UNIVERSIDADE FEDERAL DO CEARÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

VANESSA CARVALHO DO NASCIMENTO

PRIVACIDADE EM ALGORITMOS DE APRENDIZADO DE MÁQUINA: COMPARAÇÃO DE DUAS IMPLEMENTAÇÕES DE REGRESSÃO LINEAR USANDO CRIPTOGRAFIA HOMOMÓRFICA

SOBRAL

2023

VANESSA CARVALHO DO NASCIMENTO

PRIVACIDADE EM ALGORITMOS DE APRENDIZADO DE MÁQUINA: COMPARAÇÃO DE DUAS IMPLEMENTAÇÕES DE REGRESSÃO LINEAR USANDO CRIPTOGRAFIA HOMOMÓRFICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Josefran de Oliveira Bastos.

SOBRAL

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

N199p Nascimento, Vanessa Carvalho do.
Privacidade em algoritmos de aprendizado de máquina : Comparação de duas implementações de Regressão Linear usando Criptografia Homomórfica / Vanessa Carvalho do Nascimento. – 2023.
46 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Administração em Gestão Pública, Sobral, 2023.
Orientação: Prof. Dr. Antônio Josefran de Oliveira Bastos.

1. Privacidade. 2. Anonimização. 3. Machine Learning. 4. Dados distribuídos. 5. Homomorphic Encryption. I. Título.

CDD 658

VANESSA CARVALHO DO NASCIMENTO

PRIVACIDADE EM ALGORITMOS DE APRENDIZADO DE MÁQUINA: COMPARAÇÃO DE DUAS IMPLEMENTAÇÕES DE REGRESSÃO LINEAR USANDO CRIPTOGRAFIA HOMOMÓRFICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Aprovada em: 18/12/2023.

BANCA EXAMINADORA

Prof. Dr. Josefran de Oliveira Bastos (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antônio Emerson Barros Tomaz
Universidade Federal do Ceará (UFC)

Prof. Dr. José Cláudio do Nascimento
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos meus pais, pelo cuidado e apoio, e ao meu ex-professor Luciano Epifânio, por me inserir no mundo das olimpíadas de matemática.

AGRADECIMENTOS

Agradeço aos meus pais pelo amor, paciência e esforço que tiveram ao cuidar de mim.

Ao Prof. Dr. Josefran de Oliveira Bastos pelas risadas, conversas divertidas e orientações sérias.

Ao meu ex-professor Luciano Epifânio por sempre acreditar em mim e ter me dado todo apoio possível na época em que eu fazia olimpíadas de matemática.

Ao meu namorado Valfrido, pelo carinho e incentivo.

Ao meu pássaro Etzinho, por me dar paz nos momentos mais estressantes.

Ao meu amigo Francilândio por tornar os dias na faculdade mais fáceis e alegres. Nossa união nos ajudou em todo o percurso.

"Privacidade não é uma opção, e não deve ser o preço que aceitamos por apenas entrar na internet." (Gary Kovacs, 2012)

RESUMO

Privacidade de dados é uma questão essencial a ser tratada nessa era atual de complexos fluxos de dados. Métodos tradicionais de anonimização já não são mais suficientes para garantir a privacidade dos indivíduos. Novas soluções vêm surgindo e estão em contantes evolução. Dentre as principais abordagens amplamente difundas na literatura e comentadas no presente trabalho, pode-se destacar: Aprendizado Federado (AF), Computação Multipartidária Segura (CMS), Privacidade Diferencial (PD) e Criptografia Homomórfica (HE - *Homomorphic Encryption*, do inglês). Nesse contexto, é imperativo o desenvolvimento de algoritmos de Machine Learning (ML) capazes de equilibrar a utilidade e a privacidade de dados. A maioria dos trabalhos se dedicam a etapa de inferência em ML. No entanto, o cenário ideal deve também lidar com etapa de treinamento na aplicação de algoritmos que preservem a privacidade. Este trabalho compara duas implementações do modelo de regressão linear em ambas as etapas, analisando, dentre outros elementos, o erro e o tempo de execução.

Palavras-chave: privacidade; anonimização; machine learning; dados distribuídos; homomorphic encryption.

ABSTRACT

Data privacy is an essential issue to be addressed in the current era of complex data flows. Traditional anonymization methods are no longer sufficient to ensure individuals' privacy. New solutions are emerging and are constantly evolving. Among the main approaches widely discussed in the literature and explored in this work, Federated Learning (FL), Secure Multiparty Computation (SMC), Differential Privacy (DP), and Homomorphic Encryption (HE) stand out. In this context, the development of Machine Learning (ML) algorithms capable of balancing the utility and privacy of data is imperative. While most works focus on the inference stage in ML, the ideal scenario should also address the training stage in the application of algorithms that preserve privacy. This work compares two implementations of the linear regression model in both stages, analyzing elements such as error and execution time, within the broader context of ensuring data privacy and utility balance.

Keywords: privacy; anonymization; machine learning; distibuted data; homomorphic encryption.

LISTA DE FIGURAS

Figura 1 – Clientes enviando dados cifrados ao servidor. 24
Figura 2 – O servidor não possui acesso aos dados limpos, porém pode manipular a representação dos dados cifrados. 25
Figura 3 – O resultado das operações efetuadas pelo servidor é cifrado. 25
Figura 4 – Servidor enviando o resultado cifrado para os clientes. 25
Figura 5 – Crescimento do erro pode tornar a decifração falha. 27
Figura 6 – Base de dados distribuída horizontalmente. 34
Figura 7 – Esquema do sistema. 35

LISTA DE TABELAS

Tabela 1 – Gastos anual de clientes na loja. 18
Tabela 2 – Generalização do campo Idade. 19
Tabela 3 – Perturbação do campo Idade. 19
Tabela 4 – Permutação dos campos Idade e Cidade. 19
Tabela 5 – Supressão dos campos ID e Nome. 20
Tabela 6 – Mascaramento de dados. 38
Tabela 7 – Erro do Método 1 considerando o β_{limpo} com conversão de dados. 41
Tabela 8 – Erro do Método 1 considerando o β_{limpo} sem conversão de dados. 41
Tabela 9 – Diferença de erro entre os Métodos 1 e 2 considerando o β_{limpo} sem conversão de dados. 42
Tabela 10 – Tempo de execução da regressão linear. 42

LISTA DE ABREVIATURAS E SIGLAS

AF	Aprendizado Federado
CMS	Computação Multipartidária Segura
DL	*Deep Learning*
FHE	*Fully Homomorphic Encryption*
GDPR	Regulamento Geral sobre a Proteção de Dados
LGPD	Lei Geral de Proteção de Dados
ML	*Machine Learning*
PD	Privacidade Diferencial
PHE	*Partially Homomorphic Encryption*
PSC	Provedor de Serviços Criptográficos
SHE	*Somewhat Homomorphic Encryption*
SVM	*Support Vector Machine*
TFHE	*Fast Fully Homomorphic Encryption over the Torus*

LISTA DE SÍMBOLOS

ε	*privacy budget*
pk	Chave pública
sk	Chave secreta
$\lambda(n)$	Função de Carmichael
$\otimes$	Produto elemento a elemento de duas matrizes ou vetores
$\oplus$	Operação binária qualquer
$\oplus'$	Versão homomórfica de $\oplus$
e	Número de Euler

SUMÁRIO

1 INTRODUÇÃO 14
1.1 Justificativa 16
1.2 Trabalhos Relacionados 16
1.3 Objetivos 17
1.3.1 Objetivo Geral 17
1.3.2 Objetivos Específicos 17
2 FUNDAMENTAÇÃO TEÓRICA 18
2.1 Técnicas de Anonimização 18
2.1.1 Generalização 18
2.1.2 Perturbação 19
2.1.3 Permutação 19
2.1.4 Supressão 20
2.2 Privacidade Diferencial 20
2.3 Computação Multipartidária Segura 22
2.4 Aprendizado Federado 22
2.5 Breve comparação entre as abordagens anteriores 22
2.6 Criptografia Homomórfica 23
2.6.1 Introdução à criptografia homomórfica 23
2.6.2 Aspectos Fundamentais de HE 26
2.6.2.1 Ruído *26*
2.6.2.2 Classificação de esquemas *27*
2.6.2.3 Bootstrapping *28*
2.6.2.4 Key switching *28*
2.7 Sistema Paillier 29
2.8 Regressão Linear usando o Método dos Mínimos Quadrados 29
3 METODOLOGIA 32
4 DESENVOLVIMENTO 34
4.1 Método 1 34
4.1.1 Conversão de dados 35
4.1.2 Mascaramento dos dados 36

4.1.3 ***Regressão linear segura*** 37
4.1.3.1 *Inicialização* *37*
4.1.3.2 *Agregação* *37*
4.1.3.3 *Regressão* *38*
4.1.4 ***Corretude do Protocolo regressão linear*** 38
4.2 **Método 2** 39
5 **RESULTADOS** 41
5.1 **Erro** 41
5.2 **Tempo de processamento** 42
6 **CONSIDERAÇÕES FINAIS** 43
REFERÊNCIAS 44

1 INTRODUÇÃO

Na era digital atual, a questão da privacidade de dados emergiu como um tópico de alta relevância e tornou-se uma das questões sociais e culturais definidoras de nossa era (FORBES, 2019). O constante avanço tecnológico, aliado a proliferação de dispositivos conectados à internet, resultou em uma quantidade sem precedentes de informações pessoais sendo coletadas, armazenadas e compartilhadas. Aproximadamente 328.77 milhões de terabytes de dados são gerados por dia em 2023 (DUARTE, 2023). Com isso, surgem preocupações em relação à privacidade e à segurança dos dados dos indivíduos. No Brasil, a resposta a essas preocupações foi a promulgação da Lei n° 13.709/2018, conhecida com Lei Geral de Proteção de Dados (LGPD) (BRASIL, 2018).

A LGPD, inspirada no Regulamento Geral sobre a Proteção de Dados (GDPR) (European Commission, 2016) da União Europeia, foi sancionada em agosto de 2018 e entrou em vigor em setembro de 2020. Ela representa um marco legal fundamental que visa proteger a privacidade e os direitos dos cidadãos em um mundo cada vez mais orientado por dados.

Um dos principais objetivos da LGPD é conceder às pessoas um maior controle sobre suas informações pessoais. Ela define regras claras para a coleta, uso, processamento e armazenamento de dados pessoais por organizações públicas e privadas. Empresas e instituições que operam no Brasil devem se adequar a uma série de regulamentações rigorosas que buscam garantir a segurança e a privacidade dos dados de seus clientes e funcionários.

A LGPD se aplica a uma ampla gama de dados pessoais, que incluem desde informações básicas, como nome e endereço, até dados mais sensíveis, como informações médicas e financeiras. Ela estabelece que o consentimento do titular dos dados deve ser obtido para a coleta e o tratamento dessas informações, e os indivíduos têm o direito de acessar, corrigir e excluir seus dados pessoais.

Entretanto, a realidade é que a compra e venda de dados é um mercado muito lucrativo e muito atrativo, já que podem ser usados para inúmeros interesses (ECONOMIST, 2017). O capitalismo de vigilância, termo cunhado por Shoshana Zuboff, descreve o modelo econômico no qual empresas lucram com a coleta, análise e exploração dos dados pessoais dos usuários (ZUBOFF, 2018). Grandes empresas da tecnologia, como Google, Facebook e Amazon, acumularam vastos volumes de dados sobre as atividades dos indivíduos online, interesses e comportamentos. Esses dados são utilizados para direcionar anúncios, personalizar serviços e até mesmo moldar comportamentos online (SMYTH, 2019).

O capitalismo de vigilância gera preocupações significativas sobre a privacidade, uma vez que os usuários muitas vezes não têm controle sobre como seus dados são usados e compartilhados. A monetização de informações pessoais levanta questões éticas e legais sobre a exploração de dados de indivíduos em benefício das empresas.

Além disso, ao longo dos anos, ocorreram inúmeros casos de vazamento de dados em larga escala, expondo informações pessoais de milhões de pessoas. Um dos exemplos mais notórios foi o caso do Facebook e a empresa Cambridge Analytica (CADWALLADR; GRAHAM-HARRISON, 2018). Dados de aproximadamente 50 milhões de usuários do Facebook foram indevidamente coletados e usados para influenciar eleições e opiniões públicas. Outro caso notório foi do site de encontros para relacionamentos fora do casamento Ashley Madison, em que detalhes sensíveis de 37 milhões de usuários foram expostos. O impacto pessoal severo deste vazamento de dados sobre as vítimas individuais tornou-se aparente nas semanas seguintes ao incidente, quando relatos da mídia vincularam esse evento ao suicídio de um pequeno número de vítimas expostas no vazamento, além de um número crescente de tentativas de chantagem e extorsão direcionadas a indivíduos presentes nos arquivos (CROSS *et al.*, 2019). Esses incidentes de uso político e vazamentos de dados sensíveis demonstram a vulnerabilidade dos dados pessoais e a necessidade de regulamentações mais rigorosas em um ambiente digital cada vez mais interconectado.

Uma abordagem tradicional para proteger a privacidade dos dados é a anonimização. No entanto, algumas pesquisas afirmam que a anonimização tradicional não é uma garantia absoluta de proteção de dados. Com o aumento do poder computacional e técnicas avançadas de reidentificação, é possível quebrar a anonimização e vincular dados anonimizados a indivíduos reais, conforme mostrado em (AL-AZIZY *et al.*, 2016), (NARAYANAN; SHMATIKOV, 2008) e (GANTA *et al.*, 2008). Técnicas de reidentificação de dados permitem que invasores combinem dados anonimizados com informações externas para identificar pessoas específicas. Portanto, a simples remoção de nomes ou informações de identificação pessoal pode não ser suficiente para garantir a privacidade dos dados.

Um movimento global em direção a métodos mais eficazes de privacidade, que também mantenham a utilidade dos dados, é essencial. Nesse contexto, com o crescente volume de dados disponíveis, a área de *Machine Learning* (ML) tem conquistado um papel fundamental em várias aplicações, desde sistemas de recomendação até diagnóstico médico. Grande quantidade de dados favorecem o treinamento de modelos de ML eficazes, mas isso

também gera conflitos sobre como esses dados são manipulados e protegidos.

Com o aumento da conscientização sobre a importância da privacidade de dados, é imperativo desenvolver métodos que permitam a análise de dados sensíveis sem comprometer a confidencialidade das informações (ACQUISTI *et al.*, 2020). Nesse cenário, este trabalho reproduz protocolos de regressão linear segura com criptografia homomórfica baseados em Qiu *et al.* (2020) e Chen e Zheng (2022). Ambos utilizam o Método de Mínimos Quadrados e trabalham em um cenário com dados distribuídos, sendo que na primeira versão, o próprio cliente calcula os parâmetros do modelo, efetuando as etapas finais do protocolo em que é executado o cálculo da inversa de uma matriz. Isso é necessário para que essa operação seja feita usando dados limpos, devido às limitações caso fossem cifrados. Já na segunda versão, há duas outras entidades além dos clientes que efetuam os protocolos de agregação de dados e regressão, sendo necessário apenas que os clientes efetuem uma conversão de dados previamente.

1.1 Justificativa

Soluções de ML que trabalham com métodos de privacidade de dados tendem a lidar apenas com a etapa de inferência. Nesse contexto, um exemplo recorrente seria um serviço de inferência na nuvem. A nuvem possuiria um modelo pré-treinado com dados puros, e com os parâmetros obtidos do treinamento efetuaria a inferência com os dados cifrados dos clientes. Essa preferência se dá por conta do alto custo computacional e do maior erro gerado ao usar textos cifrados para treinamento.

Entretanto, soluções em que as entidades que manipulam os dados sempre lidem com textos cifrados são mais seguras e, por isso, espera-se que sejam aperfeiçoadas para que possam fornecer bons resultados em tempo viável, mantendo adequados níveis de privacidade. Dessa forma, para este trabalho foram escolhidos para implementação dois artigos que apresentam diferentes protocolos de Regressão Linear, um algoritmo muito popular de ML.

1.2 Trabalhos Relacionados

Alguns trabalhos implementam algoritmos de ML fazendo aproximações lineares de funções não lineares. Em Chen *et al.* (2018) e Han *et al.* (2019) os autores usaram uma adaptação do gradiente descendente e uma aproximação polinomial da sigmoide para obter um algoritmo de regressão logística que lida com dados cifrados tanto no treinamento quando na

inferência. Em (SARKAR *et al.*, 2023), foi proposto uma solução para identificação de tipo de câncer utilizando análise genômica com HE, também por meio de uma regressão logística. Em (PARK *et al.*, 2020) é descrita a implementação de um algoritmo para o treinamento de uma *Support Vector Machine* (SVM) que evita operações ineficientes e instabilidade numérica em um domínio cifrado. Os resultados apoiam o desenvolvimento de aplicações práticas do modelo SVM preservando a privacidade.

O artigo (PODSCHWADT *et al.*, 2022) apresenta arquiteturas de *Deep Learning* (DL) com proteção de privacidade usando *Fully Homomorphic Encryption* (FHE). Foram analisadas as alterações para tornar as ferramentas compatíveis e como essas alterações afetaram o desempenho. Também foram identificados desafios dessa configuração, como sobrecarga computacional, usabilidade e limitações impostas pelos esquemas de criptografia e suas possíveis soluções. Por fim, foram propostas métricas de avaliação que permitem uma comparação mais significativa e relevante das soluções de preservação de privacidade em DL.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem como objetivo reproduzir duas implementações de regressão linear usando criptografia homomórfica e compará-las em questão de acurácia e tempo de processamento.

1.3.2 Objetivos Específicos

- Argumentar sobre a importância da privacidade e como ela está sendo comprometida atualmente.
- Mostrar alternativas aos métodos tradicionais de anonimização de dados.
- Apresentar alguns aspectos de criptografia homomórfica e sua aplicação nas etapas de treinamento e inferência de um modelo de regressão linear com dados distribuídos.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção trata de conceitos básicos de métodos de privacidade de dados, desde anonimização tradicional até criptografia homomórfica, e de Regressão Linear por meio do Método dos Mínimos Quadrados.

2.1 Técnicas de Anonimização

A anonimização de dados é um processo importante para proteger a privacidade das informações pessoais durante a análise de dados. Diversos métodos foram desenvolvidos com o objetivo de ocultar informações sensíveis enquanto mantêm a utilidade dos dados para fins de pesquisa e análise. Nesta seção, serão explorados alguns métodos de anonimização, cada um com suas próprias características e técnicas.

As técnicas de anonimização convertem dados pessoais em dados anonimizados para reduzir o risco de divulgação não autorizada. No entanto, conforme a extensão da anonimização aumenta, a utilidade dos dados diminui. Portanto, a organização responsável pelo processo precisa escolher as técnicas de anonimização adequadas a serem aplicadas no conjunto de dados, antecipando a utilidade esperada e o risco de reidentificação.

As técnicas descritas a seguir terão como base na Tabela 1, que representa o gasto anual de clientes numa loja fictícia.

Tabela 1 – Gastos anual de clientes na loja.

ID	Nome	Idade	Cidade	Total gasto (R\$)
123	Alice	34	Sobral	139
234	Bob	19	Tianguá	20
456	Carlos	23	Forquilha	87
789	Daniel	48	Sobral	340
981	Eva	37	Forquilha	207

Fonte: elaborada pela autora.

2.1.1 Generalização

A generalização envolve a substituição de valores específicos nos dados por categorias mais amplas ou intervalos. Isso reduz a granularidade, tornando menos provável a identificação de indivíduos.

Tabela 2 – Generalização do campo Idade.

ID	Nome	Idade	Cidade	Total gasto ($R$$)
123	Alice	30 – 35	Sobral	139
234	Bob	18 – 25	Tianguá	20
456	Carlos	18 – 25	Forquilha	87
789	Daniel	45 – 50	Sobral	340
981	Eva	35 – 40	Forquilha	207

Fonte: elaborada pela autora.

2.1.2 Perturbação

O método de perturbação envolve a introdução de ruído ou alteração nos dados originais, de forma a tornar as informações individuais menos identificáveis. Por exemplo, ao adicionar um pequeno valor aleatório aos dados de idade, torna-se mais difícil identificar pessoas com idades específicas.

Tabela 3 – Perturbação do campo Idade.

ID	Nome	Idade	Cidade	Total gasto ($R$$)
123	Alice	32	Sobral	139
234	Bob	21	Tianguá	20
456	Carlos	22	Forquilha	87
789	Daniel	46	Sobral	340
981	Eva	38	Forquilha	207

Fonte: elaborada pela autora.

2.1.3 Permutação

O objetivo da permutação é reorganizar os dados de uma base de forma a representar os valores dos atributos individuais, mas, em geral, esses valores não correspondem mais aos registros originais. Ele pode ser usado quando se deseja a análise de dados agregados ou quando não há a necessidade de analisar a relação entre os atributos.

Tabela 4 – Permutação dos campos Idade e Cidade.

ID	Nome	Idade	Cidade	Total gasto ($R$$)
123	Alice	22	Tianguá	139
234	Bob	38	Forquilha	20
456	Carlos	32	Sobral	87
789	Daniel	21	Forquilha	340
981	Eva	46	Sobral	207

Fonte: elaborada pela autora.

2.1.4 Supressão

A supressão é a remoção de informações específicas dos dados. Isso pode envolver a eliminação de campos inteiros ou valores individuais.

Tabela 5 – Supressão dos campos ID e Nome.

ID	Nome	Idade	Cidade	Total gasto ($R$$)
***	***	34	Sobral	139
***	***	19	Tianguá	20
***	***	23	Forquilha	87
***	***	48	Sobral	340
***	***	37	Forquilha	207

Fonte: elaborada pela autora.

2.2 Privacidade Diferencial

A Privacidade Diferencial (PD) é uma garantia de privacidade para a entrada dos dados de um indivíduo em uma função ou sequência de funções, chamadas de mecanismo de privacidade. Ela é projetada para proteger a privacidade dos indivíduos em processos de análise estatística, como o processamento de dados de saúde ou censos, garantindo que o comportamento do mecanismo seja essencialmente inalterado, independentemente de qualquer indivíduo optar por participar ou não do conjunto de dados. A PD oferece uma forte proteção contra a divulgação não autorizada de informações pessoais.

A definição matemática da PD envolve um parâmetro ε, que mede o grau de privacidade oferecido. Quanto menor o valor de ε, maior é a proteção da privacidade. Um algoritmo aleatório M é dito $\varepsilon-$ diferencialmente privado se para quaisquer bases de dados D_1 e D_2 que se diferenciam por uma única entrada satisfizer:

$$P[M(D_1) = k] \leq e^{\varepsilon} P[M(D_2) = k] \tag{2.1}$$

para todas as possíveis saídas k.

$P[M(D1) = k]$ é a probabilidade de que, ao executar o algoritmo M no banco de dados D_1, a saída seja k. Este processo é probabilístico, ou seja, ao executá-lo várias vezes, pode ser fornecido respostas diferentes. Um exemplo desse processo pode ser numa contagem de pessoas com uma determinada doença ser adicionado algum número aleatório a essa contagem e retornar a soma final. Como o número aleatório muda cada vez que o processo é executado, os resultados irão variar.

Essa definição implica que a probabilidade de obter uma determinada saída do mecanismo para o conjunto de dados D_1 é, no máximo, e^{ε} vezes maior do que a probabilidade de obter a mesma saída para o conjunto de dados D_2. Assim, essa fórmula significa que a saída do processo é semelhante se você alterar ou remover os dados de uma pessoa. O grau de semelhança depende de ε, já que se ε estiver muito próximo de 0, então e^{ε} estará muito próximo de 1, de modo que as probabilidades serão muito semelhantes. Quanto maior ε, mais as probabilidades podem diferir. O valor ε é chamado de *privacy budget*.

Para cumprir os princípios da PD, os mecanismos de privacidade frequentemente envolvem a adição de ruído aos dados ou o uso de métodos de generalização para obscurecer atributos sensíveis. A perturbação dos dados visa torná-los impossíveis de serem restaurados e proteger a privacidade dos usuários.

Há dois tipos de modelos de PD: central (PDC) e local (PDL). No primeiro modelo, existe a ideia de um agregador central, que tem acesso aos dados originais. Um exemplo possível seria uma organização coletando dados sobre indivíduos. Neste modelo, cada usuário envia seus dados sem ruído para este agregador. O agregador pega esses dados e os transforma utilizando um mecanismo diferencialmente privado.

O mecanismo diferencialmente privado é aplicado apenas uma vez, no final do processo. O agregador pode, então, por exemplo, publicar o resultado ou compartilhá-lo com terceiros. Esse modelo possui como vantagem a precisão. No modelo central, geralmente não é necessário adicionar muito ruído para obter bons resultados com um baixo ε. Para possibilitar isso, cada usuário precisa confiar o suficiente no agregador para compartilhar dados com ele. Isso pode ser difícil, já que o agregador pode ser uma empresa ou governo não confiável. Além disso, com o modelo central, todos os dados são coletados em um só lugar. Isso aumenta o risco de falhas graves, por exemplo, se o agregador for hackeado e houver vazamento de todos os dados.

Já no modelo local, ainda há um agregador, mas ele não tem mais acesso aos dados reais. Em vez disso, cada usuário aplica um mecanismo diferencialmente privado aos seus próprios dados. E eles só enviam seus dados para o agregador depois que já estão anonimizados. Depois de coletar esses dados ruidosos, o agregador pode calcular algumas estatísticas e publicá-las. Esta última etapa não precisa ser diferencialmente privada, pois os dados são anônimos desde o início. Em teoria, o agregador poderia publicar o conjunto de dados inteiro que coletou.

A vantagem desse modelo é que ele não requer mais confiança. Como cada usuário

protege seus próprios dados, eles estão seguros mesmo se o agregador for mal-intencionado. Isso torna o modelo local adequado para situações em que a confiança é difícil de obter. Como desvantagem, tem-se que cada usuário precisa adicionar ruído aos seus próprios dados. Assim, o ruído total é muito maior. Geralmente, são necessários muitos mais usuários do que no modelo central para obter resultados úteis.

2.3 Computação Multipartidária Segura

Computação Multipartidária Segura (CMS) permite que um grupo de proprietários de dados independentes que não confiam uns nos outros ou em qualquer terceiro comum possa calcular conjuntamente uma função que depende de todas as suas entradas privadas (EVANS *et al.*, 2018). Os participantes concordam com uma função para calcular $f(x_1, x_2, \cdots, x_m)$ onde x_i é a entrada secreta da i-ésima parte e, em seguida, podem usar um protocolo CMS para calcular conjuntamente a saída dessa função em suas entradas secretas sem haver revelações de dados. Os métodos utilizados permitem também que os participantes confirmem que o resultado é de fato a saída da função nas entradas fornecidas, portanto trata-se de uma computação verificável. A CMS foi introduzido por Andrew Yao na década de 1980 e tornou-se eficiente o suficiente para ser usado na prática.

2.4 Aprendizado Federado

Aprendizado Federado (AF) é uma configuração de aprendizado de máquina que treina um modelo compartilhado globalmente em um grande número de clientes distribuídos usando um protocolo de controle eficiente com o servidor central. Nesse cenário, os dados são mantidos de forma descentralizada, somente as atualizações de parâmetros do modelo calculadas em dados locais são enviadas ao servidor, que os agrega para melhorar o modelo global (ZHENG *et al.*, 2020). Essa abordagem além de mitigar muitos dos riscos e custos sistêmicos de privacidade resultantes do aprendizado de máquina tradicional e centralizado, também aproveita os recursos de computação em dispositivos móveis.

2.5 Breve comparação entre as abordagens anteriores

Sobre PD e AF, embora ambas as ferramentas evitem o acesso direto aos dados, suas metodologias são essencialmente diferentes. A PDL é uma classificação teórica de privacidade

que pode ser alcançada por diferentes algoritmos, enquanto o AF é uma estrutura genérica de aprendizado distribuído sem um nível de privacidade comprovável teoricamente. Porém, os dois mecanismos podem ser usados em conjunto em etapas de aprendizado de máquina.

Abordagens de AF existentes ou usam CMS, que é vulnerável à inferência, ou PDL, que pode levar à baixa precisão dos resultados dado um grande número de partes com quantidades relativamente pequenas de dados cada. Combinar PD com CMS permite reduzir o crescimento de ruído à medida que o número das partes aumenta sem sacrificar a privacidade, mantendo uma taxa de confiança pré-definida.

No trabalho (TRUEX *et al.*, 2019), os autores propõem um sistema de AF que fornece garantias formais de privacidade, lida com vários cenários de confiança, e produz modelos com maior precisão quando comparados com as abordagens existentes de preservação da privacidade. Os dados nunca saem dos dispositivos dos participantes e a privacidade é garantida usando CMS e PD. Foi levada em conta a potencial inferência de participantes individuais, bem como o risco de conluio entre as entidades participantes através de um limite de confiança.

2.6 Criptografia Homomórfica

A criptografia homomórfica também é adotada para proteção de dados, provendo privacidade por meio da troca de parâmetros sob o mecanismo de criptografia durante o aprendizado de máquina. Ao contrário da proteção usando PD, os dados não são transmitidos, nem podem ser adivinhados pelos dados da outra parte. Por tais razões, este trabalho se concentrou nos recursos de criptografia homomórfica para implantar serviços com dados sensíveis. No entanto, as abordagens discutidas podem ser combinadas resultando em bons trade-offs entre precisão e privacidade.

2.6.1 Introdução à criptografia homomórfica

A criptografia homomórfica permite manipulações diretamente sobre dados cifrados sem acesso à chave privada, garantindo a confidencialidade de dados sensíveis mesmo em ambientes hostis, de zero confiança, ou em casos de ciberataque, uma vez que o servidor não possui a chave privada para decifrar os dados.

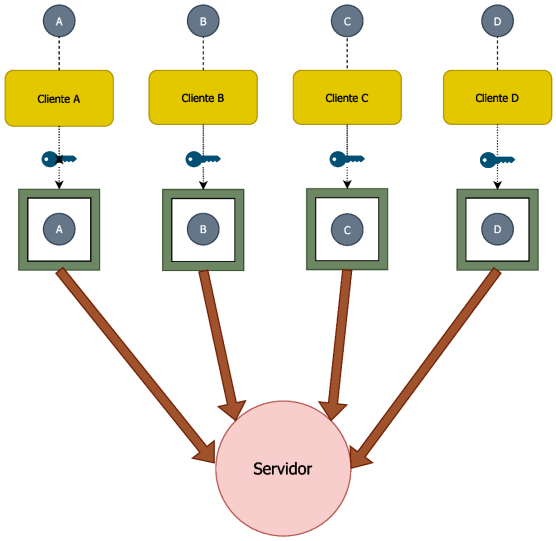
Em criptografia, ao se cifrar um texto limpo p utilizando uma chave pública pk, é obtido um texto cifrado $c = \mathrm{Enc}(p, pk)$, onde a função Enc é um algoritmo de cifração.

$\text{Dec}(c, sk) = p$ indica a operação inversa, que decifra c de volta ao texto limpo p. Se a criptografia for assimétrica, uma chave pública pk é usada para cifração e uma chave secreta sk é utilizada para a decifração. Caso a criptografia seja simétrica, uma única chave secreta é utilizada em ambas as operações.

Considere p_1 e p_2 textos limpos e seus respectivos textos cifrados $c_1 = Enc(p_1, pk)$ e $c_2 = Enc(p_2, pk)$. Na criptografia homomórfica tem-se que efetuar $p_3 = p_1 \bigoplus p_2$, onde $\bigoplus$ é uma operação sobre os textos limpos, implica em $p_3 = Dec(c_3, s_k)$ com $c_3 = c_1 \bigoplus' c_2$, em que $\bigoplus'$ é a mesma operação binária $\bigoplus$, mas sobre os textos cifrados, i.e., a versão homomórfica do operador $\bigoplus$. Em ambos os casos, um primeiro agente, detentor da informação sensível, pode cifrar p_1 e p_2 e enviar c_1 e c_2 para um segundo agente. Este poderia realizar operações homomorficamente sem ter conhecimento dos dados limpos e, por fim, retornar c_3 para o primeiro agente, que faria a decifração do resultado usando sua chave privada para obter finalmente p_3 (informação sensível).

Em pipelines de aprendizado de máquina, a etapa de inferência normalmente é implantada como serviço em infraestrutura não confiável de terceiros. Dessa forma, criptografia homomórfica se apresenta como uma solução, pois permite que a inferência seja executada sem acesso aos dados limpos, conforme é mostrado na Figura 1, na qual é ilustrado um cenário em que quatro clientes cifram seus dados, cada um com sua respectiva chave privada, e enviam o resultado para um servidor. O servidor consegue efetuar operações usando os dados cifrados, porém é incapaz de acessar os dados limpos, conforme mostrado na Figura 2.

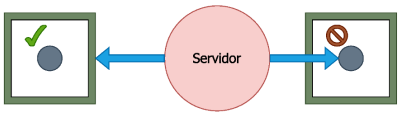
Figura 1 – Clientes enviando dados cifrados ao servidor.



Fonte: Elaborado pela autora.

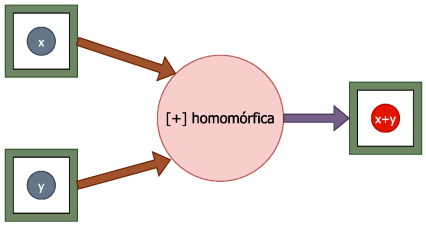
A Figura 3 indica que resultado das operações efetuadas pelo servidor também é

Figura 2 – O servidor não possui acesso aos dados limpos, porém pode manipular a representação dos dados cifrados.



Fonte: Elaborado pela autora.

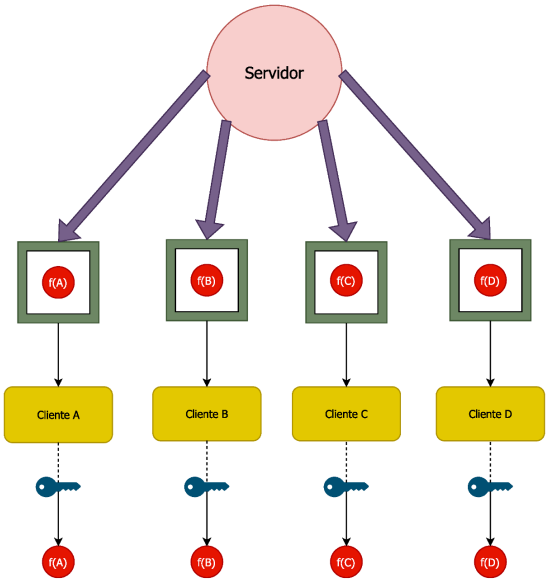
Figura 3 – O resultado das operações efetuadas pelo servidor é cifrado.



Fonte: Elaborado pela autora.

cifrado e somente o cliente, com sua chave secreta, consegue decifrar, situação apresentada na Figura 4. Contudo, para realizar a operação sobre tais representações cifradas é necessário um tipo especial de operador, a saber, a versão homomórfica do operador desejado.

Figura 4 – Servidor enviando o resultado cifrado para os clientes.



Fonte: Elaborado pela autora.

Os algoritmos de criptografia são geralmente baseados em problemas matemáticos cuja resposta é fácil de verificar, mas difícil de calcular. Para obter esta propriedade, tipicamente são empregadas funções unidirecionais (one-way functions). Por exemplo, o esquema criptográfico RSA é descrito com base no problema de fatoração de inteiros grandes em seus

fatores primos. Descobrir todos os fatores primos de um número n muito grande é uma tarefa computacionalmente difícil (ou inviável). No entanto, dito que a e b são fatores primos de n, a verificação desta afirmação é simples em termos computacionais.

O RSA funciona muito bem com computadores clássicos, pois não há soluções conhecidas para encontrar fatores primos de um número em tempo polinomial. Contudo, ao considerar computação quântica, tal esquema criptográfico pode se tornar inseguro. O algoritmo de Shor em computadores quânticos pode quebrar o RSA em tempo polinomial (BHATIA; RAMKUMAR, 2020). Por esta razão, são necessários algoritmos com resistência quântica, i.e., baseados em problemas de difícil solução também para computadores quânticos. Isso motivou o desenvolvimento da criptografia pós-quântica, em especial criptografia baseada em reticulados (lattices).

2.6.2 Aspectos Fundamentais de HE

Os valores cifrados com criptografia homomórfica sofrem dos mesmos problemas fundamentais de representação numérica. Já as versões homomórficas dos operadores sofrem de problemas fundamentais específicos relacionados à criptografia. A seguir, são apresentados tais conceitos fundamentais.

2.6.2.1 Ruído

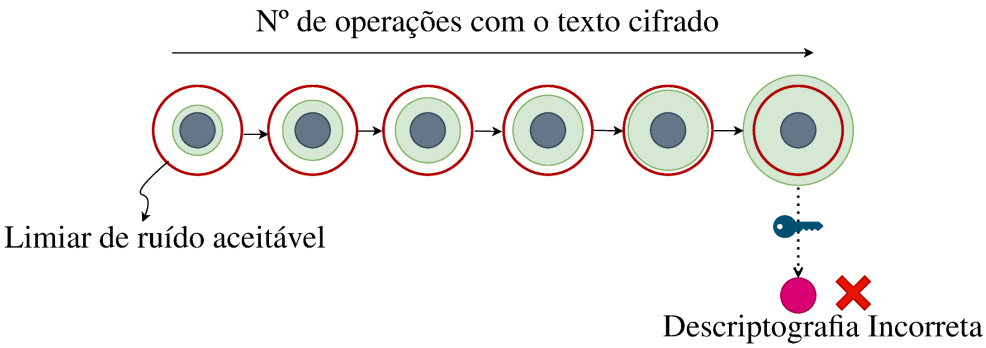
Um esquema criptográfico consiste em um conjunto de algoritmos que efetua a transformação de um texto limpo em um texto cifrado. Nesta transformação, adiciona-se ruído ao texto cifrado, a fim de tornar o processo não determinístico, o que facilitaria a quebra da criptografia por meio de algum método, e.g., ataque por dicionário.

Desta forma, operações homomórficas têm o incômodo efeito de aumentar a magnitude do ruído nos textos cifrados a cada transformação realizada, similar ao erro de representação de operações aritméticas digitais. A Figura 5 apresenta um diagrama de fluxo representando várias aplicações sucessivas de um operador homomórfico e o ruído ou erro acumulado (em verde).

Se o ruído acumulado ultrapassar um determinado limiar, a decifração será falha, podendo retornar um valor que não o esperado (ARMKNECHT *et al.*, 2015). A linha vermelha na Figura 5 representa o limiar máximo de erro para decifrar corretamente o texto cifrado.

Cada esquema criptográfico possui sua própria forma de calcular os níveis aceitáveis

Figura 5 – Crescimento do erro pode tornar a decifração falha.



Fonte: Elaborado pela autora.

de ruído e o limiar máximo para decifrar corretamente os textos cifrados. Para mais detalhes ver (MINELLI, 2018).

2.6.2.2 Classificação de esquemas

Os primeiros esquemas de criptografia homomórfica construídos admitem operações apenas de adição ou multiplicação sobre textos cifrados de forma mutuamente exclusivas, i.e., não ambas, e são nomeados como *Partially Homomorphic Encryption* (PHE) (MINELLI, 2018).

Esquemas *Somewhat Homomorphic Encryption* (SHE) permitem ambas as operações (MINELLI, 2018). Devido aos erros de representação, nos esquemas de SHE, uma quantidade limitada de operações homomórficas é permitida antes que os resultados se tornem não confiáveis e impossíveis de decifrar corretamente.

Já os esquemas FHE permitem uma quantidade arbitrária de ambas as operações (ZHI-GANG *et al.*, 2014). O FHE foi proposto pela primeira vez na década de 1970 (RIVEST *et al.*, 1978) e por muito tempo foi considerado impossível ou impraticável. Em 2009, Craig Gentry apresentou em seu trabalho de doutorado o primeiro esquema de FHE viável (GENTRY, 2009), no qual também foi introduzido o conceito de *bootstrapping* para reduzir o ruído acumulado pelas operações.

Atualmente existem diferentes versões de implementação de esquemas FHE. Em especial, a *Fast Fully Homomorphic Encryption over the Torus* (TFHE) (CHILLOTTI *et al.*, 2020a) apresenta algumas otimizações em fatores como tempo de processamento, dentre outros, e apresenta potencial de aplicação real em tempo prático. Por outro lado, dependendo do domínios dos valores a serem representados e operações sobre tais valores, outros esquemas tornam-se vantajosos. Outro exemplo de esquema eficiente é o BGV/CKKS. Também há propostas híbridas,

e.g., o framework Chimera (BOURA *et al.*, 2020) propõe alternar eficientemente entre esquemas para poder utilizar operações de diferentes naturezas e não permitidas por um único esquema.

Atualmente, há conjecturas sobre a possibilidade da extensão do esquema TFHE incorporar recursos que atualmente são presentes no BGV/CKKS. Por tal motivo, empresas como a francesa Zama têm decidido estrategicamente desenvolver pacotes de software com o esquema TFHE.

2.6.2.3 Bootstrapping

Bootstrapping é uma técnica que permite a redução do ruído de um texto cifrado (MINELLI, 2018). Com isso, remove-se a limitação da quantidade de operações sob texto cifrado que podem ser feitas sem comprometer o processo de decifração.

Basicamente, trata-se de um método que cifra duas vezes um texto limpo m e depois utiliza uma etapa de decifração (de forma homomórfica), mas com a chave secreta sk também cifrada enc_{sk}, chamada chave de bootstrapping (do inglês, bootstrapping key), resultando no texto cifrado c do texto limpo original m, mas desta vez livre do ruído acumulado por operações anteriores.

Dentre as variações de bootstrapping, há o Programmable Bootstrapping (CHILLOTTI *et al.*, 2021), que é uma extensão da versão original, desenvolvida pela Zama, por meio de uma variante do TFHE. Esta técnica permite uma operação homomórfica de qualquer função de um texto cifrado, controlando o nível de ruído. Trata-se de duas operações aplicadas em conjunto, a própria função e o bootstrapping.

2.6.2.4 Key switching

É possível efetuar a troca das chaves utilizadas na cifração. O *Key switching* é um algoritmo que converte uma mensagem cifrada sob uma chave privada (sk) para um novo texto cifrado correspondendo a mesma mensagem, mas cifrado sob a chave privada (sk'). Esse procedimento requer chaves do tipo *key-switching* (troca-chave). Para criação desta chave, é necessária a chave privada (sk).

2.7 Sistema Paillier

O sistema criptográfico de Paillier foi o algoritmo de HE utilizado nos protocolos implementados neste trabalho. Ele é semanticamente seguro e é um esquema de HE aditivo. Por tratar-se de um esquema assimétrico, ele lida com um par de chaves, pk e sk dadas por:

$$pk := (n, g)$$

$$sk := \lambda(n) = MMC(p_1 - 1, q_1 - 1)$$

Sendo $\lambda(n)$ a função de Carmichael, p_1 e q_1 primos grandes distintos, $n = p_1 \times q_1$ e $g \in \mathbb{Z}^*_{n^2}$, em que n divide a ordem de g.

Para cifrar um texto limpo $m \in \mathbb{Z}^*_n$, gera-se um valor aleatório $r \in \mathbb{Z}^*_n$. O texto cifrado c é obtido por meio da seguinte operação:

$$c = E_{pk}(m, r) = g^m \times r^n \mod n^2 \tag{2.2}$$

Sendo E_{pk} a função de cifração com a chave pública p_k.

Para decifrar um texto cifrado $c \in \mathbb{Z}^*_{n^2}$, o texto limpo m correspondente pode ser obtido fazendo:

$$m = D_{sk}(c) = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n \tag{2.3}$$

sendo D_{sk} a função decifração com a chave secreta sk e $L(u) := \frac{(u-1)}{n}$.

Neste esquema, dados $a, b \in \mathbb{Z}_n$, a seguinte propriedade é satisfeita:

$$E_{pk}(a + b) = E_{pk}(a) \times E_{pk}(b) \mod n^2 \tag{2.4}$$

Em especial, para k positivo, tem-se que:

$$E_{pk}(ka) = E_{pk}(a)^k \mod n^2 \tag{2.5}$$

2.8 Regressão Linear usando o Método dos Mínimos Quadrados

O algoritmo de Regressão Linear é um dos mais utilizados em ML para fazer predições. Seja X um conjunto de dados com n observações acerca de d atributos e Y a variável preditora,

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_d \\ X_{21} & X_{22} & \cdots & X_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nd} \end{bmatrix} \quad \text{e} \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

o objetivo da regressão é encontrar β que minimize o erro $\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$ em:

$$Y_i = \beta_1 \times X_{i1} + \beta_2 \times X_{i2} + \cdots + \beta_d \times X_{id} + \alpha + \varepsilon_i, \quad i = 1, \cdots, n \tag{2.6}$$

sendo

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}$$

Para simplificar o problema, será considerado

$$X_i = \begin{bmatrix} X_{i1} & X_{i2} & \cdots & X_{id} & 1 \end{bmatrix} \quad \text{e} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \\ \alpha \end{bmatrix}$$

Assim, o objetivo passa a ser minimizar o erro em

$$Y = X\beta + \varepsilon \tag{2.7}$$

Uma das formas mais comuns de implementação de regressão linear é usando o Método de Mínimos Quadrados. Por meio dele, o β pode ser obtido resolvendo o seguinte sistema linear:

$$A\beta = b \tag{2.8}$$

em que $A = X^T X \in \mathbb{R}^{(d+1)\times(d+1)}$ é a matriz de covariância, $b = X^T Y \in \mathbb{R}^{(d+1)}$ e

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_d & 1 \\ X_{21} & X_{22} & \cdots & X_{2d} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nd} & 1 \end{bmatrix}$$

e, portanto,

$$\beta = A^{-1} b = (X^T X)^{-1} X^T Y \tag{2.9}$$

3 METODOLOGIA

Este trabalho compara duas possíveis implementações de protocolos para a execução de regressão linear usando HE. No trabalho (CHEN; ZHENG, 2022), os autores apresentam uma tabela contendo os valores da métrica R^2 para o modelo com dados cifrados e para o modelo com dados limpos. Já no trabalho (QIU *et al.*, 2020), a métrica utilizada pelos autores para verificar a acurácia dos modelos foi o erro relativo do sistema $\text{Erro}_{\tilde{\beta}}$, dado por:

$$\text{Erro}_{\tilde{\beta}} = \frac{\|\tilde{\beta} - \beta_{\text{limpo}}\|_2}{\|\beta_{\text{limpo}}\|_2} \tag{3.1}$$

sendo β_{limpo} os parâmetros obtidos usando dados puros e $\tilde{\beta}$ obtidos pelo sistema proposto. Este trabalho apresenta resultados utilizando a métrica $\text{Erro}_{\tilde{\beta}}$, por fornecer resultados que facilitam a comparação dos métodos.

Foram considerados 10 clientes para ambos os métodos. Além disso, foram escolhidas 4 bases de dados utilizadas em (QIU *et al.*, 2020). As bases não foram normalizadas e foram executadas as mesmas modificações descritas no artigo original. Todas podem ser encontradas no repositório da UCI (DUA; GRAFF, 2017) e estão listadas abaixo:

- Auto MPG: Este conjunto de dados contém 398 registros sobre carros, com tentativas de prever as milhas por galão para cada um. Foram removidos o atributo do nome do carro e 6 registros que têm valores ausentes de cavalos de potência. No final, o conjunto de dados possui 1 atributo alvo e 7 atributos preditivos, contendo 392 registros.
- Wine Quality: Contém 4898 registros de vinho usados para prever a qualidade do vinho. Escolhemos um conjunto de dados sobre vinho branco que possui 11 atributos preditivos e 1 atributo alvo.
- Bike Sharing: Este conjunto de dados contém 17379 registros sobre aluguéis de bicicletas. Foram removidos o índice do registro, a data, a contagem de usuários casuais e a contagem de usuários registrados do conjunto de dados hour.csv. Restaram 12 atributos para prever a contagem total de bicicletas alugadas.
- Forest Fires: O conjunto de dados contém 517 registros usados para prever a área queimada da floresta. Foi removido o atributo do mês e alterado o conteúdo do atributo do dia da semana de 'mon' para 1, 'tue' para 2, etc. Finalmente, restaram 11 atributos preditivos e 1 atributo alvo.

Já para a comparação de tempo de processamento, foi considerado o tempo de processamento da etapa de regressão linear, após a etapa de agregação, que é comum a ambos

os métodos. Para a medição do tempo, foi utilizada a biblioteca *time* do Python e houve uma conversão para a unidade de minutos.

Este trabalho foi desenvolvido utilizando a linguagem de programação Python e as bibliotecas NumPy, Pandas e SymPy. Além disso, os algoritmos foram escritos utilizando os cadernos disponíveis a partir da ferramenta Jupyter Notebook.

4 DESENVOLVIMENTO

Ambos os métodos utilizam bases de dados (com m linhas e d atributos preditivos) distribuídas horizontalmente entre r clientes. Isso significa que cada informação x_i, y_i não pode ser dividida e pertence a um único cliente. Dessa forma, existem inteiros ℓ_k tal que $0 = \ell_0 < \ell_1 < \cdots < \ell_r = m$ e o cliente k ($1 \leq k \leq r$) vai possuir os dados:

$$(x_{\ell_{k-1}+1}, y_{\ell_{k-1}+1}), (x_{\ell_{k-1}+2}, y_{\ell_{k-1}+2}), \cdots, (x_{\ell_k}, y_{\ell_k})$$

Os dados do cliente k podem ser representados por uma matriz X_k e um vetor Y_k dados por:

$$X_k = [x_{\ell_{k-1}+1}, x_{\ell_{k-1}+2}, \cdots, x_{\ell_k}]^T$$

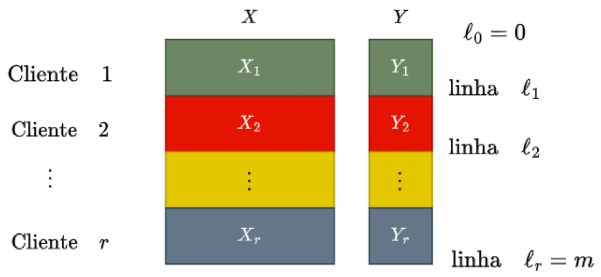
$$Y_k = [y_{\ell_{k-1}+1}, y_{\ell_{k-1}+2}, \cdots, y_{\ell_k}]^T$$

Então, tem-se que:

$$X^T = [x_1^T, \cdots, x_r^T], \quad Y^T = [y_1^T, \cdots, y_r^T]$$

A Figura 6 mostra a relação entre X e X_k, Y e Y_k.

Figura 6 – Base de dados distribuída horizontalmente.



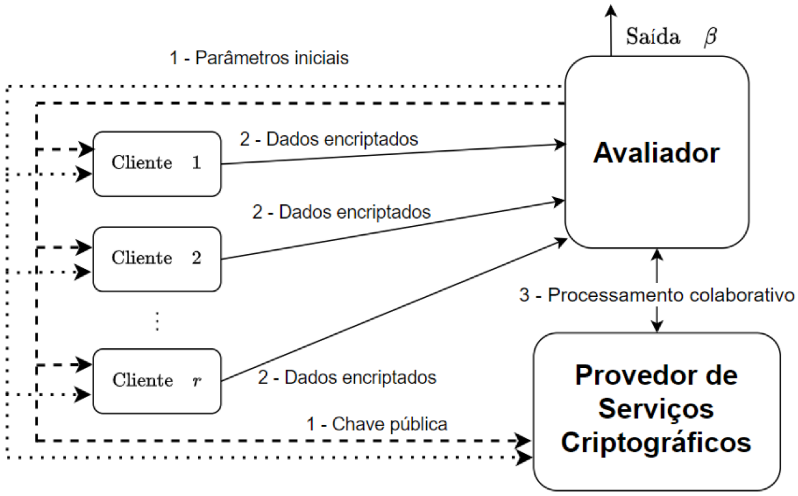
Fonte: Adaptado de (QIU *et al.*, 2020).

4.1 Método 1

A primeira implementação é baseada em (QIU *et al.*, 2020). No trabalho foi desenvolvido utilizado o sistema de Paillier e uma técnica nova de mascaramento de dados. O

framework implementado lida com 3 entidades: clientes, avaliador e o Provedor de Serviços Criptográficos (PSC), conforme mostrado na Figura 7.

Figura 7 – Esquema do sistema.



Fonte: Adaptado de (QIU *et al.*, 2020).

Os clientes são as entidades detentoras dos dados. O avaliador é a entidade que deseja fazer o treinamento e inferência do modelo. Ele pode ser uma empresa que desenvolveu um algoritmo de ML e deseja fazer o treinamento e inferência com dados reais mantendo a privacidade dos detentores desses dados. Já o PSC é quem lida com a parte da criptografia, gerando as chaves pública e privada e fazendo um processamento colaborativo com o avaliador. Em um cenário de saúde, por exemplo, diferentes hospitais (clientes) podem querer colaborar para o treinamento de um modelo de ML, desenvolvido por uma empresa A (avaliador) para prever resultados de saúde com base em dados de pacientes, sem revelar informações sensíveis de pacientes individuais utilizando os serviços de criptografia prestados por uma empresa B.

4.1.1 Conversão de dados

O sistema de Paillier aceita como entrada apenas números inteiros não negativos, tornando necessária uma etapa de conversão de dados que será implementada usando o Protocolo 1. Se todos os dados primários dos clientes forem não negativos, cada cliente pode converter seus dados em números inteiros usando a fórmula $\lfloor a \cdot 2^q \rfloor$, em que a representa seus dados primários. Se houver algum dado de entrada negativo, deve-se enviar um número real positivo M para todos os clientes, que podem usá-lo para fazer a conversão dos dados locais para números

não negativos e, em seguida, converter os dados em números inteiros. O Protocolo 1 fornece a conversão completa de dados quando há dados negativos.

Protocolo 1 Conversão dos dados

Partes:	Clientes	Avaliador	PSC
Entrada:	x_{ij}, y_i	(p,q)	pk
Saída:	$\bar{x}_{ij}, \bar{y}_i \in \mathbb{N}$		

1: Cliente k encontra o valor mínimo z_k na base negativa local $Ng = \{x_{ij}, y_i | x_{ij} \leq 0, y_i \leq 0, l_{k-1} < i \leq l_k, 1 \leq j \leq d\}$, em que $k = 1, 2, \cdots, r$.
2: Client k calcula $E(\lceil -z_k \cdot 2^q \rceil)$ e envia o resultado para o avaliador.
3: Avaliador envia todos os $E(\lceil -z_k \cdot 2^q \rceil)$ para o PSC.
4: PSC decifra todos $E(\lceil -z_k \cdot 2^q \rceil)$, encontra o máximo valor M em $\{\lceil -z_1 \cdot 2^q \rceil / 2^q, \lceil -z_2 \cdot 2^q \rceil / 2^q, \cdots, \lceil -z_r \cdot 2^q \rceil / 2^q\}$ e passa para todos os clientes.
5: Cada cliente converte seus dados locais para inteiros não negativos usando a fórmula $\bar{x}_{ij} = \lfloor (x_{ij} + M) \cdot 2^q \rfloor$ e $\bar{y}_i = \lfloor (y_i + M) \cdot 2^q \rfloor$

4.1.2 Mascaramento dos dados

Nesse sistema, o avaliador possui alguns dados que podem ser vistos como $E(x)$, inteiro c e inteiro público d. Para mascarar o dado x, o avaliador precisa calcular $E(\lfloor xc/d \rceil)$. Isso pode ser calculado fazendo $E(xc) = E(x)^c$. O problema está em como calcular $E(\lfloor xc/d \rceil)$ usando $E(xc)$ e d. Até o momento, não há uma maneira para o avaliador completar essa tarefa sozinho. No Protocolo 2 de mascaramento de dados, é realizada a divisão entre $E(xc)$ e um divisor público d. No protocolo, o $E(\lfloor r/d \rfloor)^{-1}$ é um inverso modular de $E(\lfloor r/d \rfloor)$.

Protocolo 2 (Mascaramento de dados) Calcular $E(\lfloor xc/d \rceil)$, dados $E(x)$, c e d

Partes:	Avaliador (A)	PSC (B)
Entrada:	$E(x)$, inteiros c e d	pk, inteiro d
Saída:	$E(\lfloor xc/d \rceil)$	

1: A calcula $E(xc) = E(x)^c$
2: A escolhe r aleatoriamente, calcula $E(xc + r) = E(xc) \cdot E(r)$. Então A envia $E(xc + r)$ para B.
3: B decifra $E(xc + r)$, calcula $\lfloor (xc + r)/d \rfloor$, então retorna $E(\lfloor (xc + r)/d \rfloor)$ para A.
4: A calcula $E(\lfloor r/d \rfloor)$, depois $E(\lfloor xc/d \rceil) = E(\lfloor (xc + r)/d \rfloor - \lfloor r/d \rfloor) = E(\lfloor (xc + r)/d \rfloor) \cdot E(\lfloor r/d \rfloor)^{-1}$

4.1.3 Regressão linear segura

O Protocolo 3 é a principal contribuição do artigo (QIU *et al.*, 2020). O protocolo consiste em três fases: inicialização, agregação e regressão. O Protocolo 2 de mascaramento de dados é utilizado na fase de regressão. Ao final, o Avaliador gera a saída do modelo de regressão β. No Protocolo 3, o operador $\bigotimes$ denota o produto elemento a elemento de duas matrizes ou vetores.

Protocolo 3 Regressão linear segura usando dados distribuídos

Entrada:
Cliente k: x_i, y_i ($x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $l_{k-1} < \text{i} \leq l_k$), k=1,2,$\cdots$,$r$
Avaliador: p,q - tamanho em bits da parte integral e fracionária
e - um denominador público (primo)
CSP: pk
Saída:
Avaliador: executa o modelo de regressão para obter $\beta \in \mathbb{R}^{(d+1)}$

4.1.3.1 Inicialização

O avaliador define o par (p,q) e o envia para todos os clientes. A seguir, ele escolhe um primo e de q bits que será usado para o mascaramento de dados e o compartilha com o PSC. Se todos os dados forem não negativos, cada cliente faz a conversão de seus dados para inteiros localmente. Caso contrário, todos os clientes, o avaliador e o PSC executam o Protocolo 1 para completar a conversão dos dados. Após essas etapas, todos os x_{ij} e y_i se tornam inteiros não negativos. Por fim, o PSC gera o par de chaves (pk, sk) do esquema de cifragem de Paillier e compartilha pk com todas as outras partes.

4.1.3.2 Agregação

O cliente k calcula $A_k = X_k^T X_k$ e $b_k = X_k^T Y_k$, em seguida cifra esses valores localmente. Então envia $E(A_k)$ e $E(b_k)$ para o avaliador, em que $1 \leq k \leq r$. O avaliador, por sua vez, gera a matriz $E(A)$ e o vetor $E(b)$:

$$E(A) = E\left(\sum_{k=1}^{r} A_k\right) = \bigotimes_{k=1}^{r} E(A_k) \tag{4.1}$$

$$E(b) = E\left(\sum_{k=1}^{r} b_k\right) = \bigotimes_{k=1}^{r} E(b_k) \tag{4.2}$$

4.1.3.3 Regressão

Inicialmente, o avaliador gera inteiros aleatórios v_i, s_i, t_i, w_1, w_2 no intervalo $(e, 2^{10} \times e)$ e obtém $u_i = s_i + t_i$, em que $1 \leq i \leq d+1$. Na etapa seguinte, o avaliador e o PSC executam o protocolo de mascaramento de dados, como mostrado na Tabela 6.

Tabela 6 – Mascaramento de dados.

Partes:	Avaliador	PSC
Entrada 1:	$E(a_{ij})$, $u_i v_j$, e^2	pk, e^2
Saída 1:	$E(\lfloor \frac{u_i v_j a_{ij}}{e^2} \rceil)$	
Entrada 2:	$E(b_i)$, $w_1 s_i$, e^2	pk, e^2
Saída 2:	$E(\lfloor \frac{w_1 s_i b_i}{e^2} \rceil)$	
Entrada 3:	$E(b_i)$, $w_2 t_i$, e^2	pk, e^2
Saída 3:	$E(\lfloor \frac{w_2 t_i b_i}{e^2} \rceil)$	

Fonte: Adaptado de (QIU *et al.*, 2020).

O avaliador envia as saídas 1, 2 e 3 para o PSC, que decifra os dados, obtendo:

$$\tilde{a}_{ij} = \left\lfloor \frac{u_i v_j a_{ij}}{e^2} \right\rceil, \quad \tilde{b}_{1i} = \left\lfloor \frac{w_1 s_i b_i}{e^2} \right\rceil \quad \text{e} \quad \tilde{b}_{2i} = \left\lfloor \frac{w_2 t_i b_i}{e^2} \right\rceil \tag{4.3}$$

Com isso, o PSC resolve os dois seguintes sistemas lineares, retornando $\tilde{\xi}$ e $\tilde{\eta}$ para o avaliador:

$$\tilde{A}\tilde{\xi} = \tilde{b}_1, \quad \tilde{A}\tilde{\eta} = \tilde{b}_2 \tag{4.4}$$

em que, $\tilde{A} \coloneqq \tilde{a}_{ij} \in \mathbb{R}^{(d+1)(d+1)}$, $\tilde{b}_1 \coloneqq \tilde{b}_{i1} \in \mathbb{R}^{(d+1)}$ e $\tilde{b}_2 \coloneqq \tilde{b}_{i2} \in \mathbb{R}^{(d+1)}$.

Por fim, o avaliador calcula a solução $\tilde{\beta}$ da seguinte forma:

$$\tilde{\beta} = V \left(\frac{e\tilde{\xi}}{w_1} + \frac{e\tilde{\eta}}{w_2} \right) \tag{4.5}$$

sendo, $V = diag(v_1/e, v_2/e, \cdots, v_{d+1}/e)$.

4.1.4 Corretude do Protocolo regressão linear

Definindo as matrizes diagonais S, T, U, V, a matriz $\hat{A}$ e os vetores $\hat{b}_1$ e $\hat{b}_2$ da seguinte forma:

$$S = diag(s_1/e, s_2/e, \cdots, s_{d+1}/e) \tag{4.6}$$

$$T = diag(t_1/e, t_2/e, \cdots, t_{d+1}/e) \tag{4.7}$$

$$U = S + T = diag(u_1/e, u_2/e, \cdots, u_{d+1}/e) \tag{4.8}$$

$$V = diag(v_1/e, v_2/e, \cdots, v_{d+1}/e) \tag{4.9}$$

$$\hat{A} = UAV, \quad \hat{b}_1 = \frac{w_1}{e} Sb \quad \text{e} \quad \hat{b}_2 = \frac{w_2}{e} Tb \tag{4.10}$$

em que, s_i, t_i, u_i, v_i, w_1, w_2 e e são inteiros aleatórios descritos no Protocolo 3. De acordo com a definição (4.10), os elementos $\hat{A}$, $\hat{b_1}$ e $\hat{b_2}$ são $\frac{u_i v_j a_{ij}}{e^2}$, $\frac{w_1 s_i b_i}{e^2}$ e $\frac{w_2 t_i b_i}{e^2}$, respectivamente. Nesse caso, $\hat{A}$, $\hat{b}_1$ e $\tilde{b}_2$ são as versões precisas de $\tilde{A}$, $\tilde{b}_1$ e $\tilde{b}_2$ no Protocolo 3. Fazendo $\hat{A}\xi = \hat{b}_1$ e $\hat{A}\eta = \hat{b}_2$, é possível calcular β similarmente a como foi calculado $\tilde{\beta}$ no Protocolo 3. Então, tem-se que:

$$\begin{aligned}
\beta &= V\left(\frac{e\xi}{w_1} + \frac{e\eta}{w_2}\right) \\
&= V\hat{A}^{-1}\left(\frac{e\hat{b}_1}{w_1} + \frac{e\hat{b}_2}{w_2}\right) \\
&= V\hat{A}^{-1}(Sb + Tb) \\
&= V(V^{-1}A^{-1}U^{-1})Ub \\
&= A^{-1}b
\end{aligned} \tag{4.11}$$

4.2 Método 2

O segundo método é baseado em (CHEN; ZHENG, 2022).

O trabalho descreve um protocolo na nuvem para implementação de uma regressão linear aplicando o Método de Mínimos Quadrados usando HE. Neste cenário, a nuvem possui $E(X)$ e $E(Y)$ e calcula $E(X^TX)$ e $E(X^TY)$. Entretanto, por conta da limitação do cálculo da inversa de um texto cifrado, $E(X^TX)$ e $E(X^TY)$ são enviados para a entidade detentora dos dados, que executa a decifração usando a chave secreta obtendo X^TX e X^TY e, em seguida, executa $(X^TX)^{-1}X^TY$ para obter os parâmetros do modelo. Nesse modelo há a necessidade de comunicação entre a nuvem e os detentores dos dados, que também precisarão efetuar tarefas de processamento para o cálculo dos parâmetros.

No artigo original desse método, a questão da distributividade dos dados não é tratada. Porém, por tratar-se de uma característica importante para esse cenário, este trabalho fez adaptações para que o método lidasse com dados distribuídos horizontalmente, similarmente ao executado no método anterior. Além disso, no trabalho original foi utilizada a biblioteca

SEAL. Para este trabalho, o sistema de cifração foi o de Paillier, por isso também foi utilizado o protocolo de conversão de dados apresentado na seção anterior.

5 RESULTADOS

Os resultados foram obtidos utilizando um ambiente de execução com sistema operacional Windows 11, CPU da Intel i5−1135G7 de 11ª geração com 2.4 GHz e 8 GB de RAM.

5.1 Erro

A Tabela 7 mostra o erro do Método 1 considerando o β_{limpo} com conversão de dados. Já a Tabela 8 apresenta o erro do mesmo método, mas considerando o β_{limpo} sem conversão de dados.

Tabela 7 – Erro do Método 1 considerando o β_{limpo} com conversão de dados.

Base de dados	m	d	Erro				
			$q = 10$	$q = 20$	$q = 30$	$q = 40$	$q = 50$
Auto MPG	392	7	$6.34 \cdot 10^{-8}$	$3.10 \cdot 10^{-14}$	$2.80 \cdot 10^{-20}$	$3.14 \cdot 10^{-26}$	$1.20 \cdot 10^{-29}$
Forest Fires	517	11	$1.96 \cdot 10^{-8}$	$1.33 \cdot 10^{-15}$	$7.04 \cdot 10^{-21}$	$2.29 \cdot 10^{-27}$	$2.72 \cdot 10^{-29}$
Wine Quality	4898	11	$1.89 \cdot 10^{-5}$	$5.25 \cdot 10^{-13}$	$5.22 \cdot 10^{-18}$	$8.85 \cdot 10^{-24}$	$5.00 \cdot 10^{-27}$
Bike Sharing	17379	12	$1.36 \cdot 10^{-9}$	$1.81 \cdot 10^{-15}$	$2.51 \cdot 10^{-21}$	$3.86 \cdot 10^{-29}$	$2.52 \cdot 10^{-30}$

Fonte: Elaborada pela autora.

Tabela 8 – Erro do Método 1 considerando o β_{limpo} sem conversão de dados.

Base de dados	m	d	Erro				
			$q = 10$	$q = 20$	$q = 30$	$q = 40$	$q = 50$
Auto MPG	392	7	$1.32 \cdot 10^{-4}$	$9.94 \cdot 10^{-8}$	$1.26 \cdot 10^{-10}$	$1.82 \cdot 10^{-13}$	$2.75 \cdot 10^{-13}$
Forest Fires	517	11	$9.55 \cdot 10^{-5}$	$8.86 \cdot 10^{-7}$	$9.13 \cdot 10^{-10}$	$2.96 \cdot 10^{-12}$	$2.17 \cdot 10^{-12}$
Wine Quality	4898	11	0.213	$2.12 \cdot 10^{-5}$	$8.48 \cdot 10^{-8}$	$5.60 \cdot 10^{-8}$	$5.60 \cdot 10^{-8}$
Bike Sharing	17379	12	$1.52 \cdot 10^{-2}$	$1.96 \cdot 10^{-6}$	$8.74 \cdot 10^{-9}$	$1.04 \cdot 10^{-11}$	$1.30 \cdot 10^{-11}$

Fonte: Elaborada pela autora.

Nota-se que o truncamento tem grande impacto no erro gerado pelo sistema. Isso indica a limitação do sistema ade Paillier em restringir a entrada a números inteiros não negativos, sendo necessário executar a conversão de dados caso sejam utilizados dados de ponto flutuante.

A Tabela 9 mostra a diferença de erro entre os Métodos 1 e 2 considerando o β_{limpo} sem conversão de dados.

Tabela 9 – Diferença de erro entre os Métodos 1 e 2 considerando o β_{limpo} sem conversão de dados.

Base de dados	m	d	Erro				
			$q = 10$	$q = 20$	$q = 30$	$q = 40$	$q = 50$
Auto MPG	392	7	$2.51 \cdot 10^{-9}$	$1.94 \cdot 10^{-14}$	$9.28 \cdot 10^{-17}$	$2.86 \cdot 10^{-17}$	$5.57 \cdot 10^{-17}$
Forest Fires	517	11	$-6.39 \cdot 10^{-10}$	$5.52 \cdot 10^{-14}$	$8.23 \cdot 10^{-17}$	$8.38 \cdot 10^{-17}$	$-1.76 \cdot 10^{-17}$
Wine Quality	4898	11	$1.49 \cdot 10^{-5}$	$5.25 \cdot 10^{-13}$	$6.10 \cdot 10^{-17}$	$4.32 \cdot 10^{-17}$	$-3.45 \cdot 10^{-17}$
Bike Sharing	17379	12	$1.19 \cdot 10^{-9}$	$-1.59 \cdot 10^{-17}$	$-2.54 \cdot 10^{-18}$	$1.12 \cdot 10^{-18}$	$1.84 \cdot 10^{-17}$

Fonte: Elaborada pela autora.

5.2 Tempo de processamento

A Tabela 10 apresenta o tempo de execução da etapa de regressão linear, após a etapa de agregação.

Tabela 10 – Tempo de execução da regressão linear.

Base de dados	m	d	Tempo de execução da regressão linear (min)				
			$q = 10$	$q = 20$	$q = 30$	$q = 40$	$q = 50$
Auto MPG (M1)	392	7	5.67	5.11	5.28	5.11	2.96
Auto MPG (M2)	392	7	$9.70 \cdot 10^{-2}$	$9.85 \cdot 10^{-2}$	0.101	0.112	0.100
Forest Fires (M1)	517	11	12.43	8.05	10.56	9.56	9.86
Forest Fires (M2)	517	11	0.235	0.218	0.233	0.266	0.270
Wine Quality (M1)	4898	11	8.99	7.16	9.39	6.83	6.29
Wine Quality (M2)	4898	11	0.242	0.253	0.305	0.337	0.345
Bike Sharing (M1)	17379	12	9.26	11.17	10.80	11.74	13.87
Bike Sharing (M2)	17379	12	0.272	0.357	0.267	0.856	0.944

Fonte: Elaborada pela autora.

A partir das tabelas, notam-se as diferenças entre os métodos. Ambos necessitam que o detentor dos dados execute o protocolo de conversão, já que se trata do uso do sistema de Paillier, que necessita de dados inteiros não negativos. No entanto, para o Método 2, o detentor precisa executar a etapa final, que decifra X^TX e X^TY, calcula a inversa de X^TX e multiplica o resultado por X^TY decifrado, o que exige mais poder de processamento. Já no Método 1 nenhuma outra etapa além da conversão de dados necessita do cliente, já que todo o resto do processo é executado entre o avaliador e o PSC, diferente do Método 2, que utiliza apenas a entidade do avaliador, além do próprio cliente. Por outro lado, o Método 1 implica mais tempo de processamento, que envolve a etapa de mascaramento de dados para a execução da parte final da regressão linear.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresenta um dos algoritmos mais utilizados em aprendizado de máquina, como prova de conceito para validar transformações homomórficas como alternativa para proteger dados em ambientes hostis. Foram desmistificadas algumas terminologias e principais conceitos de criptografia homomórfica, e foram apresentados os resultados obtidos pelos modelos descritos.

Os métodos apresentados indicam que é possível obter baixos erros ao executar uma regressão linear usando dados cifrados. O Método 2 lida apenas com os clientes e um avaliador e exige que os clientes colaborem com parte do processamento do cálculo dos parâmetros. Já o Método 1 utiliza uma terceira unidade chamada PSC e os clientes só precisam efetuar a conversão de dados. No entanto, o Método 1 possui maior tempo de processamento. Além disso, nota-se a limitação do sistema de Paillier, por lidar apenas com entradas inteiras e não negativas, gerando erro de truncamento pela utilização da conversão de dados.

Criptografia homomórfica é uma tecnologia com potencial de se tornar indispensável nos debates sobre privacidade e uso de dados. Porém, no momento atual, tal tema ainda é relativamente pouco discutido. Os materiais disponíveis, em geral, são complexos e repletos de conceitos matemáticos avançados que intimidam um leitor leigo. Devido a isto, diversas iniciativas da academia e indústria privada têm apresentado alternativas a fim de tornar a implementação mais facilitada. A exemplo, pode-se citar a empresa Zama, que implementa bibliotecas que facilitam o uso de criptografia homomórfica (CHILLOTTI *et al.*, 2020b), e a padronização de um protocolo para uso da tecnologia de criptografia homomórfica liderada pela comunidade acadêmica (LAUTER, 2021).

Como sugestão para trabalhos futuros, seria interessante a integração do treinamento usando DP com a inferência usando FHE, ideia desenvolvida no trabalho (SÉBERT *et al.*, 2022).

REFERÊNCIAS

ACQUISTI, A.; BRANDIMARTE, L.; LOEWENSTEIN, G. Secrets and likes: The drive for privacy and the difficulty of achieving it in the digital age. **Journal of Consumer Psychology**, Wiley Online Library, v. 30, n. 4, p. 736–758, 2020.

AL-AZIZY, D.; MILLARD, D.; SYMEONIDIS, I.; O'HARA, K.; SHADBOLT, N. A literature survey and classifications on data deanonymisation. In: SPRINGER. **Risks and Security of Internet and Systems: 10th International Conference, CRiSIS 2015, Mytilene, Lesbos Island, Greece, July 20-22, 2015, Revised Selected Papers 10**. [*S. l.*], 2016. p. 36–51.

ARMKNECHT, F.; BOYD, C.; CARR, C.; GJØSTEEN, K.; JÄSCHKE, A.; REUTER, C. A.; STRAND, M. A guide to fully homomorphic encryption. **Cryptology ePrint Archive**, 2015.

BHATIA, V.; RAMKUMAR, K. An efficient quantum computing technique for cracking rsa using shor's algorithm. In: IEEE. **2020 IEEE 5th international conference on computing communication and automation (ICCCA)**. [*S. l.*], 2020. p. 89–94.

BOURA, C.; GAMA, N.; GEORGIEVA, M.; JETCHEV, D. Chimera: Combining ring-lwe-based fully homomorphic encryption schemes. **Journal of Mathematical Cryptology**, De Gruyter, v. 14, n. 1, p. 316–338, 2020.

BRASIL. **Lei Geral de Proteção de Dados Pessoais**. 2018. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm.

CADWALLADR, C.; GRAHAM-HARRISON, E. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. **The guardian**, v. 17, n. 1, p. 22, 2018. Disponível em: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

CHEN, B.; ZHENG, X. Implementing linear regression with homomorphic encryption. **Procedia Computer Science**, Elsevier, v. 202, p. 324–329, 2022.

CHEN, H.; GILAD-BACHRACH, R.; HAN, K.; HUANG, Z.; JALALI, A.; LAINE, K.; LAUTER, K. Logistic regression over encrypted data from fully homomorphic encryption. **BMC medical genomics**, Springer, v. 11, p. 3–12, 2018.

CHILLOTTI, I.; GAMA, N.; GEORGIEVA, M.; IZABACHÈNE, M. Tfhe: fast fully homomorphic encryption over the torus. **Journal of Cryptology**, Springer, v. 33, n. 1, p. 34–91, 2020.

CHILLOTTI, I.; JOYE, M.; LIGIER, D.; ORFILA, J.-B.; TAP, S. Concrete: Concrete operates on ciphertexts rapidly by extending tfhe. In: **WAHC 2020–8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography**. [*S. l.*: *s. n.*], 2020. v. 15.

CHILLOTTI, I.; JOYE, M.; PAILLIER, P. Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. In: SPRINGER. **International Symposium on Cyber Security Cryptography and Machine Learning**. [*S. l.*], 2021. p. 1–19.

CROSS, C.; PARKER, M.; SANSOM, D. Media discourses surrounding 'non-ideal'victims: The case of the ashley madison data breach. **International Review of Victimology**, SAGE Publications Sage UK: London, England, v. 25, n. 1, p. 53–69, 2019.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: http://archive.ics.uci.edu/ml.

DUARTE, F. **Amount of Data Created Daily (2023)**. 2023. https://explodingtopics.com/blog/data-generated-per-day.

ECONOMIST, T. The world's most valuable resource is no longer oil, but data. **The Economist**, 6 de Maio 2017. Disponível em: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

European Commission. **Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)**. European Commission, 2016. Disponível em: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

EVANS, D.; KOLESNIKOV, V.; ROSULEK, M. *et al.* A pragmatic introduction to secure multi-party computation. **Foundations and Trends® in Privacy and Security**, Now Publishers, Inc., v. 2, n. 2-3, p. 70–246, 2018.

FORBES. **Data privacy will be the most important issue in the next decade**. 2019. https://www.forbes.com/sites/marymeehan/2019/11/26/data-privacy-will-be-the-most-important-issue-in-the-next-decade/?sh=30d1d86e1882.

GANTA, S. R.; KASIVISWANATHAN, S. P.; SMITH, A. Composition attacks and auxiliary information in data privacy. In: **Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining**. [*S. l.*: *s. n.*], 2008. p. 265–273.

GENTRY, C. **Fully homomorphic encryption scheme**. Tese (Doutorado) – Stanford University, 2009.

HAN, K.; HONG, S.; CHEON, J. H.; PARK, D. Logistic regression on homomorphic encrypted data at scale. In: **Proceedings of the AAAI conference on artificial intelligence**. [*S. l.*: *s. n.*], 2019. v. 33, n. 01, p. 9466–9471.

LAUTER, K. E. **Private AI: Machine Learning on Encrypted Data**. 2021. Cryptology ePrint Archive, Report 2021/324. https://ia.cr/2021/324.

MINELLI, M. **Fully homomorphic encryption for machine learning**. Tese (Doutorado) – Université Paris sciences et lettres, 2018.

NARAYANAN, A.; SHMATIKOV, V. Robust de-anonymization of large sparse datasets. In: IEEE. **2008 IEEE Symposium on Security and Privacy (sp 2008)**. [*S. l.*], 2008. p. 111–125.

PARK, S.; BYUN, J.; LEE, J.; CHEON, J. H.; LEE, J. He-friendly algorithm for privacy-preserving svm training. **IEEE Access**, IEEE, v. 8, p. 57414–57425, 2020.

PODSCHWADT, R.; TAKABI, D.; HU, P.; RAFIEI, M. H.; CAI, Z. A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption. **IEEE Access**, IEEE, v. 10, p. 117477–117500, 2022.

QIU, G.; GUI, X.; ZHAO, Y. Privacy-preserving linear regression on distributed data by homomorphic encryption and data masking. **IEEE Access**, IEEE, v. 8, p. 107601–107613, 2020.

RIVEST, R. L.; ADLEMAN, L.; DERTOUZOS, M. L. *et al.* On data banks and privacy homomorphisms. **Foundations of secure computation**, Citeseer, v. 4, n. 11, p. 169–180, 1978.

SARKAR, E.; CHIELLE, E.; GURSOY, G.; CHEN, L.; GERSTEIN, M.; MANIATAKOS, M. Privacy-preserving cancer type prediction with homomorphic encryption. **Scientific reports**, Nature Publishing Group UK London, v. 13, n. 1, p. 1661, 2023.

SÉBERT, A. G.; SIRDEY, R.; STAN, O.; GOUY-PAILLER, C. Protecting data from all parties: Combining fhe and dp in federated learning. **arXiv preprint arXiv:2205.04330**, 2022.

SMYTH, S. M. The facebook conundrum: is it time to usher in a new era of regulation for big tech? **International Journal of Cyber Criminology**, International Journal of Cyber Criminology, v. 13, n. 2, p. 578–595, 2019.

TRUEX, S.; BARACALDO, N.; ANWAR, A.; STEINKE, T.; LUDWIG, H.; ZHANG, R.; ZHOU, Y. A hybrid approach to privacy-preserving federated learning. In: **Proceedings of the 12th ACM workshop on artificial intelligence and security**. [*S. l.*: *s. n.*], 2019. p. 1–11.

ZHENG, H.; HU, H.; HAN, Z. Preserving user privacy for machine learning: local differential privacy or federated machine learning? **IEEE Intelligent Systems**, IEEE, v. 35, n. 4, p. 5–14, 2020.

ZHI-GANG, C.; JIAN, W.; LIQUN, C.; XIN-XIA, S. Review of how to construct a fully homomorphic encryption scheme. **International Journal of Security and Its Applications**, 2014.

ZUBOFF, S. **The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power**. 1st. ed. [*S. l.*: *s. n.*], 2018. ISBN 1610395697.