PhishFL: Uma solução Federada para Detecção de Phishing Baseada em BERT

Lucca F. T. Nolasco[1], **Andher Paulo C. Santana**[2], **Rodolfo I. Meneguette**[3]
Vinícius P. Gonçalves[3], **André Luiz M. Serrano**[3], **Geraldo P. Rocha Filho**[1]

[1]Universidade Estadual do Sudoeste da Bahia – UESB

[2]Universidade de Brasília – UnB

[3]Universidade de São Paulo – USP

202310250@uesb.edu.br, andher.santana@aluno.unb.br

{meneguette@icmc.usp.br, vpgvinicius, andrelms}@unb.br

geraldo.rocha@uesb.edu.br

Resumo. *Este trabalho aborda o problema da detecção automática de e-mails de phishing, que representa uma das principais ameaças à segurança digital. Métodos tradicionais de detecção, baseados em regras estáticas ou em dados centralizados, apresentam limitações quanto à adaptação e à preservação da privacidade dos usuários. Diante disso, é proposto o PhishFL, uma solução que utiliza Federated Learning (FL) para o treinamento distribuído de modelos de classificação textual, permitindo que múltiplos clientes colaborem na construção de um modelo global sem o compartilhamento direto de dados sensíveis. Os resultados demonstram que o modelo federado é capaz de realizar a detecção de e-mails de phishing com desempenho competitivo em relação à abordagem centralizada. Ainda, observa-se que o aumento do número de clientes impacta diretamente a estabilidade e a acurácia do modelo, evidenciando o trade-off entre desempenho e privacidade no contexto de FL.*

Abstract. *This work addresses the problem of automatic detection of phishing emails, which represents one of the main threats to digital security. Traditional detection methods, based on static rules or centralized data, have limitations regarding adaptation and preservation of user privacy. Therefore, PhishFL is proposed, a solution that uses Federated Learning (FL) for the distributed training of text classification models, allowing multiple clients to collaborate in building a global model without the direct sharing of sensitive data. The results demonstrate that the federated model is capable of detecting phishing emails with competitive performance compared to the centralized approach. Furthermore, it is observed that increasing the number of clients directly impacts the stability and accuracy of the model, highlighting the trade-off between performance and privacy in the context of FL.*

1. Introdução

O uso de e-mails como meio de comunicação continua sendo amplamente difundido em ambientes corporativos e pessoais, tornando-se um dos principais vetores para ataques

cibernéticos baseados em engenharia social [Wang et al. 2020, de Andrade et al. 2025]. Dentre esses ataques, o *phishing* destaca-se como uma das ameaças mais recorrentes, explorando vulnerabilidades humanas para induzir usuários a fornecer informações sensíveis, tais como credenciais de acesso e dados financeiros [Cuchta et al. 2019, de Andrade et al. 2025, Serrano et al. 2026]. Embora mecanismos tradicionais operem com regras estáticas, limitando sua adaptação, avanços em *Machine Learning* (ML) e Processamento de Linguagem Natural (PLN) têm viabilizado modelos mais eficazes para análise textual [Junnarkar et al. 2021], com destaque para arquiteturas Transformer [Turc et al. 2019, Bhargava et al. 2021].

Salienta-se, entretanto, que, apesar do potencial dessas abordagens, o problema para detectar *phishing* ainda enfrenta desafios relacionados à privacidade, escalabilidade e heterogeneidade dos dados. Em cenários reais, os e-mails encontram-se naturalmente distribuídos entre diferentes usuários e dispositivos, frequentemente sujeitos a restrições de compartilhamento por questões legais e de segurança. Nesse contexto, abordagens centralizadas tornam-se limitadas, pois exigem a agregação dos dados em um único repositório, aumentando o risco de exposição de informações sensíveis e dificultando a escalabilidade. Tais limitações reforçam a importância de paradigmas distribuídos, como o *Federated Learning* (FL), que permitem a construção de modelos globais sem o compartilhamento direto de dados sensíveis [Mammen 2021, de Oliveira et al. 2023].

Diversos trabalhos têm investigado a detecção de e-mails de *phishing* utilizando algoritmos tradicionais de ML, bem como abordagens baseadas em redes neurais profundas [Livara and Hernandez 2022, Rathee and Mann 2022, Andrade et al. 2024]. Embora essas soluções apresentem resultados promissores, a maioria delas adota uma arquitetura centralizada, desconsiderando aspectos fundamentais tais como privacidade dos dados, distribuição geográfica das informações e escalabilidade do treinamento. Abordagens mais recentes exploram modelos híbridos combinando BERT e redes neurais recorrentes, porém ainda mantendo dependência de centralização dos dados [Chinta et al. 2025, Alhuzali et al. 2025]. Diante desse cenário, observa-se uma lacuna na literatura quanto ao uso integrado de FL com modelos de linguagem avançados, tais como o BERT, para a detecção de e-mails de *phishing*.

Diante dessa lacuna, este trabalho propõe o PhishFL, uma solução federada para resolver o problema de detecção de e-mails de *phishing*, utilizando um modelo BERT ajustado para classificação textual. O PhishFL permite que o treinamento ocorra de forma distribuída nos dispositivos dos usuários, preservando a privacidade dos dados e promovendo maior diversidade de padrões de aprendizado. O modelo global é construído a partir da agregação dos parâmetros locais, possibilitando aprendizado colaborativo sem compartilhamento de dados brutos. Motivado por essas observações, este trabalho apresenta as seguintes contribuições: (i) a proposição de uma arquitetura federada para detecção de e-mails de *phishing* baseada em modelos BERT; e (ii) a avaliação do impacto do particionamento de dados e da variação do número de clientes no desempenho do modelo.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta como o PhishFL foi modelado. A Seção 3 apresenta os resultados obtidos para avaliar o PhishFL. Por fim, a Seção 4 apresenta as conclusões e os trabalhos futuros.

2. Solução Proposta

Esta seção apresenta o PhishFL, uma solução baseada em FL para tratar do problema de detecção de e-mails de *phishing*, na qual um modelo BERT foi modelado como classificador textual. Diferentemente de abordagens tradicionais, nas quais os dados são centralizados para processamento, o PhishFL realiza o treinamento de forma distribuída diretamente nos dispositivos dos clientes, utilizando os e-mails localmente disponíveis.

A Figura 1 apresenta o fluxo de funcionamento do PhishFL. Inicialmente, um modelo inicial é distribuído pelo servidor central (Passo 1). Após isso, os clientes dividem os dados disponíveis em conjuntos de treino e de teste. Com esses dados, cada dispositivo cliente utiliza as informações coletadas para o treinamento local de um modelo BERT (Passo 2) sem que os dados saiam do seu local de origem, dificultando a exposição e vazamento de dados sensíveis. Após o treinamento local, os dispositivos clientes transmitem os parâmetros atualizados de seus modelos ao servidor central (Passo 3), que realiza a agregação dos resultados com o FedAvg (Passo 4) e o redistribui aos clientes.

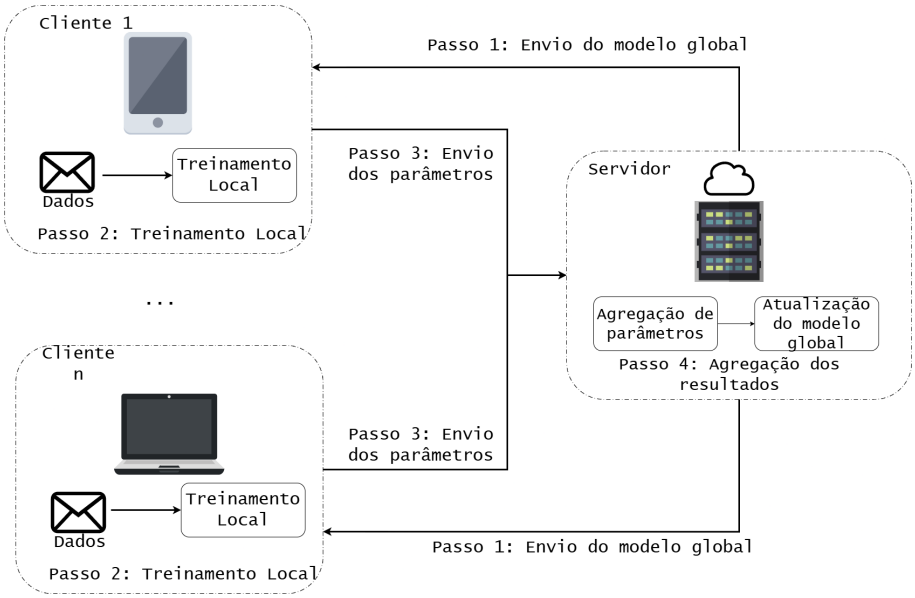


Figura 1. Visão geral do funcionamento do PhishFL

2.1. Formulação da Solução

Considere um conjunto de N clientes $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$, onde cada cliente c possui um conjunto de dados local $D_c = \{(x_i, y_i)\}$ composto por e-mails x_i e seus respectivos rótulos $y_i \in \{0, 1\}$, indicando se o e-mail é legítimo ou uma tentativa de *phishing*. O objetivo do PhishFL é aprender um modelo global parametrizado por w que minimize a função de perda agregada sobre todos os clientes, sem que os dados locais sejam compartilhados. Esse problema pode ser formalizado como:

$$\min_{w} \; F(w) = \sum_{c=1}^{N} \frac{|D_c|}{|D|} F_c(w) \tag{1}$$

onde $|D_c|$ representa o número de amostras no cliente c, $|D| = \sum_{c=1}^{N} |D_c|$ é o total de dados, e $F_c(w)$ é a função de perda local definida como:

$$F_c(w) = \frac{1}{|D_c|} \sum_{(x_i, y_i) \in D_c} \ell(f_w(x_i), y_i) \tag{2}$$

em que $f_w(\cdot)$ representa o modelo BERT parametrizado por w, e $\ell(\cdot)$ é a função de perda, tipicamente a entropia cruzada para classificação binária.

O processo de otimização é realizado de forma distribuída por meio do algoritmo *Federated Averaging* (FedAvg). Em cada rodada de comunicação r, um subconjunto de clientes $S^{(r)} \subseteq \mathcal{C}$ é selecionado para participar do treinamento. Cada cliente atualiza localmente o modelo global recebido $w^{(r-1)}$ por meio de múltiplas épocas de gradiente descendente, produzindo um modelo local $w_c^{(r)}$. Após as atualizações locais, o servidor central agrega os modelos recebidos de acordo com $w^{(r)} = \sum_{c \in S^{(r)}} \frac{|D_c|}{\sum_{k \in S^{(r)}} |D_k|} w_c^{(r)}$. Esse procedimento é repetido ao longo de R rodadas até a convergência do modelo global.

2.2. Modelo de Classificação Federado

O objetivo do PhishFL é detectar se um e-mail possui alta probabilidade de ser uma tentativa de *phishing* à partir de seu texto. Para realizar essa classificação, foi utilizado o modelo BERT adaptado utilizado por [Bhargava et al. 2021] e [Turc et al. 2019]. Esse modelo foi escolhido, em contra partida ao BERT tradicional, pois possui um *fine-tuning* para classificação binária à partir de entradas textuais

O Algoritmo 1 apresenta o funcionamento do PhishFL. O servidor começa inicializando um modelo global para detecção de *phishing* (por exemplo, uma rede neural com pesos $w^{(0)}$). Em cada rodada r, escolhe clientes disponíveis e envia a eles o modelo global mais recente $w^{(r-1)}$. Cada cliente treina esse modelo localmente usando apenas os textos dos próprios e-mails D_c por E épocas, gerando um modelo atualizado w_c (sem compartilhar os dados brutos). Em seguida, o cliente envia ao servidor apenas os pesos atualizados (e, no exemplo, também o tamanho do seu conjunto de dados n_c). O servidor então combina (agrega) todas as atualizações recebidas por meio de uma média ponderada (FedAvg), formando o novo modelo global $w^{(r)}$. Por fim, esse modelo global atualizado é enviado novamente aos clientes, repetindo o ciclo até completar R rodadas.

2.3. Dataset

O dataset utilizado nesta pesquisa é composto por e-mails rotulados quanto à sua legitimidade, contendo duas colunas principais: o texto do e-mail e a respectiva classificação (legítimo ou *phishing*) [Chakraborty 2023]. Ao todo, o conjunto de dados possui 18.650 amostras, sendo aproximadamente 60% e-mails legítimos e 40% classificados como *phishing*. No processo de pré-processamento, a variável de saída foi convertida para formato categórico, atribuindo-se valor 1 para e-mails legítimos e 0 para e-mails de *phishing*. Além disso, os textos foram preparados para entrada no modelo BERT, incluindo a tokenização por meio do método *WordPiece*, que permite a decomposição em subpalavras, favorecendo a representação de termos desconhecidos. A Tabela 1 apresenta exemplos de amostras do dataset após o pré-processamento.

Algorithm 1 Treinamento Federado para Detecção de E-mails de *phishing*

Require: $\mathcal{C} = \{c_1, \ldots, c_N\}$ clientes; R rodadas; E épocas locais; η taxa de aprendizado

Servidor inicializa o modelo global $w^{(0)}$
for $r \leftarrow 1$ **até** R **do**
  Servidor seleciona $S^{(r)} \subseteq \mathcal{C}$ e envia $w^{(r-1)}$ aos clientes
  for all $c \in S^{(r)}$ **em paralelo do**
    $w_c \leftarrow w^{(r-1)}$
    for $e \leftarrow 1$ **até** E **do**
      $w_c \leftarrow w_c - \eta \cdot \nabla \ell(w_c; D_c)$
    end for
    Cliente envia w_c e $n_c \leftarrow |D_c|$ ao servidor
  end for
  $w^{(r)} \leftarrow \dfrac{\sum_{c \in S^{(r)}} n_c \cdot w_c}{\sum_{c \in S^{(r)}} n_c}$
end for
return $w^{(R)}$

Tabela 1. Amostra de dados após pré-processamento

E-Mail Text	E-Mail Type
software at incredibly low prices (86 % lower) [...]	0
On Sun, Aug 11, 2002 at 11:17:47AM +0100, wintermute [...]	1
Question?Do you want a different job? [...]	0
sle 31 call for papers , sle 31 , st andrews , scotland , [...]	1

3. Avaliação de desempenho

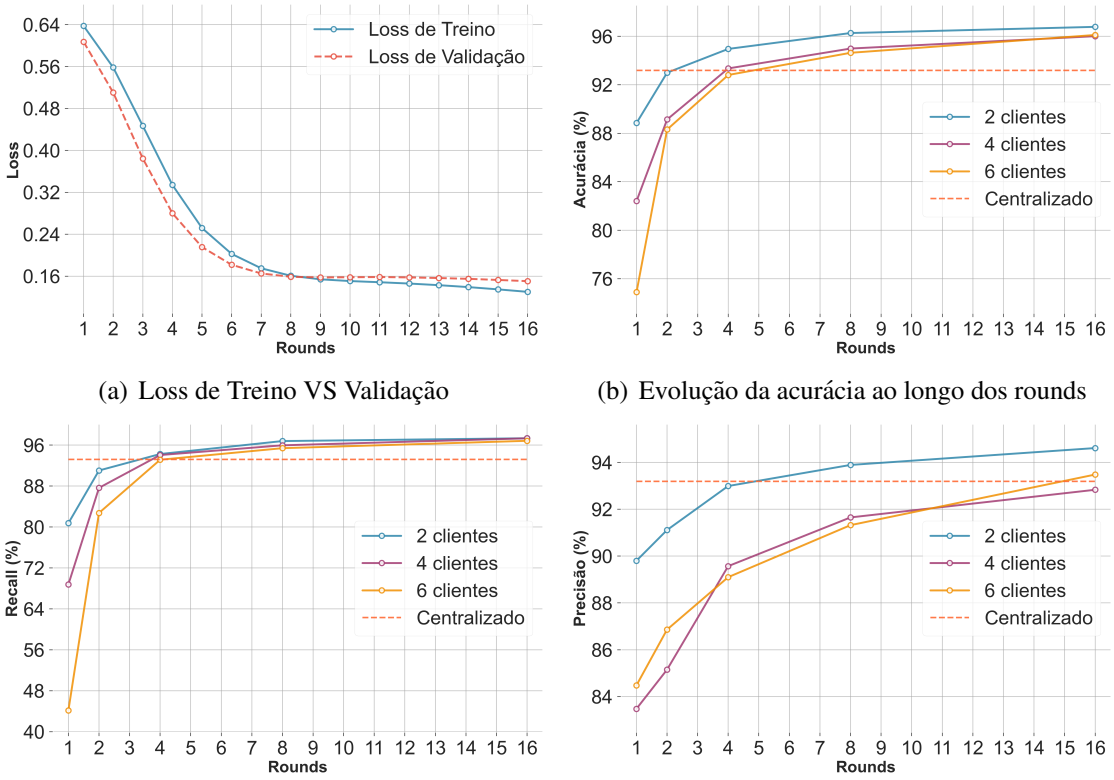
3.1. Configuração dos Experimentos

Nesta seção, avalia-se o desempenho do PhishFL na detecção de e-mails de *phishing*, em comparação com um modelo BERT treinado de forma centralizada. Os experimentos foram conduzidos utilizando a linguagem *Python* em conjunto com o *framework Flower*. Para investigar o comportamento do modelo em ambientes distribuídos, variou-se o número de clientes participantes em 2, 4 e 6, mantendo-se constantes os demais parâmetros experimentais, a fim de permitir comparações justas entre os diferentes níveis de distribuição dos dados.

A preparação dos dados foi realizada por meio de uma divisão estratificada do conjunto original, sendo 80% destinados ao treinamento e 20% ao teste. Posteriormente, os dados foram distribuídos de forma uniforme entre os clientes. O processo de treinamento federado foi executado ao longo de 16 *rounds* de comunicação entre clientes e servidor. Durante esse processo, foram avaliadas as seguintes métricas: (i) loss médio de treino dos clientes; (ii) acurácia; (iii) precisão; (iv) *recall*; e (v) tempo médio de treino dos clientes. Essas métricas permitem analisar tanto a qualidade preditiva quanto o custo computacional do modelo, possibilitando uma comparação direta entre as abordagens federada e centralizada. A seguir será apresentado o impacto dos resultados alcançados.

3.2. Impactos dos Resultados Obtidos

Na Figura 2(a) é apresentada a comparação entre o progresso do Loss de treino e validação ao longo de 16 rounds para a configuração de 6 clientes por representar o cenário com maior número de participantes. Observa-se que tanto o *loss* de treino quanto o de validação diminuem rapidamente dos *round* 1 ao *round* 7 enquanto aprendem padrões dominantes dos dados. Depois, passam a decrescer lentamente até se estabilizarem para realizar refinamentos leves. O comportamento semelhante entre as curvas do *loss* de treino e validação apontam uma boa capacidade de generalização do PhishFL.



(a) Loss de Treino VS Validação

(b) Evolução da acurácia ao longo dos rounds

(c) Evolução do *recall* ao longo dos rounds

(d) Evolução da precisão ao longo dos rounds

Figura 2. Métricas de desempenho do PhishFL

É apresentada na Figura 2(b) a comparação na evolução da acurácia do modelo agregado ao longo dos 16 *rounds* e do modelo centralizado de controle. Observa-se uma evolução rápida nos primeiros 4 rounds, que reduz sua intensidade até o round 8 e se estabiliza até o round 16. Esse comportamento e o das demais métricas abaixo pode ser explicado pelo caráter iterativo do FL no qual a agregação das sucessivas atualizações dos clientes tende a aprimorar a capacidade de classificação do PhishFL até um patamar de estabilização. No pior caso federado, houve ume melhora de 2,79 pontos percentuais em relação ao modelo centralizado.

Em relação ao *Recall*, o PhishFL superou o modelo centralizado já no *round* 4, como apresentado na Figura 2(c), com um crescimento estável até o *round* 16. Na classificação de e-mails de *phising,* um falso negativo apresenta graves riscos à segurança do usuário, tornando essencial um valor alto para o *recall*, característica presente no PhishFL. No pior caso, superou o modelo centralizado em 3,62 pontos percentuais.

A Figura 2(d) apresenta a comparação do crescimento da Precisão ao longo dos 16 *rounds* com o modelo centralizado. Observa-se uma curva de crescimento mais acentuada que demora mais a se estabilizar, chegando a valores próximos apenas ao fim do experimento. Esse comportamento pode ser explicado devido à menor quantidade de dados por cliente e à quantidade de e-mails maliciosos ser superior à de e-mails seguros.

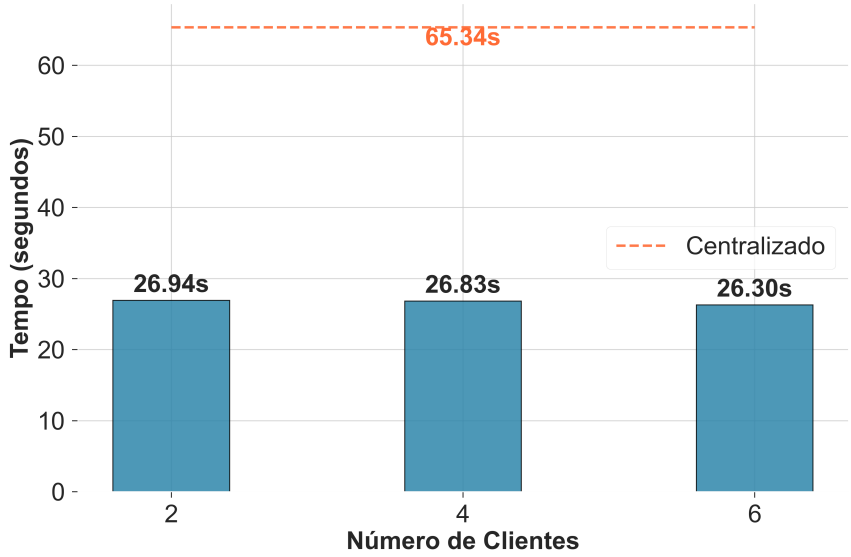


Figura 3. Comparação entre o tempo de treino do modelo centralizado e os tempos médios de treino dos clientes

Os resultados, apresentados na Figura 3, indicam que o treinamento centralizado apresentou o maior custo temporal, com 65,34 segundos. Em contraste, no cenário federado, observou-se um tempo médio de treinamento por cliente substancialmente menor, sendo de 26,94 s para 2 clientes, 26,83 s para 4 clientes e 26,30 s para 6 clientes. O propósito desta comparação é evidenciar o impacto do particionamento e do paralelismo do aprendizado federado sobre a duração do treinamento local, além de situar tais valores frente ao modelo centralizado de controle.

4. Conclusão

Este artigo propôs o PhishFL, uma solução para a detecção de e-mails de *phishing* baseada em aprendizado federado. Utilizando um modelo BERT para classificação, a solução foi avaliada em diferentes configurações de treinamento e comparada à abordagem centralizada. Os resultados evidenciaram desempenho consistente, com destaque para o *recall*, além de estabilidade e convergência ao longo das rodadas. Observou-se, ainda, eficiência computacional nas configurações federadas analisadas. Dessa forma, o PhishFL mostrou-se uma abordagem eficaz para a detecção de *phishing*, ao conciliar bom desempenho preditivo e treinamento descentralizado.

Referências

Alhuzali, A., Alloqmani, A., Aljabri, M., and Alharbi, F. (2025). In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets. *Applied Sciences*, 15(6):3396.

Andrade, C. A., Rocha Filho, G. P., Meneguette, R. I., Maranhão, J. P. A., Sant'Ana, R., Duarte, J. C., Serrano, A. L. M., and Gonçalves, V. P. (2024). Fortunate: Decrypting and classifying malware by variable length instruction sequences. In *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*, pages 1–9. IEEE.

Bhargava, P., Drozd, A., and Rogers, A. (2021). Generalization in nli: Ways (not) to go beyond simple heuristics.

Chakraborty, S. (2023). Phishing email detection.

Chinta, P. C. R., Moore, C. S., Karaka, L. M., Sakuru, M., Bodepudi, V., and Maka, S. R. (2025). Building an intelligent phishing email detection system using machine learning and feature engineering. *European Journal of Applied Science, Engineering and Technology*, 3(2):41–54.

Cuchta, T., Blackwood, B., Devine, T. R., Niichel, R. J., Daniels, K. M., Lutjens, C. H., Maibach, S., and Stephenson, R. J. (2019). Human risk factors in cybersecurity. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, SIGITE '19, page 87–92, New York, NY, USA. Association for Computing Machinery.

de Andrade, C. A. B., Rocha Filho, G. P., Meneguette, R. I., Sant'Ana, R., Duarte, J. C., Serrano, A. L. M., Neumann, C., and Gonçalves, V. P. (2025). Forensics: Deciphering and detecting malware through variable-length instruction sequences. *Journal of Internet Services and Applications*, 16(1).

de Oliveira, J. A., Gonçalves, V. P., Meneguette, R. I., de Sousa Jr, R. T., Guidoni, D. L., Oliveira, J. C., and Rocha Filho, G. P. (2023). F-nids—a network intrusion detection system based on federated learning. *Computer Networks*, 236:110010.

Junnarkar, A., Adhikari, S., Fagania, J., Chimurkar, P., and Karia, D. (2021). E-mail spam classification via machine learning and natural language processing. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 693–699.

Livara, A. and Hernandez, R. (2022). An empirical analysis of machine learning techniques in phishing e-mail detection. In *2022 International Conference for Advancement in Technology (ICONAT)*, pages 1–6.

Mammen, P. M. (2021). Federated learning: Opportunities and challenges.

Rathee, D. and Mann, S. (2022). Detection of e-mail phishing attacks - using machine learning and deep learning. *International Journal of Computer Applications*, 14:513–535.

Serrano, A. L. M., Rodrigues, G. A. P., Rocha Filho, G. P., Gonçalves, V. P., Bonacin, R., Bispo, G. D., Peixoto, M. G. M., and Meneguette, R. I. (2026). Efficient and lightweight phishing detection: A case for sustainable cybersecurity with tf-idf and lightgbm. *IEEE Access*, 14:55458–55471.

Turc, I., Chang, M., Lee, K., and Toutanova, K. (2019). Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Wang, Z., Sun, L., and Zhu, H. (2020). Defining social engineering in cybersecurity. *IEEE Access*, 8:85094–85115.