**ESTUDO DE CASO PARA MÉTODOS DE APLICAÇÃO DA LEI GERAL DE PROTEÇÃO DE DADOS**

Juan Cardoso da Silva, Ronaldo Celso Messias Correia

Departamento de Matemática e Computação (DMC)

Universidade Estadual Paulista “Júlio Mesquita Filho” (UNESP)

Faculdade de Ciências e Tecnologia (FCT)

Presidente Prudente - SP, Brasil

[**juan.c.silva@unesp.br**](mailto:juan.c.silva@unesp.br)**,** [**ronaldo.correia@unesp.br**](mailto:ronaldo.correia@unesp.br)

**Resumo – Com a ampliação das leis protetoras privacidades para o mundo digital ao redor do mundo como GDPR (General Data Protection Regulation) e a LGPD (Lei Geral de Proteção de Dados – Surgiu um problema relacionado a proteção de dados privados onde sistemas não têm mecanismos adequados para proteção desses dados ou os mecanismos criados/utilizados não são suficientes para o tratamento de dados. Neste trabalho é demonstrado como modelos de aprendizado de máquina supervisionados são capazes de identificar os dados, julgar o contexto dos dados e realizar uma operação de encriptação como mecanismo de segurança, garantindo a integridade dos dados no contexto da lei. No artigo é explorado como o aprendizado de máquina pode realizar o processo de garantir o cumprimento da lei, a quantidade de detecções e a capacidade de julgar dados a partir de uma base de dados gerados.**

**Abstract – With the growth of regulations that focus in protecting privacy around the world such as the GDPR (General Data Protection Regulation) and the LGPD (Lei Geral de Proteção de Dados) – This emerged one problem related with the protection of private date on systems where doesn’t have enough mechanism to protect and process it or said mechanism isn’t enough to process data properly according with the law. In this work it’s shown how supervised machine learning models are capable of identifying data, judge its context and execute one operation of encryption as an answer for the security mechanism, providing data integrity in the law context. In this paper it’s explored how machine learning is capable of ensuring compliance with the law, how much of the detections are properly made and the capacity of judging data from a generated database.**

# INTRODUÇÃO

Com a evolução da tecnologia focada em conectar computadores se tornou disponível para as pessoas nos anos 90, surgiu a World Wide Web e futuramente, nomeado Web 1.0 – Sendo um sucesso mundial e fornecendo diversos sites focados em encontrar informações espalhadas na internet, geralmente contendo sites estáticos e sem muitos detalhes e funcionalidades, também, essa versão da web fornece atualizações para sistemas e softwares, tais como o rudimentar Windows Update do Windows 95 e sites de terceiros dedicados para jogos como Quakeworld (hoje em dia continuado por fãs, o web site original está fechado), nessa época a internet vivia sem rumo sobre sua funcionalidade concreta no futuro, até que em 1999 surgiu o primeiro uso do termo para sua segunda iteração, a Web 2.0

A Web 2.0 foi pensada na interatividade entre pessoas na internet além do acesso e busca de informações, com isso diversos sites novos começaram a nascer focados em fornecer serviços como os de construção de *Wikis*, Aplicações Web, *Mashups*, Redes Sociais, Eventos Colaborativos (caridades e doações para determinados motivos tais quais crowdsourcing e crowdfunding, ambos para financiar tipos de projetos, tais como *Patreon* e *Kickstarter*), Conteúdo gerado de usuários (*Youtube*), Blogs, Streaming Curadoria Social (*Reddit* e *Instagram*). Com esse serviço disponível agora em mãos dos usuários e popularizado ao ponto de dados sensíveis como cartões de créditos, emails e outras informações pessoais circulando, era inevitável a chegada de agentes indesejáveis atacando a rede em busca de acesso a esses dados, buscando lucro em tentativas de se apropriar desses dados.

Os ataques desses agentes eventualmente resultaram na criação de leis progenitoras, focadas em tratar ataques relacionados a privacidades, como a Lei de Combate a Crimes Cibernéticos (12.737/2012) criada para tratar o caso da Carolina Dieckmann. Em 2016 um evento no Reino Unido onde o Facebook indevidamente estava utilizando dados privados de alunos em uma universidade, acabou resultando na *General Data Protection Regulation* (GDPR), dois anos depois, o congresso brasileiro criou sua resposta em forma de lei, utilizando a lei 12.737/2012 e bases da GDPR para criar a Lei Geral de Proteção de dados (LGPD). Após sua implementação em maio de 2021, diversos métodos de implementação da lei foram adotados para respeitá-la, de novas opções de gerenciamento de cookies até como os dados são manipulados no em sistemas.

Neste artigo é apresentado como a LGPD pode ser utilizada em conjunto de modelos de machine learning, identificando os dados relevantes e encriptando eles, deixando os dados irrelevantes sem encriptação, por fim, comparar os resultados dos modelos a apresentados, suas limitações e assegurar os incisos mencionados sejam respeitados.

O objetivo principal deste artigo é demonstrar a capacidade dos modelos de máquinas supervisionados e seu desempenho quanto às identificações dos dados privados e públicos, como também a capacidade de realizar previsões para encaminhar esses dados para um algoritmo de encriptação responsável por gerar uma base dados contendo dados privados encriptados e dados públicos vistos a olho nu, pois não se encaixam dentro do contexto dos incisos do artigo 6 da lei apresentada.

Os objetivos secundários deste artigo são mostrar como as configurações dos modelos de máquinas supervisionados se comportam com configurações diferentes nos modelos e como essas configurações modificam a capacidade de identificar e realizar previsão dos dados privados e procurar uma aplicabilidade deste método no mundo real.

Na seção III Metodologias é mostrado como foi desenvolvido os modelos para solucionar o problema apresentado, na seção IV é apresentado o estudo de caso como o modelo se comporta para solucionar o problema, em V é apresentado o desempenho dos modelos, VI é a seção de limitações e discussões de como alguns dos processos apresentados na seção III pode ter impactado o projeto, em VII as possíveis aplicações de um projeto como este, na seção VIII os trabalhos futuros e por fim as conclusões tiradas após a experimentação e desenvolvimento do projeto.

1. **FUNDAMENTAÇÃO TEÓRICA**

Esta seção é dedicada a discussão sobre os conceitos básicos relacionados aos temas da LGPD, Machine Learning e encriptação.

# LEI GERAL DE PROTEÇÃO DE DADOS

Criada em 2018 e entrado em vigor em 2021, a Lei Geral de Proteção de Dados 13.709/2018 (abreviando como LGPD) possui o objetivo de garantir os direitos de privacidade e proteção de dados sensíveis de usuários e o livre desenvolvimento da personalidade pessoal natural – a lei atua também como um mecanismo de segurança judiciária, promovendo padronizações, regulamentos, práticas de proteção de dados pessoais/sensíveis e a punição por multas e até prisões, caso ocorra não realização da lei.

Pela lei, dados sensíveis/privados são informações relacionadas diretamente à intimidade pessoal de um indivíduo, opiniões políticas, convicções religiosas, filiação a organizações religiosas/políticas/filosóficas, etnia, dados relacionados à saúde, dados genéticos (como biometria e afins), CPF e todo outro tipo de informação do qual possa ser vinculada a vida íntima de uma pessoa.

A lei garante ao cidadão a capacidade de deleção dos dados sensíveis/privados, revogar consentimento do uso de tais dados, transferir os dados para outro serviço similar ao serviço atual, acesso aos dados para atualizá-los e garantir a consistência dos dados no meio digital.

Além disso a lei [5], também dispõe significados para como esses dados podem ser manipulados, pela lei, o tratamento de dados é entendido como qualquer tipo de operação onde os dados pessoais recebem interação por parte de um profissional, tais como acessar um dado, alterar, deletar, produzir, coletar, classificar, acessar, reproduzir, copiar, processar, armazenar e quais queres outras manipulações possíveis com os dados.

A lei também determina a existência de um operador e um controlador, operadores são pessoas das quais possam tomar decisões finais a respeito de como o dado vai ser manipulado e os operadores realizam as manipulações, ambos podem ser pessoas naturais ou jurídicas, sendo elas do âmbito público ou privado [5].

Pelas considerações do Art. 6 [5], o tratamento de dados pessoais deve seguir alguns princípios, para este projeto, os princípios considerados para atingir o objetivo procurado são transparência, adequação, finalidade e responsabilização.

Os princípios seguem como [5]:

1. **Finalidade:** realização do tratamento para propósitos legítimos, específicos, explícitos e informados ao titular, sem possibilidade de tratamento posterior de forma incompatível com essas finalidades;
2. **Adequação:** compatibilidade do tratamento com as finalidades informadas ao titular, de acordo com o contexto do tratamento;
3. **Necessidade:** limitação do tratamento ao mínimo necessário para a realização de suas finalidades, com abrangência dos dados pertinentes, proporcionais e não excessivos em relação às finalidades do tratamento de dados;
4. **Livre Acesso:** garantia, aos titulares, de consulta facilitada e gratuita sobre a forma e a duração do tratamento, bem como sobre a integralidade de seus dados pessoais;
5. **Qualidade de dados:** garantia, aos titulares, de exatidão, clareza, relevância e atualização dos dados, de acordo com a necessidade e para o cumprimento da finalidade de seu tratamento;
6. **Transparência:** garantia, aos titulares, de informações claras, precisas e facilmente acessíveis sobre a realização do tratamento e os respectivos agentes de tratamento, observados os segredos comercial e industrial;
7. **Segurança:** utilização de medidas técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão;
8. **Prevenção:** adoção de medidas para prevenir a ocorrência de danos em virtude do tratamento de dados pessoais;
9. **Não discriminação:** impossibilidade de realização do tratamento para fins discriminatórios ilícitos ou abusivos;
10. **Responsabilização:** demonstração, pelo agente, da adoção de medidas eficazes e capazes de comprovar a observância e o cumprimento das normas de proteção de dados pessoais e, inclusive, da eficácia dessas medidas.

# MACHINE LEARNING

*Machine Learning*, uma categoria pertencente à área de IA tem como objetivo “treinar” máquinas com intenção de resolver problemas, automatizar tarefas e identificar padrões.

Com as capacidades modernas da *Machine Learning* aumentando, a área deixou de apenas ser estruturação e dedução de dados, evoluindo para entender padrões indefinidos usando técnicas de treinamento do qual cada padrão aprendido utilizado, pode ser utilizado para analisar, outros padrões indefinidos, compreender dados e automatizar tarefas. Hoje me dia, com a capacidade computacional da nuvem e a abrangente quantidade de dados disponíveis, ou até mesmo gerados por outras ML para treinar ML, demonstrando a capacidade de ser aplicada em diversos cenários diferentes para chegar a um resultado esperado/procurado, ajudando em várias áreas de atuações nas profissões, com as maiores contribuições da ML sendo para as áreas da saúde e áreas de processamento de imagens. Este trabalho foca em trazer essas ideias de aplicações de ML para área de direito aplicada, mais especificamente na aplicação de LGPD.

Utilizando *Deep Learning* podemos usar a ML para aprender a não só reconhecer e identificar padrões, como utilizar para realizar avaliações de dados, seguindo algumas categorias de avaliações.

# TENSORFLOW

*TensorFlow*, é um framework de algoritmos de *Deep* *Learning* e *Machine* *Learning*, desenvolvido pela *Google* *Brains* e tornado *open-source* em 2015, sendo uma biblioteca aberta utilizando computação numérica em larga escala

O *TensorFlow* cria grafos/estruturas permitindo o fluxo destes sejam processados, cada vértice do grafo possui a representatividade de operações matemáticas e cada aresta, um array multidimensional, chamado de *tensor*.

A biblioteca permite funcionar localmente, consumindo GPU e CPU da máquina do programador ou utilizando uma máquina na nuvem, com um maior poder computacional, neste caso, utilizando uma TPU resultando em maior poder computacional. No desenvolvimento do projeto, no entanto, foi utilizado GPU em conjunto da CPU localmente em um Tensorflow instalado localmente na máquina e rodando dentro de um ambiente curado no Jupyter-Notebook onde foram realizados tratamento de dados, criação de modelos e testes.

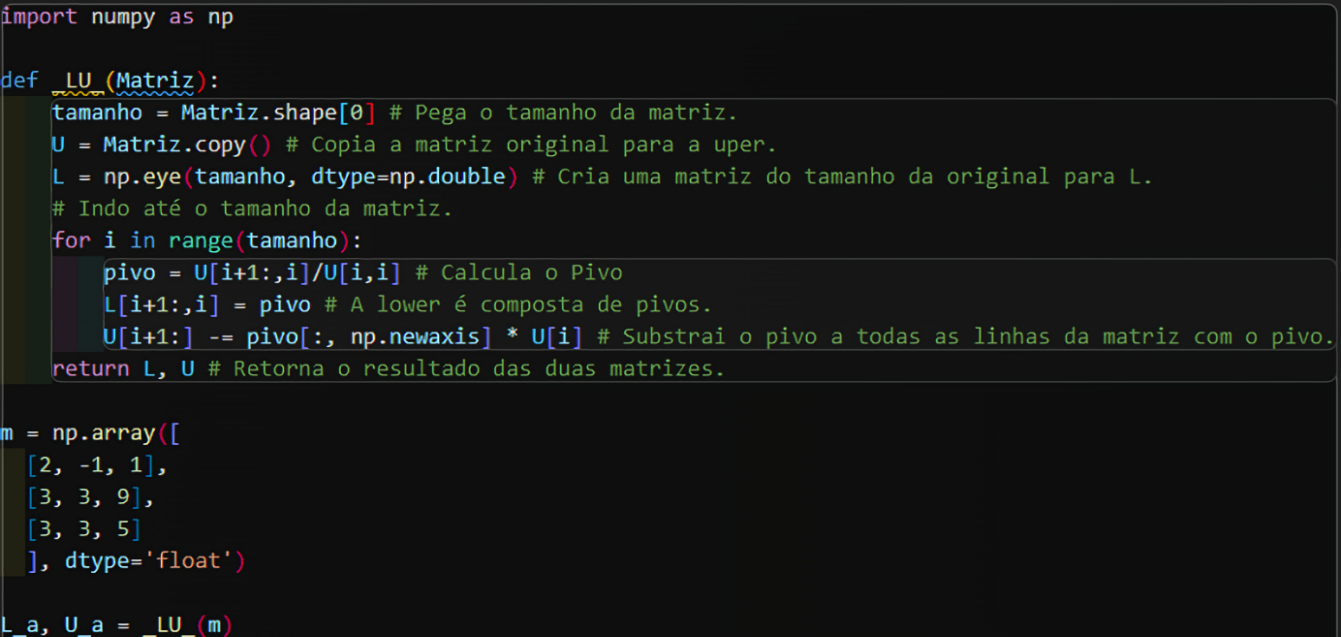
# JUPYTER NOTEBOOK

Jupyter-Notebook é um ambiente para executar e projetar códigos na web utilizando python como sua linguagem de programação, cada notebook é dividido em seções acompanhadas com áreas de textos contendo markdown text para anotações e títulos. Cada seção dentro de um notebook (chamada de documentos) possui um input e um output, servindo como um espaço de log para cada documento e permitir verificar resultados anteriores como permitir uma modularidade maior para corrigir e escrever código.

# MATPLOTLIB, NUMPY E PANDSA

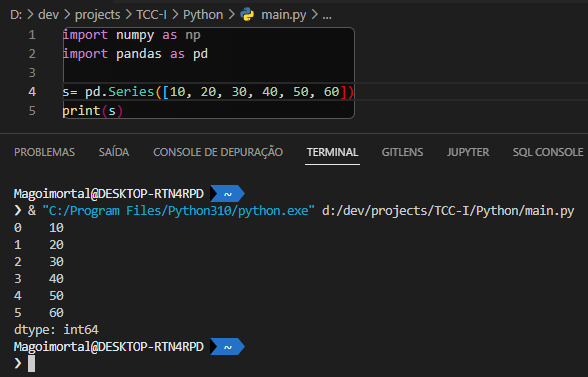
Numpy é uma biblioteca de Python criada utilizando vetorização invés de loops, arrays e indexação, utilizando código de máquina pré compilado em C baseada no objeto *array* da biblioteca para realizar as operações com velocidade e consistência, pois todas as operações necessárias estão pré-compiladas e armazenadas pronto para serem utilizadas.

Figura 1 Exemplo de operação com matrizes utilizando NumPy.



Fonte: Elaborado pelo Autor

Figura 2 Exemplo de uma estrutura de série utilizando os pandas.



Fonte: Elaborado pelo autor.

Pandas permite o cientista poder manipular os dados sem necessitar preocupar com tratamento das estruturas, complexidade e tempo de velocidade dentro de um algoritmo, facilitando a operações/manipulações com dados, estruturando-os para uso em conjunto de outras aplicações, como aprendizado de máquina, visualização de dados ou até mesmo Cálculo Numérico.

Matplotlib é uma biblioteca implementada em Python para visualização de dados em forma gráfica, permitindo visualização de dados conforme a necessidade do programador ao realizar testes, experimentos, pesquisas e trabalhar com resultados obtidos de projetos.

# CRIPTOGRAFIA

RSA é um algoritmo de criptografia desenvolvido em 1976 logo depois do primeiro algoritmo de compartilhamento de chaves chamado Diffie-Hellman apresentado. O RSA permite a encriptação de dados e a desencriptarão dos mesmos utilizando uma chave pública, privada e um números primos para geração dessas chaves.

Primeiro selecionamos um valor P e Q diferentes e quando multiplicado sejam maiores que 400, depois calculamos o valor N com multiplicação de P por Q, em seguida calculamos o euler totiente:

Depois seleciona um inteiro E que seja co-primo do nosso euler totiente e procedemos a calculamos nosso D:

Sabendo nosso D e E podemos criar as chaves públicas e privadas e fazer as encriptações de ida e volta conforme a necessidade:

Processo de ida:

Processo de volta

# FAKER API

Faker API é uma API de geração de dados utilizada para construções de projetos, testes unitários e criação de backends – cada dado gerado pelo fake é razoável em relação a um dado “real”. A API foi escrita primeiramente em PERL e depois portada para outras linguagens de programação, a API permite também que a formação desses dados seja baseada no país determinado pelo programador.

# FUNÇÕES DE ATIVAÇÃO

Para criar um modelo de Machine Learning utilizando Neural Networks (NN) usa-se funções de ativação em suas entradas, saídas e principalmente nos *hiddens layers* (camadas escondidas), sendo essa a camada de computação e cálculos dos pesos, sendo responsável por pegar os dados de entradas, computá-los e ejetá-los para saída

Em cada camada existe um nodo contendo funções de ativações como outras configurações, como conexões, dimensões de entradas e total de saídas, os nodos de entrada tem como foco em providenciar dados para as camadas ocultas, por isso, os modelos desenvolvidos para apresentar neste artigo podem ter funções recorrentes entre eles, já as camadas escondidas possuem funções de ativação diferenciadas entre modelos, cada função realiza um cálculo diferente para avaliar os dados que passaram pelo pré-processamento e passar para a camada de saída onde as funções de ativação dessa camada são utilizadas para mostrarem os resultados da camada anterior, importante realçar as camadas escondidas são abstraídas e não são expostas de qualquer maneira.

# TRABALHOS RELACIONADOS

A IBM realizou um trabalho onde foi desenvolvido um sistema automatizado na nuvem para detecção de irregularidades em uma base de dados, chamado de QRadar [3]. Este sistema realiza varreduras para confirmar se existem casos onde os dados sensíveis foram deletados com sucesso da base, se existir vestígios ou uma possibilidade de a deleção não ocorrer com sucesso, um trigger é ativado para realizar outra varredura mais profunda, os modelos apresentados anteriormente, realizar encriptação dos dados para estar de acordo com o inciso de permanência dos dados, já que caso necessário por investigação, eles precisam estar disponíveis, entretanto, os modelos não possui a capacidade de varreduras profundas como o modelo da IBM.

Neste trabalho de pesquisa [4], também da IBM, o modelo de ML utilizado por pelos pesquisadores usa escolhas binárias para os dados coletados serem minimizados de acordo com a GDPR, no caso, estão a utilizar dado de um hospital para escolher quais dados podem ser deletados baseado nos parâmetros escolhidos. Comparado aos modelos, apesar de um resultado mais condizente com a LGPD, existe uma porcentagem pequena de valores não encriptados como falsos negativos, onde neste projeto, pelo escopo mais simples, foi possível atingir o corte dos dados necessários dos salários para aparecer na dashboard de dados.

# METODOLOGIAS

Importante realçar, neste artigo foi-se utilizado dados projetados pelo programador para realizar o treinamento e teste dos modelos a serem apresentados, tais dados foram criados com o Faker API e exportado para um CSV usando Python.

O conceito para o funcionamento do modelo seria utilizar algumas bases das LGPD, como a aplicação dos incisos I, VII e VIII para criar o modelo e algoritmos necessários para avaliar e identificar dados sensíveis de uma base de dados, usar o vetor de previsão da função de ativação do modelo para determinar a veracidade dos dados quanto o definido pela classe na base de dados e baseado nesse vetor resultado, passar a base junto com a previsão do modelo para encriptar os dados e deixar os dados irrelevantes de fora desse processo, o processo de encriptação utilizado permite encriptar e desencriptar os dados, garantindo o cumprimento da LGPD.

No projeto foram feitos várias bases de dados utilizando uma API de geração de dados sensíveis, inicialmente apenas uma base simplificada contendo apenas duas classes, uma contendo o dado e outra contendo o resultado do dado, para comparar com a previsão do modelo, depois foram criados bases com mais classes para dados e a classe contendo o resultado para comparação, o primeiro modelo feito focado em verificar dados sensíveis utiliza bases de teste e bases de treino, como também bases para verificar a previsão do modelo.

O processo generalizado do modelo pode ser entendido como: alimentar o modelo com uma base treino e uma base teste, verificar se a acurácia está dentro do aceitável (maior que 80%), então verificar através de gráficos e distribuições de probabilidade algum tipo de outlier ou discrepância, por fim o modelo pode fazer uma previsão e usa-la em conjunto com o algoritmo de encriptação para gerar a base resultado com dados sensíveis encriptados e dados irrelevantes inalterados.

A utilização do Faker API é um dos primeiros pontos a serem discutidos na aplicação do projeto, por ser uma API de geração de dados razoáveis porém falsos – os dados gerados não são relacionados a pessoas reais apesar de terem todas características de dados sensíveis (relacionados a pessoas reais e sua privacidade, como definido na LGPD), isso permite a experimentação mais aberta e sem precisar utilizar empresas fornecedoras de dados privados sem gastar recursos, como arriscar vestígios de dados sensíveis nos melhores pesos dos modelos *de machine leraning.*

Utilizando RSA é possível pegar os dados fornecidos pelo Faker API e encripta-los de uma maneira onde é possível realizar o retorno dos dados para serem visualizados como a lei LGPD pede, utilizando-a em conjunto com os modelos, cada dado avaliado será tratado como condizente com o dado sensível ou não, se for, será encriptado.

Após a instalação e preparação das API’s, foi feito uma classe especializada para criação de dados privados, a classe funciona utilizando um atributo determinador do país dos dados a serem gerados, no caso como o foco é no Brasil, será utilizado ‘pt-br’ no atributo da classe, permitindo acessar chamadas do Faker para gerar dados especificados pelo programador, como CPF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorítimo 1: Sensitive Data Generator. Elaborado pelo autor. | | | | | |
|  | ***Input:*** *Nome do arquivo(string), título (lista de strings), tamanho (em inteiro).* | | | | |
|  | ***Output:*** *Criação de um arquivo csv com N colunas e M quantidade de linhas de dados.* | | | | |
| **1:** | ***classe*** *sensitive-data-generator****:*** | | | | |
| **2:** |  | *iniciar(lingua)****:*** | | | |
| **3:** |  |  | *config-tipo-de-dados ← língua* | | |
| **4:** |  | ***fim fun*** |  | | |
| **5:** |  | *gerar-dados-sensiveis-aleatorios()****:*** | | | |
| **6:** |  |  | *gerador ←* ***novo*** *gerador(****self****.config-tipo-de-dados)* | | |
| **7:** |  |  | ***se (converte-para-inteiro(****random(0,1))* ***<*** *1****)*** *então****:*** | | |
| **8:** |  |  |  | ***retorne lista(lista(****gerador.dados-sensiveis()****)****, 1****)*** | |
| **9:** |  |  | ***fim*** |  | |
| **10:** |  |  | ***se-não****:* | | |
| **11:** |  |  |  | ***retorne lista(lista****(gerador.dados-não-sensiveis()****)****, 0****)*** | |
| **12:** |  |  | ***fim*** |  | |
| **13:** |  | ***fim*** |  |  | |
| **14:** |  | *escrever-dados-em-csv(nome-arquivo, titulo, tamanho)****:*** | | | |
| **15:** |  |  | ***abra-arquivo(****nome, ’w’, encoding ← ‘uft-8’****)******como*** *arquivo\_csv****:*** | | |
| **16:** |  |  |  | *escritor-arquivo ←****novo******escritor-csv****(arquivo\_csv)* | |
| **17:** |  |  |  | ***se (titulo******==******NULL) então:*** | |
| **18:** |  |  |  |  | *escritor-arquivo.****escreve-coluna****(‘A’, ‘B’, ’C’... N)* |
| **19:** |  |  |  | ***fim*** |  |
| **20:** |  |  |  | ***se-não:*** | |
| **21:** |  |  |  |  | *escritor-arquivo.****escreve-coluna****(titulo)* |
| **22:** |  |  |  | ***fim*** |  |
| **23:** |  |  |  | ***para******(inicializar*** *i← 0* ***até*** *tamanho****)******faça****:* | |
| **24:** |  |  |  |  | *escritor-arquivo.****escreve-coluna****(gerar-dados-sensiveis-aleatorios ())* |
| **25:** |  |  |  | ***fim*** |  |
| **26:** |  |  | ***fim*** |  |  |
| **27:** |  | ***fim*** |  |  |  |
| **28:** | ***fim*** |  |  |  |  |

O algoritmo acima utiliza como gerador o Faker API, passando o parâmetro de língua como ‘pt-br’ é possível gerar dados baseados no Brasil utilizando um total de 4 colunas, sendo a última dedicada ao tipo de dado (1 para sensível/privado e 0 para não sensível/privado), a variável de nome tamanho controla o total de linhas a serem geradas, no estudo foram feitas 1000 linhas de dados aleatoriamente alternados entre sensíveis/privados e não sensíveis/privados.

Após a criação dessa classe foi iniciado o desenvolvimento do primeiro modelo como um teste para entender como utilizar o Tensorflow e seu conjunto de ferramentas para desenvolver o modelo teste, sendo ele a primeira base para os outros modelos finais onde serão realizados os resultados e as comparações entre os dados.

O primeiro modelo foi desenvolvido em Jupyter-Notebook, seguindo o tutorial do Keras [inserir bibliografia] como uma base a ser utilizada para desenvolver o modelo teste, terminado de manusear o treinamento pelo Keras, começou o desenvolvimento do modelo teste, em ele foi gerado um CSV contendo apenas dois tipos de dados, um sensível/privado e um outro genérico. Após a geração os dados são processados para permitir a interpretação deles baseados no resultado do processamento, uma vez que o modelo por si não é capaz se interpretar letras, palavras ou até mesmos frases, utilizando do conjunto de ferramentas fornecidas pelo Tensorflow e o Keras, é possível compilar cada palavra e associar um valor a elas, permitindo serem interpretadas no modelo teste.

Outro aspecto importante para o desenvolvimento foi a utilização de um arquivo de funções criadas para mostrar o desempenho e resultado dos testes e dos treinos dos modelos, neste arquivo apresenta o Matplotlib importado sendo utilizado nas funções para gerar os gráficos necessários.

Os gráficos gerados foram utilizados para medir e avaliar o desempenho dos modelos conforme os testes eram feitos e código reavaliado, também, foram usados para verificar como o modelo final configurado de maneiras diferentes se comportaram com os dados de entrada, isto é, como as funções de ativações e de saída influenciaram no desempenho geral na identificação e classificação dos dados apresentados.

Com os preparativos feitos e o primeiro modelo criado, foi-se utilizado um método de pré-processamento de palavras transformando-as em valores passíveis de interpretação pelo modelo, uma vez que a IA não interpreta o mundo em associação visual e sim em associação de valores representativos de um objeto alvo. Esse processamento dos dados é utilizado no modelo final e suas variantes, mudando apenas a quantidade de valores de entradas presentes no processamento.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorítimo 2: Leitura e Preprocessamento. Elaborado pelo autor. | | | |
|  | ***Input:*** *Uma lista tirada de um database possuindo colunas e linhas* | | |
|  | ***Output:*** *Duas variáveis (classes X e Y) contendo os valores pré-processados.* | | |
| **1** | *X ←* ***read-csv****(string caminho-para-csv);* | | |
| **2** | *X ← X.****fillna****(media(X)* ***inplace*** *← True);* | | |
| **3** | *colunas ← lista(X);* | | |
| **4** | ***para*** *index* ***em*** *colunas* ***faça:*** | | |
| **5** |  | ***se*** *X[index].dtypes* ***==*** *‘object’* ***então:*** | |
| **6** |  |  | *X[index] ←* ***categorizar****(****fatorizar****(X[index])[0])* |
| **7** |  | **fim** |  |
| **8** | *Le ←* ***preprocessing****.****LabelEncoder****();* | | |
| **9** | ***para*** *i* ***em*** *colunas* ***faça****:* | | |
| **10** |  | ***Se*** *X[index].dtypes* ***==*** *‘object’* ***então:*** | |
| **11** |  |  | *X[index] ← Le.****fit-transform****(X[index])* |
| **12** |  | **fim** |  |
| **13** | **fim** |  |  |
| **14** | *Y ← X[“Classe”];* | | |
| **15** | *X ←****remover****([“Classe”,* ***axis*** *←1,* ***inplace*** *←True]);* | | |
| **16** | *X ←X.****values*** | | |
| **17** | *Y ←Y.****values****;* | | |
| **18** | ***imprimir****(X);* | | |
| **19** | ***imprimir*** *(“\n”);* | | |
| **20** | ***imprimir****(Y);* | | |

Após o pré-processamento, um modelo base, chamado de Alpha, é criado para verificar o desempenho geral do conceito utilizado no modelo, como também gerar suas derivações usando outras funções de ativações e de saídas. Cada layer do modelo utiliza uma rede conhecida como Dense Networksendo uma Neural Network densamente conectada, no caso deste modelo, cada conexão seria um tensor, as funções de ativação são ReLU (*Rectified Linear Unit*) e a de saída Sigmoide utilizada para converter processamento do modelo em outputs de probabilidade válidos para serem utilizados como avaliadores da database no contexto da LGPD. O modelo vai utilizar um otimizador Adam (este utiliza os primeiros e segundos momentos do gradiente para realizar a adaptação da curva de aprendizado para cada peso na Neural Network), passamos o parâmetro *Binary Crossentropy* utilizado para acompanhar as perdas métricas e penalizar o modelo no seu desempenho por cometer tais erros, como as penalidades baseadas em métricas, tais como True Positives (Positivos Verdadeiros), False Positives (Falsos Positivos), True Negatives (Negativos Verdadeiros), False Negatives (Falsos Negativos), por fim mais duas métricas adicionadas para análise no matplotlib, acurácia e precisão.

Tabela 1 – Apresentação dos componentes do Modelo Alpha em forma e tabela, cada linha representa um layer de Neural Network, divididas em conexões (tensors) por função de ativação.

|  |  |
| --- | --- |
| **Quantidade de Tensors** | **Tipo Ativação** |
| 12 | ReLU |
| 12 | ReLU |
| 8 | ReLU |
| 4 | ReLU |
| 1 | Sigmoid |

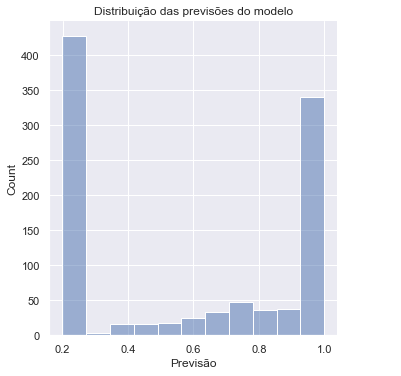
Fonte: Elaborado pelo autor

Construído o modelo, passamos para a fase de treinamento utilizando as classes X e Y em conjunto de alguns parâmetros de treinamento, o *EalyStopping* é utilizado para monitorar o modelo durante o processo e para o treinamento quando ele perceber a falta de ganho de desempenho utilizando o parâmetro de paciência (*patience*), neste caso de valor 10 e focando em restaurar os melhores pesos sempre que possível para evitar um treinamento mais longo. Feito isso é utilizado 100 épocas para treinar o modelo utilizando um *batch* de 50 em 50 amostras, nesse teste base do modelo Alpha foi obtido um resultado relativamente satisfatório de ~89,59% de acurácia e uma perca de ~28,59%

Depois de treinado e testado, é feito uma distribuição gráfica de probabilidade para verificar a quantidade de previsões feitas e a possibilidade de discrepâncias, assim verificando se o vetor resposta contendo as probabilidades está satisfazendo os requisitos para poder passar os resultados em conjunto dos dados, para o algoritmo de encriptação assim gerando um novo database com os dados privados encriptados e os irrelevantes permanecendo sem alteração. Nesta versão de modelo os resultados foram suficientemente para passar para o próximo estágio (modelos com funções de ativações diferentes), a encriptação dos dados baseados na probabilidade de ele estar correto.

Nas análises das probabilidades, não foram detectadas discrepâncias nas porcentagens das distribuições de probabilidade, entretanto existiu uma quantidade de Falsos Negativos e Falsos Positivos, dos quais serão discutidos na seção de limitação neste artigo.

Figura 3 - Probabilidades calculadas por quantidade de dados.



Fonte: Elaborado pelo autor

Tabela 2 – Aproximação da distribuição de previsões do modelo Alpha baseado no gráfico de barras acima.

|  |  |  |  |
| --- | --- | --- | --- |
| **Distribuição das previsões** | | | |
| **< 0.2** | **< 0.6** | **< 0.8** | **< 1** |
| ~450 | ~50 | ~50 | ~350 |

Fonte: Elaborado pelo autor

Observação: Existem probabilidades intermediárias entre o limite superior de 0.2 até o limite inferior de 0.8 não representadas na tabela por serem valores irrisórios devido ao fato de como o algoritmo de encriptação funciona, mas estão representados graficamente para entender a distribuição.

Ao analisar utilizando IRQ, Z-score e análises de limites e ), concluiu-se que os dados do resultado sem outliers ou dados discrepantes o suficiente para necessitar refazer toda a sessão de análise visando podar tais outliers de um jeito que não prejudique o desenvolvimento do projeto.

Feito isso, é feito uma leitura da base de dados sem pré-processamento para realizar o processo de encriptação pelo algoritmo, usando o array de predicts (previsões de probabilidades) para determinar se as colunas devem serem encriptadas ou não, foi assumido uma aproximação para binária para determinar a encriptação, se muito próximo de 1, é realizado a encriptação, do contrário a encriptação é rejeitada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorítimo 3: Encriptação dos dados | | | | |
|  | **Input:***Um array contendo os predicts(probabilidades) e o array de classes pura.s* | | | |
|  | **Output:***Um arquivo com as classes relevantes encriptadas.* | | | |
| **1** | **abrir-arquivo**(nome-arquivo, ‘w’,) **como** arquivo\_csv: | | | |
| **2** |  | escritor *←* escreve-coluna(array-titulo-das-colunas); | | |
| **3** |  | encriptador *←* algoritmo-criptografia(); | | |
| **4** |  | encriptador.calcularChaves(); | | |
| **5** |  | **para** index **até** 999 **faça**: | | |
| **6** |  |  | **se** (aproximação(previsão[index]) == classes[index]) **então**: | |
| **7** |  |  |  | R1 *←* encriptador.encriptar(X[titulo[1]][index]); |
| **8** |  |  |  | R2 *←* encriptador.encriptar(X[titulo[2]][index]); |
| **9** |  |  |  | R3 *←* encriptador.encriptar(X[titulo[3]][index]); |
| **10** |  |  |  | encriptado\_r1 *←* remover-espaço(R1); |
| **11** |  |  |  | encriptado\_r2 *←* remover-espaço(R2); |
| **12** |  |  |  | encriptado\_r3 *←* remover-espaço(R3); |
| **13** |  |  |  | escritor.**escreve-coluna**(encript\_r1, encript\_r2, encript\_r3) |
| **14** |  |  | **fim** | |
| **15** |  |  | **Se-não então:** | |
| **16** |  |  |  | escritor.**escreve-coluna**(X) |
| **17** |  |  | **fim** | |
| **18** |  | **fim** |  | |

Depois foi feito um loop para criar e analisar 10 modelos baseados no Alpha apresentado, com o foco de verificar a porcentagem de acerto e se é viável continuar com a estrutura dele para gerar os próximos mais distinguidos, a média feita com os desempenhos resultou em um total de 91.99% de acurácia em média, com isso o desenvolvimento dos modelos finais foi iniciado e testado.

Usando o modelo Alpha como base e utilizando diferentes tipos de funções de ativações como hard sigmoide, ReLU, sigmoide homogênea e tangente – Foi realizado o mesmos processos apresentados anteriormente no artigo, variando as funções com objetivo de verificar os desempenhos e qual dos modelos vai apresentar uma parte de identificação de dados, sendo as detecções True Positives (Positivos Verdadeiros), False Positives (Falsos Positivos), True Negatives (Negativos Verdadeiros), False Negatives (Falsos Negativos), para facilitar, abreviamos as categorizações acima respectivamente de TP, FP, TN e FN.

# APRESENTAÇÃO E DESEMPENHO DOS MODELOS.

O modelo final segue os mesmos preparos do modelo Alpha com a variação das funções de ativações para obter um desempenho de identificação de dados melhor, utilizando funções de ativações do tipo *hard sigmoid* com o filtro binário de ser ou não um dado privado para determinar o resultado do algoritmo de encriptação.

Foram desenvolvidos um total de quatro modelos com funções de ativações variadas, cada modelo acessa os mesmos dados de teste e de treino para depois passarem pelo processo de avaliação de encriptação, como também amostragem de desempenho através de gráficos.

A primeira variação desse tipo de modelo é o modelo utilizando um tensor de 12 conexões com uma função de ativação ReLU e para as demais conexões a utilização do Hard Sigmoide para calcular os melhores pesos e realizar a saída das previsões, abaixo uma tabela descritiva do modelo.

Tabela 3 – Apresentação dos componentes do modelo 1 em forma e tabela.

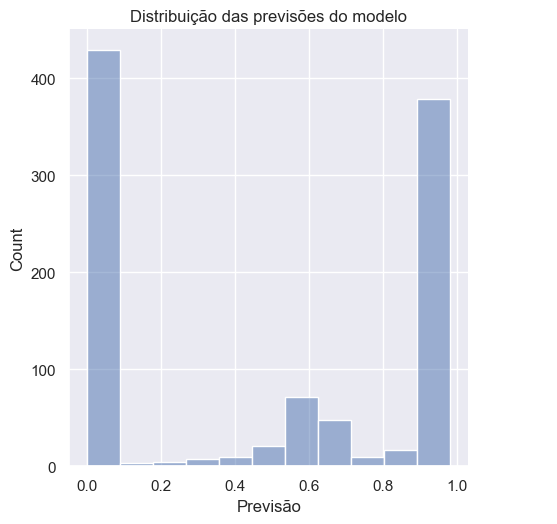
|  |  |
| --- | --- |
| **Quantidade de Tensors** | **Tipo Ativação** |
| 12 | ReLU |
| 8 | Hard Sigmoid |
| 8 | Hard Sigmoid |
| 8 | Hard Sigmoid |
| 1 | Hard Sigmoid |

Fonte: Elaborado pelo autor

As funções de ativações deste modelo trabalham utilizando uma reta linear para diferenciar os dados de entrada, transportando-os para a camada de processamento do qual utiliza uma variante da função de Heaviside, podendo determinar em seu processamento saídas entre 0 e 1 após o processamento.

Após esse processamento/treinamento, o modelo é capaz de realizar previsões com cada saída em forma de porcentagem, indicando as chances daquela previsão está de acordo com o resultado real de cada dado, no caso, 0 ou 1 se não pertenceu ou pertence a categoria de dados sensíveis/privados respectivamente. O gráfico abaixo é uma amostragem da distribuição de previsões no modelo 1 e uma tabela mostrando as detecções e suas categorias.

Figura 4 – Quantidade de dados por valor de precisão do modelo 1



Fonte: Elaborado pelo autor

Tabela 4 – Comparação entre as detecções dos modelos apresentados no modelo 1, utilizando os parâmetros TP, FP, TN e FN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo 1: Hard Sigmoid.** | | | |
| **TP** | **FP** | **TN** | **FN** |
| 484 | 66 | 431 | 16 |

Fonte: Elaborado pelo autor

O modelo 2 utiliza funções de ativações baseadas na *Rectified Linear Activation Unit*, permitindo a ativação dos neurônios em tempo de processamento maior que outras quando os inputs não resultam em 0, diferentes das outras quais podem ter problemas para começarem a processar os dados na camada escondida , com isso os resultados desta versão apresentou resultados onde não existiam Falsos Negativos e apenas o parâmetros de Falsos Positivos acabaram sendo gerados, este podendo inferir no resultado de encriptação.

Mais adiante neste artigo pode-se verificar a distribuição das previsões em forma gráfica como também a tabela de dados e como foram categorizados pelo modelo, existe a possibilidade de as funções de ativação terem sido vítimas da camada de saída.

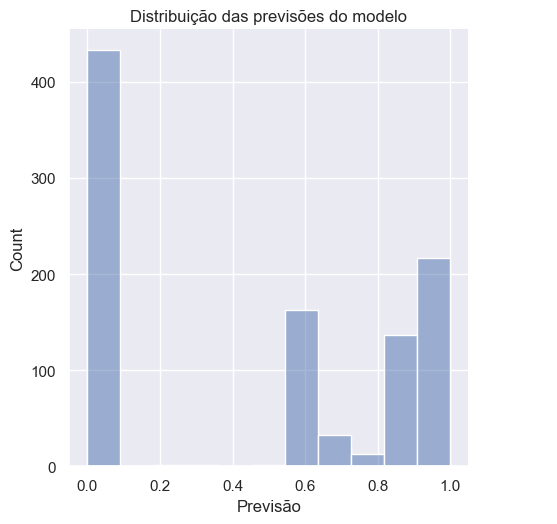
Sendo um Hard Sigmoide, essa camada pode não pegar todos os pontos de avaliação por ser uma função de ativação considerada não-suave, alguns dados podem terem sido evitados pela presença da avaliação binária.

Tabela 5 – Apresentação dos componentes do Modelo 2 em forma e tabela.

|  |  |
| --- | --- |
| **Quantidade de Tensors** | **Tipo Ativação** |
| 12 | ReLU |
| 12 | ReLU |
| 8 | ReLU |
| 4 | ReLU |
| 1 | Hard Sigmoid |

Fonte: Elaborado pelo autor

Figura 5– Quantidade de dados por valor de precisão do modelo 2.



Fonte: Elaborado pelo autor

Ainda assim o modelo teve um resultado grandioso quando comparado ao seu “irmão” anterior, principalmente olhando para os resultados gráficos nas previsões acima e nas detecções na tabela abaixo.

Tabela 6 – Comparação entre as detecções dos modelos apresentados na versão final, utilizando os parâmetros TP, FP, TN e FN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo 2: Hard Rectified Linear Unit (ReLU).** | | | |
| **TP** | **FP** | **TN** | **FN** |
| 521 | 73 | 402 | 0 |

Fonte: Elaborado pelo autor

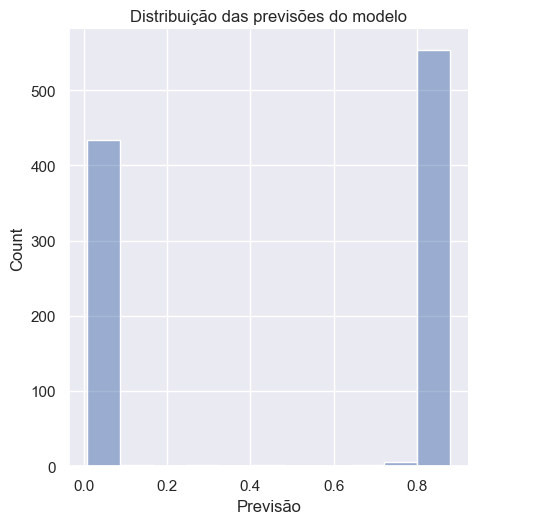
O modelo 3 é um modelo mais homogêneo usando apenas funções Sigmoide, sendo a mais famosa e permitindo uma flexibilidade maior, podendo ser utilizadas nas camadas de entrada, escondidas e nas camadas de saídas para parte de previsão.

Tabela 7 – Apresentação dos componentes do Modelo 3 em forma e tabela.

|  |  |
| --- | --- |
| **Quantidade de Tensors** | **Tipo Ativação** |
| 12 | Sigmoid |
| 8 | Sigmoid |
| 8 | Sigmoid |
| 8 | Sigmoid |
| 1 | Sigmoid |

Fonte: Elaborado pelo autor

Figura 6– Quantidade de dados por valor de precisão do modelo 3



Fonte: Elaborado pelo autor

Tabela 8 – Comparação entre as detecções dos modelos apresentados na versão final, utilizando os parâmetros TP, FP, TN e FN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo 3: Pure Sigmoid** | | | |
| **TP** | **FP** | **TN** | **FN** |
| 517 | 4 | 399 | 75 |

Fonte: Elaborado pelo autor

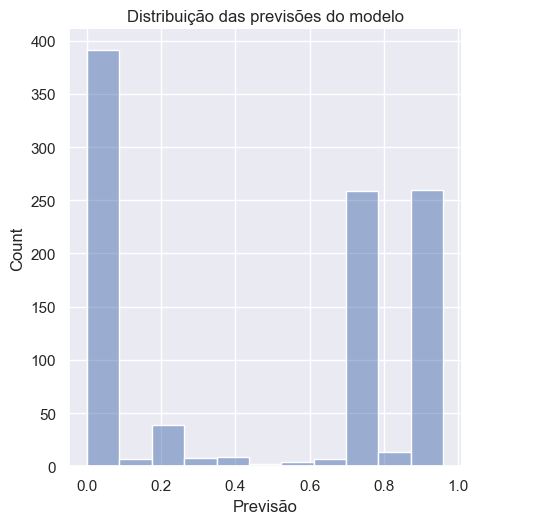
Olhando apenas para a distribuição de detecções, este modelo obteve a melhor distribuição entre os modelos de forma geral, se aproximando do resultado desejado para o algoritmo de encriptação realizar a sua função, porem existe casos de detecções fora da curva que foram categorizados errados. O modelo 4 é o mais heterogêneo de todos já apresentados, contendo três funções de ativação, ReLU para entradas, tangente para a camada escondida e uma *Hard Sigmoid* para a saída, essas escolhas foram feitas para entradas e saídas devido a como a Tangente se comporta na camada escondida, produzindo resultados entre [-1, 1], a ReLU vai pegar as entradas e trabalhar com elas num intervalo de [0, 1] e a saída usando *Hard Sigmoid* volta em a mostrar os resultados em [-1, 1], abaixo a tabela contendo a configuração do modelo.

Tabela 9 – Apresentação dos componentes do Modelo 4 em forma e tabela.

|  |  |
| --- | --- |
| **Quantidade de Tensors** | **Tipo Ativação** |
| 12 | ReLU |
| 8 | Tangente |
| 8 | Tangente |
| 4 | Tangente |
| 1 | Hard Sigmoid |

Fonte: Elaborado pelo autor

Figura 7– Quantidade de dados por valor de precisão do modelo 4



Fonte: Elaborado pelo autor

Tabela 10 – Comparação entre as detecções dos modelos apresentados na versão final, utilizando os parâmetros TP, FP, TN e FN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo 4: Tangente.** | | | |
| **TP** | **FP** | **TN** | **FN** |
| 521 | 83 | 393 | 0 |

Fonte: Elaborado pelo autor

Observando a tabela de detecções, mesmo alcançando o objetivo de não obter detecções falsas nos negativos, o modelo teve uma queda quando comparado ao modelo 2, onde obteve o menor resultado entre os dois quando olhando na coluna de Falsos Positivos podemos ver uma quantidade maior nas detecções do modelo 2.

Em todas as encriptações feitas pelos modelos utilizando o RSA customizado, as encriptações foram de acordo com as tabelas de detecções de cada modelo, incluindo os Falsos Positivos e Falsos Negativos, esse desenvolvimento será discutido na seção de limitações a seguir.

# LIMITAÇÕES E DISCUSSÃO

No desenvolvimento do projeto não existiu muitas oportunidades de trabalhar com dados reais e tangíveis, as maiorias das oportunidades para manipula-los necessitaria de recurso monetário indisponível para o desenvolvimento desse projeto, então foi-se tomado a iniciativa de utilizar a API chamada de Faker para desenvolver dados privados/sensíveis dos quais possam ser tangíveis o suficiente para desenvolver os modelos e treina-los em método de treinamento supervisionado para realizar o desejado e discutido no artigo, identificar o dado, encripta-lo se for sensível, deixa-lo visível se for irrelevante perante a lei.

Existe a possibilidade de os dados gerados apresentarem um bias para os modelos fazendo com que eles não consigam diferenciar adequadamente os tipos dos dados gerados, isso sendo um resultado da utilização da Faker API, como o método de pré-processamento (convertendo os valores baseados numa tabela dentro do Keras).

Outra das limitações desse projeto foi trabalhar com a API antiga do Tensorflow de processamento de texto, a nova API lançada no ano de 2023 apresenta uma capacidade de maior de processamento, interpretação e avaliação de texto – Dentro do contexto desse artigo, se a versão tivesse disponibilizada no tempo de desenvolvimento do projeto, os resultados dos modelos poderiam ser mais satisfatórios e apresentar variações mais interessantes.

O que pode ser dito também a respeito da capacidade de detecção de cada modelo quando observando as tabelas na seção de metodologia, é a capacidade de detecção, no caso pode ter ocorrido uma falta maior de coesão dos parâmetros das classes e no modelo para melhorar e possivelmente zerar essas detecções de Falsos Positivos e Falsos Negativos, olhando para o lado do tempo para desenvolver todos os modelos (em torno de 6 meses), os resultados apresentados foram interessantes e satisfatórios, apesar dos pontos apresentados.

Em desenvolvimento dessa área e desses tipos de modelos, é possível detectar a presença de picos de perdas/loss sendo onde o modelo pode ter dificuldade de validar os dados e acabam resultando em penalidades, perdas de melhores pesos e até avaliações erradas dos dados durante todo o processo.

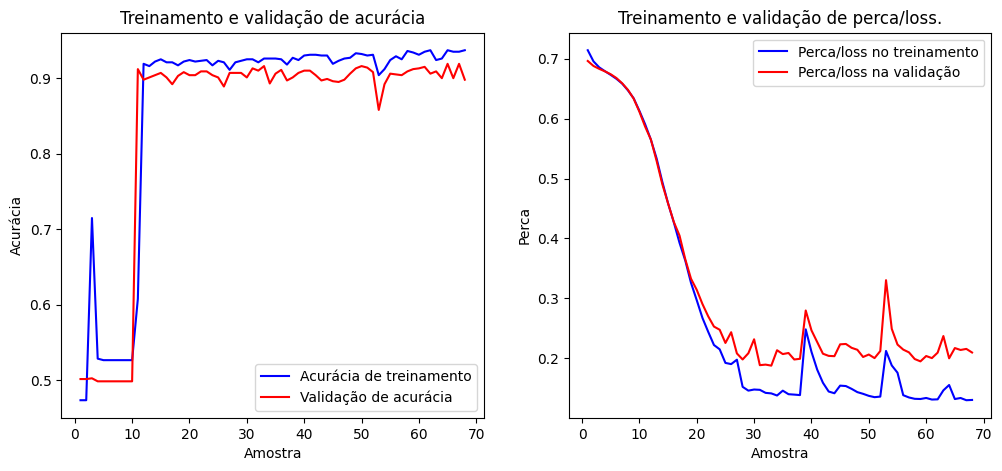
Abaixo os gráficos comparativos entre cada modelo respectivamente:

Figura 8 - Desempenho gráfico do modelo 1

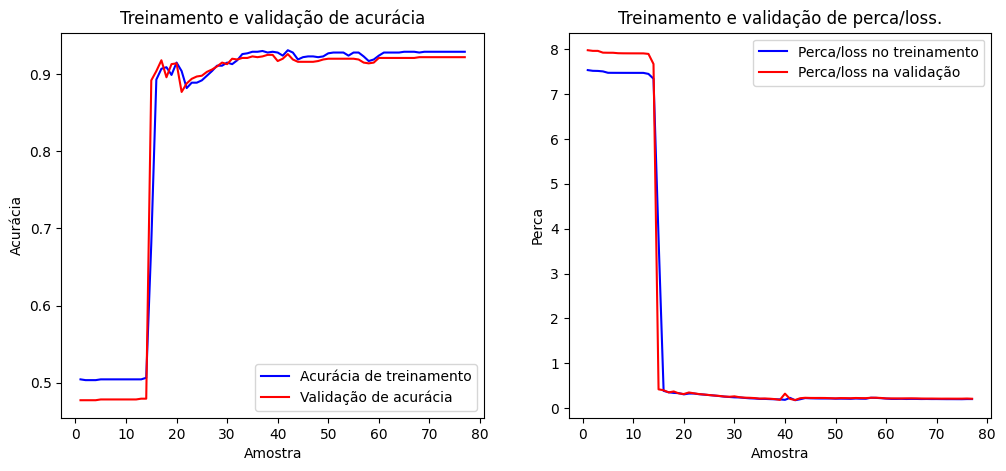


Figura 9 - Desempenho gráfico do modelo 2

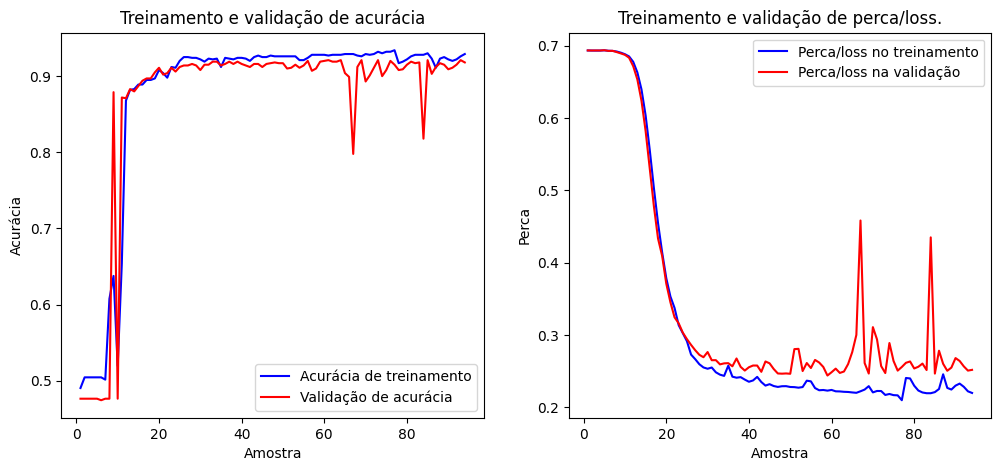


Figura 10 - Desempenho gráfico do modelo 3

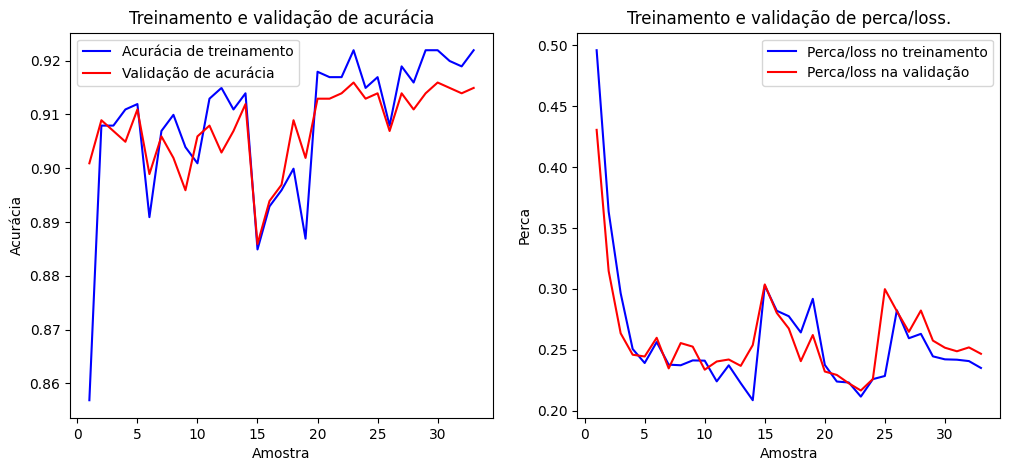


Figura 11 - Desempenho do modelo 4

Os modelos 1, 3 e 4 foram vítimas dessas percas, principalmente na parte de avaliação, que é a parte mais relevante para a verificação das capacidades de cada um.

Em contrapartida o modelo 2 não obteve tais variações durante todo o seu processo de treinamento e validação, mas por não permitir análise de valores abaixo de zero durante o processamento na camada escondida, ocorreu detecções de FP e FN respectivamente, mesmo obtendo o melhor desempenho de distribuição de previsões e pouca perca durante sua execução.

Mesmo com o desempenho basicamente intocável, o modelo 2 ainda teve problemas na detecção, voltando ao ponto de que os parâmetros e as classes apresentem problemas, podendo necessitar de mais configurações nas classes ou mais parâmetros mais complexos no treinamento.

Outro problema para o desenvolvimento do projeto foi a limitação física em hardware do computador utilizado para o desenvolvimento do projeto, já que muito dos modelos desenvolvidos disponíveis utilizam milhares e até milhões de parâmetros em conjunto de diversos processadores CUDA (*Compute Unified Device Architecture*) para mastigar e compreender os dados inseridos nos modelos.

# APLICAÇÕES

Uma aplicação para esse modelo seria como um intermediador no database para encriptar os dados quando a detecção deles forem feitas, para evitar que eles possam ser vazados indevidamente. Considere uma situação onde invasores da rede de um sistema obterão acesso interno ao sistema, um administrador vigiando a rede ou até mesmo um sistema integrado de segurança poderia ativar o modelo para encriptar os dados sensíveis em uma chamada ao invés de fornecê-los desvalidos para os invasores, uma vez que dados encriptados não são fáceis de quebrarem e dependendo do nível de encriptação, até mesmo impossíveis dentro de um determinado tempo, ou seja, em tempo de causar um dano real aos donos do sistema, já que estes tem tempo para ativar precauções, planos de contingência e avisar usuários do sistema sobre o comprometimento dos dados.

Outra possível aplicação é a utilização do modelo para a limitar acesso de dados para usuários de dashboards de dados tais como a Microsoft PowerBI, assim limitando o acesso das informações privadas por nível de usuário e tipo de requisição de dados, garantindo que apenas os operadores corretos de cada dado podem ter acesso aos dados.

# CONTINUAÇÃO E POSSÍVEIS MELHORIAS

Caso um possível desenvolvedor se interesse em contribuir para essa ideia apresentada neste artigo, recomenda-se procurar aprimoramentos dos parâmetros de análise, de entrada em cada layer de entrada e utilização, se possível, de ambientes de processamento conectados na nuvem, pois este projeto foi realizado localmente utilizando um computador relativamente capacitado para desenvolvimento e teste dos modelos, esta limitação não ocorre em ambientes computacionais na nuvem, pois estes funcionam baseados no tempo de uso de cada modelo e o tempo de processamento dos dados.

Outra melhoria é a utilização das novas ferramentas, API’s e frameworks desse projeto, nos seis meses de desenvolvimento até a escrita do artigo, recentemente houve uma melhoria nas capacidades de processamento de dados textuais e de valores pelo Tensorflow, devido a apresentação de competidores tais como o ChatGPT.

Mencionando ChatGPT, poderia ser feito uma API’s intermediária entre a IA e os dados, já que o ChatGPT possui uma quantidade de parâmetros aproximado em milhões, tendo uma capacidade maior de entender e processar os tipos de dados pessoais, um adendo e aviso a este método de desenvolvimento, o ChatGPT não possui uma categoria de privacidade nos seus pesos, ou seja, pode existir a chance dos dados sensíveis se manterem dentro do modelo até após sua execução e desligamento, resultando em um evento do qual um outro usuário qualquer pode fazer uma requisição aleatória e o ChatGPT acabar entregando os dados sensíveis como foram os casos de programadores pedindo ajudas com códigos e a IA entregando códigos protegidos por *Defense Contract Management Agency* (DCMA), não só desrespeitando a lei, como entregando códigos protegidos por Propriedade Intelectual (IP), caso o responsável pelo ChatGPT resolvam esse problema, a utilização dessa IA é interessante.

# CONCLUSÕES

Neste artigo foi apresentado como a internet integrada a serviços para usuários se tornou mais comum e cada vez mais presente na vida de maneira geral e as consequências desses sistemas não utilizarem mecanismos (seja de direito ou da parte da computação) até a criação da LGPD, Machine Learning e como a área de Ciências da Computação junto com Direito se intersectam para garantir a integridade dos dados sensíveis de usuários em diversos sistemas integrados na internet.

Dito isso foi desenvolvido utilizando Python junto de Jupyter Notebook, para criar modelos de máquinas, algoritmos de encriptação e usar de diversos frameworks, bibliotecas e API’s para o desenvolvimento desses modelos, visando utilizar o melhor das ciências da computação para integrar com essa área de direito que surgiu e se torna mais presentes nas nossas vidas pessoais e profissionais.

O Faker API foi importante no desenvolvimento desse projeto não só pela geração de dados utilizados para o desenvolvimento dos modelos, como na influência individual em cada modelo para a detecção dos dados.

Os modelos apresentados tiveram a capacidade de computar e compreender os dados para gerar resultados encriptados de dados pessoais e deixar dados irrelevantes em abertos, pois estes não são categorizados pela lei como dados sensíveis, por não haver uma ligação pessoal entre a informação presente na database e a uma pessoa, onde o dado diz algo sobre a vida íntima da pessoa em questão. Com isso pode se afirmar que existe sim um espaço do qual a área de IA pode preencher para ajudar profissionais de direito e profissionais de computação a realizarem seus trabalhos com facilidade e automação, como também garantir a integridade da lei, assim como expandir as áreas de oportunidades no mercado de trabalho pela existência dessa nova intersecção entre as áreas apresentadas.

Os modelos construídos, testados e estudados nesse artigo são exemplos de como a área de IA pode ajudar nessa aplicação da lei, como facilitar o trabalho judicial na detecção de dados não respeitando a integridade da lei, como a aplicação de incisos da lei para garantir a proteção dos dados e protegê-los com encriptação.

# REFERÊNCIA BIBLIOGRÁFICA

**[1] Lei 'Carolina Dieckmann', que pune invasão de PCs, entra em vigor.** G1 Globo, São Paulo, dia 1, junho de 2013. Disponível em: <https://g1.globo.com/tecnologia/noticia/2013/04/lei-carolina-dieckmann-que-pune-invasao-de-pcs-passa-valer-amanha.html> Acesso em: 27, agosto de 2022

**[2]** Leal da Silva, Julia; **Tomada de Decisão Automatizada e Controle pela LGPD.** IAPD, 20, janeiro de 2021. Disponível em: <https://iapd.org.br/decisao-automatizada-lgpd-direito-aexplicacao/> Acesso em: 24, agosto de 2022.

**[3]** Fortunato, Caroline, **Using QRadar for LGDP**, 19, julho de 2019, Disponível em:<https://www.proof.com.br/wp-content/uploads/2019/08/Using-QRadar-for-LGPD.pdf>

**[4]** Goldsteen, Abigail; Ezov Gilad; Shmelkin, Ron; Moffie, Micha; Farkash, Ariel; **Data minimization for GDPR compliance in machine learning models**, p. 1-15, 2021, Disponível em: <<https://link.springer.com/article/10.1007/s43681-021-00095-8>>

**[5]** Roberto Fernandes Castilho, José; Andrade Gomes, Henrique; **Legislação Básica de Direito da Informática**, 2° edição reformulada e atualizada, São Paulo, Editora Pillares.

**[6] General Data Protection Regulation GDPR:** OJ L 119, 04.05.2016; cor. OJ L 127, 23.5.2018. Disponível em: <<https://gdpr-info.eu/>> Acesso em: 27, agosto de 2022

**[7]** Stallings, William; Brown, Lawrie; **Computer Security: Principles and Practice Second Edition**, 2° edição reformulada e atualizada, São Paulo, Pearson; 2ª edição.

**[8]** Mota, José; **Da Web 2.0 ao E-Learning 2.0: Aprender na rede,** p.1-199, 2009. Disponível em: <https://repositorioaberto.uab.pt/handle/10400.2/1381>

**[9]** Filipe Lima Rapôso, Cláudio; Melo de Lima, Haniel; Ferreira de Oliveira Junior, Waldecy; Aragão Ferreira Silva, Paola; Elaine de Souza Barros, Elaine; **LGPD - LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS EM TECNOLOGIA DA INFORMAÇÃO: Revisão Sistemática**, p. 1-10. 2019. Disponível em:<https://revistas.cesmac.edu.br/index.php/administracao/article/view/1035>

**[10]** Spadaccini de Teffé, Chiara; Viola, Mario; **Tratamento de dados pessoais na LGPD: estudo sobre as bases legais**, p. 1-38, 2020. Disponível em: <https://civilistica.emnuvens.com.br/redc/article/view/510>

**[11]** Tankard, Colin; **What the GDPR means for business**, p. 1-8, 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1353485816300563?casa\_token=pjOHlq5iYsAAAAA:dhlGukGxSanwjnBY9aPtw36O1CslOJG1wZ7wTEryTJHi5QtFVM2G6kc8CwuemdVrUDRl2tRqJoA>

**[12]** Goldsteen, Abigail; Ezov Gilad; Shmelkin, Ron; Moffie, Micha; Farkash, Ariel; **Data minimization for GDPR compliance in machine learning models**, p. 1-15, 2021. Disponível em: <https://link.springer.com/article/10.1007/s43681-021-00095-8>

**[13]** Shanmugam, Divya; Shabanian, Samira; Diaz, Fernando; Finck, Michèle, Biega, Asia; **Learning to Limit Data Collectionvia Scaling laws: A computional Interpolation for the Legal Principle of Data Minimization**, p. 1-11, 2022. Disponível em: < https://arxiv.org/abs/2107.08096 >

**[14]** D. R. Ignatius Moses Setiadi, A. Faishal Najib, E. H. Rachmawanto, C. Atika Sari, K. Sarker and N. Rijati, "**A Comparative Study MD5 and SHA1 Algorithms to Encrypt REST API Authentication on Mobile-based Application**," 2019 International Conference on Information and Communications Technology (ICOIACT), 2019, pp. 206-211, doi: 10.1109/ICOIACT46704.2019.8938570. Disponível em: <https://ieeexplore.ieee.org/document/8938570>

**[15]** Boutaba, Raouf; Mohammad A., Salahuddin; Limam, Noura; Ayoubi, Sara; Shahriar, Nashid; Estrada-Solano Felipe; Caicedo M. Oscar; **A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities**, p. 1-99, 2018. Disponível em: <https://jisajournal.springeropen.com/articles/10.1186/s13174-018-0087-2#Sec2>

**[16]** Fatih, Ertam; Galip, Aydin; **Data classification with deep learning using TensorFlow**, p. 1-4, 2017. Disponível em: <https://ieeexplore.ieee.org/document/8093521>

**[17]** Nguyen Quang-Hung; Hieu Doan; Nam Thoai; **Performance Evaluation of Distributed Training in TensorFlow 2**, p. 1-5, 2020. Disponível em: < https://ieeexplore.ieee.org/document/9353085 >

**[18]** PYTHON SOFTWARE FOUNDATION. **Python Language** Site: Documentation, 2022. Disponível em: <https://www.python.org/doc/> Acesso em: 24 de agosto de 2022.

**[19]** PANDAS **Pandas DataFrame** Site: About, 2022. Página sobre nós. Disponível em: <https://pandas.pydata.org/about/> Acesso em: 24 de agosto de 2022.

**[20]** PANDAS **Pandas DataFrame** Site: Documentation, 2022. Disponível em: <https://pandas.pydata.org/about/> Acesso em: 24 de agosto de 2022.

**[21]** PANDAS **10 minutes to pandas** Site: User Guide, 2022. Disponível em: <https://pandas.pydata.org/docs/user\_guide/10min.html#viewing-dataAcesso em: 24 de agosto de 2022.

**[22]** NUMPY PROJECT AND COMMUNITY. **What is Numpy?** Site: Documentation, 2022. Página de documentação. Disponível em:

<https://numpy.org/doc/stable/user/whatisnumpy.html> Acesso em: 24 de agosto de 2022.

**[23]** J. D. Hunter, "**Matplotlib: A 2D Graphics Environment**," in Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55. Disponível em:<https://ieeexplore.ieee.org/document/4160265>

**[24]**MDN Web Docs, **Mozilla Developer Network Web Docs**, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/>

**[25]** CLOUD GOOGLE. **Considerações sobre dados confidenciais em conjuntos de dados de aprendizado de máquina.** Site: Documentação, 2022. Disponível em: <<https://cloud.google.com/architecture/sensitive-data-and-ml-datasets>> Acesso em: 27, agosto de 2022.

**[26]** D Rachmawati, J T Tarigan and A B C Ginting; **A comparative study of Message Digest 5(MD5) and SHA256 algorithm**, p. 1-7, 2018. Disponível em <<https://iopscience.iop.org/article/10.1088/1742-6596/978/1/012116/meta>>

**[27]** Song, Congzheng; Ristenpart, Thomas; Shmatikov, Vitaly; **Machine Learning Models that Remember Too Much**, p. 1-15, 22, setembro de 2017. Disponível em: <<https://arxiv.org/pdf/1709.07886.pdf>>

**[28]** Vieira Souza, Iuri; **Aplicações de software desenvolvidas no contexto da Inteligência Artificial (IA), Machine Learning e Big  
Data e o direito dos cidadãos de acordo com a Lei  
Geral de Proteção de Dados (LGPD)**, p. 1-83, 2021. Disponível em: < https://bdm.unb.br/bitstream/10483/30275/1/2021\_IuriSousaVieira\_tcc.pdf>

**[29]** Goldsteen, Abigail; Ezov Gilad; Shmelkin, Ron; Moffie, Micha; Farkash, Ariel; **Data minimization for GDPR compliance in machine learning models**, p. 1-15, 2021, Disponível em: <<https://link.springer.com/article/10.1007/s43681-021-00095-8>>

**[30]** . Pavani and P. Sriramya, "**Enhancing Public Key Cryptography using RSA, RSA-CRT and N-Prime RSA with Multiple Keys**" 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 1-6, doi: 10.1109/ICICV50876.2021.9388621. Disponível em: <https://ieeexplore.ieee.org/document/9388621>

**[31]** NVIDIA, “O QUE É COMPUTAÇÃO ACELERADA POR DE PLACAS DE VÍDEO?” Disponível em: <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/

Acessado em Novembro/2017> Acesso em: 14/06/2023

**[32]** Barney, B. (2010). **“Introduction to parallel computing.” Lawrence Livermore National**,Laboratory, 6(13):10

**[33]** SANDERS, J., KANDROT, E., **“Cuda by example - An Introduction to General-Purpose GPU Programming”**.

**[34]** Defense Contract Management Agency, US Government, Site: Governamental Disponível em: < https://www.dcma.mil/ > Acesso em: 16/06/2023

**[35]** McCallum**,** Shiona; **ChatGPT Banned in Italy over privacy** **concerns**, Site: Notícia, abril de 2023, Disponível em: <https://www.bbc.com/news/technology-65139406> Acesso em: 16/06/2023

***Palavras Chaves (Keywords) –* Lei Geral de Proteção de dados (LGPD); Proteção de dados; Métodos de aplicação da LGPD, Manipulação e minimização de dados; Machine Learning; Inteligência Artificial; Methods of applying GDPR; Data protection.**