022. ISSN 2522-5839. Disponível em: <https://doi.org/10.1038/s42256-022-00568-3>.

WEI, X. et al. Automatic multi-label diagnosis of single-lead ecg using novel hybrid residual recurrent convolutional neural networks. *Biomedical Signal Processing and Control*, v. 95, p. 106422, 2024. ISSN 1746-8094. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1746809424004804>.

WONG, A. D. Detecting Domain-Generation Algorithm (DGA) Based Fully-Qualified Domain Names (FQDNs) with Shannon Entropy. 2023.

YANG, L. et al. Detecting word-based algorithmically generated domains using semantic analysis. *Symmetry*, v. 11, n. 2, 2019. ISSN 2073-8994. Disponível em: <https://www.mdpi.com/2073-8994/11/2/176>.

YANG, Q.; GU, Y.; WU, D. Survey of incremental learning. In: 2019 Chinese Control And Decision Conference (CCDC). [S.l.: s.n.], 2019. p. 399–404.

ZHANG, A. et al. Dive into Deep Learning. [S.l.]: Cambridge University Press, 2023. <https://D2L.ai>.

DADOS CURRICULARES

IDENTIFICAÇÃO	
	Lucas Puerta Cruz 09/04/1993
Nacionalidade	brasileira
Nome em citações bibliográficas	Cruz, Lucas Puerta Cruz, Lucas
Currículo Lattes	http://lattes.cnpq.br/6866425119763705
ORCID	https://orcid.org/0009-0001-8802-946X
Outro identificador	URL
FORMAÇÃO ACADÊMICA	
2012/2015	Sistemas de informação Faculdade de informática e administração Paulista - FIAP
PRODUÇÃO BIBLIOGRÁFICA	
PARTICIPAÇÃO EM BANCAS E ORIENTAÇÕES	
Bancas de trabalhos de conclusão	
Orientações	
PARTICIPAÇÃO EM EVENTOS CIENTÍFICOS	