Uma Arquitetura de Aprendizado Federado Homomórfica para Treinamento de LSSVM usando CKKS e Decomposição QR de Householder

Victor Faria Fernandes[1], **Luis Antonio Kowada**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

victor_fernandes@id.uff.br, luis@ic.uff.br

Abstract. *This work proposes a One-Shot Homomorphic Federated Learning architecture for training Least Squares Support Vector Machines (LSSVM) using the CKKS cryptographic scheme. By reducing training to a linear system, we employ the Householder QR decomposition to solve KKT conditions directly on encrypted data, intrinsically avoiding branching operations incompatible with homomorphic encryption. We evaluate the architecture using OpenFHE on two datasets — Iris and the Breast Cancer Wisconsin Diagnostic dataset (WDBC) — under both IID and non-IID (Dirichlet) client partitioning, scaling from 40 to 225 clients. The federated FHE solution closely tracks its plaintext federated counterpart across all configurations (relative parameter error below* 3×10^{-4} *for WDBC and below* $3{,}5\%$ *for Iris' polynomial-kernel classes), reaching* 100% *accuracy on the linearly separable Iris class and up to* $95{,}6\%$ *accuracy on WDBC under non-IID partitioning. This approach completely eliminates multiple communication rounds, ensuring robust data privacy and algorithmic stability across dataset scale, client count, and data distribution.*

Resumo. *Este trabalho propõe uma arquitetura de Aprendizado Federado Homomórfico One-Shot para o treinamento de Máquinas de Vetores de Suporte de Mínimos Quadrados (LSSVM) usando o esquema CKKS. Ao reduzir o treinamento a um sistema linear, emprega-se a decomposição QR de Householder para resolver o sistema KKT sobre dados cifrados, evitando ramificações estruturais incompatíveis com a criptografia homomórfica. A arquitetura é avaliada via OpenFHE em dois datasets — Iris e o Breast Cancer Wisconsin Diagnostic (WDBC) — sob particionamento IID e não-IID (Dirichlet) entre clientes, com escalabilidade testada de 40 a 225 clientes. A solução FHE federada acompanha de perto sua contraparte federada em texto claro em todas as configurações (erro relativo dos parâmetros abaixo de* 3×10^{-4} *para o WDBC e abaixo de* $3{,}5\%$ *para as classes de kernel polinomial do Iris), atingindo* 100% *de acurácia na classe linearmente separável do Iris e até* $95{,}6\%$ *de acurácia no WDBC sob particionamento não-IID. A abordagem elimina múltiplas rodadas de comunicação, garantindo privacidade robusta de dados e estabilidade algorítmica através de diferentes escalas de dataset, números de clientes e distribuições de dados.*

1. Introdução

A criptografia homomórfica (FHE) consolida-se como tecnologia crítica para aprendizado de máquina seguro. Destaca-se o esquema CKKS [Cheon et al. 2017], que processa dados cifrados em formato de ponto flutuante, fundamentando-se na intratabilidade do problema RLWE (*Ring Learning With Errors*) [Lyubashevsky et al. 2013]. Alinhado a essa capacidade de computação sobre dados privados, este trabalho propõe uma arquitetura de treinamento federado homomórfico para a variante *Least Squares Support Vector Machine* (LSSVM) [Suykens and Vandewalle 1999]. A escolha da LSSVM justifica-se pela simplificação do processo de otimização: o treinamento é reduzido à solução de um sistema linear, em contraste com a programação quadrática convexa exigida pelo SVM convencional, viabilizando a integração eficiente com criptografia homomórfica.

De forma intuitiva, uma SVM busca o hiperplano $w \cdot x + b = 0$ que separa duas classes de dados maximizando a distância (margem) até os pontos mais próximos de cada classe, onde o vetor w é normal a esse hiperplano. A variante LSSVM substitui as restrições de desigualdade e a função de perda *hinge* — não-polinomiais — por restrições de igualdade e um termo de erro quadrático, aproximando o problema de uma regressão. Essa reformulação faz com que as condições de otimalidade de Karush–Kuhn–Tucker (KKT), que caracterizam a solução ótima do problema, colapsem em um único sistema linear, em vez de exigirem um processo iterativo de programação quadrática. A decomposição QR de Householder é então empregada para resolver esse sistema de forma direta e numericamente estável, sem recorrer à inversão explícita de matrizes — operação particularmente custosa e instável no domínio homomórfico.

As contribuições deste trabalho são: (i) a formulação do circuito aritmético FHE para resolução do sistema de Karush–Kuhn–Tucker (KKT) [Nocedal and Wright 2006] da LSSVM via decomposição QR de Householder no domínio cifrado; (ii) uma estratégia de cálculo de limites de Chebyshev em texto claro antes da cifragem, garantindo estabilidade das avaliações não-lineares; e (iii) a avaliação experimental da arquitetura em dois *datasets* (Iris e WDBC), sob particionamento IID e não-IID (Dirichlet) entre clientes, com escalabilidade testada de 40 a 225 clientes e extensão a *kernel* polinomial homogêneo de grau 2 para classes não linearmente separáveis.

2. Trabalhos Relacionados

A intersecção entre criptografia homomórfica e aprendizado de máquina pode ser organizada em dois paradigmas distintos, conforme classificação de [Iezzi 2020]: *Private Prediction as a Service* (PPaaS), no qual a predição sobre dados cifrados ocorre com um modelo pré-treinado em texto claro; e *Private Training as a Service* (PTaaS), no qual o próprio treinamento do modelo ocorre sobre dados cifrados. O presente trabalho insere-se no segundo paradigma, que impõe requisitos computacionais substancialmente maiores, mas oferece garantias de privacidade mais abrangentes.

PPaaS: Inferência sobre dados cifrados

Trabalhos como o de [Graepel et al. 2012] demonstraram a viabilidade da inferência homomórfica para classificadores lineares, paradigma no qual o treinamento ocorre em texto claro e apenas a predição recebe dados cifrados. Mais recentemente, [Buchanan and Ali 2025] avaliaram SVMs com o esquema CKKS via OpenFHE nesse

mesmo modelo (PPaaS), reportando um *overhead* de $\approx 10^3\times$ em relação à inferência em texto claro — custo fortemente influenciado pela dimensão do anel e tamanho do módulo. Contudo, em todas essas abordagens, o servidor acessa o modelo treinado em texto claro. Essa exposição dos parâmetros (e, indiretamente, dos dados de treinamento) representa uma limitação crítica para cenários federados com rigorosos requisitos de privacidade.

PTaaS: Treinamento sobre dados cifrados

O treinamento homomórfico impõe desafios severos por exigir circuitos aritméticos profundos. [Chen et al. 2018] exploraram a regressão logística via gradiente descendente com aproximações de Chebyshev (desafio análogo ao da Seção 5.3). Para SVM, [Park et al. 2020] viabilizaram o primeiro treinamento FHE adotando a formulação LS-SVM (compatível com HE) e resolvendo o sistema linear via **gradiente descendente iterativo** em cenário de cliente único. Contudo, métodos iterativos — limitação também presente nas soluções de [Han and Ki 2020] para sistemas lineares gerais — exigem profundidade multiplicativa crescente e são sensíveis à taxa de aprendizado. O presente trabalho inova estruturalmente ao estender o modelo para o cenário federado (k clientes) utilizando um método **direto** (decomposição QR de Householder). Essa abordagem resolve o sistema KKT em uma passagem única e determinística, substituindo as sucessivas iterações por uma profundidade multiplicativa fixa e previsível (Seção 5.1).

Decomposição QR e Aprendizado Federado sob FHE

Em computação multipartidária segura (SMPC), [Hartebrodt and Röttger 2023] demonstraram que a transformação de Householder vaza dados dos clientes durante agregações parciais, sendo evitada nesse domínio. Contudo, a arquitetura aqui proposta contorna essa limitação estruturalmente: ao executar a decomposição inteiramente sob FHE (CKKS) no servidor central, não há reconstrução de textos claros ou exposição de somas parciais.

No contexto federado, abordagens tradicionais baseadas em privacidade diferencial (DP) [McMahan et al. 2019, Geyer et al. 2018] tendem a degradar a precisão do modelo devido à injeção de ruído. Por outro lado, soluções recentes que unem FHE e aprendizado federado, como o protocolo xMK-CKKS [Ma et al. 2022], mantêm o paradigma iterativo do *FedAvg*. Essas abordagens exigem dezenas de rodadas de comunicação e expõem o modelo a ataques de reconstrução de gradientes [Zhu et al. 2019].

Diferentemente dessas abordagens iterativas, este trabalho inova ao combinar a resolução direta do sistema KKT (via Householder) com um modelo federado *One-Shot*. Essa união exige apenas uma rodada de transmissão por cliente para os parâmetros finais, mitigando vetores de ataque, reduzindo a sobrecarga de rede e dispensando a complexidade de esquemas de chaves múltiplas (*multi-key*).

3. Fundamentação Teórica

A segurança dos esquemas de criptografia homomórfica fundamenta-se em problemas matemáticos estabelecidos, como o LWE (*Learning With Errors*) e sua variante em anéis, o RLWE (*Ring Learning With Errors*) [Lyubashevsky et al. 2013]. Tais problemas possuem reduções quânticas para os piores casos do SVP (*Shortest Vector Problem*), conferindo-lhes resistência a ataques de computação quântica. Essa segurança é sustentada pelo fato de que problemas de reticulados (*lattices*), como o SVP e o SIS (*Short*

Integer Solution) [Micciancio and Goldwasser 2002], não possuem algoritmos quânticos eficientes conhecidos até o momento.

Além desse nível de segurança, o diferencial da criptografia homomórfica reside na capacidade de processar dados em estado cifrado. Atualmente, existem diferentes famílias de criptossistemas projetadas para tipos específicos de dados e operações.

3.1. O Esquema CKKS

O esquema CKKS [Cheon et al. 2017], fundamentado na intratabilidade do problema RLWE, destaca-se pelo processamento homomórfico de vetores de números reais e complexos. Suas principais propriedades incluem: (i) suporte a operações aritméticas nativas; (ii) aproximação de funções contínuas por meio de interpolação polinomial (ex., Chebyshev); e (iii) eficiência computacional via empacotamento (*batching*), que permite a execução de operações vetoriais SIMD (*Single Instruction, Multiple Data*).

A cifra opera em um anel de polinômios ciclotômicos e engloba funções criptográficas fundamentais: geração de chaves ($KeyGen$, produzindo a chave pública pk, privada sk e de avaliação evk), além de codificação e cifragem sob um fator de escala Δ. No domínio homomórfico, as operações centrais para este trabalho são a adição (Add), a multiplicação ($Mult$) — que exige a chave evk para relinearização — e a reescala ($Rescaling$), passo crucial para conter o crescimento exponencial de Δ após sucessivas multiplicações.

Essas operações formam a base para a resolução do sistema linear da LSSVM em estado cifrado. Para abstrair a manipulação algébrica de baixo nível, a implementação deste trabalho utiliza a biblioteca OpenFHE [Badawi et al. 2022], que também provê suporte nativo à paralelização via OpenMP.

3.2. LSSVM

A formulação clássica da Máquina de Vetores de Suporte (SVM), em formato primal, consiste em um problema de programação quadrática convexa [Boyd and Vandenberghe 2004]. Geometricamente, busca-se o hiperplano $w \cdot x + b = 0$ que separa as duas classes com a maior margem possível, onde w é o vetor normal ao hiperplano; maximizar essa margem equivale a minimizar $\|w\|^2$. Como dados reais raramente são perfeitamente separáveis, introduzem-se variáveis de folga ξ_i, que permitem que pontos individuais violem a margem a um custo proporcional. Assim, o problema consiste em maximizar a margem de separação minimizando $\frac{1}{2}\|w\|^2$ sujeito a restrições de desigualdade com variáveis de folga (ξ_i). A variante *Least Squares Support Vector Machine* (LSSVM) segue uma lógica distinta: em vez de penalizar violações de margem via a função de perda *hinge* (não-polinomial), ela substitui as restrições de desigualdade por igualdades e adota a soma dos erros quadráticos como função de custo [Suykens and Vandewalle 1999], aproximando o problema de uma regressão linear regularizada. Isso reduz o treinamento à solução do sistema KKT (*Karush-Kuhn-Tucker*) [Nocedal and Wright 2006] e contorna a programação quadrática convexa.

Dada uma base com N amostras $\{x_i, y_i\}_{i=1}^{N}$, onde $x_i \in \mathbb{R}^d$ e $y_i \in \{-1, 1\}$, a LSSVM formula o problema primal como:

$$\min_{w,b,e} \mathcal{J}(w,e) = \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \tag{1}$$

sujeito a:

$$y_i(w^T\phi(x_i) + b) = 1 - e_i, \quad i = 1, \ldots, N \tag{2}$$

onde w é o vetor de pesos, b o viés, e_i a variável de erro para x_i, γ o parâmetro de regularização e $\phi(x_i)$ o mapeamento para espaço de alta dimensão.

Como o objetivo (1) é quadrático e as restrições (2) são afins, as condições de otimalidade de primeira ordem (KKT) — que anulam simultaneamente o gradiente do Lagrangiano em relação a w, b, e e aos multiplicadores α — resultam diretamente em um sistema de equações lineares, sem necessidade de um processo iterativo. Aplicando as condições KKT e o *Kernel trick* — $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ — constrói-se a matriz $\Omega_{ij} = y_i y_j K(x_i, x_j)$, resultando no sistema linear:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \tag{3}$$

onde $\mathbf{1} \in \mathbb{R}^N$, I é a identidade $N \times N$, $\alpha \in \mathbb{R}^N$ é o vetor de multiplicadores de Lagrange e $Y \in \mathbb{R}^N$ o vetor de rótulos.

Neste trabalho, adota-se mapeamentos explícitos em formato primal: kernel linear para problemas linearmente separáveis e kernel polinomial homogêneo de grau 2 ($K(x_i, x_j) = (x_i^T x_j)^2$) para classes não-linearmente separáveis, expandido como produto cartesiano de características. Em ambos os casos, recupera-se w explicitamente a partir dos multiplicadores α:

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{4}$$

Ao materializar w na memória, a inferência para novas amostras x reduz-se a $w^T\phi(x) + b$, dispensando a reavaliação do kernel sobre as amostras de treinamento.

3.3. Decomposição QR de Householder

Uma reflexão de Householder é uma transformação ortogonal $H = I - 2\,\frac{vv^T}{v^T v}$, definida por um vetor v, que reflete um vetor x em torno de um hiperplano de modo a zerar todas as suas entradas exceto a primeira: $Hx = \pm\|x\|_2\, e_1$. Aplicando sucessivamente uma reflexão de Householder a cada coluna de uma matriz A, é possível triangularizá-la, obtendo $Q^T A = R$ com Q ortogonal e R triangular superior — a decomposição $A = QR$.

Essa decomposição permite resolver um sistema linear $Ax = c$ sem calcular A^{-1}: basta computar $Q^T c$ e, em seguida, obter x por retrosubstituição em $Rx = Q^T c$, já que R é triangular. No contexto deste trabalho, essa propriedade é decisiva: como discutido na Seção 3, a inversão explícita de matrizes é numericamente instável e proibitivamente custosa no domínio homomórfico, enquanto a decomposição de Householder reduz

o problema a um número fixo de operações primitivas por coluna — normas, produtos internos e uma única raiz quadrada e um único recíproco por pivô —, todas passíveis de aproximação polinomial estável via limites de Chebyshev pré-calculados (Seção 5.3).

4. Metodologia e Algoritmo Proposto

Propõe-se uma arquitetura de Treinamento Delegado Seguro (*Secure Outsourced Training*) do tipo *One-Shot*, integrada ao esquema CKKS, conforme ilustrado na Figura 1. O algoritmo opera sobre subproblemas de classificação binária (estratégia *One-vs-Rest* — OvR): os clientes limitam-se à construção e cifragem dos dados locais, delegando ao servidor o treinamento homomórfico e a agregação dos modelos.

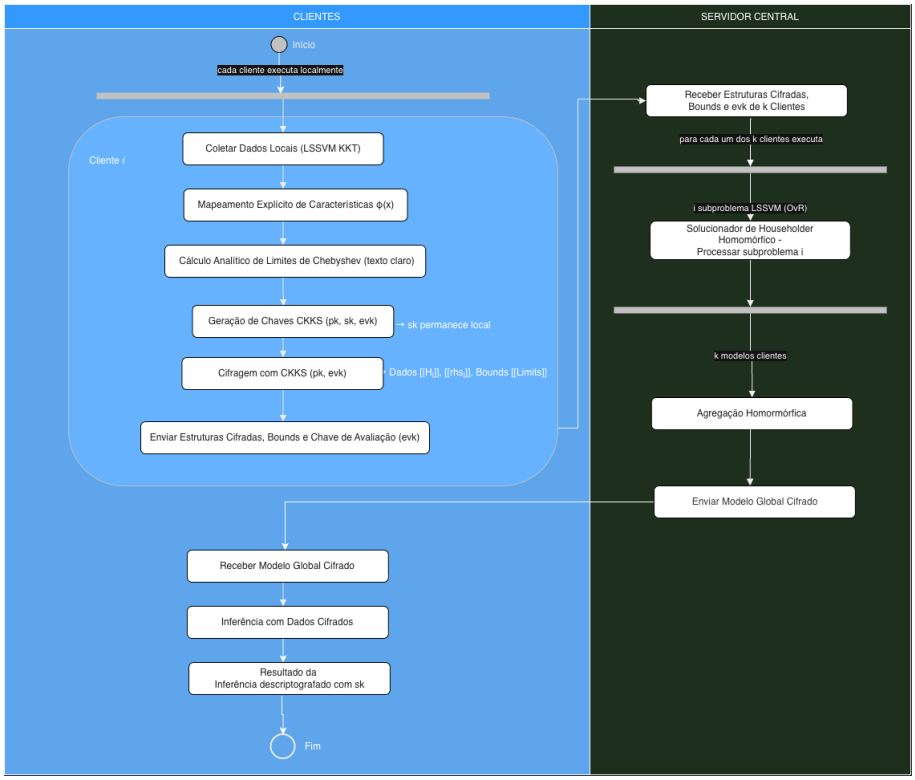

Figura 1. Fluxo de processamento entre k clientes e o servidor central. Utiliza-se um único contexto criptográfico e chaves CKKS compartilhadas (pk, sk, evk), distribuídos previamente de forma confiável (Seção 5.2).

A metodologia divide-se em quatro etapas: pré-processamento local, cifragem, treinamento centralizado e agregação homomórfica.

4.1. Pré-processamento e Mapeamento Explícito de Características

O uso de funções de kernel implícitas no contexto federado exigiria a transmissão dos vetores de suporte ao servidor, violando a premissa de privacidade. Para contornar esse obstáculo, aplica-se uma função de mapeamento explícita $\phi(x)$, que projeta as amostras diretamente no espaço primal. Para um kernel polinomial homogêneo de grau d, as interações entre características são pré-calculadas localmente como produto cartesiano de monômios de grau d, garantindo que w possua dimensionalidade idêntica para todos os clientes e viabilizando a combinação linear no servidor.

4.2. Cifragem Local e Treino no Servidor Central

Considere k clientes, cada um com subconjunto disjunto de dados. O processo ocorre nas seguintes etapas:

- **Construção e Cifragem Local:** O cliente i monta a matriz H_i e o vetor rhs_i, cifra-os com o contexto CKKS pré-estabelecido — par de chaves (pk, sk) único e compartilhado entre os clientes, conforme detalhado na Seção 5.2 — e transmite $[\![H_i]\!]$, $[\![rhs_i]\!]$ e as chaves de avaliação ao servidor.
- **Resolução Cifrada:** O servidor resolve o sistema KKT cifrado via decomposição QR de Householder, cuja compatibilidade homomórfica foi estabelecida na Seção 3.3.
- **Saída:** O servidor devolve $[\![w_i]\!]$ e $[\![b_i]\!]$ para cada modelo i, em estado cifrado.

4.3. Agregação Federada (One-Shot FedAvg)

Após o treinamento dos k modelos, o servidor agrega os parâmetros por média homomórfica. Esta formulação pressupõe que cada cliente i contribui com o mesmo número de amostras ($n_i = N/k$); em cenários heterogêneos, a agregação deve ser ponderada por n_i/N:

$$[\![w_{global}]\!] = \frac{1}{k}\sum_{i=1}^{k}[\![w_i]\!] \qquad (5) \qquad\qquad [\![b_{global}]\!] = \frac{1}{k}\sum_{i=1}^{k}[\![b_i]\!] \qquad (6)$$

Como a técnica é *One-Shot*, não há múltiplas rodadas de comunicação, reduzindo a sobrecarga de rede. O modelo global pode ser avaliado no servidor ou distribuído aos clientes para inferência em domínio cifrado ou em texto claro após decifragem local.

5. Análise de Complexidade e Segurança

5.1. Análise de Complexidade

A maior carga computacional reside no servidor central, originada pela profundidade multiplicativa exigida pelo circuito aritmético das transformações de Householder. A Tabela 1 detalha o consumo de níveis por categoria de operação em função da dimensão da matriz, contextualizando a profundidade configurada de 186 níveis adotada nos experimentos.

Tabela 1. Profundidade multiplicativa estimada por tamanho de matriz ($D_{\text{sqrt}} = D_{\text{inv}} = D_{\text{backsub}} = 8$; fator de segurança $\times 1{,}3$). As avaliações de Chebyshev consomem $\delta = 2\lceil \log_2(D+1) \rceil = 8$ níveis.

Componente	4×4	5×5	6×6
QR: $n\,(\delta_{\text{sqrt}} + \delta_{\text{inv}} + 1)$	68	85	102
Retrosubstituição.: $n\,(\delta_{\text{backsub}} + 1)$	36	45	54
Overhead de base	8	8	8
Total $\lceil \cdot \times 1{,}3 \rceil$	**146**	**180**	**214**

A profundidade utilizada nos experimentos foi parametrizada com base nestes cálculos. Para o cenário com $k = 40$ clientes, os clientes processam matrizes de dimensão 4×4 (estimativa analítica de 146 níveis), enquanto o baseline com cliente único comparativo emprega matrizes 5×5 (estimativa de 180 níveis). A configuração empírica

de 186 níveis adotada absorve ambos os cenários e fornece margem de segurança para as operações do OpenFHE, garantindo a execução completa do circuito sem necessidade de *bootstrapping* intermediário.

A agregação homomórfica dos k modelos exige exatamente $k-1$ adições homomórficas e uma multiplicação por escalar $(1/k)$, escalando linearmente com k e representando custo negligenciável frente ao treinamento.

5.2. Análise de Segurança

O modelo de ameaça assume um servidor honesto-curioso (*honest-but-curious*) [Malekzadeh et al. 2021], que segue o protocolo corretamente, mas tenta inferir dados a partir dos textos cifrados ($[\![H_i]\!]$, $[\![rhs_i]\!]$) e dos intervalos de Chebyshev recebidos em texto claro. Como os clientes compartilham um único par de chaves, a confidencialidade interna depende do não-vazamento da chave privada (sk). Ataques ativos, como adulteração de mensagens, fogem ao escopo e constituem trabalhos futuros (Seção 7).

A privacidade dos dados locais reduz-se à segurança IND-CPA do esquema CKKS, fundamentada na dificuldade computacional do problema RLWE [Cheon et al. 2017]. Como o servidor acessa unicamente a chave de avaliação e os textos cifrados, qualquer capacidade de inferência implicaria a quebra estrutural do CKKS. Por fim, embora os intervalos de Chebyshev revelem meta-informações limitadas (magnitude de H_i), cenários com requisitos estritos de privacidade podem substituí-los por limites globais conservadores pré-acordados, aceitando-se, em contrapartida, certa imprecisão nas aproximações polinomiais.

Modelo de chaves. A arquitetura utiliza um único par de chaves (pk, sk) compartilhado entre todos os clientes, preterindo esquemas *multi-key*. Essa escolha viabiliza a agregação homomórfica direta dos parâmetros com baixo custo computacional. Contudo, como a chave privada (sk) é acessível a todos, o modelo de ameaça assume que os participantes são honesto-curiosos, pois qualquer cliente poderia decifrar os dados dos demais. Consequentemente, a abordagem é ideal para cenários de consórcio com confiança mútua (como redes de hospitais ou filiais corporativas), sendo inadequada para ambientes com clientes mutuamente desconfiados.

A arquitetura mitiga os seguintes vetores de ataque:

1. **Privacidade dos Dados:** O servidor opera como um executor cego, processando o treinamento exclusivamente sobre dados cifrados, sem qualquer acesso ao texto claro.
2. **Isolamento de Chaves:** O servidor utiliza apenas pk e evk. A chave privada (sk) permanece restrita aos clientes, impossibilitando a decifragem de resultados intermediários.
3. **Superfície de Ataque Reduzida:** Ao exigir uma única transmissão por cliente, a abordagem *One-Shot* elimina estruturalmente os ataques de reconstrução de gradientes, comuns em arquiteturas iterativas [Zhu et al. 2019].
4. **Integridade Polinomial:** O cálculo dinâmico de limites (Seção 5.3) evita o crescimento explosivo do erro na aproximação de Chebyshev, prevenindo ataques de negação de serviço (DoS) e o envenenamento do modelo.

5.3. Parametrização e Cálculo dos Limites de Chebyshev

Para garantir a estabilidade das avaliações não-lineares, este trabalho implementa uma estratégia de cálculo dos limites de aproximação em texto claro antes da cifragem. O domínio $[a, b]$ deve ser definido antes da transição para o domínio cifrado. O cálculo ocorre nas duas operações centrais do método de Householder:

- **Raiz Quadrada:** O domínio é obtido por simulação em texto claro. O limite superior é a soma dos quadrados dos elementos da coluna k ($\sum x_i^2$), definindo o intervalo $[0, \sum x_i^2]$.
- **Inversão de Pivôs Diagonais:** Simula-se a decomposição QR em texto claro e extrai-se cada diagonal $d = R_{i,i}$, aplicando margem de $\pm 10\%$ para absorver ruído homomórfico:
  - Diagonais positivas: $lo = d \times 0{,}90, \quad hi = d \times 1{,}10$.
  - Diagonais negativas: $lo = -|d| \times 1{,}10, \quad hi = -|d| \times 0{,}90$.

  Essa formulação garante $lo < hi$ e que o intervalo não intercepte a origem, requisito para estabilidade da inversão homomórfica.

Vale observar que a convenção de sinal adotada (sign $= +1$ fixo) implica que, após a reflexão na coluna k, o pivô diagonal satisfaz $R_{k,k} = -\|x\|_2 < 0$ para qualquer dado não nulo. Portanto, na prática, todos os intervalos são negativos e o domínio de aproximação de Chebyshev para $1/t$ nunca cruza a origem por construção, desde que $\|x\|_2$ seja suficientemente distante de zero.

Os limites são calculados por cada cliente sobre seu próprio bloco de dados e transmitidos ao servidor junto com os dados cifrados, dispensando decifrações intermediárias.

6. Resultados Experimentais

6.1. Ambiente de Execução e Parametrização

Todos os experimentos — tanto em texto claro quanto no domínio cifrado (FHE) — foram executados em um MacBook Air 13" (Apple M4, 2025) com 24 GB de RAM e 494 GB de armazenamento, sob macOS Sequoia 15.3. As operações homomórficas utilizaram a biblioteca OpenFHE v1.5.1, com paralelização configurada em 4 *threads* via OpenMP (`OMP_NUM_THREADS=4`). Por terem sido conduzidos no mesmo hardware, os fatores de overhead discutidos a seguir constituem uma comparação direta. O código-fonte integral, bem como os scripts configurados para reproduzir os resultados apresentados nesta seção, estão disponíveis publicamente no repositório: `https://github.com/victorfariafernandes/homomorphic-cli/tree/sbseg-public-release`.

O cenário experimental simula $k = 40$ clientes com particionamento disjunto dos dados. O dataset Iris [Fisher 1936] foi embaralhado e estratificado em 120 amostras de treinamento (distribuídas uniformemente entre os clientes, resultando em 3 amostras por cliente) e 30 amostras para validação centralizada, com classificação multiclasse via estratégia *One-vs-Rest* (OvR) sobre 3 classes. As quatro características (comprimento e largura de sépala e pétala, em centímetros) são variáveis numéricas contínuas, e o rótulo de classe é categórico nominal, correspondendo a uma das três espécies de *Iris*.

Como segunda base de avaliação, utiliza-se também o dataset *Breast Cancer Wisconsin (Diagnostic)* (WDBC) [Wolberg et al. 1993], composto por 569 amostras de

biópsias de tumores mamários. Cada amostra é descrita por 30 características numéricas contínuas extraídas de imagens digitalizadas de núcleos celulares (ex.: raio, textura, perímetro, área e simetria, cada uma reportada como média, erro-padrão e valor extremo entre os núcleos observados em uma imagem), e o rótulo de classe é binário (*benigno* ou *maligno*) — configurando, ao contrário do cenário multiclasse OvR do dataset Iris, um problema de classificação de duas classes.

A parametrização do modela e da criptografia adotou regularização LSSVM $\gamma = 1{,}1$ e *kernels* adaptados à separabilidade dos dados: linear para a Classe 0 e polinomial homogêneo de grau 2 (com expansão explícita) para as Classes 1 e 2. O treinamento empregou a variante orientada a linhas do algoritmo de Householder, utilizando polinômios de Chebyshev de grau 8 para aproximação de raízes e inversões — grau escolhido por equilibrar a precisão necessária sem onerar excessivamente a profundidade multiplicativa. Na granularidade dos dados, a distribuição de 120 amostras entre 40 clientes resultou em 3 amostras locais por participante, gerando matrizes H_i de dimensão 4×4.

Para suportar a profundidade multiplicativa das transformações de Householder e das avaliações de Chebyshev de grau 8 sem *bootstrapping* intermediário, os parâmetros do esquema CKKS utilizados nos experimentos (com $k = 40$) foram configurados da seguinte maneira: o nível de segurança foi definido como `HEStd_NotSet`, e a dimensão do anel (N) escolhida foi de 32.768. O módulo total foi estabelecido em 9.360 bits (sendo $60{+}186{\times}50$), utilizando um fator de escala (Δ) de 50 bits e suportando uma profundidade multiplicativa de 186 níveis. Por fim, o esquema contou com 16.384 slots disponíveis, e o tempo necessário para o *setup* do contexto foi de $14{,}8$ s.

Tabela 2. Tempos de execução do treinamento e inferência federados em texto claro e FHE por classe ($k = 40$, dataset Iris)

Modo	Classe	Kernel	Treino/cliente	Inferência
texto claro	0 — setosa	linear	≈13,1 µs	5,1 µs
	1 — versicolor	poly grau 2	≈16,7 µs	6,2 µs
	2 — virginica	poly grau 2	≈12,9 µs	8,7 µs
FHE CKKS	0 — setosa	linear	≈73,8 s	25,79 s
	1 — versicolor	poly grau 2	≈97,0 s	35,75 s
	2 — virginica	poly grau 2	≈111,9 s	33,70 s

A Tabela 2 evidencia o substancial *overhead* computacional do treinamento federado em FHE, que chega a ser $\approx 5{,}6 \times 10^6$ vezes mais lento que em texto claro. Os tempos médios por cliente escalam com a complexidade do *kernel*: $73{,}8$ s para o modelo linear (Classe 0) e até $111{,}9$ s para o polinomial de grau 2 (Classes 1 e 2), devido à expansão dimensional. Esse elevado custo, também observado na inferência (fator de $\approx 5{,}1 \times 10^6$), é intrínseco à profundidade multiplicativa de 186 exigida pela fatoração *Householder QR*. Esse cenário impõe um *overhead* de duas a três ordens de magnitude superior a abordagens focadas apenas em inferência [Buchanan and Ali 2025], refletindo o grande desafio de realizar o treinamento completo no domínio cifrado.

6.2. Escalabilidade

Para avaliar o comportamento da arquitetura sob variação do número de clientes e da dimensionalidade dos dados, executou-se cada dataset em duas configurações de k: Iris com $k = 40$ e $k = 80$, e WDBC com $k = 150$ e $k = 225$. A Tabela 3 resume o tempo médio de treinamento por cliente, o pico médio de memória residente (RSS) por processo, o custo de comunicação por cliente (*uplink*) e o tempo total de execução para cada configuração.

Tabela 3. Escalabilidade: tempo de treinamento, memória e comunicação em função do número de clientes k (classe 0 de cada dataset).

Dataset	k	Treino/cliente médio (s)	Pico RSS médio (MiB)	*Uplink*/cliente (KiB)	Tempo total (min)
Iris	40	73,8	≈2.260	22.812,2	42,9
Iris	80	199,8	≈5.623	113.717,4	185,5
WDBC	150	432,9	≈5.987	77.593,7	235,3
WDBC	225	250,9	≈5.864	101.414,4	213,8

Os dois datasets exibem tendências opostas ao dobrar (aproximadamente) o número de clientes: no Iris, o tempo médio por cliente *aumenta* de $73{,}8$ s ($k = 40$) para $199{,}8$ s ($k = 80$), apesar de cada cliente processar menos amostras — consistente com contenção no *pool* fixo de 5 *workers*, cujo tempo de fila cresce com o dobro de clientes mesmo com o custo do sistema 4×4 praticamente constante; já no WDBC o tempo médio *diminui* de $432{,}9$ s ($k = 150$) para $250{,}9$ s ($k = 225$), pois cada cliente recebe uma fração menor das 569 amostras totais, e essa redução no custo de construção de H_i compensa a maior contenção entre *workers*. Em ambos os casos, o *uplink* por cliente cresce com k, como esperado, mas o tempo total não degrada proporcionalmente, sugerindo que o paralelismo absorve parte do crescimento — distinção relevante entre custo por cliente (sensível à contenção) e custo total do sistema (atenuado pela paralelização) para o planejamento de implantações com mais clientes do que o testado aqui.

6.3. Precisão do Modelo

A Tabela 4 consolida as métricas de classificação e o erro relativo para os *datasets* Iris e WDBC. O desempenho foi avaliado sob quatro configurações: **Padrão** (particionamento IID entre clientes, modelo proposto), **Dirichlet** (particionamento não-IID via distribuição de Dirichlet, $\alpha = 0{,}5$), **Escala** (aumento do número de clientes para $k = 80$ no Iris e $k = 225$ no WDBC) e **Baseline (média)** (cliente único com poucas amostras por classe, $N = 2$, semente 42). Em vez de replicar as métricas absolutas da contraparte federada em texto claro, a tabela evidencia a equivalência numérica da abordagem proposta ao apresentar diretamente o erro relativo ($\|w_{FHE} - w_{plain}\| / \|w_{plain}\|$) dos parâmetros cifrados para cada cenário analisado.

Classe 0 — Setosa (caso primário, kernel linear)

A solução FHE federada com $k = 40$ clientes atingiu 100% em acurácia, precisão e F1 para a Classe 0, resultado idêntico às referências em texto claro — tanto federada quanto com dados completos — e reproduzido de forma consistente em todas as demais configurações avaliadas (Dirichlet e $k = 80$; Tabela 4). O erro relativo dos parâmetros

Tabela 4. Métricas de classificação e erro relativo no conjunto de validação para os *datasets* Iris (multiclasse OvR) e WDBC (binário).

Dataset / Classe	Abordagem	Acurácia	Precisão	F1	Erro Relativo†
Dataset Iris					
0 — setosa (linear)	**Padrão (IID, $k = 40$)**	$\mathbf{100{,}00\%}$	$\mathbf{100{,}00\%}$	$\mathbf{100{,}00\%}$	$\mathbf{1{,}08 \times 10^{-1}}$
	Dirichlet ($\alpha = 0{,}5$, $k = 40$)	$100{,}00\%$	$100{,}00\%$	$100{,}00\%$	$1{,}49 \times 10^{-5}$
	Escala (IID, $k = 80$)	$100{,}00\%$	–	–	$4{,}59 \times 10^{-5}$
	Baseline (média)	$96{,}67\%$	$100{,}00\%$	$94{,}74\%$	–
1 — versicolor (poly grau 2)	**Padrão (IID, $k = 40$)**	$86{,}67\%$	$71{,}43\%$	$83{,}33\%$	$1{,}29 \times 10^{-2}$
	Dirichlet ($\alpha = 0{,}5$, $k = 40$)	$83{,}33\%$	$66{,}67\%$	$80{,}00\%$	$5{,}65 \times 10^{-6}$
	Escala (IID, $k = 80$)	$90{,}00\%$	–	–	$9{,}15 \times 10^{-6}$
	Baseline (média)	$33{,}33\%$	$33{,}33\%$	$50{,}00\%$	–
2 — virginica (poly grau 2)	**Padrão (IID, $k = 40$)**	$76{,}67\%$	$58{,}82\%$	$74{,}07\%$	$3{,}46 \times 10^{-2}$
	Dirichlet ($\alpha = 0{,}5$, $k = 40$)	$73{,}33\%$	$56{,}25\%$	$69{,}23\%$	$6{,}72 \times 10^{-6}$
	Escala (IID, $k = 80$)	$73{,}33\%$	–	–	$1{,}31 \times 10^{-5}$
	Baseline (média)	$80{,}00\%$	$70{,}00\%$	$70{,}00\%$	–
Acurácia multiclasse (OvR)*		$86{,}67\%$	–	–	–
Dataset WDBC					
Binária (linear)	**Padrão (IID, $k = 150$)**	$\mathbf{91{,}23\%}$	$\mathbf{86{,}36\%}$	$\mathbf{88{,}37\%}$	$\mathbf{2{,}38 \times 10^{-4}}$
	Dirichlet ($\alpha = 0{,}5$, $k = 150$)	$95{,}61\%$	$93{,}02\%$	$94{,}12\%$	$2{,}84 \times 10^{-4}$
	Escala (IID, $k = 225$)	$89{,}47\%$	–	–	$2{,}90 \times 10^{-4}$
	Baseline (média)	$37{,}72\%$	$28{,}36\%$	$34{,}86\%$	–

† Erro relativo calculado por $||w_{FHE} - w_{plain}||/||w_{plain}||$. Aplica-se apenas às soluções FHE.

foi de $1{,}08 \times 10^{-1}$ na configuração IID, mas situou-se na faixa de 10^{-5} sob Dirichlet e sob $k = 80$ (Tabela 4). Essa diferença aparentemente grande entre configurações não se traduz em qualquer degradação de acurácia — 100% em todos os casos — e é consistente com o fato de que o erro relativo depende diretamente da norma $\|w_{plain}\|$ de referência, que varia entre partições distintas dos dados; para uma classe tão facilmente separável quanto a *setosa*, mesmo uma diferença absoluta pequena entre w_{FHE} e w_{plain} pode gerar um quociente relativo elevado sem qualquer impacto prático na fronteira de decisão. Esse ponto reforça a importância de interpretar o erro relativo em conjunto com a acurácia, não isoladamente.

Esses resultados validam a hipótese central do trabalho: para problemas linearmente separáveis, a arquitetura federada homomórfica proposta preserva integralmente a qualidade do modelo treinado em texto claro, sem degradação atribuível à criptografia. A propriedade de separabilidade linear da *setosa* — cujas dimensões de pétala diferem substancialmente das demais espécies — produz um sistema KKT bem condicionado (números de condição entre $6{,}9$ e $93{,}1$ nos clientes, no cenário IID com $k = 40$), favorável à estabilidade numérica das transformações de Householder no domínio cifrado.

Classes 1 e 2 — Efeito da federação, do particionamento não-IID e da escala (kernel polinomial)

Para as Classes 1 (*versicolor*) e 2 (*virginica*), avaliadas com *kernel* polinomial de grau 2, o modelo FHE federado (IID, $k = 40$) alcançou acurácias de $86{,}67\%$ e $76{,}67\%$. Esses resultados superam substancialmente o *baseline* de cliente único e aproximam-se da referência centralizada em texto claro ($90{,}00\%$). Os ínfimos erros relativos confirmam que o decréscimo de acurácia decorre da federação (treinamento fragmentado), e não da crip-

tografia. Cenários não-IID (Dirichlet, $\alpha = 0{,}5$) e de maior escala ($k = 80$) mostraram apenas variações moderadas (ex., queda para $83{,}33\%$ e $73{,}33\%$ no Dirichlet), evidenciando que a distribuição das amostras impacta mais do que o número de clientes. A acurácia multiclasse agregada atingiu $86{,}67\%$ para $k = 40$.

O desempenho aquém da referência centralizada deve-se a dois fatores: a sobreposição não-linear inerente entre essas espécies [Fisher 1936] e a instabilidade de matrizes mal-condicionadas [Golub and Van Loan 2013]. Primeiramente, o teto de $90{,}00\%$ no cenário centralizado indica que o grau 2 é estruturalmente insuficiente para uma separação total (demandando *kernels* de grau superior ou RBF). Em segundo lugar, a fragmentação dos dados eleva o número de condição (κ) das matrizes locais, atingindo valores próximos a 1.400 sob *kernel* polinomial, em contraste aos 93 da Classe 0 (linear). Contudo, esse impacto é restrito a poucos pontos percentuais, não causando uma degradação severa no modelo, seja sob IID, Dirichlet ou maior escala.

WDBC — Efeito da federação, do particionamento não-IID e da escala

Para o *dataset* WDBC (*kernel* linear, classificação binária), a solução FHE federada (IID, $k = 150$) alcançou $91{,}23\%$ de acurácia, $86{,}36\%$ de precisão e $88{,}37\%$ de F1. Essas métricas aproximam-se da referência centralizada em texto claro (acurácia de $95{,}61\%$) e superam amplamente o *baseline* de cliente único ($37{,}72\%$), confirmando que a federação compensa efetivamente a escassez de amostras locais. O erro relativo dos parâmetros ($2{,}38 \times 10^{-4}$) manteve-se ínfimo, não afetando a capacidade de classificação.

Sob particionamento não-IID (Dirichlet, $\alpha = 0{,}5$), a acurácia subiu de forma contraintuitiva para $95{,}61\%$ (com erro de $2{,}84 \times 10^{-4}$). Esse aumento provavelmente decorre da variância amostral em vez de um benefício estrutural do modelo não-IID, o que demandará investigações futuras com múltiplas sementes. Já na avaliação de escala para $k = 225$ clientes, notou-se uma degradação suave da acurácia (para $89{,}47\%$) com erro estável ($2{,}90 \times 10^{-4}$). Em contraste com o comportamento não-monotônico do Iris, essa queda consistente ilustra o impacto direto da fragmentação progressiva dos dados.

7. Conclusão

Este trabalho apresentou uma arquitetura de treinamento federado homomórfico para LSSVM baseada no esquema CKKS, integrando aprendizado federado One-Shot e decomposição QR de Householder para treinamento sobre dados cifrados sem exposição das amostras locais dos clientes.

Os resultados experimentais demonstraram erro relativo dos parâmetros tipicamente na faixa de 10^{-6} a 10^{-2} para as classes de kernel polinomial do Iris e em torno de 3×10^{-4} para o WDBC, entre os parâmetros cifrados e os calculados em texto claro, confirmando a viabilidade da abordagem tanto para kernels lineares quanto para kernels polinomiais homogêneos de grau 2, sob particionamento IID e não-IID (Dirichlet), e para uma faixa de 40 a 225 clientes. Na Classe 0 do Iris (setosa, linearmente separável) e no WDBC (kernel linear, classificação binária), a solução FHE federada manteve-se próxima das referências em texto claro em todas as configurações avaliadas. A decomposição QR de Householder demonstrou compatibilidade com o domínio FHE, conforme estabelecido na Seção 3.3.

As principais limitações do trabalho são: (i) a degradação de acurácia nas Classes 1 e 2 do Iris frente à referência de dados completos, atribuível à sobreposição não-linear entre as espécies *versicolor* e *virginica* e ao número de condição elevado das matrizes locais sob fragmentação de dados — efeito presente tanto em texto claro quanto sob FHE, e portanto não atribuível à criptografia, mas ainda relevante para a escolha de kernel em implantações futuras; (ii) a melhora de acurácia observada no WDBC sob particionamento Dirichlet frente ao cenário IID, provavelmente atribuível à variância amostral entre partições aleatórias distintas e não a um efeito sistemático do particionamento não-IID, o que exige validação com múltiplas sementes antes de ser generalizada; e (iii) o escopo do modelo de ameaça, limitado ao servidor honest-but-curious sem considerar clientes maliciosos ou colisão entre participantes. Trabalhos futuros devem endereçar essas limitações, além de investigar a extensão a kernels não-lineares implícitos no domínio cifrado (ex., RBF), a avaliação em datasets de escala industrial e com número ainda maior de clientes, otimizações de parametrização do CKKS, aceleração por GPU e mecanismos de verificação dos limites de Chebyshev submetidos pelos clientes.

Referências

Badawi, A. A., Alexandru, A., Bates, J., Bergamaschi, F., Cousins, D. B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Pascoe, C., Polyakov, Y., Quah, I., R.V., S., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., and Zucca, V. (2022). OpenFHE: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915. `https://eprint.iacr.org/2022/915`.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Buchanan, W. J. and Ali, H. (2025). Evaluation of privacy-aware support vector machine (SVM) learning using homomorphic encryption. arXiv:2503.04652v1.

Chen, H., Gilad-Bachrach, R., Han, K., Huang, Z., Jalali, A., Laine, K., and Lauter, K. (2018). Logistic regression over encrypted data from fully homomorphic encryption. *BMC Medical Genomics*, 11(4):81.

Cheon, J. H., Kim, A., Kim, M., and Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. pages 409–437. Springer.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Geyer, R. C., Klein, T., and Nabi, M. (2018). Differentially private federated learning: A client level perspective.

Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4th edition.

Graepel, T., Lauter, K., and Naehrig, M. (2012). ML confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology (ICISC)*, Lecture Notes in Computer Science, pages 1–21. Springer.

Han, K. and Ki, D. (2020). Better bootstrapping for approximate homomorphic encryption. page 364–390, Berlin, Heidelberg. Springer-Verlag.

Hartebrodt, A. and Röttger, R. (2023). Privacy of federated qr decomposition using additive secure multiparty computation. *IEEE Transactions on Big Data*, 10(1):31–40.

Iezzi, M. (2020). Practical privacy-preserving data science with homomorphic encryption: An overview. In *IEEE International Conference on Big Data*, pages 3979–3988.

Lyubashevsky, V., Peikert, C., and Regev, O. (2013). On ideal lattices and learning with errors over rings. *J. ACM*, 60(6).

Ma, J., Naas, S.-A., Sigg, S., and Lyu, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*.

Malekzadeh, M., Borovykh, A., and Gündüz, D. (2021). Honest-but-curious nets: Sensitive attributes of private inputs can be secretly coded into the classifiers' outputs. CCS '21, page 825–844, New York, NY, USA. Association for Computing Machinery.

McMahan, H. B., Andrew, G., Erlingsson, U., Chien, S., Mironov, I., Papernot, N., and Kairouz, P. (2019). A general approach to adding differential privacy to iterative training procedures.

Micciancio, D. and Goldwasser, S. (2002). *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.

Park, S., Byun, J., Lee, J., Cheon, J. H., and Lee, J. (2020). HE-friendly algorithm for privacy-preserving SVM training. *IEEE Access*, 8:57414–57425.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9:293–300.

Wolberg, W. H., Mangasarian, O. L., Street, N., and Street, W. N. (1993). Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. arXiv:1906.08935.