

Comparando Semelhanças em Impressões Digitais usando Inteligência Artificial

Marco C. Chaul¹, Alessandro R. Silva¹, Sandrerley R. Pires²

¹Instituto Federal de Goiás

²Universidade Federal de Goiás

chaul@neokoros.com, alessandro.rodrigues@ifg.edu.br, sandrerley@ufg.br

Abstract

Fingerprint recognition algorithms are complex and interdependent processes with multiple stages. The efficiency of each stage affects the quality of the subsequent ones, necessitating optimization at every step. Recently, Artificial Intelligence (AI) has played a pivotal role in improving these processes. This paper specifically evaluates AI's impact on the final stage, which involves determining the similarity between input fingerprints and those stored in a database, ultimately identifying the source finger. The paper discusses the use of intelligent models and presents the achieved results. In conclusion, it underscores the vital role of AI in advancing and refining traditional fingerprint analysis algorithms.

KEYWORDS: Fingerprints, Fingerprint Recognition, Fingerprint Recognition Algorithms.

Resumo

Algoritmos de reconhecimento de impressões digitais são complexos e envolvem várias etapas interdependentes. Recentemente, a Inteligência Artificial (IA) tem se destacado como uma ferramenta poderosa para melhorar esses processos. Este artigo se concentra em avaliar como a IA pode aprimorar a etapa final: a identificação de semelhanças entre impressões digitais de entrada e as armazenadas no banco de dados, determinando se são da mesma origem. Detalhes sobre os modelos de IA usados e os resultados obtidos são discutidos ao longo do artigo, juntamente com a importância e eficácia dos recursos de IA no avanço e aprimoramento dos algoritmos tradicionais utilizados na análise de impressões digitais.

PALAVRAS-CHAVES: Impressões Digitais, Reconhecimento de Impressões Digitais, Algoritmos de Reconhecimento de Impressões Digitais.

1. Introdução

O processo de reconhecimento de impressão digital envolve uma série de etapas fundamentais. Estas incluem o isolamento e realce da imagem da impressão digital, a aplicação de filtros de identificação de direção, a linearização da imagem, a detecção de minúcias e a obtenção de um conjunto de relações espaciais entre as minúcias, culminando na criação de uma “assinatura” única para cada impressão digital [Maltoni et al., 2009]. Cada uma dessas etapas é vital para a eficácia geral do sistema, exigindo que cada atividade seja meticulosamente analisada e refinada para maximizar a precisão e a eficiência do sistema.

Este trabalho se concentra especificamente na fase final do pipeline de reconhecimento: a comparação da “assinatura” da impressão digital em questão com as “assinaturas” das impressões digitais armazenadas no banco de dados. Essa tarefa pode ser realizada utilizando diversas abordagens, seja por modelos heurísticos, determinísticos ou por modelos estatísticos. De acordo com [Xu et al., 2019] a melhoria dessa etapa pode levar a um aumento significativo na qualidade do algoritmo de reconhecimento, melhorando sua competitividade em relação a outros algoritmos disponíveis no mercado. Recentemente, a incorporação de técnicas de inteligência artificial tem demonstrado um impacto significativo no aumento da eficiência desses algoritmos, abrindo novos caminhos para avanços

futuros[Jain et al., 1997]. Neste estudo, utilizamos de modelos de regressão, além de uma rede neural profunda clássica totalmente conectada [Goodfellow, 2016], como métodos de comparação de assinaturas.

Este trabalho busca, portanto, desenvolver e avaliar a aplicabilidade de modelos inteligentes no apoio ao pipeline clássico de reconhecimento de impressões digitais. O objetivo é avaliar o impacto desses modelos no processo de reconhecimento e contribuir para o avanço do campo com um aumento esperado na qualidade do processo de reconhecimento.

Neste artigo, apresentamos um novo método de comparação de assinaturas baseado em um conjunto de recursos específicos. O nosso método é baseado na ideia de representar cada assinatura como um vetor de recursos. Os recursos são características das assinaturas que são relevantes para a comparação.

Este artigo é dividido em cinco seções, começando com esta introdução. A seção dois descreve as técnicas utilizadas neste estudo, enquanto a seção três detalha a abordagem proposta. A seção quatro apresenta e discute os resultados obtidos e a seção cinco oferece as conclusões deste trabalho.

2. Etapas do Processo de Reconhecimento de Impressão Digital

A extração de impressões digitais é um processo utilizado para capturar e identificar características únicas, chamadas minúcias, presentes nas pontas dos dedos humanos. Este processo é sumarizado a seguir:

Captura da imagem (passo 1): A captura inicial da impressão digital é realizada usando um dispositivo de digitalização, como um scanner de impressão digital ou um sensor embutido em um dispositivo eletrônico. A imagem capturada é uma representação visual das cristas e vales presentes na superfície da ponta do dedo.

Antes de extrair as minúcias, a imagem da impressão digital passa por uma série de etapas de pré-processamento (passo 2) para melhorar a qualidade e remover ruídos. Isso pode envolver a aplicação de filtros para melhorar o contraste, reduzir manchas e remover distorções. Depois, a imagem pré-processada é convertida em uma imagem binária (passo 3), onde os pixels são classificados como branco ou preto. Isso é feito usando um algoritmo de limiarização, onde um valor de limite é definido para distinguir entre as partes claras e escuras da imagem.

A imagem binarizada é segmentada (passo 4), sendo dividida em regiões menores, isolando as áreas de interesse que contêm as impressões digitais. Isso pode ser feito detectando os contornos dos objetos presentes na imagem ou usando algoritmos de segmentação que analisam a conectividade dos pixels.

A detecção das cristas e sulcos (passo 5) é realizada por meio de algoritmos de processamento de imagens que analisam a topologia das regiões segmentadas. Esses algoritmos buscam características específicas para identificar as cristas e sulcos, como mudanças abruptas na direção do fluxo das linhas ou padrões repetitivos.

Uma vez que as cristas e sulcos são identificados, as informações sobre sua orientação, forma e comprimento podem ser utilizadas para extrair as minúcias (passo 6). As minúcias são pontos de referência específicos ao longo das cristas, como bifurcações

(pontos onde uma crista se divide em duas) e terminações (pontos onde uma crista termina).

A detecção de minúcias ocorre por meio de algoritmos que analisam as mudanças na orientação das cristas e a proximidade entre elas. Esses algoritmos podem utilizar técnicas como a filtragem de orientação, como o de Gabor [Gabor, 1946], que calcula a orientação predominante das cristas em cada região, e a análise de vizinhança, que verifica a presença de pontos de interesse, como bifurcações e terminações.

Representação e armazenamento: As minúcias extraídas são representadas por meio de vetores ou descritores que capturam suas características essenciais, como posição, orientação e tipo. Essas informações são usadas para criar uma representação compacta e única da impressão digital, geralmente chamada de modelo de impressão digital. Esse modelo é então armazenado em um banco de dados para fins de comparação e identificação futura [Maltoni et al., 2009, Jain et al., 1997].

A etapa seguinte é a comparação de duas representações que foram extraídas previamente (passo 7).

3. Metodologia

Este trabalho utiliza de modelos de regressão e uma Rede Neural Profunda Totalmente Conectada (DNN). A proposta consiste em aplicar técnicas de modelagem para superar as limitações dos métodos existentes e alcançar resultados mais precisos e confiáveis. A combinação dos modelos de regressão e DNN visa explorar as capacidades de aprendizado e generalização dessas técnicas, proporcionando uma análise mais refinada e robusta das amostras de impressões digitais. Ao comparar os resultados desses modelos com as soluções atuais, espera-se evidenciar os benefícios e aprimoramentos alcançados por meio da aplicação de técnicas de inteligência artificial.

3.1. Dataset Utilizado

A escolha de um dataset adequado é fundamental para o desenvolvimento e treinamento de algoritmos de inteligência artificial. Neste estudo, utilizamos como base de dados principal os conjuntos DB1, DB2, DB3 e DB4 do FVC2006, que consistem em 6720 imagens de impressões digitais. Esse dataset é amplamente utilizado na literatura ([Zhang et al., 2015], [Chen et al., 2015]). Essas imagens foram coletadas de diferentes fontes, sendo o DB1 proveniente de sensores de campo elétrico, o DB2 de sensores óticos, o DB3 de sensores térmicos, e o DB4 composto por imagens geradas artificialmente pelo software SFinGe v3.0. Essas bases de dados do FVC são amplamente reconhecidas e utilizadas na comunidade científica devido à sua qualidade e validação rigorosa. Foram adicionadas imagens autorais, contendo milhares de amostras adicionais, para aprimorar o desempenho e a generalização do modelo proposto. Essa diversidade e quantidade de dados proporcionam um ambiente rico para o treinamento e avaliação dos algoritmos de reconhecimento de impressões digitais.

3.2. Extração de características

O dataset FVC contém múltiplas amostras de uma mesma impressão digital, capturadas em momentos diferentes e com variações de ângulo e deformação. Para realizar a análise de similaridade, analisamos milhares de pares de amostras genuínas e falsas, extraindo para cada entrada um conjunto de características. Entre as características utilizadas estão

	source	k	g	p	cs	cm	ncj	nck	nj	nk	mtd	aj	ak	al	dg	pn	atj	atk	angulo	time	p1	p2	Pontuacao	
0	DB4	16961.0	1.0	309.0	61.0	20.0	12.0	18.0	41.0	42.0	5.0	29684.0	31134.0	41689.0	15.606983	3074.0	55615.0	54532.0	-21.0	41.0	0.232288	3.669725	0.373678	
1	DB2	8026.0	1.0	562.0	163.0	30.0	9.0	9.0	42.0	49.0	3.0	38029.0	37915.0	49920.0	25.510130	73170.0	54451.0	69698.0	-5.0	113.0	0.437318	21.951220	0.602179	
2	DB4	7484.0	1.0	0.0	20.0	12.0	31.0	27.0	52.0	51.0	4.0	17366.0	18593.0	45560.0	8.636554	67.0	59099.0	59480.0	15.0	10.0	0.054299	0.343675	0.185531	
3	DB4	7458.0	1.0	180.0	58.0	18.0	34.0	28.0	59.0	53.0	5.0	30456.0	28969.0	42197.0	14.068550	2215.0	46373.0	50799.0	-4.0	45.0	0.103614	0.679245	0.267151	
4	DB4	10759.0	1.0	364.0	49.0	17.0	5.0	9.0	26.0	29.0	3.0	35808.0	36163.0	39403.0	12.666974	2425.0	45479.0	49662.0	17.0	9.0	0.383289	12.565217	0.421244	
...
55739	DB3	16062.0	1.0	343.0	69.0	25.0	24.0	15.0	59.0	62.0	4.0	40697.0	40040.0	72213.0	20.274916	4109.0	86011.0	91982.0	0.0	19.0	0.170858	3.453039	0.405611	
55740	DB4	18226.0	0.0	0.0	5.0	6.0	16.0	28.0	52.0	95.0	5.0	5027.0	5031.0	23836.0	3.299037	0.0	75207.0	67835.0	-46.0	10.0	0.007287	0.160000	0.022430	
55741	DB3	2321.0	1.0	-16.0	15.0	11.0	17.0	20.0	38.0	42.0	2.0	18352.0	18608.0	50962.0	5.726457	20.0	58312.0	68709.0	2.0	7.0	0.075815	0.707602	0.161031	
55742	DB3	15989.0	1.0	337.0	86.0	26.0	14.0	12.0	59.0	41.0	11.0	38758.0	37923.0	58354.0	20.330322	8421.0	92600.0	61252.0	1.0	59.0	0.279454	7.952941	0.462791	
55743	DB3	2937.0	1.0	0.0	19.0	11.0	35.0	39.0	68.0	53.0	3.0	20157.0	19277.0	56691.0	5.928074	45.0	87396.0	61742.0	0.0	7.0	0.033574	0.177160	0.104563	

Tabela 1. Características calculadas.

a quantidade de minúcias em cada impressão, a quantidade de minúcias equivalentes entre elas, a qualidade de cada região, a área em comum, área de total, distorção e outras características que são segredos industriais. No total, selecionamos 30 características que são calculadas nas etapas anteriores e definimos duas saídas para o treinamento determinar se duas amostras são do mesmo dedo ou não.

A Tabela 1 ilustra um exemplo das características extraídas, onde a coluna “g” indica se a amostra é genuína (1) ou não (0), e a coluna “Pontuacao” representa um valor entre 0 e 1, indicando a similaridade obtida com a solução da Equação 1. Essa equação foi criada através de heurística com pesquisas e o conhecimento adquirido ao longo do tempo. Como referência teórica temos [Kock, 2016]. Os melhores algoritmos existentes utilizam sua própria equação e ela não é disponibilizada publicamente. A coluna “Pontuacao” é o cálculo utilizado atualmente para determinar a similaridade no algoritmo. Na tabela temos com j e k as amostras e podemos citar cs (segmentos comuns), cm (minúcias comuns), ncj e nck (número de minúcias que não casaram em j e em k), nj e nk (número de minúcias em j e k), mtd (cálculo interno), aj e ak (área das minúcias em j e k), al (área interseção), atj e atk (área total de j e k), dg (desvio do grafo), sendo o restante das colunas obtidos através de equações internas do algoritmo de comparação.

$$\text{Pontuacao} = \min \left(1.0, \frac{2.5 \cdot \text{cm}}{2.5 \cdot \text{cm} + \text{sn}c_j + \text{sn}c_k \cdot \left(\frac{\text{cm}}{\text{cm}+8.0} \right) \cdot \left(\frac{\text{dg}}{\max(\text{cm}, 1.0)} \right)} \right)$$

Se $\text{cm} < 8$ e $\text{sn}c_j + \text{sn}c_k > 0$, então:

$$\text{Pontuacao} = \min \left(1.0, \text{Pontuacao} \cdot \left(\frac{\text{cm}}{\frac{\text{sn}c_j + \text{sn}c_k}{2}} \right) \right)$$

Cálculo da Pontuação (1)

A Figura 1 apresenta a topologia da rede DNN proposta para a análise dessas características. Essa arquitetura é o resultado de várias testes e é uma proposição própria com o melhor resultado alcançado.

3.3. Modelos de Regressão

Após a extração das características e criação do dataset, realizamos a etapa de treinamento utilizando diferentes modelos de regressão.

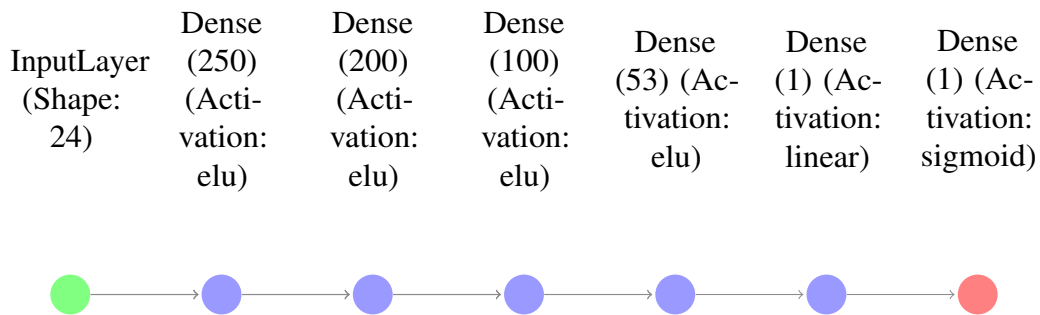


Figura 1. Modelo da rede para cálculo da similaridade.

Exploramos amplamente as capacidades dos modelos, ajustando cuidadosamente seus parâmetros e considerando treinamentos adicionais com polinômios de graus 2 e 3 para o MLPRegressor, a fim de capturar relações não lineares entre as características, buscando encontrar o modelo que proporcionasse o melhor desempenho e precisão em nossa tarefa de regressão.

O primeiro passo foi carregar a base de dados contendo as informações das amostras de impressões digitais. Em seguida, foi realizada a separação de 10% dos dados para a validação do modelo, garantindo que houvesse um conjunto de dados independente para avaliação posterior. Durante os testes a escolha de usar 90% para treinamento se mostrou mais eficaz que a abordagem de 80%.

Nos 90% dos dados utilizados para treinamento foram utilizadas estratégias de validação cruzada em k-fold com 5 execuções e também grid-search para determinação dos melhores hiper parâmetros.

Os modelos de regressão foram treinados com as configurações padrão no k-fold, exceto para o modelo MLPRegressor, que testamos vários hiperparâmetros pelo grid-search e a melhor foi um tamanho de lote de 48 e um tamanho de camada oculta de (80, 60).

Com base nas colunas “Pontuacao” e “g”, criamos uma nova coluna chamada “PontuacaoN” (Pontuação Ideal), com o objetivo de estabelecer uma referência ideal de similaridade. Essa coluna foi criada de forma heurística para aprimorar o resultado da coluna “Pontuacao” e indicar o valor alvo ideal, além do 0 e 1 da coluna “g”, que desejamos alcançar com os modelos. A equação original é falha em alguns casos e a normalização visa ajustar os valores para os que queremos obter. Como veremos serão utilizadas as colunas “g” e “PontuacaoN” como alvos para os estudos realizados nesse artigo. No final saberemos qual será o melhor.

Em seguida, calculamos a correlação emparelhada das colunas do nosso dataset em relação à coluna “G” e também em relação à coluna “PontuacaoN” (Figura 2) . Esse cálculo nos permitiu identificar quais colunas apresentavam uma correlação significativa com as saídas desejadas, tanto em relação à similaridade real quanto à similaridade ideal.

Com base nas correlações identificadas, foram realizados dois treinamentos: um para o modelo com o target “G” e outro com o target “PontuacaoN”. O dataset foi filtrado para conter apenas as 24 colunas com maior correlação absoluta.

Utilizamos a metodologia de Grid Search para determinar os melhores hiper parâmetros para o modelo MLPRegressor. Avaliamos diversas combinações possíveis de

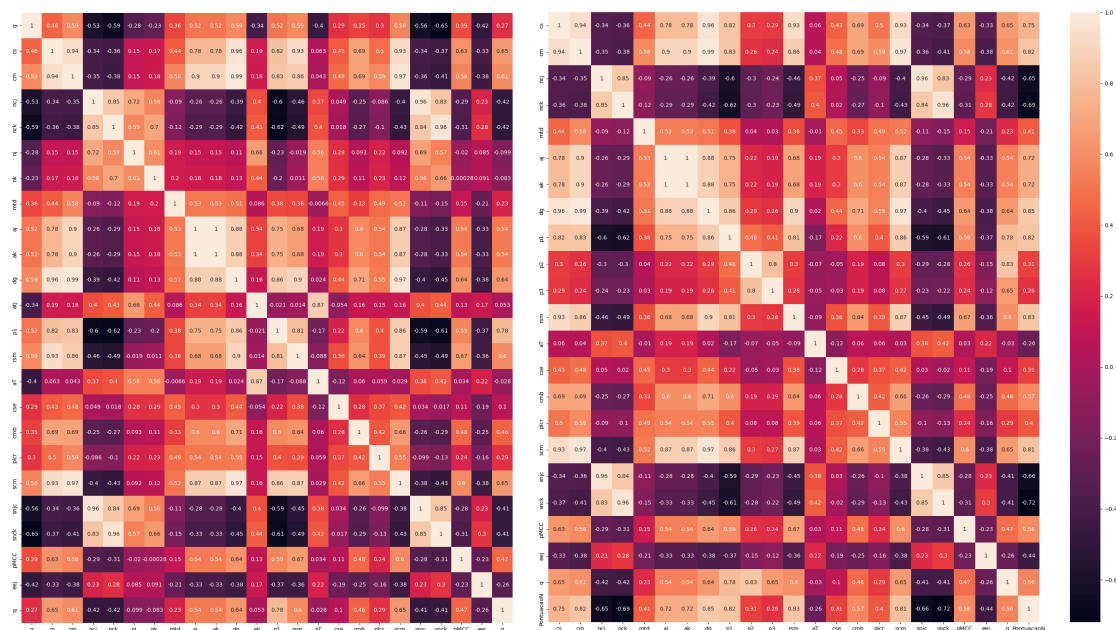


Figura 2. Correlação com G (esq) e PontuacaoN (dir).

valores dos hiper parâmetros e para esse estudo escolhemos a métrica R^2 como índice de desempenho.

Após a obtenção da melhor configuração de hiper parâmetros para o MLPRegressor, o modelo com base no ajuste fino, foi treinado 5 vezes e também criamos modelo acrescentando características polinomiais de grau 2 e grau 3.

Para determinar a significância estatística, utilizamos o teste t de duas amostras ([Kutner et al., 2004]). O teste t de duas amostras é um teste estatístico usado para comparar a média de duas populações. No nosso caso, estamos interessados em comparar a média dos valores de perda dos modelos treinados com as características originais, os modelos treinados com polinômios de grau 2 e os modelos treinados com polinômios de grau 3.

A hipótese nula do teste t de duas amostras é que não há diferença significativa entre as médias das duas populações. A hipótese alternativa é que há uma diferença significativa entre as médias das duas populações.

O valor p do teste t de duas amostras é 0,027. Este valor é menor que o nível de significância padrão de 0,05. Portanto, rejeitamos a hipótese nula e concluímos que há uma diferença significativa entre as médias dos valores de perda dos modelos treinados com as características originais, os modelos treinados com polinômios de grau 2 e os modelos treinados com polinômios de grau 3.

Em particular, os modelos treinados com polinômios de grau 2 e 3 têm valores de perda menores do que os modelos treinados com as características originais. Isso significa que os modelos treinados com polinômios de grau 2 e 3 são capazes de fazer previsões mais precisas do que os modelos treinados com as características originais. Optamos pelo polinômio de grau 2 devido ao consumo de memória elevado do grau 3.

Além do MLPRegressor, a cada execução, treinamos outros modelos de regressão,

utilizando como target a saída do MLPRegressor Grau 2. Foram utilizados os modelos LinearRegression, DecisionTreeRegressor, RandomForestRegressor, KNeighborsRegressor, Ridge, Lasso, ElasticNet, RANSACRegressor, HuberRegressor e TheilSenRegressor.

A partir dos resultados obtidos por esses modelos, selecionamos os 3 melhores com base na métrica de R^2 score. Esses 3 modelos foram então combinados, baseados nos seus Valores R^2 , em um modelo VotingRegressor, que implementa a estratégia de votação, na qual cada regressor contribui com uma previsão ponderada e a previsão final é determinada pela média dessas previsões ponderadas. O resultado obtido foi 0.9971.

Para o treinamento dos modelos DNN (Deep Neural Network) foram realizadas duas estratégias distintas utilizando diferentes targets e configurações. O primeiro treinamento do modelo DNN teve como alvo a coluna “G”, que representa a similaridade binária (0 ou 1). Utilizamos as características de entrada extraídas da correlação.

No segundo treinamento, o alvo do modelo DNN foi o resultado obtido pelo VotingRegressor. Essa abordagem busca explorar a capacidade do VotingRegressor em capturar características relevantes, processá-las e fornecer essas informações para o modelo DNN.

A melhoria ou piora percentual em relação ao modelo heurístico será calculada usando a Equação 2:

$$\text{Melhoria Percentual} = \left(\frac{\text{Resultado Modelo} - \text{Resultado Heurístico}}{\text{Resultado Heurístico}} \right) \times 100$$

Melhoria Percentual (2)

4. Resultados e Análise

Utilizando a base de treinamento, avaliamos as métricas de desempenho verificando a capacidade dos modelos em capturar a similaridade entre as impressões digitais como apresentado na Tabela 2. Selecionamos os três melhores para criar o Voting Regressor. A Tabela 3 mostra a avaliação dos modelos finais utilizando a base de treino e a base de validação com o modelo escolhido para aplicação prática. Essa etapa nos permitiu avaliar a capacidade dos modelos de generalizar e fazer previsões precisas em dados não vistos anteriormente.

MLPRegressor2, r2: 0.9978	RandomForestRegressor, r2: 0.9953
DecisionTreeRegressor, r2: 0.9875	KNeighborsRegressor, r2: 0.9826
LinearRegression, r2: 0.8877	Ridge, r2: 0.8868
HuberRegressor, r2: 0.8792	TheilSenRegressor, r2: 0.8083
RANSACRegressor, r2: 0.7545	Lasso, r2: -0.0001
ElasticNet, r2: -0.0001	

Tabela 2. Resultado dos modelos de regressão.

Ao final dessa metodologia de análise, obtivemos os resultados dos modelos de regressão e do modelo DNN. Esses resultados foram avaliados com base nas métricas de desempenho e na capacidade de capturar a similaridade entre as impressões digitais.

	Tipo	Limiar	Verdadeiras	Falsas	Erros	% melhoria
Heurística	Treino	0,135	44.111	9.329	2.304	
DNN 0 ou 1	Treino	0,988	45.758	9.329	657	3,7%
Vot. Regressor	Treino	0,525	45.164	9.329	1.251	2,4%
DNN + VR	Treino	0,135	40.957	9.329	5.458	-7,2%
Heurística	Validação	0,390	4.984	957	249	
Vot. Regressor	Validação	0,129	5.151	957	82	3,4%

Tabela 3. Resultados Treino e Validação.

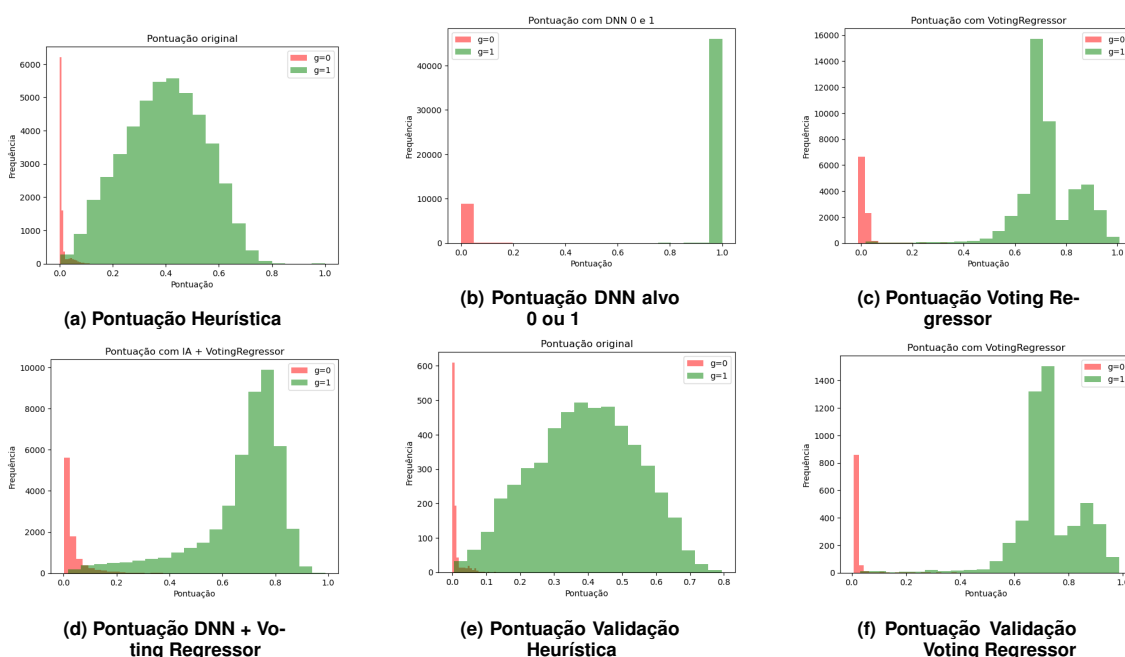


Figura 3. Histogramas das saídas

4.1. Análise da similaridade dos parâmetros espaciais obtidos durante o treinamento dos modelos

Utilizamos um dataset composto por 55.744 linhas de características, das quais 46.415 linhas correspondem a amostras genuínas (verdadeiras) e 9.329 linhas correspondem a comparações de impressões digitais de diferentes dedos (falsas).

Nosso objetivo é eliminar completamente falsos positivos, ou seja, eliminar casos onde duas impressões digitais de dedos diferentes são classificadas como genuínas (falsos positivos), além de minimizar a exclusão de amostras genuínas (falsos negativos), ou seja, amostras genuínas classificadas como não-genuínas.

Com base nos resultados obtidos, analisamos não somente sua saída absoluta, mas a capacidade de ser usado em ambientes de produção. Para isso localizamos o valor da pontuação (limiar) que separa a saída do modelo em duas classes distintas e cujo foco é eliminar todos os falsos positivos. O limiar de cada modelo é obtido no ponto onde o valor da pontuação é tal que o número de falsos positivos acima desse valor é 0 (zero).

Os histogramas da Figura 3 representam graficamente a separação entre amostras genuínas (verdes), falsas (vermelhas) e a região de sobreposição. Quando mais afastado

estiver o conjunto verde do vermelho melhor o resultado.

No histograma da solução heurística original, utilizada como base para os estudos de melhoria, (Figura 3a), podemos observar que, ao usar um limiar de 0.135, foram removidas 2.304 amostras genuínas.

No histograma da pontuação DNN com alvo na coluna “g” (Figura 3b), podemos observar que, ao usar um limiar de 0,988, foram removidas apenas 657 amostras genuínas.

No histograma da pontuação do VotingRegressor (Figura 3c), podemos observar que, ao usar um limiar de 0,525, foram removidas 1.251 amostras genuínas.

No histograma da pontuação do modelo DNN cujo alvo do treinamento foi o resultado do VotingRegressor (Figura 3d), podemos observar que, ao usar um limiar de 0,469, foram removidas 5.458 amostras genuínas.

O modelo DNN com alvo 0 e 1 apresentou o melhor desempenho em termos da quantidade de amostras genuínas que não foram removidas. No entanto, o limiar utilizado foi extremamente alto, o que pode gerar problemas em aplicações práticas. Ao investigar a causa desse limite tão alto, identificamos que o erro está nas etapas anteriores, especificamente na extração das minúcias das impressões digitais. Esse problema fez com que algumas amostras falsas atingissem uma classificação muito próxima de 1 (um). Já estamos trabalhando um novo modelo de DNN para a etapa de extração de minúcias.

Para aplicação em produção, o melhor modelo foi a estratégia do VotingRegressor, que conseguiu identificar 1.053 impressões digitais adicionais em comparação com uma solução anteriormente desenvolvida usando uma abordagem heurística.

Substituindo os valores na Equação 2 temos:

$$\text{Melhoria Percentual} = \left(\frac{45.164 - 44.111}{44.111} \right) \times 100 = 2.4\%$$

Esse resultado mostra a eficácia do modelo VotingRegressor na análise da similaridade dos parâmetros espaciais obtidos a partir das minúcias das impressões digitais. Com base nessa melhoria percentual de 2.4%, podemos concluir que o modelo do VotingRegressor possui um grande potencial para aplicações práticas na identificação e verificação biométrica de impressões digitais.

4.2. Resultados com a base de Validação

A etapa de validação foi fundamental para verificar a eficácia dos modelos desenvolvidos. Durante essa fase, os modelos foram avaliados usando uma base de dados de validação separada, na qual os modelos não tinham conhecimento prévio.

Os resultados obtidos na validação foram equivalentes aos resultados do treinamento, o que demonstra a capacidade dos modelos de generalizar o aprendizado para novos conjuntos de dados. Essa validação prática é essencial para garantir que os modelos sejam efetivos em situações reais.

A heurística foi capaz de identificar corretamente 4.984 das 5.233 amostras genuínas, enquanto o modelo do VotingRegressor obteve um desempenho melhor, identificando 5.151 amostras genuínas. Essa diferença de desempenho reforça a superioridade do modelo do VotingRegressor.

5. Conclusão

A utilização de recursos de Inteligência Artificial, como os modelos do VotingRegressor e DNN, demonstrou um avanço significativo na etapa de cálculo da similaridade no processo de verificação de impressões digitais. Durante a análise dos resultados da similaridade dos parâmetros espaciais obtidos a partir das minúcias das impressões digitais, constatamos a eficácia e o potencial dos modelos do VotingRegressor e DNN, além de destacar a importância de melhorar a qualidade dos dados de entrada.

Como trabalhos futuros, este estudo evidenciou a necessidade de aprimorar as etapas de segmentação da digital e extração de minúcias para obter resultados mais precisos e confiáveis na identificação e verificação de impressões digitais. Segundo Champod et al. [champod] e Maltoni et al. [maltoni09], a segmentação da digital e a extração de minúcias desempenham um papel crucial no processo de verificação biométrica. A adoção de técnicas mais avançadas nessas etapas é fundamental para garantir a precisão e confiabilidade dos modelos desenvolvidos. Nesse sentido, Goodfellow [Goodfellow] e Mitchell [mitchell1997] destaca o potencial do uso de técnicas de aprendizado profundo, como redes neurais convolucionais, para aprimorar essas etapas iniciais, enquanto Russell e Norvig [russell2016] exploram as aplicações mais amplas da inteligência artificial.

Referências

- Chen, X., Zhang, Y., Zhao, J., and Wang, S. (2015). Fingerprint presentation attack detection using deep learning. *IEEE Transactions on Information Forensics and Security*, 10(10):2430–2443.
- Gabor, D. (1946). Theory of communication. In *Journal of IEE (London)*, volume 93, pages 429 – 457. IJCB.
- Goodfellow, I. (2016). *Deep Learning*. The Mit Press, second edition.
- Jain, A., Hong, L., Pankanti, S., and Bolle, R. (1997). An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 1(85):1365 – 1388.
- Kock, A. J. d. (2016). The study of similarity score calculation methods for minutia-based fingerprint matching algorithms. Master's thesis, Stellenbosch University.
- Kutner, M., Neter, J., Nachtsheim, C. J., and Li, W. (2004). *Applied Linear Statistical Models*. McGraw-Hill/Irwin, New York, 5 edition.
- Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. (2009). *Handbook of Fingerprint Recognition*. Springer, second edition.
- Xu, J., Zhang, L., and Wang, J. (2019). A new fingerprint matching algorithm based on convolutional neural networks. *IEEE Access*, 7:100883–100892.
- Zhang, Y., Zhou, Y., Zhao, J., and Wang, S. (2015). Fingerprint minutiae extraction using convolutional neural networks. *Pattern Recognition*, 48(1):40–50.