Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Verificação de Assinaturas Online com Aprendizagem de Métricas: Explorando o Impacto da Dispersão de Limiares Locais Ótimos

João Pedro Felix de Almeida

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Bruno Luiggi Macchiavello Espinoza

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Verificação de Assinaturas Online com Aprendizagem de Métricas: Explorando o Impacto da Dispersão de Limiares Locais Ótimos

João Pedro Felix de Almeida

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Bruno Luiggi Macchiavello Espinoza (Orientador)
CIC/UnB

Prof. Dr. Pedro Garcia Freitas
CIC/UnB

Prof. Dr. Eduardo Peixoto Fernandes da Silva
ENE/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 25 de Julho de 2023

Dedicatória

Dedico este trabalho aos meus pais e a minha irmã que sempre me incentivaram a estudar.

Agradecimentos

O trabalho aqui apresentado só pôde ser realizado graças ao conjunto de dados DeepSign que foi gentilmente cedido pela Universidade Autônoma de Madrid (UAM-ES).

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Este trabalho apresenta uma nova função de perda para aprendizagem de métricas no contexto de verificação biométrica de assinaturas *online* que tem por objetivo, ao mesmo tempo, permitir que o modelo seja robusto em relação à alta variação intraclasse enquanto gera representações profundas que respeitem uma mesma distribuição a fim de tornar a distância entre duas assinaturas de uma mesma pessoa semelhante a distância entre duas assinaturas de outra pessoa qualquer, isso é feito para permitir que o sistema atinja bons resultados ao operar com um limiar global. O modelo que serviu de referência para este trabalho realiza a classificação de uma assinatura como sendo original ou falsificada a partir do uso de limiares e, embora apresente ótimos resultados utilizando um limiar local, seus resultados pioram consideravelmente ao utilizar um limiar global. Esta função de perda, juntamente com a substituição de dois sinais de entrada, propostas neste trabalho, tem como objetivo diminuir a dispersão entre os limiares locais ótimos a fim de aproximá-los e tornar o limiar global ótimo mais expressivo, de modo que o sistema apresente resultados melhores com relação ao *Equal Error Rate* (EER). O modelo proposto foi testado em todas as tarefas do conjunto de dados DeepSign e em relação ao modelo base apresentou melhoras em relação à variância dos limiares globais entre 26.39% e 99.05% e em relação à amplitude entre 14.79% e 60.79%. Tais melhoras foram acompanhadas por melhorias no EER de modo a superar o estado da arte em algumas tarefas. Com relação ao modelo de referência o modelo proposto apresentou melhora relativa de 7.87% no cenário com assinaturas realizadas com caneta *stylus*, ataques com falsificações profissionais e quatro assinaturas de referência, além de melhoras relativas de 14.44% e 7.26% com quatro e uma referência, respectivamente, no cenário em que as assinaturas foram realizadas com dedo e ataques com falsificações profissionais.

Palavras-chave: biometria, verificação de assinaturas *online*, aprendizagem de métricas

Abstract

This work presents a novel loss function for metric learning in the context of online signature biometric verification whose goal is, at the same time, allow the model to be robust regarding high inter-class variance while generating similar deep representations from different signatures of the same user that respect the same distribution in order to make the distance between two signatures of the same person similar to the distance between two signatures of any other person, this is done to allow the system achieve good results when operating with a global threshold. The reference model used in this work performs the classification of a signature as genuine or forgery using thresholds and, albeit it presents good results using local threshold, they decrease when using a global threshold. This novel loss function, together with the change of two input signals, proposed in this work, aims to decrease the dispersion of optimal local thresholds to make the optimal global threshold more expressive, resulting in better *Equal Error Rate* (EER). The proposed model was tested in all tasks of the DeepSign dataset and, in relation to the base model, presented improvements from 26.39% to 99.05% in variance and from 14.79% to 60.79% in amplitude. These improvements were accompanied by improvements in the EER allowing the proposed model to surpass the state of the art in some tasks. In relation to the reference model, the proposed model presented an improvement of 7.87% in the stylus scenario, four reference signatures and attacks with skilled forgeries, in addition with relative improvements of 14.44% and 7.26% in the finger scenario with four and one, respectively, reference signatures and attacks with skilled forgeries.

Keywords: biometrics, online signature verification, metric learning

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CRAN Rede Adaptativa Recorrente Convolucional — do inglês *Convolutional Recurrent Adaptive Network*.

DTW *Dynamic Time Warping*.

EB-DBA *Euclidean barycenter-based DTW barycenter averaging*.

EER *Equal Error Rate*.

FAR Taxa de Falsa Aceitação — do inglês *False Accept Rate*.

FRR Taxa de Falsa Rejeição — do inglês *False Reject Rate*.

FSL Aprendizagem com Poucas Instâncias — do inglês *Few Shot Learning*.

GARU *Gated Auto Regressive Units*.

GRU *Gated Recurrent Unit*.

LSTM Memória de Curto e Longo Prazo — do inglês *Long Short Term Memory*.

MMD Máxima Discrepância Média — do inglês *Maximum Mean Discrepancy*.

SGD Gradiente Descendente Estocástico — do inglês *Stochastic Gradient Descent*.

SVM Máquina de Vetor de Suporte — do inglês *Suport Vector Machine*.

Capítulo 1

Introdução

Quer seja para firmar contratos ou simplesmente assegurar o recebimento de uma correspondência, o uso de assinaturas sempre esteve entre os métodos mais aceitos pela sociedade para garantir o envolvimento de alguém em algo [1].

Contudo, a ampla aceitação e consequentemente o alto uso de assinaturas como forma de reconhecimento motivou que pessoas mal-intencionadas passassem a imitar a assinatura de outras com o intuito de obter alguma vantagem ilegal, o que exigiu ainda no século XVIII que pessoas se especializassem na comparação de assinaturas a fim de verificar se elas pertenciam ao mesmo autor ou não [2].

Naturalmente, foi apenas uma questão de tempo até que esse passasse a ser, também, um problema computacional com destaques para os sistemas propostos, ainda na década de 1970, em [3, 4, 5], que já apontavam a possibilidade de resolução do problema a partir de algoritmos computacionais. Após quase 50 anos da publicação desses trabalhos os avanços na área foram expressivos, embora ainda hoje o uso de assinaturas em sistemas biométricos apresente resultados inferiores quando comparado com outras biometrias [6] em termos de acurácia, o que apenas evidencia o espaço para crescimento na área.

A depender de como a assinatura é armazenada, existem duas categorias de reconhecimento biométrico de assinaturas: *offline*, quando a assinatura é salva como uma imagem, e *online*, quando a assinatura é descrita por meio de sequências temporais como as que indicam a trajetória e pressão aplicada no dispositivo de captura. Este trabalho tem como foco a verificação de assinaturas *online* que tem como principal técnica empregada o algoritmo *Dynamic Time Warping* (DTW), responsável por medir a discrepância entre sequências temporais, mesmo que elas possuam tamanhos distintos. Hoje, a maioria dos trabalhos com resultados que compõem o estado da arte consistem em técnicas que buscam aumentar o poder de discriminação do DTW [7, 8, 9, 10] que com as modificações necessárias já é capaz, mesmo sem fazer uso de inteligência artificial, de apresentar bons resultados [11].

Tendo em vista que assinaturas são dados sensíveis, o processo de criação e compartilhamento de conjuntos de dados tende a ser bastante burocrático, o que durante muitos anos dificultou o uso de abordagens que fazem uso de aprendizagem profunda, marcadas pelo grande volume de dados exigido, na área de verificação de assinaturas *online*. Contudo, os recentes esforços realizados em [9] resultaram no conjunto de dados DeepSign que contém assinaturas originais e falsificações profissionais de pouco mais de 1500 usuários e é hoje o maior conjunto de dados de assinaturas *online* disponível para pesquisadores e um dos principais responsáveis pelo surgimento de técnicas que fazem uso dessa classe de algoritmos. Além disso, o DeepSign também apresenta um rígido protocolo de avaliação a fim de permitir uma comparação justa entre diferentes trabalhos, que também prevê que os usuários utilizados durante a fase de treinamento sejam diferentes dos utilizados em produção com o intuito de facilitar a implantação prática dos sistemas. Dessa forma, a partir do uso de uma ou quatro, a depender da tarefa, assinaturas originais de referência, o sistema deve ser capaz de determinar se uma assinatura em análise é original ou não.

Essa última restrição tem como um dos principais efeitos o desencorajamento do uso de classificadores específicos por usuário tendo em vista o pequeno número de assinaturas de referência disponível e a não existência de amostras negativas, isto é, falsificações profissionais, e faz com que as principais abordagens que fazem uso de redes profundas se baseiem em redes extratoras de características a fim melhorar o desempenho do DTW [7, 9, 10]. O principal reflexo disso, no entanto, é que o sistema precisa operar de maneira independente do usuário de modo que é comum, como no caso dos trabalhos citados anteriormente, que eles operem a partir do uso de pontuações de similaridade ou dissimilaridade a partir de um limiar: a depender de como a pontuação das assinaturas se posiciona com relação ao limiar a assinatura em análise é classificada como sendo original ou falsificada.

O sistema utilizado como referência neste trabalho é o DsDTW [10], cujos resultados fazem parte do estado da arte, que propõe, através do uso de aprendizagem de métricas, um sistema que a partir de um conjunto de assinaturas de referência determina uma pontuação de dissimilaridade entre as mesmas e a assinatura em análise. O DsDTW faz uso de um limiar global e classifica as assinaturas que possuírem pontuação inferior ao limiar como sendo originais e como falsificações no caso contrário. O problema observado com esse sistema e que é o principal objeto de estudo neste trabalho é o fato de que o DsDTW é capaz de criar, na imensa maioria dos usuários presentes no conjunto de testes do DeepSign, representações profundas cujas pontuações de dissimilaridade admitem uma separação perfeita entre assinaturas originais e falsificações quando considerado um limiar específico por usuário, isto é, um limiar local. Contudo, esses limiares específicos por usuário são no geral consideravelmente diferentes uns dos outros de modo que, ao fazer

uso de um limiar único global, mesmo que ótimo, o sistema acaba por errar a classificação de assinaturas que ao utilizar um limiar local ótimo por usuário o mesmo não erraria.

O objetivo desta monografia é apresentar uma solução que diminui os efeitos desse problema, isto é: modificar o DsDTW para que o sistema crie representações profundas de modo que os limiares ótimos específicos para cada usuário passem a ser mais próximos uns dos outros, ou seja, diminuir a dispersão dos limiares ótimos locais. Com isso, a expectativa é que o limiar ótimo global seja mais semelhante aos limiares ótimos locais e, consequentemente, que os resultados do sistema melhorem. Além disso, esta monografia também apresenta uma visão geral sobre biometria e o uso de sistemas de verificação biométrica baseados em assinaturas *online*, além de abordar algumas das principais técnicas utilizadas na solução do problema.

Esta monografia está dividida em cinco capítulos: este, que é o primeiro e apresenta a motivação do trabalho e outros quatro. O Capítulo 2 apresenta conceitos básicos relacionados a área de biometria com foco em sistemas de verificação que trabalham com assinaturas *online*, apresentando algumas das principais técnicas utilizadas para a solução do problema, incluindo uma breve revisão de trabalhos relacionados e uma visão detalhada do modelo DsDTW, que serve de referência para este trabalho. O Capítulo 3 detalha o problema relacionado à diferença dos limiares ótimos locais e global descrito anteriormente e apresenta a solução aqui proposta. O Capítulo 4 apresenta, com detalhes, o conjunto de dados DeepSign e seu rígido protocolo de avaliação, além dos resultados obtidos juntamente com algumas considerações acerca dos mesmos. Por fim, o Capítulo 5 apresenta minhas conclusões sobre o trabalho desenvolvido e algumas sugestões de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo descreve brevemente conceitos básicos relacionados à biometria e à verificação de assinaturas online, além de abordar algumas das técnicas utilizadas na elaboração da solução proposta e alguns trabalhos relacionados. No final do capítulo também é apresentada a arquitetura que serve de base para este trabalho.

2.1 Biometria

Biometria é a ciência de estabelecer a identidade de um indivíduo a partir de atributos físicos, químicos e comportamentais de uma pessoa [2]. Tais atributos serão denominados neste trabalho como simplesmente impressão biométrica. Devido a distinção natural entre impressões biométricas, a dificuldade de serem forjadas e dos usuários detentores da impressão as repudiarem já que elas se baseiam não naquilo que o usuário sabe, como senhas, e sim em quem o usuário é [12], a biometria é uma ótima opção em sistemas que exigem alto nível de segurança como os bancários e eleitorais.

Entre os principais usos de reconhecimento biométrico, duas tarefas merecem destaque: a identificação e a verificação. Um sistema de identificação biométrica busca responder à pergunta "quem é você?" e para isso realiza comparações entre a impressão biométrica em análise e as amostras existentes no banco de dados, de modo a retornar o identificador do usuário uma vez que uma correspondência seja encontrada, caracterizando assim uma comparação $1 : N$ (lê-se um para N) [13]. Em contrapartida, a verificação biométrica busca responder à pergunta "você é de fato quem afirma ser?" e para isso conduz uma comparação apenas entre a(s) amostra(s) cadastrada(s) daquele usuário com a impressão biométrica em análise, caracterizando assim uma comparação $1 : 1$ (lê-se um para um) [13].

A construção de um sistema biométrico depende de fatores como o contexto em que ele será utilizado e da impressão (ou das impressões no caso de sistemas multibiométricos) que será utilizada. Alguns exemplos de impressões biométricas são:

1. Impressões físicas: digital, face, íris, geometria da mão e palmar.
2. Impressões comportamentais: modo de andar, assinatura e modo de digitar.
3. Impressões químicas: odor e DNA.

Já no que diz respeito ao contexto, é necessário avaliar se o usuário quer ou não ser reconhecido. Em um sistema cooperativo, como um que dá acesso a uma conta bancária, é natural que o usuário busque cooperar com o sistema ao tentar recriar as condições da impressão biométrica no momento do cadastro [13]. Contudo, em um sistema de reconhecimento negativo, como um que identifica foragidos da justiça, é natural que o usuário busque atrapalhar o sistema aplicando, por exemplo, força excessiva em um sensor de impressões digitais [14].

Outro fator determinante na construção de um sistema biométrico adequado a um contexto específico é se ele é aberto ou secreto. Quando o usuário sabe que está passando por um processo de reconhecimento, seja ele de identificação ou verificação, diz-se que o sistema é aberto, caso ele não saiba, o sistema é então classificado como secreto [14].

A cooperação do usuário e a ciência do processo de reconhecimento biométrico são determinantes na escolha da impressão biométrica que será utilizada. Enquanto sistemas de reconhecimento facial podem atuar de maneira secreta, os que lidam com assinaturas necessitam que o usuário realize a assinatura, o que o torna impossível de ser utilizado nesses contextos.

No entanto, apesar de sistemas biométricos apresentarem um potencial de segurança maior ao buscar reconhecer o usuário não pelo que ele sabe, mas sim por quem ele é, deve-se ter em mente que dificilmente existe uma correspondência perfeita entre duas impressões biométricas de um mesmo usuário, pois fatores como condições físicas e psicológicas podem afetar sua impressão biométrica [12].

Por isso, sistemas biométricos geralmente atuam com o objetivo de obter uma pontuação entre as amostras de referência e a amostra em análise e, a depender de como ela se posiciona em relação a um limiar, é tomada a decisão se aquela comparação é verdadeira, no caso em que a impressão em análise e a de referência são correspondentes, ou falsa, no caso em que a impressão em análise e a de referência não são correspondentes [15].

Dessa forma, é esperado que um sistema biométrico cometa erros [13]. Como um sistema de identificação pode ser visto como uma série de N ocorrências de um sistema de verificação, é suficiente analisar as possibilidades de erro de um sistema de verificação biométrica:

- Aceitar uma impressão falsa: as impressões erroneamente aceitas pelo sistema contribuem para a Taxa de Falsa Aceitação — do inglês *False Accept Rate* (FAR). No cenário que impressões referentes ao mesmo usuário recebem pontuação próxima a zero (medida de dissimilaridade), a tendência é que, conforme o limiar aumente, a FAR também aumente. No cenário que a pontuação é muito maior que zero (medida de similaridade), a tendência é que, conforme o limiar aumente, a FAR diminua.
- Rejeitar uma impressão verdadeira: as impressões erroneamente rejeitadas pelo sistema contribuem para a Taxa de Falsa Rejeição — do inglês *False Reject Rate* (FRR). No cenário que impressões referentes ao mesmo usuário recebem pontuação próxima a zero (medida de dissimilaridade), a tendência é que, conforme o limiar aumente, a taxa de falsa rejeição diminua. No cenário que a pontuação é muito maior que zero (medida de similaridade), a tendência é que, conforme o limiar aumente, a taxa de falsa rejeição também aumente.

Independentemente de o sistema trabalhar com uma medida de similaridade ou de dissimilaridade, a métrica comumente utilizada para avaliar um sistema biométrico é o ponto em que as taxas de falsa aceitação e de falsa rejeição são iguais, isto é, o EER. Dessa forma, quanto menor o valor do EER, melhor [16].

Este trabalho aborda o uso de assinaturas feitas à mão como impressão biométrica no cenário de verificação, onde o objetivo é determinar se uma assinatura foi, de fato, realizada pelo usuário cadastrado, o que caracteriza um sistema biométrico cooperativo e aberto.

2.2 Verificação de Assinaturas

Assinaturas feitas à mão estão entre as principais formas de realizar autenticação de um indivíduo nos dias de hoje, não apenas pelo seu poder de discriminação mas também devido ao seu histórico de aceitação tanto socialmente quanto legalmente [8]. Justamente por isso, ao longo dos anos diversos sistemas biométricos baseados em assinaturas, em especial de verificação, foram propostos [10, 11, 17].

Apesar disso, especialmente devido ao fato de se tratar de uma biometria comportamental, ainda existem muitos desafios a serem superados na área. Grande parte disso se deve ao fato de que é natural que a assinatura de uma pessoa sofra variações a depender de fatores como caneta, seja ela esferográfica ou *stylus*, posição, humor e até mesmo o tempo dedicado para a realização da assinatura, fazendo assim que ocorram, em grande escala, as chamadas variações intraclasse, pois ocorrem entre as assinaturas realizadas por um mesmo usuário [2].

Além disso, a fim de construir um sistema seguro, é necessário que o sistema saiba distinguir assinaturas de diferentes usuários, especialmente daqueles que tenham interesse em se passar por outra pessoa. Por isso, é necessário que o sistema seja robusto o suficiente para saber lidar com falsificações profissionais, que naturalmente se parecerão com as assinaturas originais, o que gera, em pequena escala, as chamadas variações interclasse, pois ocorrem entre as assinaturas realizadas por diferentes usuários.

Dessa forma, ao mesmo tempo que o sistema de verificação de assinaturas não deve rejeitar assinaturas originais, ele precisa rejeitar assinaturas falsificadas, o que caracteriza um problema de otimização de duas métricas opostas: ao relaxar o sistema tornando-o mais tolerante a variações, com o intuito de não rejeitar assinaturas originais, a tendência é que assinaturas falsificadas também sejam beneficiadas e consequentemente aceitas. Ao restringir o sistema de modo a não aceitar grandes variações entre assinaturas, com o intuito de não aceitar falsificações, a tendência é que assinaturas originais também sejam prejudicadas e consequentemente rejeitadas.

A depender de como a assinatura é armazenada, existem duas categorias principais de verificação de assinaturas [18]:

1. *Offline*: quando a assinatura é salva como uma imagem, o que permite que ela seja feita, por exemplo, em papel comum com uma caneta esferográfica.
2. *Online*: quando ao invés de uma imagem, a assinatura é capturada por algum dispositivo dedicado, como mesas digitalizadoras, ou genérico, como *tablet* e celular, e salva como uma sequência numérica que descreve as características do processo de construção da assinatura.

Os sistemas que trabalham com assinaturas *offline* operam com imagens e comumente fazem uso de técnicas gerais de visão computacional e processamento de imagens. Uma vez que esta abordagem está fora do escopo deste trabalho, ao leitor interessado é recomendada a leitura de [19] para maiores detalhes acerca dos avanços na área. Em relação aos sistemas que trabalham com assinaturas *online*, foco deste trabalho, existem duas abordagens principais [7]:

1. Paramétricas: que buscam descrever as assinaturas a partir de parâmetros gerais como número de vezes que a caneta foi levantada e tamanho da assinatura.
2. Funcionais: que descrevem a assinatura como funções que variam ao longo do tempo como a trajetória e pressão da caneta.

A abordagem funcional tradicionalmente apresenta melhores resultados se comparada com a paramétrica [2] e é a abordagem utilizada neste trabalho.

2.3 Aprendizagem de Máquina

Aprendizagem de máquina é uma subárea de inteligência artificial que tem como principal objetivo trabalhar a questão de "como" construir um programa de computador que consiga melhorar sua performance através da experiência, de maneira formal [20]:

> **Definição 1** Um programa de computador aprende com a experiência **E** a respeito de uma classe de tarefas **T** e medida de performance **P** se sua performance nas tarefas **T**, medidas por **P**, melhoram com a experiência **E**.

No contexto de verificação de assinaturas *online*, por exemplo, o problema poderia ser modelado da seguinte forma:

- Tarefa **T**: determinar se duas assinaturas pertencem ou não a um mesmo usuário.
- Medida de performance **P**: EER.
- Experiência de treinamento **E**: um conjunto de dados composto por funções temporais que descrevem as assinaturas bem como a informação de quem é seu autor.

A fim de alcançar esse objetivo, existem quatro tipos principais de aprendizagem de máquina:

1. Supervisionada: em que o objetivo é aprender a mapear uma entrada para uma saída baseado em um conjunto de dados rotulados que associa pares de entrada e saída [21].
2. Não supervisionada: o processo de aprendizagem não é supervisionado uma vez que os dados não possuem rótulos que determinem o que eles são. É tipicamente utilizada para descobrir padrões em dados (mineração de dados) [21].
3. Semissupervisionada: uma espécie de meio termo das abordagens anteriores em que os dados não rotulados são usados para melhorar o desempenho do modelo treinado com dados rotulados [21].
4. Reforço: em que a aprendizagem é baseada em recompensas (que podem ser boas ou ruins) e o objetivo é o que sistema aprenda a tomar decisões que maximizem as boas recompensas e minimizem as más recompensas [22].

Neste trabalho, a solução proposta para o problema de verificação de assinaturas *online* será baseada em aprendizagem supervisionada, uma vez que o conjunto de dados aqui utilizado é rotulado de maneira adequada para esse tipo de abordagem.

Naturalmente, existem diversas classes de algoritmos dentro da área de aprendizagem de máquina supervisionada que a depender da tarefa se demonstram mais ou menos adequados, como os algoritmos baseados em instâncias, regressão, árvores de decisão, métodos *baysianos* e redes neuronais [23]. É justamente essa última classe de algoritmos que será utilizada neste trabalho.

2.4 Redes Neuronais

As redes neuronais surgiram inspiradas no cérebro humano e são constituídas por unidades de processamento capazes de armazenar experiências através de um processo de aprendizagem e que podem se comunicar umas com as outras de modo a simular o processo de sinapse [24].

A ideia por trás dessas redes é que elas funcionem como uma espécie de composição de funções comumente chamadas de camadas, onde a saída de uma das funções, isto é, de uma das camadas, serve como entrada para outra e o tamanho dessa cadeia é que determina a profundidade do modelo (terminologia que dá origem ao termo aprendizagem profunda) [25].

A depender da maneira de como as informações fluem pelo modelo, existem dois tipos de redes neuronais principais [25]: as *feedfoward*, em que as informações fluem apenas do ponto de entrada até o ponto de saída e as recorrentes, em que as camadas intermediárias conseguem se comunicar tanto com suas camadas anteriores quanto com as posteriores.

2.4.1 Camadas Lineares

Camadas lineares são camadas *feedfoward* em que todos os neurônios se comunicam uns com os outros através de parâmetros (pesos) que são otimizados a partir do processo de aprendizagem [26]. Seja A uma matriz de pesos no formato $[S, E]$ em que S e E são, respectivamente, o tamanho da saída e entrada da camada, x o vetor de entrada e b um vetor de viés (opcional), uma camada linear realiza a seguinte operação:

$$y = xA^{\intercal} + b. \tag{2.1}$$

A Figura 2.1 ilustra a influência dos dados de entrada na saída de uma possível camada linear.

2.4.2 Redes Convolucionais

Redes neuronais convolucionais são redes neuronais *feedfoward* que realizam uma operação de convolução em pelo menos uma de suas camadas e são apropriadas para dados

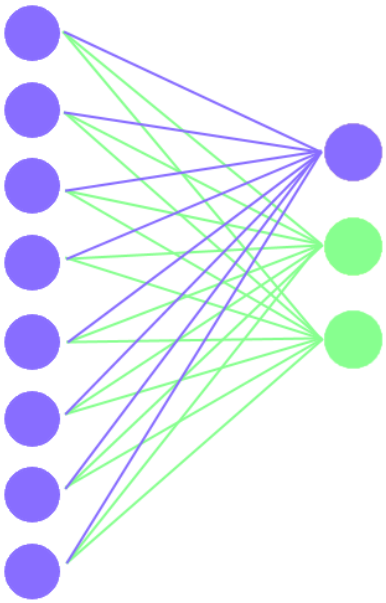

Figura 2.1: Influência dos neurônios de entrada (à esquerda) na saída (à direita) de uma camada linear. Note que todos os neurônios da entrada influenciam em todos os neurônios da saída.

que possuem topologia em formato de grelha, como sequências temporais ou imagens, além de permitirem seu uso com entradas de tamanhos variáveis [27]. Uma típica rede convolucional faz uso de três camadas principais: convolução, amostragem e função de ativação, no caso ReLU , que serão explicadas no decorrer desta seção.

Camada de Convolução

A camada de convolução faz uso de um núcleo, tipicamente menor que o dado, para extrair características da entrada. A implicação direta de um núcleo menor que o dado é que a convolução gera interações esparsas, pois evita que todos os neurônios interajam entre si, como aconteceria em uma típica rede neuronal baseada, por exemplo, em camadas lineares [25]. Naturalmente, isso reduz consideravelmente a quantidade de informações a serem armazenadas.

Além disso, o mesmo núcleo é utilizado em toda a operação de convolução, o que caracteriza o compartilhamento de parâmetros, que tem como principal efeito tornar a convolução uma operação equivariante à translação, o que significa que mudanças na entrada da função (camada) geram as mesmas mudanças na saída, isto é, a função f é equivariante à função g se $f(g(x)) = g(f(x))$ [25]. A Figura 2.2 ilustra a aplicação de uma operação de convolução em uma entrada de uma dimensão, comumente utilizadas para o

processamento de sequências temporais como assinaturas *online*, com núcleo de tamanho três e passo de tamanho um.

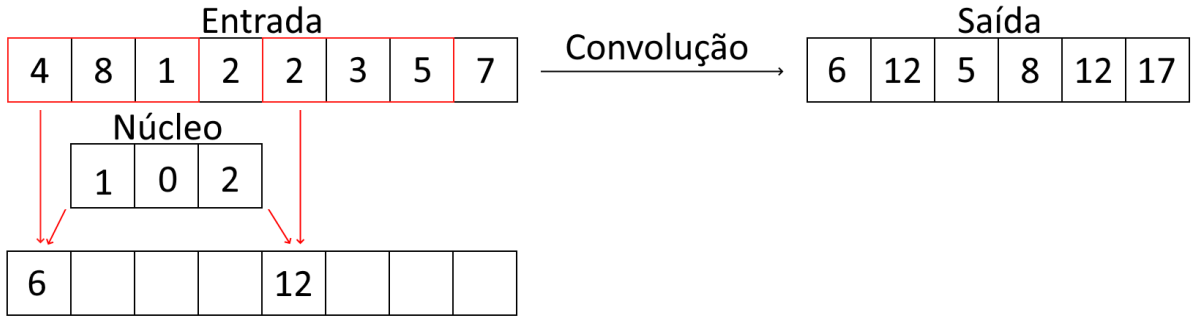

Figura 2.2: Exemplo de convolução de uma dimensão.

Função de ativação ReLU

A função de ativação ReLU é outra das principais camadas presentes em redes neuronais convolucionais, onde é responsável por determinar a importância do neurônio na etapa de aprendizagem. A ReLU pode ser definida da seguinte forma:

$$\text{ReLU}(x) = \max(x, 0), \tag{2.2}$$

ou seja, uma função de ativação ReLU basicamente substitui as entradas negativas por zero. Devido as melhorias tanto na velocidade de treinamento do modelo quanto na performance em termos de métricas como a acurácia quando comparadas a outras funções de ativação [28], a ReLU é hoje a principal função de ativação utilizada em redes neuronais convolucionais [27].

Amostragem

Um dos principais problemas associados à operação de convolução é que as características extraídas são bastantes sensíveis com relação à sua posição na entrada e uma solução para minimizar esse problema e alcançar um grau de invariância a translações locais é a partir das camadas de amostragem [29].

As camadas de amostragem atuam basicamente diminuindo o tamanho da amostra da entrada a partir de uma janela de tamanho fixo. Essa janela é deslocada sobre os dados respeitando o tamanho do passo, de maneira semelhante ao que acontece com os núcleos de uma convolução, aplicando a operação apropriada a depender da técnica de amostragem. Entre os diversos tipos existentes de amostragem, dois em particular

possuem relevância neste trabalho: as amostragens máxima e média. As Figuras 2.3 e 2.4 ilustram, respectivamente, essas duas abordagens em um contexto cujos dados de entrada possuem apenas uma dimensão, a janela é de tamanho dois e o passo de tamanho um.

Figura 2.3: Exemplo de amostragem máxima de uma dimensão.

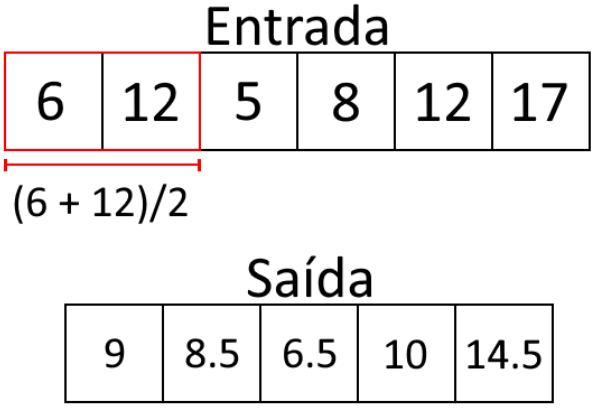

Figura 2.4: Exemplo de amostragem média de uma dimensão.

2.4.3 GRU e GARU

Gated Recurrent Unit (GRU) é um tipo de rede recorrente proposta inicialmente em [30] com o objetivo de solucionar o problema do desaparecimento do gradiente em sequências muito longas, característico das redes recorrentes tradicionais [27], através de equações, comumente chamadas de portões, que controlam o fluxo das informações na rede. A GRU pode ser definida da seguinte forma:

$$
\begin{aligned}
z_t &= \text{sigmoide}(W_z \cdot x_t + U_z \cdot y_{(t-1)} + b_z), \\
r_t &= \text{sigmoide}(W_r \cdot x_t + U_r \cdot y_{(t-1)} + b_r), \\
\bar{y}_t &= \tanh(W_y \cdot x_t + r_t \times U_y \cdot y_{(t-1)} + b_z), \\
y_t &= z_t \times y_{(t-1)} + (1 - z_t) \times \bar{y}_t,
\end{aligned}
\tag{2.3}
$$

em que z, r, x e y são, respectivamente, o portão de atualização, o portão de esquecimento, a entrada e a saída da GRU, W e U são matrizes de pesos, b é um vetor de viés, $(\cdot)$ se refere à multiplicação matricial e $(\times)$ à multiplicação elemento a elemento. O portão de atualização controla as informações que devem ser propagadas pela rede, podendo inclusive determinar que toda a informação deve ser propagada, o que por sua vez resolve o problema do desaparecimento do gradiente. Já o portão de esquecimento, como o nome sugere, controla quais informações a rede pode ignorar.

Diversas variações da GRU já foram propostas ao longo dos anos [31, 32], mas entre elas vale destacar a *Gated Auto Regressive Units* (GARU), que já demonstrou bons resultados em sistemas de verificação de assinaturas *online* [7]:

$$
\begin{aligned}
r_t &= \text{sigmoide}(W_r \cdot x_t + U_r \cdot y_{(t-1)} + b_r), \\
y_t &= \tanh(W_y \cdot x_t + U_y \cdot (r_t \times y_{t-1}) \times U_y).
\end{aligned}
\tag{2.4}
$$

A principal diferença entre a GRU e a GARU se encontra na ausência do portão de atualização na GARU, já que a intenção dos autores era que a rede sempre atualizasse os dados ao invés de realizar um controle dos mesmos através do portão de atualização. Em resumo, a GARU nada mais é que uma simplificação da GRU.

2.5 Aprendizagem com Poucas Instâncias

Aprendizagem com Poucas Instâncias — do inglês *Few Shot Learning* (FSL), é uma subárea de aprendizagem de máquina que tem por objetivo aprender novos conceitos a partir de poucos dados rotulados [33], situação recorrente no contexto biométrico uma vez que não é interessante para o usuário fornecer um número muito alto de amostras de impressões biométricas durante a etapa de cadastro. Esse objetivo é, no geral, atingido por meio de duas maneiras principais [34]:

1. Meta-aprendizagem: onde a ideia é que o modelo aprenda a partir de diferentes tarefas e conjuntos de dados a fim de que possa ser usado em um conjunto de dados diferente do qual utilizou para aprender.

2. Aprendizagem de métricas: onde a intenção é que o modelo aprenda uma métrica de similaridade (ou dissimilaridade) entre diferentes amostras.

Este trabalho faz uso de aprendizagem de métricas que é tipicamente alcançada através de uma função capaz de mapear instâncias similares para pontos em um espaço multidimensional próximos uns dos outros enquanto instâncias dissimilares são mapeadas em pontos distantes uns dos outros [35]. No geral, os modelos que trabalham com aprendizagem de métricas funcionam a partir de redes neuronais siamesas que consistem em redes gêmeas, isto é, cuja arquitetura é igual e seus pesos e parâmetros são compartilhados, que posteriormente são unidas por alguma função que computa uma métrica entre as características extraídas [36]. A ideia é que duas ou mais amostras passem simultaneamente pela rede e, em seguida, a métrica seja calculada a fim de determinar o grau de similaridade entre essas amostras.

2.5.1 Otimização em FSL

Os modelos de FSL que trabalham com aprendizagem de métricas são, no geral, otimizados com base em funções de perda que medem a similaridade (ou dissimilaridade) entre diferentes instâncias de amostras a fim de induzir a rede a mapear as instâncias pertencentes a uma mesma classe de maneira próxima ao mesmo tempo que força instâncias de classes diferentes a serem mapeadas de maneira distante umas das outras. A intuição que motiva essa abordagem é que instâncias de uma mesma classe são naturalmente mais semelhantes entre si que instâncias de classes diferentes. Dessa forma, instâncias de uma mesma classe terão distâncias menores entre si do que quando comparadas com instâncias de outras classes.

Quando o valor da função de perda é superior a zero, a rede, através de um algoritmo de otimização, como o Gradiente Descendente Estocástico — do inglês *Stochastic Gradient Descent* (SGD), que é o utilizado neste trabalho, irá atualizar seus parâmetros a fim de que na próxima iteração do treinamento o valor da função de perda diminua. No caso do SGD, por exemplo, essa atualização é feita a partir do cálculo do gradiente de cada um dos parâmetros com respeito a uma amostragem dos dados, que é então multiplicado pelo valor da taxa de aprendizagem, de modo que o resultado é então subtraído dos antigos valores dos parâmetros a fim de determinar os novos [25].

Cada iteração do algoritmo de otimização corresponde a um lote, uma amostra dos dados de treinamento, e a união de todos os lotes forma uma época, ou seja, uma época é composta por todo o conjunto de treinamento. Tendo em vista que a rede processa todos os elementos do lote simultaneamente, o tamanho de um lote é, no geral, muito menor que

o tamanho da época a fim de diminuir o custo computacional associado ao processamento desses dados [29].

A fórmula 2.5 descreve o cálculo do gradiente $\hat{g}$ com respeito a uma amostragem dos dados de tamanho m em que L é a função de perda que mede a performance da função f, que representa o modelo, configurada com os parâmetros θ com respeito à entrada x e saída ideal y. A fórmula 2.6, por sua vez, descreve a atualização dos parâmetros θ com respeito ao gradiente $\hat{g}$ e a taxa de aprendizagem ϵ.

$$\hat{g} \leftarrow \frac{1}{m}\nabla_\theta \sum_{i=1}^{m} L(f(x^i;\theta), y^i), \tag{2.5}$$

$$\theta \leftarrow \theta - \epsilon\hat{g}. \tag{2.6}$$

2.5.2 Funções de Perda para Aprendizagem de Métricas

Como abordado na seção 2.5.1, o principal objetivo das funções de perda em aprendizagem de métricas é induzir a rede a mapear instâncias de uma mesma classe de maneira próxima ao mesmo tempo que instâncias de classes diferentes são mapeadas de maneira distante umas das outras. Entre as principais funções de perda utilizadas em aprendizagem de métricas com esse objetivo, vale citar a *Contrastive Loss* e a *Triplet Loss*.

A *Contrastive Loss*, proposta inicialmente em [37], trabalha com pares de amostras e tem por objetivo mapear as amostras de pares positivos, isto é, aqueles que devem ser considerados da mesma classe (como sendo da mesma pessoa no contexto biométrico) de maneira próxima, enquanto empurra pares negativos, aqueles que pertencem a classes diferentes (pessoas diferentes no contexto biométrico), para que fiquem a uma distância mínima $m \in \mathbb{R}^+$ das amostras positivas [38]. A perda de um par a partir da *Contrastive Loss* pode ser definida da seguinte forma:

$$y\mathtt{d} + (1-y)\max(m - \mathtt{d}, 0), \tag{2.7}$$

em que y assume o valor 1 se ambos os elementos do par são da mesma classe e -1 caso contrário, $\mathtt{d}$ é o valor da distância entre os membros do par e $m \in \mathbb{R}^+$ é uma margem.

A *Triplet Loss*, proposta inicialmente em [39], em contrapartida, trabalha com três amostras sendo elas a âncora, uma amostra positiva que possui a mesma classe da âncora e uma amostra negativa que possui uma classe diferente da âncora. O objetivo da *Triplet Loss* é criar uma separação de, no mínimo, o valor de uma margem $m \in \mathbb{R}^+$ entre as distâncias da âncora para a amostra positiva e da âncora para a amostra negativa. A fórmula básica da *Triplet Loss* é a seguinte:

$$\max(0, d(a,p) + m - d(a,n)), \tag{2.8}$$

em que $m \in \mathbb{R}^+, a, p, n,$ e d se referem, respectivamente, à margem, âncora, amostra positiva, amostra negativa e uma medida de distância.

Como é possível observar, no caso da *Contrastive Loss* a função tenta forçar que as representações profundas de amostras positivas tenham distâncias próximas a zero, o que dependendo do problema, especialmente naqueles marcados por grandes variações intra-classe, pode ser algo extremamente difícil de ser alcançado e que possivelmente impedirá que o modelo convirja uma vez que a função de perda estará frequentemente indicando alterações drásticas nos parâmetros da rede. A *Triplet Loss*, por sua vez, busca forçar que as distâncias entre amostras positivas e a âncora sejam menores que as distâncias entre a âncora e as amostras negativas ao mesmo tempo que faz com que exista uma separação entre essas distâncias de pelo menos o valor da margem. Dessa forma, a *Triplet Loss* permite maiores variações intraclasse que a *Constrative Loss*, pois ela só se importa que exista uma separação e não com onde essa separação estará.

2.6 DTW e *Soft* DTW

O algoritmo DTW foi inicialmente proposto em [40] e tem como objetivo principal determinar o quão similares são duas sequências temporais e ao longo do tempo se firmou como uma das principais técnicas utilizadas em sistemas de verificação biométrica de assinaturas *online* [7, 8, 9, 10, 11, 17, 41].

Para isso, o algoritmo realiza diversas comparações entre pontos distintos nas duas sequências a fim de obter o alinhamento ótimo entre os dois sinais, admitindo inclusive que um ponto em alguma das sequências seja levado em conta mais de uma vez, o que permite que as sequências analisadas possuam tamanhos diferentes, o que difere de abordagens mais simples, como o cálculo de uma simples distância euclidiana entre duas sequências temporais. As Figuras 2.5 e 2.6 ilustram, respectivamente, um possível alinhamento obtido através do cálculo de uma distância euclidiana e do DTW entre duas sequências temporais.

Dadas duas sequências temporais A e B de tamanhos, respectivamente, $[1..m]$ e $[1..n]$ e seja M a matriz de tamanho [1..m, 1..n] utilizada para o cálculo do DTW onde a coordenada i, j se refere ao DTW entre o ponto i na sequência A e o ponto j na sequência B e seja d uma medida de distância entre dois pontos x e y quaisquer, a matriz M pode ser definida da seguinte forma:

$$M[i,j] = d(A[i], B[j]) + \min(X, Y, Z), \tag{2.9}$$

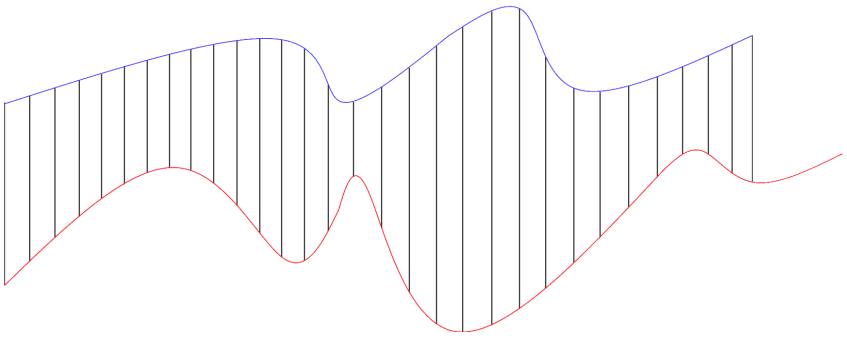

Figura 2.5: Distância Euclidiana.

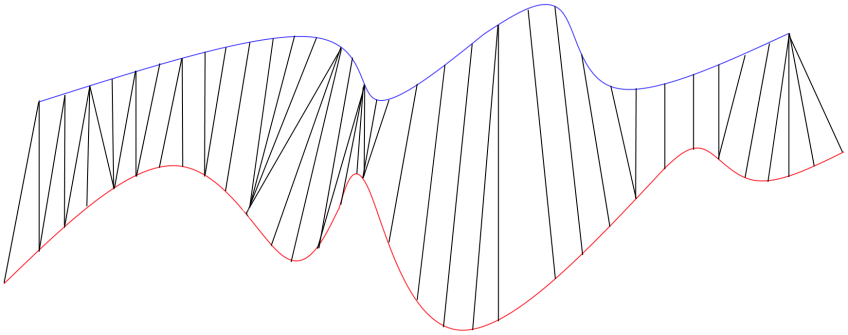

Figura 2.6: DTW.

onde:

$$\begin{aligned} X &= d(A[i-1], B[j]) \quad \text{se } (i-1) > 0, 0 \text{ do contrário,} \\ Y &= d(A[i], B[j-1]) \quad \text{se } (j-1) > 0, 0 \text{ do contrário,} \\ Z &= d(A[i-1], B[j-1]) \quad \text{se } (i-1) > 0 \text{ e } (j-1) > 0, 0 \text{ do contrário.} \end{aligned} \tag{2.10}$$

A distância total do DTW entre as sequências A e B será, então, o valor contido em $M[m, n]$. O ponto negativo do DTW, no entanto, se encontra no fato de a recursão do algoritmo dinâmico envolver o uso do operador não suave de mínimo, o que faz com que o gradiente ou subgradiente não fique bem definido [10], o que inviabiliza, por exemplo, o uso do DTW como função de perda.

É justamente daí que surge o *Soft* DTW, uma variação diferenciável do clássico DTW ao substituir o operador de mínimo tradicional por um operador de mínimo suave, que pode ser definido da seguinte forma [42]:

$$\min a_1, ..., a_n = \begin{cases} \min_{1,...,n} a_i, & \text{se } \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} \exp^{\frac{-a_i}{\gamma}}, & \text{se } \gamma > 0. \end{cases} \tag{2.11}$$

Note que ao usar $\gamma = 0$ o algoritmo se comporta exatamente da mesma maneira que o DTW.

2.7 Trabalhos Relacionados

Em [41] foi proposto um sistema de verificação de assinaturas *online* a partir do algoritmo clássico de aprendizagem de máquina dos K vizinhos mais próximos com a medida DTW. Dado um conjunto de assinaturas de referência e uma assinatura para análise, o algoritmo calcula o valor do DTW entre a assinatura em análise e cada uma das referências de modo que a que apresentar o menor valor é considerada como sendo a assinatura mais próxima. Em seguida, é feita a média aritmética dos valores da medida DTW entre essa assinatura mais próxima e cada uma de suas K assinaturas de referência mais próximas, que por sua vez é multiplicado por uma constante previamente definida. A ideia é que para que uma assinatura em análise seja considerada original, a distância entre ela e a assinatura de referência mais próxima seja menor que a distância média entre as K assinaturas mais próximas da assinatura de referência.

Em [17] é proposta uma espécie de extensão do algoritmo anterior ao trabalhar com o algoritmo dos JK vizinhos mais próximos ao invés de apenas os K vizinhos mais próximos. Aqui, inicialmente são selecionados os J vizinhos mais próximos de maneira semelhante a abordagem anterior. Em seguida, é feita a média das distâncias entre cada um desses J vizinhos e seus respectivos K vizinhos mais próximos. Caso a distância entre esse J-ésimo vizinho e a assinatura em análise seja inferior a distância entre o J-ésimo vizinho e seus respectivos K vizinhos mais próximos, é computado um voto para que a assinatura seja classificada como original. Caso a distância seja igual ou superior, é computado um voto para que a assinatura seja classificada como falsa. Uma vez que cada J-ésimo vizinho tem direito a um voto, a classificação final da assinatura é feita de acordo com o rótulo (original ou falsificação) mais votado.

O principal ponto negativo das abordagens anteriores é que elas no geral demandam uma grande quantidade de assinaturas de referência para que funcionem bem, além de exigir uma grande quantidade de execuções do algoritmo DTW entre as assinaturas de referência, o que pode torná-las impraticáveis.

Em [11] é proposto o algoritmo *Euclidean barycenter-based DTW barycenter averaging* (EB-DBA), que tem por objetivo calcular, utilizando as assinaturas de referência, um modelo que expressa uma espécie de assinatura média. Tendo em vista que as assinaturas

são compostas por uma sequência numérica que varia ao longo do tempo, ou seja, uma sequencial temporal, para que esse modelo seja expressivo é necessária uma reamostragem a fim de que as assinaturas passem a ter o mesmo tamanho e também que seja dada maior importância para os pontos mais semelhantes entre elas. Assim, para determinar se uma assinatura em análise é original ou não, basta realizar o cálculo do DTW entre a assinatura em análise e o modelo previamente calculado. Tendo em vista que o modelo pode ser obtido a qualquer momento e então armazenado, a inferência necessita apenas de uma execução do DTW. É importante ressaltar que os resultados obtidos nesse trabalho superam diversas abordagens que fazem uso de inteligência artificial de maneira geral.

Já no que diz respeito a abordagens que fazem de aprendizagem profunda, vale citar [43] onde é proposto o uso de uma rede convolucional de uma dimensão que atua como extratora de características seguida de uma rede recorrente que determina se uma assinatura é original ou falsificada.

Em [44] é proposta uma arquitetura composta por uma rede convolucional em conjunto com Memória de Curto e Longo Prazo — do inglês *Long Short Term Memory* (LSTM). A partir das coordenadas X e Y de cada uma das assinaturas é construída uma imagem que é dividida em diversos segmentos que são tratados sequencialmente pela rede, que é treinada com assinaturas originais e falsificações profissionais de diversos usuários. O vetor de características resultante passa então por um decodificador que busca reconstruir a imagem original a fim de calcular assim a função de perda do modelo a partir da diferença entre a imagem original e a reconstruída, o que incentiva que as características extraídas tenham forte relação com a assinatura original. A ideia é que isso gere um extrator de características capaz de distinguir entre uma assinatura original e uma falsificação sem depender de uma classe, ou seja, sem necessitar que o usuário esteja previamente cadastrado no sistema. A saída dessa rede é então utilizada por uma Máquina de Vetor de Suporte — do inglês *Suport Vector Machine* (SVM), previamente treinada para cada um dos usuários do sistema utilizando para isso assinaturas de referência fornecidas pelo usuário e assinaturas aleatórias de outros usuários que atuam como falsificações aleatórias, ou seja, simulando um cenário em que um usuário tenta adivinhar a assinatura de outro. O ponto negativo dessa abordagem é ignorar informações que poderiam contribuir positivamente para o sistema, como as informações relativas à maneira como a pressão exercida na superfície de captura variou ao longo do tempo.

Em [7] o problema de verificação de assinaturas online foi tratado como um problema de meta-aprendizagem em que cada cliente é considerado uma tarefa. Inicialmente é realizado uma espécie de pré-processamento da assinatura através do cálculo de um descritor que serve de entrada para uma GARU. Em seguida, uma camada é responsável por identificar o cliente e projetar as características extraídas no espaço de tarefas adequado. Por

fim, a dissimilaridade entre as características extraídas de duas assinaturas é calculada através do DTW.

Em [45] foi proposto um sistema de verificação de assinaturas online que pode ser treinado sem o uso de falsificações profissionais através da inserção de perturbações no sinal de uma assinatura de referência. Essas assinaturas com perturbações passam então por uma rede neuronal convolucional com *multi-head attention* e as características extraídas servem de base para um verificador que utiliza distância euclidiana para determinar a similaridade entre duas assinaturas.

Em [9] as sequências que descrevem as assinaturas passam pelo processo de alinhamento através do DTW e servem, posteriormente, como entrada para uma rede recorrente siamesa baseada em variações da GRU a fim de gerar uma métrica de dissimilaridade entre a assinatura de referência e a assinatura em análise a fim de permitir a classificação através de um limiar.

O sistema utilizado como referência neste trabalho também faz uso de aprendizagem profunda e é descrito com detalhes na próxima seção.

2.8 DsDTW

A arquitetura do sistema de verificação biométrica de assinaturas *online* utilizado como base neste trabalho é o DsDTW proposto em [10]. Esta seção irá apresentar os blocos básicos que o compõe.

Inicialmente as assinaturas, que são descritas por meio de sequências numéricas que indicam dados como a trajetória da caneta e pressão aplicada no dispositivo de captura, servem como entrada para uma rede Rede Adaptativa Recorrente Convolucional — do inglês *Convolutional Recurrent Adaptive Network* (CRAN), responsável por aprender representações profundas das assinaturas, isto é, extrair características que facilitem a discriminação entre diferentes assinaturas. Durante a etapa de treinamento as representações profundas geradas passam por uma amostragem tardia e é então calculada a função de perda com base no *Soft* DTW que indicará ao modelo quais modificações em seus pesos e parâmetros devem ser realizadas. Durante a etapa de teste, a saída da CRAN serve como entrada para um verificador baseado no DTW tradicional responsável por determinar se uma assinatura é original ou não. A Figura 2.7 ilustra a arquitetura geral e a Tabela 2.1 apresenta os detalhes do modelo DsDTW.

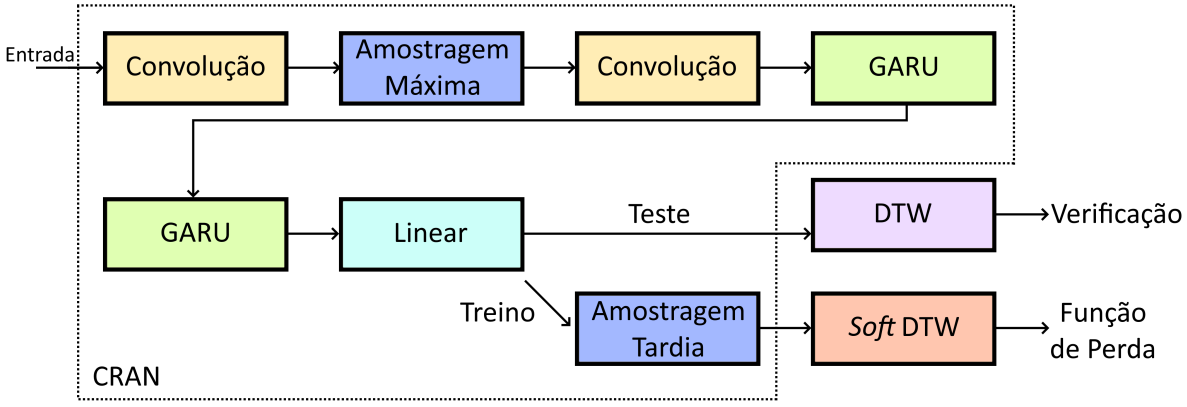

Figura 2.7: Arquitetura do modelo DsDTW.

Camada	Configuração
Convolução 1D	c: 64, k: 7, s: 1, p: 3
Amostragem Máxima	k: 2, s: 2
Convolução 1D	c: 128, k: 3, s: 1, p: 1
Dropout	prob: 0,1
Rede Recorrente	GARU (seção 2.4.3): 128
Dropout	prob: 0,1
Rede Recorrente	GARU (seção 2.4.3): 128
Linear	c: 64
Amostragem Média Tardia	k: 2, s: 2

Tabela 2.1: Arquitetura da CRAN.
Fonte: [10]

”c”, ”k”, ”s”, ”p” e ”prob” denotam, respectivamente, o número de canais, tamanho do núcleo (*kernel*), tamanho do passo (*stride*), tamanho do preenchimento (*padding)* e probabilidade. As convoluções são seguidas pela função de ativação ReLU.

2.8.1 Sequências de Entrada

Como será abordado com detalhes no Capítulo 4, as assinaturas realizadas com caneta *stylus* presentes no DeepSign e que foram utilizadas para o treinamento do modelo DsDTW são descritas, entre outras, pelas sequências temporais que descrevem a trajetória da caneta (coordenadas x e y) e pressão exercida na superfície do dispositivo de captura.

A partir desses dados, é necessário calcular as seguintes sequências temporais, conforme proposto em [10], que servem de entrada para o modelo:

1. Velocidade horizontal: v_x.
2. Velocidade vertical: v_y.
3. Magnitude da velocidade: $v = \sqrt{v_x^2 + v_y^2}$.
4. Ângulo do caminho da tangente: $\theta = \arctan(v_y/v_x)$.
5. $\cos(\theta)$.
6. $\sin(\theta)$.
7. Pressão: p.
8. Derivada discreta de v: $\dot{v}$.
9. Derivada discreta de θ: $\dot{\theta}$.
10. Logaritmo do raio da curvatura: $\rho = \log(v/\dot{\theta})$.
11. Aceleração centrípeta: $c = v \times \theta$.
12. Aceleração total: $a = \sqrt{\dot{v}^2 + c^2}$.

Todas as funções temporais são normalizadas para apresentarem média zero e variância um e as derivadas discretas dos sinais temporais $\dot{v}$ e $\dot{\theta}$ são calculadas a partir de uma regressão de segunda ordem da seguinte forma [46]:

$$\dot{f}(i) = \frac{\sum_{\epsilon=1}^{2} \epsilon(f(i+\epsilon) - f(i-\epsilon))}{2\sum_{\epsilon=1}^{2} \epsilon^2}. \tag{2.12}$$

Tendo em vista que normalmente as assinaturas dentro de um lote possuirão tamanhos diferentes, todas elas passam por um processo de preenchimento (*padding)*, ao adicionar zeros, que faz com que todas as assinaturas passem a ter o mesmo tamanho da maior assinatura dentro do lote a fim de que possam ser processadas pelo modelo conforme as configurações apresentadas na seção 2.8. Esse preenchimento é posteriormente ignorado pela GARU e no cálculo do DTW e *Soft* DTW.

2.8.2 Dissimilaridade Entre as Assinaturas

Enquanto a classificação do DsDTW é feita utilizando o DTW tradicional, a função de perda do modelo é calculada com base no *Soft* DTW. A fórmula (2.13) representa a dissimilaridade entre duas assinaturas X e Y utilizando o DTW e a (2.14) utilizando o *Soft* DTW:

$$\mathrm{dte}(X, Y) = \frac{\mathrm{dtw}(f(X), f(Y))}{|f(X)| + |f(Y)|}, \tag{2.13}$$

$$\mathrm{dtr}(X, Y) = \frac{\mathrm{dtw}_\gamma(\varphi(f(X)), \varphi(f(Y)))}{|\varphi(f(X))| + |\varphi(f(Y))|}, \tag{2.14}$$

onde $f(\cdot)$ denota a saída da rede CRAN, $\varphi(\cdot)$ a amostragem tardia, $|\cdot|$ o tamanho da assinatura, dtw é o DTW tradicional e dtw_γ é o *Soft* DTW com fator de suavização γ.

Note que a amostragem tardia é utilizada apenas durante o treinamento, uma vez que ela é capaz de não apenas diminuir o consumo de memória mas também melhorar os resultados quando aplicada dessa forma [10].

2.8.3 Verificador

O verificador utilizado no DsDTW tem como objetivo dar uma pontuação para a assinatura em análise baseado em um conjunto de assinaturas de referência e, caso essa pontuação seja inferior a um limiar, a assinatura é classificada como sendo original e falsificada no caso contrário. Não é necessário definir previamente o limiar uma vez que o objetivo é traçar as curvas das métricas FRR e FAR com relação a variação do limiar e assim determinar o ponto onde elas se intersectam, que é justamente o ponto onde ocorre o EER, principal métrica de avaliação em sistemas biométricos. Dessa forma, o limiar ótimo será exatamente aquele onde ocorre a intersecção das duas curvas e, consequentemente, o EER. As curvas mencionadas anteriormente serão definidas a partir do teste de todas as pontuações calculadas pelo verificador como sendo o limiar de modo que cada pontuação é fruto da análise de uma assinatura.

Assim, dado um conjunto de n assinaturas de referência do usuário k $\{X_1^k ... X_n^k\}$, é calculada a média dos valores do dte entre todos os pares possíveis como $\bar{d}_k$. Caso exista apenas uma assinatura de referência, d_k assume o valor de 1. Em seguida, dada uma assinatura Y que deve ser analisada, as seguintes pontuações devem ser calculadas e somadas a fim de determinar a pontuação final de uma assinatura:

$$s_{ave}^k(Y) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{dte}(X_i^k, Y) / \sqrt{\bar{d}_k}, \tag{2.15}$$

$$s_{min}^k(Y) = \min_{i \ldots n} \ \mathrm{dte}(X_i^k, Y) / \sqrt{\bar{d}_k}. \tag{2.16}$$

Dessa forma, dado um limiar th previamente definido, caso $s_{ave}^k(Y) + s_{min}^k(Y) < th$ a assinatura é classificada como sendo original, caso contrário é classificada como sendo uma falsificação.

2.8.4 Função de Perda

O DsDTW utiliza uma variação da função de perda *Triplet Loss*, abordada na seção 2.5.2, com o *Soft* DTW como medida de distância.

Dado um conjunto de n_e escritores, cada escritor possui um minilote composto por uma assinatura original que será utilizada como âncora, n_p assinaturas originais que funcionarão como amostras positivas e n_n falsificações, que podem ser profissionais ou não, que funcionarão como amostras negativas. Dessa forma, para cada escritor existem $n_p \times n_n$ triplas possíveis, mas apenas as que a perda da tripla resultar em um número maior que zero serão levadas em conta no valor final da função de perda.

A perda de uma tripla $(X_a^k,\, X_{p,i}^k,\, X_{n,j}^k)$ que se refere ao k-ésimo escritor ($k \in [1, n_e]$) com relação à sua respectiva âncora, i-ésima amostra positiva e j-ésima amostra negativa é definida da seguinte maneira [10]:

$$l(X_a^k, X_{p,i}^k, X_{n,j}^k) = \max(0, \mathrm{dtr}(X_a^k, X_{p,i}^k) + m - \mathrm{dtr}(X_a^k, X_{n,j}^k)), \tag{2.17}$$

em que $m \in \mathbb{R}^+$ é uma margem e dtr é definido conforme apresentado na seção 2.8.2. A perda de um minilote é dada da seguinte forma:

$$L_k = \frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_n} l(X_a^k, X_{p,i}^k, X_{n,j}^k)}{1 + \sum_{i=1}^{n_p} \sum_{j=1}^{n_n} \mathbb{I}(l(X_a^k, X_{p,i}^k, X_{n,j}^k) > 0)}, \tag{2.18}$$

em que $\mathbb{I}$ é uma função indicadora: caso a condição seja satisfeita, a saída é 1, caso contrário, a saída é 0. Por fim, a perda total de um lote composto por k minilotes, associados a um único escritor cada, é definida conforme abaixo:

$$L = \frac{1}{n_e} \sum_{k=1}^{n_e} (L_k + \frac{\lambda}{n_p} \sum_{i=1}^{n_p} \mathrm{dtr}(X_a^k, X_{p,i}^k)), \tag{2.19}$$

em que λ controla a força do termo que representa a variação dentro de uma mesma classe, isto é, a variação existente entre as assinaturas originais de um mesmo usuário.

Capítulo 3

Metodologia

Este capítulo descreve, em detalhes, o problema atacado neste trabalho e as modificações propostas em cima do modelo DsDTW a fim de diminuir os impactos causados por ele.

3.1 O Problema do Limiar Global

O modelo DsDTW [10] tem como objetivo criar representações profundas das funções temporais que caracterizam uma assinatura de modo a aumentar o poder de discriminação do DTW para permitir uma boa classificação das assinaturas em meio a versões originais e falsificações.

Naturalmente, assinaturas idênticas apresentarão pontuação, conforme descrito na seção 2.8.3, igual a zero, enquanto assinaturas semelhantes, geralmente diferentes assinaturas originais de um mesmo usuário, apresentarão pontuações baixas, isto é, relativamente próximas a zero. Da mesma forma, é esperado que as pontuações de falsificações aleatórias (assinaturas originais de outros usuários) sejam bem maiores que as pontuações de assinaturas originais. De maneira semelhante, também é de se esperar que as pontuações de falsificações profissionais fiquem em algum lugar entre os dois cenários anteriores.

Como brevemente apresentado na seção 1, ao considerar um único usuário o DsDTW é comumente capaz de criar representações profundas cujas pontuações de assinaturas originais e falsificações possam ser perfeitamente separadas a partir de um limiar específico, isto é, existe um valor tal que todas as pontuações de assinaturas originais são menores que ele e todas as falsificações são maiores ou iguais a ele. O reflexo disso é que o EER do sistema considerando este único usuário com este limiar é zero. Apenas para se ter uma ideia, dos 442 usuários presentes no conjunto de testes de assinaturas realizadas com caneta *stylus* do DeepSign, o modelo DsDTW aqui replicado para servir de base para as alterações propostas a partir da próxima seção foi capaz de encontrar uma separação perfeita considerando limiar específico por usuário em 397 deles no cenário em que as

falsificações foram realizadas por profissionais e quatro assinaturas de referência estavam disponíveis.

Contudo, deve-se ter em mente que o objetivo do DsDTW é operar com um limiar global comum a todos os usuários e, ao analisar esses limiares ótimos e perfeitos por usuário foi constatado que eles tendem a ser consideravelmente diferentes uns dos outros de modo que o limiar ótimo de um usuário que realiza uma separação perfeita e atinge EER de 0% quando considerado este usuário, pode resultar em um EER de 100% para outro usuário, mesmo que este possua um limiar ótimo que também realize uma separação perfeita e atinja EER de 0%. A Figura 3.1 ilustra essa situação.

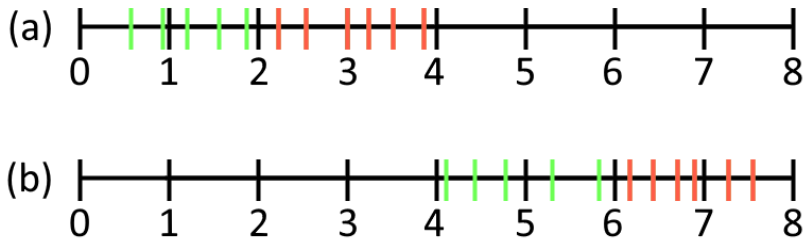

Figura 3.1: Disposição das pontuações de assinaturas originais (em verde) e de falsificações (em vermelho) para os usuários (a) e (b).
Note que na situação (a) o uso do limiar 2 e que na situação (b) o uso do limiar 6 criam uma separação perfeita entre originais e falsificações, o que caracteriza um EER com limiar local igual a 0% nos dois casos. Contudo, utilizar 2 como limiar no caso (b) faria com que todas as assinaturas fossem classificadas como falsificações, enquanto utilizar 6 no caso (a) faria com que todas as assinaturas fossem classificadas como originais. Embora exista um limiar ótimo perfeito em cada situação, não existe um limiar em comum que permita uma boa separação nos dois casos.

A escolha de um limiar único para todos os usuários no sistema é importante pois evita a necessidade de que, durante a etapa de cadastro, sejam fornecidas falsificações profissionais para cada um dos usuários a fim de determinar um bom limiar local para cada um deles, o que dificultaria muito colocar o sistema em prática. Dessa forma, ao obter um limiar que funcione bem no conjunto de testes, a tendência é que esse limiar também funcione bem em produção.

Além disso, é preciso ter em mente que o DsDTW opera em produção com usuários que ele não teve acesso durante o treinamento, o que o torna genérico e dispensa a necessidade de, por exemplo, a realização de um ajuste fino sempre que for necessário analisar assinaturas de um novo usuário. Tudo isso dificulta o uso de um limiar específico por usuário. Por isso, o objetivo deste trabalho é fazer com que o DsDTW gere representações profundas cujos limiares ótimos específicos por usuário sejam mais semelhantes entre

si a fim de diminuir os impactos do problema exposto. Em termos estatísticos: o objetivo deste trabalho é modificar o DsDTW com o intuito de fazer com que a dispersão dos limiares ótimos locais referentes a cada um dos usuários presentes no conjunto de testes do DeepSign diminua. A expectativa é que, ao tornar os limiares ótimos locais mais semelhantes, o limiar ótimo global se torne mais expressivo e o sistema apresente menor EER quando comparado ao sistema base.

3.1.1 A Origem do Problema

O problema exposto está intimamente ligado com a função de perda utilizada no DsDTW e abordada na seção 2.8.4, pois a *Triplet Loss* tem por objetivo, a partir de uma assinatura original que funciona como âncora, aproximar as amostras positivas (assinaturas originais) enquanto afasta as amostras negativas (falsificações) a pelo menos o valor de uma margem positiva. A única restrição é de que a distância entre a âncora e a amostra positiva seja menor que a distância entre a âncora e a amostra negativa. A Figura 3.2 ilustra o processo de otimização obtido a partir da *Triplet Loss*. Portanto, a magnitude das distâncias não é levada em conta, o que permite situações como as apresentadas na Figura 3.1 de modo que as distâncias entre assinaturas originais de um usuário possam assumir valores maiores que as distâncias entre assinaturas originais e falsificações de outros usuários.

Tudo isso dificulta a escolha de um limiar global e levanta o questionamento de se a *Triplet Loss* é de fato adequada ou não para o problema de verificação de assinaturas *online*. O ponto, no entanto, é que esse comportamento apresentado pela *Triplet Loss* na realidade é desejado.

A grande vantagem da *Triplet Loss*, conforme abordado na seção 2.5.2, é ser robusta a variações intraclasse. Como previamente discutido na seção 2.2 é natural, especialmente por se tratar de uma biometria comportamental, que mesmo entre assinaturas originais de um mesmo usuário existam grandes variações, especialmente quando comparados usuários habituados em realizar assinaturas com usuários que possuem pouca prática. Dessa forma, caso o sistema não tolere variações intraclasse, o modelo não será capaz de encontrar um bom limiar mesmo considerando apenas um único usuário, isto é, ele não será capaz de determinar com precisão quais assinaturas são originais e quais são falsificadas nem mesmo utilizando um limiar específico por usuário.

Em resumo, ao mesmo tempo que é importante que o modelo aprenda a tolerar variações intraclasse para permitir uma boa separação entre assinaturas falsificadas e originais, também é importante que a distância entre duas assinaturas de um usuário seja semelhante à distância entre duas assinaturas de outro usuário qualquer, pois assim será possível escolher um limiar global que realize uma boa separação para todos os usuários. Note que, ao usar a *Triplet Loss*, não é necessário colocar restrições com relação à semelhança das

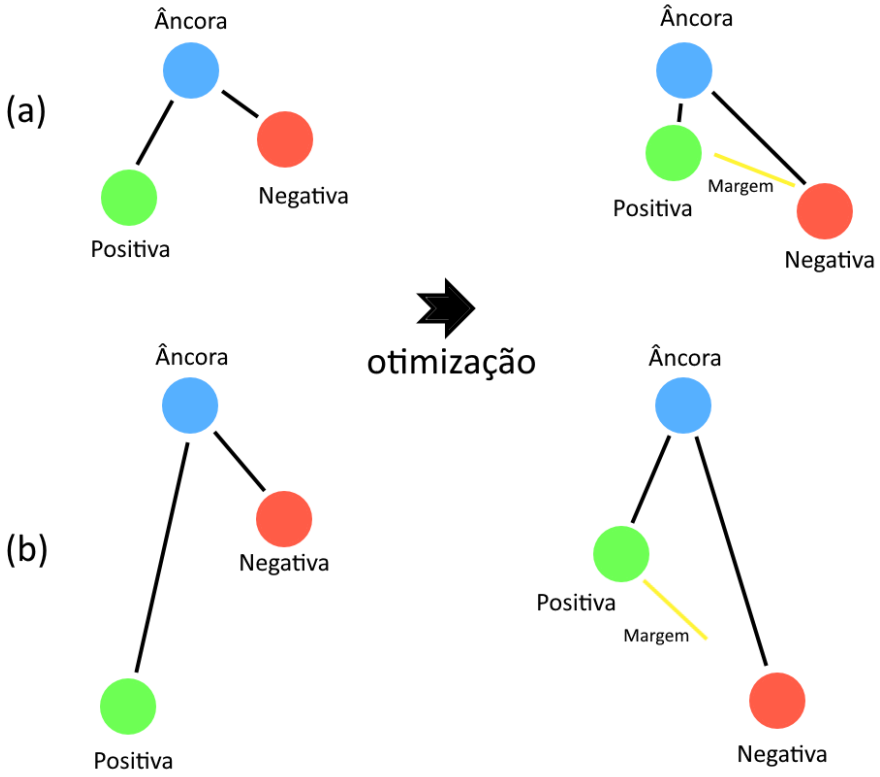

Figura 3.2: Processo de otimização da *Triplet Loss*.
Observe que, após a otimização, a distância entre a âncora e a amostra negativa no caso (a) é muito menor que a mesma distância no caso (b). O mesmo acontece em relação à distância entre a âncora e a amostra negativa. Note ainda que entre a amostra positiva e a negativa existe uma distância de, no mínimo, o valor da margem em amarelo após a otimização em ambos os casos

distâncias de falsificações, pois a própria *Triplet Loss* já busca garantir que as distâncias entre a âncora e assinaturas originais sejam sempre menores que as distâncias entre a âncora e falsificações.

3.2 Mudanças nos Sinais de Entrada

As funções temporais descritas na seção 2.8.1, com exceção da que se refere à pressão, são todas definidas com base nas velocidades horizontais e verticais da assinatura *online*, o que tende a gerar forte correlação entre essas velocidades e as demais funções temporais. Por isso, propomos a substituição da velocidade horizontal e vertical pelas coordenadas x e y normalizadas com relação ao centroide da seguinte forma:

$$\hat{x} = \frac{(x_i - x_g)}{(x_{max} - x_{min})}, \tag{3.1}$$

$$\hat{y} = \frac{(y_i - y_g)}{(y_{max} - y_{min})}, \tag{3.2}$$

em que x_i e y_i denotam o i-ésimo elemento, x_g e y_g o centroide, x_{max} e y_{max} os pontos de máximo e x_{min} e y_{min} os pontos de mínimo de cada uma das funções temporais. De acordo com os testes realizados, essa mudança reduziu um pouco da correlação uma vez que agora é utilizado o sinal puro e não a derivada discreta do dado, embora inegavelmente ainda exista uma correlação entre essas funções temporais e as outras nove que são calculadas com as velocidades vertical e horizontal da assinatura *online*. Os experimentos também revelaram que essa mudança não apenas fornece mais informações ao modelo e aumenta o poder de discriminação do mesmo mas também ajuda na melhora dos resultados ao contribuir para mitigar o problema do limiar global.

3.3 Função de Perda Proposta

A fim de diminuir o impacto do problema apresentado na seção 3.1, esta seção apresenta uma nova função de perda e detalha as várias partes que a compõe.

3.3.1 Máxima Discrepância Média

Máxima Discrepância Média — do inglês *Maximum Mean Discrepancy* (MMD), é uma medida distância entre distribuições [47] proposta inicialmente em [48] e que ganhou destaque no problema de adaptação de domínio [49, 50], que tem por objetivo aproximar a distribuição dos dados pertencentes a um domínio fonte a distribuição dos dados pertencentes a um domínio alvo [51].

Seja $\mathbb{E}_{X\sim P}[(\varphi(X))]$ e $\mathbb{E}_{Y\sim Q}[\varphi(Y)]$ a esperança de $\varphi(X)$ e $\varphi(Y)$ com relação às distribuições P e Q sobre um conjunto χ onde $X \sim P$ quer dizer que X tem distribuição P e $Y \sim Q$ que Y tem distribuição Q, a MMD pode ser definida da seguinte forma:

$$\text{MMD(X, Y)} = \|\mathbb{E}_{X\sim P}[\varphi(X)] - \mathbb{E}_{Y\sim Q}[\varphi(Y)]\|_{\chi}. \tag{3.3}$$

A depender da função φ escolhida, a MMD apresentará comportamentos diferentes. O uso de $\varphi(x) = x$ com $\chi = \mathbb{R}^d$, por exemplo, resulta em:

$$\begin{aligned} \text{MMD(X, Y)} &= \|\mathbb{E}_{X\sim P}[\varphi(X)] - \mathbb{E}_{Y\sim Q}[\varphi(Y)]\|_{\mathbb{R}^d}, \\ &= \|\mathbb{E}_{X\sim P}[X] - \mathbb{E}_{Y\sim Q}[Y]\|_{\mathbb{R}^d}, \\ &= \|\mu_P - \mu_Q\|_{\mathbb{R}^d}. \end{aligned} \tag{3.4}$$

Assim, a MMD entre X e Y será zero quando as suas médias forem iguais. O problema dessa abordagem, no entanto, é que ela não leva em conta fatores como, por exemplo, a variância das duas distribuições. Assim, a fim de garantir que a MMD só será zero quando as duas distribuições forem exatamente idênticas, uma possibilidade é aplicar um truque de núcleo utilizando a função $\varphi \to \mathcal{H}$ em que $\mathcal{H}$ é um *Reproducing Kernel Hilbert Space* e, admitindo $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$, definir a MMD como:

$$\begin{aligned} \text{MMD}^2(\text{X, Y}) &= \|\mathbb{E}_{X\sim P}[\varphi(X)] - \mathbb{E}_{Y\sim Q}[\varphi(Y)]\|^2_{\mathcal{H}}, \\ &= \langle \mathbb{E}_{X\sim P}[\varphi(X)], \mathbb{E}_{X\sim P}[\varphi(X)] \rangle_{\mathcal{H}} + \langle \mathbb{E}_{Y\sim Q}[\varphi(Y)], \mathbb{E}_{Y\sim Q}[\varphi(Y)] \rangle_{\mathcal{H}} \\ &\qquad - 2\langle \mathbb{E}_{X\sim P}[\varphi(X)], \mathbb{E}_{Y\sim Q}[\varphi(Y)] \rangle_{\mathcal{H}}, \\ &= \mathbb{E}_{X,X\sim P} k(X, X) + \mathbb{E}_{Y,Y\sim Q} k(Y, Y) - 2\mathbb{E}_{X\sim P, Y\sim Q} k(X, Y). \end{aligned} \tag{3.5}$$

Um núcleo que satisfaz as propriedades desejadas e é o utilizado neste trabalho é o gaussiano, definido da seguinte forma [48]:

$$k(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) = \exp\left(\frac{-1}{\sigma^2}[x_i^\intercal x_i - 2x_i^\intercal x_j + x_j^\intercal x_j]\right). \tag{3.6}$$

Assim, a MMD funciona como uma medida de discrepância de modo que quanto mais próximo de zero é o valor, mais semelhantes são as distribuições.

Ao fazer com que o modelo gere representações profundas que compartilhem de uma mesma distribuição, a tendência é que isso influencie o modelo a mapear as assinaturas de maneira mais consistente entre diferentes usuários, o que se espera que aumente o poder de discriminação de um limiar ótimo global.

3.3.2 A Ponderação das Triplas

O papel da margem presente na Fórmula 2.17 é buscar que as distâncias entre a âncora e a amostra positiva estejam a, no mínimo, o valor da margem da distância entre a âncora e a amostra negativa, que por sua vez pode ser uma falsificação profissional ou aleatória, isto é, uma assinatura original de outro usuário.

Idealmente a margem deve assumir um valor tão grande quanto possível, pois isso fará com que o modelo treinado seja capaz de distinguir de maneira eficiente se uma assinatura é original ou não. Contudo, a tendência é que um valor de margem muito grande indique alterações muito drásticas para o modelo, o que provavelmente dificultará a convergência do mesmo. Outra situação que pode indicar alterações muito drásticas se encontra na presença de pontos fora da curva, especialmente os oriundos de falsificações aleatórias.

Um fato que foi observado durante os experimentos é que durante o processo de aprendizagem, especialmente nas épocas iniciais, por vezes acontecia de uma tripla com falsificação aleatória, que correspondia a um ponto fora da curva com relação às demais triplas com falsificações aleatórias, indicasse uma alteração muito drástica no modelo por não respeitar a margem, pois as representações profundas geradas estavam muito semelhantes, fato que prejudicava a convergência do modelo.

Justamente por isso, é proposta a seguinte modificação na fórmula 2.17:

$$l(X_a^k, X_{p,i}^k, X_{n,j}^k) = \begin{cases} \max(0, \mathrm{dtr}(X_a^k, X_{p,i}^k) + m - \mathrm{dtr}(X_a^k, X_{n,j}^k))(\alpha), \\ \qquad \text{se } X_{n,j}^k \text{ é uma falsificação profissional,} \\ \max(0, \mathrm{dtr}(X_a^k, X_{p,i}^k) + m - \mathrm{dtr}(X_a^k, X_{n,j}^k))(1-\alpha), \\ \qquad \text{se } X_{n,j}^k \text{ é uma falsificação aleatória,} \end{cases} \tag{3.7}$$

em que $\alpha \in [0, 1]$ é uma constante que controla a influência das triplas com falsificações profissionais e aleatórias. Observe que isso só é possível pois o sistema, em tempo de treinamento, tem acesso as informações que determinam se uma assinatura é original ou se é uma falsificação profissional ou aleatória e o objetivo é dar maior importância para falsificações profissionais que para falsificações aleatórias.

3.3.3 A Variância Entre as Distâncias

Como abordado na seção 3.1, ao fazer com que as pontuações de assinaturas originais de um usuário sejam semelhantes as pontuações de assinaturas originais de outros usuários, uma vez que a *Triplet Loss* busca garantir que a distância entre a âncora e a amostra positiva seja menor que a distância entre a âncora e a amostra negativa, o comportamento final do modelo será que as pontuações de assinaturas originais fiquem semelhantes, independentemente do usuário, o que facilita o estabelecimento de um bom limiar global. Dessa forma, também é proposto o uso da variância das distâncias (utilizando o *Soft* DTW) de assinaturas originais dos usuários presentes em um lote na função de perda.

3.3.4 A Fórmula Final

Diante do exposto nas seções anteriores e levando em conta que a MMD trabalha com duas distribuições, a função de perda proposta neste trabalho funciona com lotes compostos por dois minilotes. Seja (X_a^k, $X_{p,i}^k$, $X_{n,j}^k$) a tripla formada pela âncora, i-ésima amostra positiva e j-ésima amostra negativa do escritor k, a perda de uma tripla pode ser definida da seguinte forma:

$$l(X_a^k, X_{p,i}^k, X_{n,j}^k) = \begin{cases} \max(0, \mathrm{dtr}(X_a^k, X_{p,i}^k) + m - \mathrm{dtr}(X_a^k, X_{n,j}^k)) * \alpha, \\ \qquad \text{se } X_{n,j}^k \text{ é uma falsificação profissional,} \\ \max(0, \mathrm{dtr}(X_a^k, X_{p,i}^k) + m - \mathrm{dtr}(X_a^k, X_{n,j}^k)) * (1-\alpha), \\ \qquad \text{se } X_{n,j}^k \text{ é uma falsificação aleatória,} \end{cases} \tag{3.8}$$

em que $\alpha \in [0, 1]$ é uma constante que controla a influência das triplas com falsificações profissionais e aleatórias, $m \in \mathbb{R}^+$ é uma margem e dtr é definido na Fórmula 2.14. A perda de um escritor considerando n_n amostras negativas e n_p amostras positivas é definida da seguinte forma:

$$L_k = \frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_n} l(X_a^k, X_{p,i}^k, X_{n,j}^k)}{1 + \sum_{i=1}^{n_p} \sum_{j=1}^{n_n} \mathbb{I}(l(X_a^k, X_{p,i}^k, X_{n,j}^k) > 0)}, \tag{3.9}$$

em que $\mathbb{I}$ é uma função indicadora: caso a condição seja satisfeita, a saída é 1, caso contrário, a saída é 0. A perda da *Triplet Loss* com relação a um lote é definida da seguinte forma:

$$L = \frac{1}{2} \sum_{i=1}^{2} (L_k + \frac{\lambda}{n_p} \sum_{i=1}^{n_p} \mathrm{dtr}(X_a^k, X_{p,i}^k)), \tag{3.10}$$

em que λ controla a força do termo que representa a variação existente entre as assinaturas originais de um mesmo usuário. Por fim, a função de perda proposta é definida da seguinte forma:

$$T = L * p + \mathrm{MMD}(S_1, S_2) * q + \sigma^2(g_1 \parallel g_2) * r \tag{3.11}$$

em que S_k corresponde as assinaturas dentro de um minilote, excetuando as falsificações aleatórias, ou seja: assinaturas originais (incluindo a âncora) e falsificações profissionais, g_k corresponde as distâncias entre a âncora e cada uma das amostras positivas do usuário k, $(\parallel)$ é o operador de concatenação, σ^2 é o operador de variância e $p, q, r \in \mathbb{R}^+$ são constantes que controlam o impacto de cada uma das parcelas na função de perda total.

O motivo de a MMD ignorar as falsificações aleatórias é evitar que uma mesma assinatura contribua mais de uma vez na otimização sendo uma delas como amostra original e outras como amostras negativas (falsificações aleatórias).

Capítulo 4

Experimentos e Discussão

Este capítulo descreve o conjunto de dados utilizado neste trabalho e seu rígido protocolo de avaliação que busca viabilizar uma comparação justa entre diferentes trabalhos. Também são apresentados os diversos testes realizados neste trabalho a fim de analisar a eficácia das mudanças propostas em relação ao modelo de referência, ao modelo replicado e ao estado da arte.

4.1 Conjunto de Dados

Um dos maiores problemas relacionados ao uso de abordagens que envolvam redes neuronais profundas para verificação de assinaturas *online* se encontra na falta de conjunto de dados grandes o suficiente para tornar isso viável, principalmente pelo fato de assinaturas serem dados sensíveis, o que não apenas torna mais burocrático o processo de aquisição das assinaturas mas também a disponibilização para pesquisadores.

Neste projeto, foi utilizado o conjunto de dados DeepSign [9], que consiste na união dos conjuntos MCYT [52], BiosecurID [53], Biosecure DS2 [54] e e-BioSign DS1, além de um novo conjunto chamado e-BioSign DS2. Justamente por isso, o DeepSign é hoje o maior conjunto de assinaturas *online* disponível, totalizando 1526 usuários e mais de 70 mil assinaturas.

Além disso, o DeepSign possui dois cenários operacionais distintos: um em que as assinaturas foram realizadas a partir de uma caneta *stylus* e outro onde foram realizadas utilizando um dedo. No entanto, apenas 4088 das assinaturas presentes foram realizadas com dedo, o que pode tornar inviável algumas abordagens que envolvam um treinamento do zero utilizando apenas essas assinaturas.

Outro ponto fundamental para permitir a construção de sistemas robustos de verificação biométrica de assinaturas *online* é a presença de falsificações profissionais, que podem ocorrer de duas maneiras:

1. Falsificações estáticas: em que o falsificador teve acesso a uma imagem da assinatura original para se basear.

2. Falsificações dinâmicas: em que o falsificador além da imagem da assinatura resultante também teve acesso a um vídeo da realização da assinatura original.

Além disso, no contexto do DeepSign uma falsificação é dita aleatória se ela corresponde a uma assinatura original de algum usuário diferente da qual ela está sendo comparada. Falsificações aleatórias são utilizadas para simular o cenário onde o impostor busca adivinhar como é uma assinatura.

Por fim, tendo em vista a variação natural entre assinaturas de um mesmo usuário é interessante que as assinaturas sejam coletadas em dias diferentes e de preferência com um intervalo de algumas semanas entre as sessões de captura, prática que é adotada na maioria dos conjuntos de dados, incluindo os presentes no DeepSign. A Tabela 4.1 resume as principais características dos conjuntos de dados presentes no DeepSign.

Conjunto de Dados	**nº de assinaturas originais por usuário**	**nº de falsificações profissionais por usuário**	**nº de usuários**	**nº de sessões de captura**	**Dispositivo de captura**	**Tipo de falsificação**
MCYT	25	25	330	5	Wacom Intuos A6 USB	Estáticas
BiosecurID	16	12	400	4	Wacom Intuos 3	Estáticas e dinâmicas
Biosecure DS2	30	20	650	2	Wacom Intuos 3	Dinâmicas
e-BioSign DS1	8	6	65	2	Wacom STU-500 STU-530 DTU-1031 Samsung ATIV7 Galaxy Note 10,1	Estáticas e dinâmicas
e-BioSign DS2	8	6	81	2	Wacom STU-530 Samsung Galaxy Note 10.1 Samsung Galaxy S3	Dinâmicas

Tabela 4.1: Principais características dos conjuntos de dados presentes no DeepSign. Fonte: [9]

Os sinais capturados em cada um dos conjuntos de dados presentes no DeepSign são os seguintes:

1. MCYT: coordenadas espaciais X e Y, pressão, ângulos da caneta (azimute, ângulo de altitude) e carimbos de tempo. Todas as assinaturas foram realizadas com caneta *stylus*.

2. BiosecurID e Biosecure DS2: coordenadas espaciais X e Y, pressão, ângulos da caneta (azimute, ângulo de altitude), carimbos de tempo e toque na tela (se a caneta estava tocando ou não a tela naquele momento). Todas as assinaturas foram realizadas com caneta *stylus*.

3. e-BioSign DS1 e e-BioSign DS2: coordenadas espaciais X e Y, pressão, carimbos de tempo e toque na tela para as assinaturas capturadas com *stylus*. As informações

presentes nas assinaturas capturadas com o dedo são as mesmas, com exceção da pressão e toque na tela que não estão presentes.

Cada assinatura é salva como uma sequência temporal em um arquivo de texto em que a primeira linha corresponde a um número N de instantes capturados naquela assinatura e as demais N linhas se referem, cada uma, a um instante diferente de captura de modo que, a depender da ordem própria de cada conjunto de dados, os dados (coordenadas X e Y, pressão, etc.), são dispostos como números inteiros separados por espaço.

4.2 Protocolo de Avaliação

A fim de viabilizar uma comparação justa entre diferentes sistemas de verificação, o DeepSign conta com um protocolo de avaliação bastante rígido e bem definido.

O DeepSign é, por padrão, dividido em subconjuntos para treinamento e para teste, sendo que 1084 usuários estão presentes no conjunto de treinamento e os demais 442 no conjunto de teste no cenário com caneta *stylus*. No cenário com assinaturas realizadas com dedo existem 75 usuários no conjunto de treinamento e 69 no conjunto de teste. É importante ressaltar que os usuários presentes no conjunto de teste são diferentes dos presentes no conjunto de treino para melhor avaliar a capacidade de generalização do sistema.

Além disso, as comparações realizadas para o cálculo do EER, principal métrica na avaliação de sistemas biométricos, são pré-estabelecidas de acordo com os seguintes cenários:

1. $4vs1$ ou $1vs1$: que diz respeito à quantidade de assinaturas de referência que o sistema tem à disposição. No cenário $4vs1$ são disponibilizadas quatro assinaturas originais que o modelo utilizará como referência para determinar se outra assinatura (diferente das quatro de referência) é original ou não. O cenário $1vs1$ é similar, porém o modelo só tem à disposição uma assinatura para utilizar como referência.
2. Falsificações profissionais ou aleatórias: que diz respeito se as comparações que o sistema deveria prever como sendo falsificações são profissionais ou aleatórias, conforme definido na seção 4.1.
3. Assinaturas capturadas com *stylus* ou dedo: que diz respeito à maneira como a assinatura foi capturada.

Assim, o protocolo de avaliação do DeepSign é baseado em um arquivo que estabelece quais assinaturas de quais usuários devem ser comparadas umas com as outras com base nos cenários descritos acima, o que permite uma comparação justa entre os resultados obtidos por diferentes sistemas de verificação.

4.3 Geração de Épocas e Lotes

Tendo em vista que a rede utilizada faz uso de triplas e a função de perda leva em conta tanto assinaturas originais, quanto falsificações profissionais e aleatórias, cada lote é composto por minilotes que seguem a mesma estrutura utilizada pelo DsDTW original [10]:

1. Uma âncora: assinatura original que será utilizada como referência para o cálculo da função de perda.
2. Cinco amostras positivas: cinco assinaturas originais que pertencem ao mesmo usuário da âncora escolhida.
3. Dez amostras negativas: sendo cinco falsificações profissionais de assinaturas pertencentes ao usuário da âncora e cinco assinaturas originais de usuários diferentes do usuário da âncora, que representam as falsificações aleatórias.

Ou seja, cada minilote exige, necessariamente, seis assinaturas originais (cinco positivas e a âncora) e cinco falsificações profissionais, fatores que impedem o uso de todas as assinaturas presentes para treinamento sem realizar repetições. É o que acontece, por exemplo, no conjunto de dados MCYT que, para cada usuário, existem 25 assinaturas originais e 25 falsificações profissionais. As 25 assinaturas originais permitem a criação de quatro minilotes sem que haja a repetição de assinaturas, o que totaliza o uso de 24 assinaturas originais e 20 falsificações profissionais. Assim, cinco falsificações profissionais e uma assinatura original acabam por não serem utilizadas na época para cada usuário do conjunto de dados MCYT, situação que ocorre em todos os demais conjunto de dados de acordo com o número de assinaturas originais e falsificações profissionais existentes. Note que, quando se trata da escolha de falsificações aleatórias, o número de assinaturas presentes no conjunto de dados não é um problema tendo em vista que elas consistem em qualquer assinatura original de algum usuário diferente do usuário da âncora.

Contudo, cada minilote é gerado de maneira totalmente aleatória seguindo as restrições referentes às escolhas da âncora e das amostras positivas e negativas, o que acaba por permitir que assinaturas que não foram utilizadas em uma época possam ser utilizadas em outra época. É importante destacar que os critérios adotados para geração de minilotes não foram apresentados no artigo de referência e que, portanto, podem ser diferentes dos critérios aqui utilizados.

4.4 Resultados e Discussão

Com o intuito de permitir uma comparação justa dos impactos das mudanças propostas neste trabalho, as próximas seções além de apresentarem os resultados obtidos pelo modelo de referência [10] também apresentam os resultados aqui obtidos na intenção de recriar o modelo original e os resultados obtidos após a implementação das alterações propostas em cima do modelo replicado. Além disso, o modelo original realiza o treinamento utilizando apenas assinaturas realizadas com caneta *stylus*, embora avalie o sistema utilizando, também, o conjunto de testes de assinaturas realizadas com dedo. Por isso, essa metodologia também é a adotada neste trabalho. No entanto, uma vez que as assinaturas realizadas com dedo não possuem informação referente à pressão, a prática adota no trabalho de referência e também neste trabalho é assumir que ela é constante em 1. Contudo, a seção 4.5 apresenta um processo de ajuste fino utilizando assinaturas realizadas com dedo e reporta os resultados alcançados.

Conforme será apresentado nas próximas subseções os resultados obtidos pelo modelo construído na intenção de replicar o modelo de referência não foram iguais aos reportados em [10], sendo eles melhores ou piores a depender do cenário analisado. Diversos fatores podem contribuir para essa diferença, como os critérios para a geração de lotes, minilotes e épocas que não foram reportados no artigo original e que, portanto, podem ser diferentes, além do fato de que, mesmo com o uso de uma semente aleatória, os resultados obtidos a partir de um modelo construído com PyTorch não são reproduzíveis [55].

Os experimentos foram realizados utilizando uma placa de vídeo GTX 1080Ti, um processador i7-8700 e um ambiente Python 3.9.0 com PyTorch 1.13.1 [55]. Assim como no trabalho de referência [10], foi utilizado o otimizador SGD com *momentum* constante em $0,9$ e a taxa de aprendizagem inicial foi de $0,01$ com decaimento exponencial de $0,9$ após cada época. O valor de γ escolhido para o *Soft* DTW foi de 5, o λ utilizado na equação 3.10 do modelo proposto e 2.19 do modelo replicado foi de $0,01$ e um lote foi constituído por dois minilotes, ou seja, 32 assinaturas. Os valores de α na equação 3.8 e de p, q e r na equação 3.11 foram de, respectivamente, $0,9, 1,1$ e $0,1$. Todos os resultados reportados foram obtidos a partir do uso de uma mesma semente aleatória (333) a fim de tornar, na medida do possível, a comparação mais justa. O modelo que busca replicar o proposto em [10] foi treinado durante 25 épocas enquanto o modelo com as modificações propostas foi treinado durante apenas 15 épocas, uma vez que os testes previamente realizados demonstraram que não havia necessidade de um treinamento maior.

4.4.1 Cenário 4vs1

A Tabela 4.2 apresenta a dispersão dos limiares ótimos locais, expressa pela variância e amplitude, no cenário em que existem quatro assinaturas de referência e as assinaturas foram realizadas com caneta *stylus*. A tabela evidencia que as alterações propostas no modelo com o objetivo de aproximar os limiares surtiram efeito uma vez que o modelo proposto apresenta menor variância e amplitude tanto considerando falsificações profissionais quanto aleatórias. Em relação à variância, a melhora relativa no cenário com falsificações profissionais foi de 60,96%, enquanto no cenário com falsificações aleatórias foi de 26,39%. Já em relação à amplitude, no cenário com falsificações profissionais a melhora relativa foi de 44,15% e de 14,79% no cenário com falsificações aleatórias. Essas medidas comprovam que as modificações propostas de fato diminuíram a dispersão dos limiares ótimos locais neste cenário.

Modelo	Variância		Amplitude	
	Profissionais	Aleatórias	Profissionais	Aleatórias
DsDTW Replicado	0,04978	0,01262	1.3834	0,5888
Proposto	**0,01943**	**0,00929**	**0,7726**	**0,50171**

Tabela 4.2: Medidas de dispersão dos limiares ótimos específicos por usuário no cenário com caneta *stylus* e quatro amostras de referência.

A Tabela 4.3, por sua vez, apresenta os valores do EER no conjunto DeepSign no cenário em que as assinaturas foram realizadas com caneta *stylus* e havia quatro assinaturas de referência à disposição. O modelo aqui replicado não conseguiu alcançar os resultados reportados pelo modelo original, sendo pior que o mesmo tanto em relação a falsificações profissionais quanto aleatórias. Contudo, as modificações implementadas em cima desse modelo resultaram em uma melhora global no EER de 0,43% e relativa de 15,52% no caso de falsificações profissionais e global de 0,21% e relativa de 14,19% no caso de falsificações aleatórias. Graças a isso, as modificações aqui propostas conseguiram superar o modelo original em 0,2% no caso de falsificações profissionais, o que significa uma melhora relativa de 7,87% no EER. A melhora dos resultados com relação ao modelo replicado mostra que, de fato, a aproximação dos limiares locais ótimos, traduzida pela menor variância dos mesmos, contribui para a melhoria do sistema neste cenário.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	2,54	**0,97**
DsDTW replicado	2,77	1,48
Proposto	**2,34**	1,27

Tabela 4.3: EER (em %) no cenário com caneta *stylus* e quatro assinaturas de referência.

A Tabela 4.4 apresenta as medidas de dispersão no cenário com assinaturas realizadas com dedo e quatro referências. Observe que, novamente, as modificações aqui propostas resultaram na redução da dispersão dos limiares ótimos específicos por usuário tanto considerando assinaturas profissionais quanto aleatórias. Em relação à variância, a melhora relativa no cenário com falsificações profissionais foi de 33,3% e de 49,15% no cenário com falsificações aleatórias. Já em relação à amplitude a melhora nesses dois cenários foi de, respectivamente, 25,53% e 31,35%. Esses dados comprovam que o objetivo de reduzir a dispersão dos limiares ótimos locais também foi alcançado neste cenário. O reflexo dessas mudanças no EER é apresentado na Tabela 4.5.

Modelo	Variância		Amplitude	
	Profissionais	Aleatórias	Profissionais	Aleatórias
DsDTW Replicado	0,01886	0,01005	0,86444	0,57978
Proposto	**0,01258**	**0,00511**	**0,64377**	**0,39802**

Tabela 4.4: Medidas de dispersão dos limiares ótimos específicos por usuário no cenário com dedo e quatro assinaturas de referência.

Diferente do cenário com caneta *stylus*, o modelo replicado conseguiu superar o de referência quando as falsificações eram aleatórias, embora tenha apresentado piora no cenário em que as falsificações eram profissionais. As mudanças propostas neste trabalho foram traduzidas em uma melhora global de 0,51% e relativa de 6,74% no caso de falsificações profissionais com relação ao modelo replicado, embora tenham resultado em uma piora global de 0,3% com relação ao modelo replicado no cenário de falsificações aleatórias. Mesmo assim, o modelo proposto apresentou uma melhora relativa de 4,42% em relação ao modelo de referência no cenário de falsificações aleatórias.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	**6,99**	1,81
DsDTW Replicado	7,57	**1,43**
Proposto	7,06	1,73

Tabela 4.5: EER (em %) no cenário com dedo e quatro assinaturas de referência

Ao comparar as Tabelas 4.3 e 4.5 chama atenção os altos valores de EER presentes na segunda, especialmente no cenário que leva em conta falsificações profissionais. Entre as causas dessas discrepâncias estão a ausência da informação de pressão nas assinaturas realizadas com dedo e o fato de os sistemas terem sido treinados apenas com assinaturas realizadas com caneta *stylus*, como dito na seção 4.4. Além disso, deve-se levar em conta que, no geral, realizar assinaturas com dedo é mais difícil que com uma caneta devido à falta de costume da maioria das pessoas, o que tende a aumentar ainda mais as variações intraclasse.

Falsificações Profissionais

As Tabelas 4.6 e 4.7 apresentam, respectivamente, os resultados obtidos em cada um dos subconjuntos de dados presentes no DeepSign no caso de assinaturas realizadas com caneta *stylus* e dedo em que as falsificações eram profissionais e quatro assinaturas de referência estavam disponíveis. O conjunto DeepSignDB corresponde a união de todos os conjuntos e as comparações com falsificações aleatórias podem incluir assinaturas originais de conjuntos diferentes do qual as referências fazem parte.

Ambas as tabelas mostram que a média do EER considerando um limiar específico por usuário é inferior ao EER utilizando um limiar global. Isto é reflexo do problema mencionado no Capítulo 3 que, embora atenuado conforme mostrado na seção anterior através da redução da dispersão dos limiares ótimos específicos por usuário, ainda se demostra presente. O grande destaque vai para os conjuntos eBioSignDS1_W1 e eBioSignDS2_W2 que apresentam média de EER local de zero, ou seja, que considerando um limiar específico por usuário o sistema nunca erra, mas que ainda assim apresentam EER de 2,857% e 1,667%, respectivamente, ao utilizar um limiar global ótimo.

Conjunto de dados	**EER (%)**	**Média de EER (%) local**
DeepSignDB	**2,335**	**0,770**
MCYT	2,108	0,752
BiosecurID	1,168	0,221
BiosecureDS2	2,244	1,012
eBioSignDS1_W1	2,857	0,000
eBioSignDS1_W2	3,690	2,024
eBioSignDS1_W3	4,881	0,595
eBioSignDS1_W4	3,095	0,238
eBioSignDS1_W5	5,119	2,619
eBioSignDS2_W2	1,667	0,000

Tabela 4.6: Cenário 4vs1 com caneta *stylus* e falsificações profissionais.

Conjunto de dados	EER (%)	Média de EER (%) local
DeepSignDB	**7,06**	**4,207**
eBioSignDS1_W4	11,19	4,167
eBioSignDS1_W5	9,64	2,976
eBioSignDS2_W5	4,88	2,262
eBioSignDS2_W6	3,16	1,286

Tabela 4.7: Cenário 4vs1 com dedo e falsificações profissionais.

Observe que as médias de EER utilizando limiar local reportadas na Tabela 4.7 são consideravelmente superiores as apresentadas na Tabela 4.6. Este é um indicativo de que a variação intraclasse nas assinaturas realizadas com dedo é maior que a de assinaturas realizadas com caneta *stylus*.

Falsificações Aleatórias

As Tabelas 4.8 e 4.9 apresentam os resultados do modelo proposto em cada um dos subconjuntos de dados presentes no DeepSign no cenário com quatro referências e assinaturas realizadas com caneta *stylus* e dedo, respectivamente. Assim como no caso das falsificações profissionais é possível observar que os valores da média do EER utilizando limiar local são consideravelmente inferiores que seus respectivos valores de EER utilizando limiar global, com destaque para os subconjuntos eBioSignDS1_W1, eBioSignDS1_W3 e eBioSignDS1_W4 no cenário com caneta *stylus* e eBioSignDS1_W4, eBioSignDS2_W1 e eBioSignDS2_W2 no cenário com dedo, que apresentam média de EER local igual a zero, embora o EER com limiar global seja superior a zero.

Conjunto de dados	EER	Média de EER local
DeepSignDB	**1,265**	**0,290**
MCYT	0,007	0,001
BiosecurID	0,818	0,037
BiosecureDS2	1,573	0,396
eBioSignDS1_W1	2,416	0,000
eBioSignDS1_W2	2,311	0,861
eBioSignDS1_W3	0,861	0,000
eBioSignDS1_W4	0,007	0,000
eBioSignDS1_W5	2,941	1,155
eBioSignDS2_W2	0,651	0,609

Tabela 4.8: Cenário 4vs1 com caneta *stylus* e falsificações aleatórias.

Conjunto de dados	**EER (%)**	**Média de EER (%) local**
DeepSignDB	**1,73**	**0,268**
eBioSignDS1_W4	0,74	0,000
eBioSignDS1_W5	3,47	0,168
eBioSignDS2_W5	0,90	0,000
eBioSignDS2_W6	0,08	0,000

Tabela 4.9: Cenário 4vs1 com dedo e falsificações aleatórias.

Observe ainda que na Tabela 4.9 a média do EER local no conjunto DeepSignDB por inteiro é superior a soma das médias dos conjuntos que o compõe. Isso significa que ao comparar falsificações aleatórias de conjuntos de dados diferentes do conjunto das referências surgiram novos casos de falsa aceitação, o que fez com que a média do DeepSignDB aumentasse com relação aos demais conjuntos.

4.4.2 Cenário 1vs1

A Tabela 4.10 apresenta as medidas de dispersão dos limiares ótimos específicos por usuário no conjunto de dados DeepSign considerando o cenário em que as assinaturas foram realizadas com caneta *stylus* e uma referência estava disponível. Assim como nos cenários em que havia quatro referências disponíveis, o modelo proposto apresentou menor dispersão que o modelo replicado em todos os casos. No que diz respeito à variância, a melhora nos cenários com falsificações profissionais e aleatórias foi de, respectivamente, 99,24% e 99,05%. Em relação à amplitude as melhoras foram de, respectivamente, 57,29% e 47,76%. O reflexo das mudanças no EER pode ser observado na Tabela 4.11.

Modelo	Variância		Amplitude	
	Profissionais	Aleatórias	Profissionais	Aleatórias
DsDTW Replicado	0,03678	0,01262	0,26491	0,20172
Proposto	**0,00028**	**0,00012**	**0,11313**	**0,10536**

Tabela 4.10: Medidas de dispersão dos limiares ótimos específicos por usuário no cenário com caneta *stylus* e uma assinatura de referência.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	**4,04**	**1,69**
DsDTW Replicado	4,89	2,33
Proposto	4,22	2,19

Tabela 4.11: EER (em %) no cenário com caneta *stylus* e uma assinatura de referência.

A Tabela 4.11 mostra que neste cenário o modelo replicado não conseguiu alcançar os resultados do artigo de referência, fato que se manteve nos resultados apresentados pelo modelo replicado com as modificações propostas. Contudo, ao comparar os resultados do modelo proposto com o replicado, é possível observar que houve uma diminuição de 0,67% e 0,14% no EER, que correspondem a uma melhora de 13,70% e 6,01%, nos cenários com falsificações profissionais e aleatórias, respectivamente.

Já em relação ao cenário em que as assinaturas foram realizadas com dedo e uma assinatura de referência estava à disposição, a Tabela 4.12 apresenta a dispersão dos limiares ótimos específicos por usuário no conjunto de dados DeepSign. Novamente as medidas de dispersão comprovam que as modificações propostas foram capazes de aproximar mais os limiares ótimos locais uns dos outros. No cenário com falsificações profissionais a variância apresentou uma melhora de 79,63% enquanto no cenário com falsificações aleatórias a melhora foi de 65,52%. Em relação à amplitude as melhoras foram de, respectivamente, 60,79% e 47,43%.

Modelo	Variância		Amplitude	
	Profissionais	Aleatórias	Profissionais	Aleatórias
DsDTW Replicado	0,00054	0,00029	0,13368	0,07059
Proposto	**0,00011**	**0,00010**	**0,05241**	**0,03711**

Tabela 4.12: Medidas de dispersão dos limiares ótimos específicos por usuário no cenário com dedo e uma assinatura de referência.

Tendo em vista que em todos os cenários do protocolo de avaliação do DeepSign a dispersão dos limiares ótimos locais apresentou melhoras entre 26,39% e 99,05% com relação à variância e entre 14,79% e 60,79% com relação à amplitude, podemos concluir que, de maneira geral, o objetivo deste trabalho de diminuir a dispersão dos limiares ótimos locais dos usuários presentes no conjunto de dados DeepSign foi alcançado.

A Tabela 4.13 mostra que o modelo replicado novamente apresentou resultados piores que o modelo de referência. Contudo, em relação ao modelo replicado, as mudanças propostas resultaram em uma diminuição global do EER de 1,02% no cenário com falsificações profissionais, que equivale a uma melhora de 7,40%, e uma piora de global 1,06%, que equivale a 33%, no cenário com falsificações aleatórias.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	**11,84**	**2,89**
DsDTW Replicado	13,79	3,18
Proposto	12,77	4,24

Tabela 4.13: EER (em %) no cenário com dedo e uma assinatura de referência.

Observe que os únicos casos que o modelo proposto apresentou pioras com relação ao replicado foram aqueles cujas assinaturas foram realizadas com dedo e as falsificações consideradas eram aleatórias. Uma vez que a dispersão dos limiares ótimos específicos por usuário diminuiu e essa queda de desempenho não aconteceu no cenário com caneta *stylus* e falsificações aleatórias é provável que o modelo proposto seja mais sensível à informação de pressão, ausente em assinaturas realizadas com dedo, que o modelo replicado e o de referência. Além disso, tendo em vista que essa piora não aconteceu em relação ao cenário com falsificações profissionais, é possível que uma diminuição no valor de α na Fórmula 3.8 a fim de aumentar a importância das falsificações aleatórias contribua positivamente para a performance nesse cenário.

Falsificações Profissionais

As Tabelas 4.14 e 4.15 apresentam os resultados obtidos pelo modelo proposto em cada um dos subconjuntos de dados que compõem o DeepSign considerando, respectivamente, os cenários em que as assinaturas foram realizadas com caneta *stylus* e dedo e apenas uma referência estava disponível. Observe que, diferente do cenário com quatro assinaturas de referência, em nenhum dos subconjuntos de assinaturas a média do EER local ótimo foi de zero, o que evidencia o quanto o maior número de assinaturas de referência contribui positivamente para a resolução do problema.

Conjunto de dados	**EER (%)**	**Média de EER local (%)**
DeepSignDB	**4,219**	**1,956**
MCYT	3,821	2,328
BiosecurID	1,673	0,623
BiosecureDS2	4,393	2,106
eBioSignDS1_W1	7,143	3,363
eBioSignDS1_W2	5,238	1,815
eBioSignDS1_W3	6,637	1,905
eBioSignDS1_W4	6,131	2,827
eBioSignDS1_W5	8,869	5,387
eBioSignDS2_W2	3,929	2,292

Tabela 4.14: Cenário 1vs1 com caneta *stylus* e falsificações profissionais.

Conjunto de dados	EER (%)	Média de EER (%) local
DeepSignDB	**12,768**	**8,021**
eBioSignDS1_W4	18,631	8,661
eBioSignDS1_W5	14,435	10,714
eBioSignDS2_W5	9,286	5,565
eBioSignDS2_W6	7,649	2,054

Tabela 4.15: Cenário 1vs1 com dedo e falsificações profissionais.

Falsificações Aleatórias

As Tabelas 4.16 e 4.17 apresentam os resultados obtidos em cada um dos subconjuntos de dados presentes no DeepSign considerando, respectivamente, os cenários com caneta *stylus* e dedo em que as falsificações consideradas eram aleatórias e apenas uma referência estava à disposição. Observe que diferente do caso com quatro assinaturas de referência, nenhum dos subconjuntos nestes cenários apresentou média de EER local de zero.

Conjunto de dados	EER	Média de EER local
DeepSignDB	**2,188**	**1,211**
MCYT	1,130	0,621
BiosecurID	0,827	0,553
BiosecureDS2	2,909	1,339
eBioSignDS1_W1	3,929	1,964
eBioSignDS1_W2	3,225	1,229
eBioSignDS1_W3	1,786	0,357
eBioSignDS1_W4	2,868	1,003
eBioSignDS1_W5	6,418	2,442
eBioSignDS2_W2	3,214	2,595

Tabela 4.16: Cenário 1vs1 com caneta *stylus* e falsificações aleatórias.

Conjunto de dados	EER (%)	Média de EER (%) local
DeepSignDB	**4,242**	**1,460**
eBioSignDS1_W4	0,930	0,399
eBioSignDS1_W5	5,888	1,949
eBioSignDS2_W5	3,030	0,888
eBioSignDS2_W6	2,090	1,413

Tabela 4.17: Cenário 1vs1 com dedo e falsificações aleatórias.

4.5 Ajuste Fino

Devido ao pequeno número de assinaturas realizadas com dedo disponíveis para treinamento e com o intuito de analisar o quanto elas seriam capazes de contribuir para a performance do modelo com relação ao EER, foi realizado um ajuste fino as utilizando com a mesmas configurações apresentadas na seção 4.4 para o modelo proposto.

Foram realizados dois treinamentos: um para o cenário com uma referência e outro para o cenário com quatro referências. Os pesos iniciais foram aqueles cujos resultados foram reportados na seção 4.4.

4.5.1 Cenário 4vs1

A Tabela 4.18 revela que o ajuste fino resultou em uma melhora global de 1,08% no caso de falsificações profissionais e uma piora global de 0,99% com relação ao modelo sem ajuste fino. Graças a isso, o modelo com ajuste fino foi capaz de superar o modelo de referência no cenário com falsificações profissionais em 14,44%, enquanto o modelo sem ajuste fino apresentou melhora com relação ao cenário com falsificações aleatórias.

Os dados sugerem que o modelo com ajuste fino acabou por criar representações profundas de assinaturas originais de usuários diferentes de maneira mais semelhante, o que é traduzido em pontuações menores para falsificações aleatórias (pontuações oriundas de comparações entre a referência e assinaturas originais de outros usuários) e consequentemente maior EER neste cenário. Entre as possíveis soluções está, para o caso de assinaturas realizadas com dedo, realizar tanto o treinamento base como o ajuste fino utilizando maior ponderação para triplas em que a falsificação é aleatória.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	6,99	1,81
DsDTW Replicado	7,57	**1,43**
Proposto sem Ajuste Fino	7,06	1,73
Proposto com Ajuste Fino	**5,98**	2,72

Tabela 4.18: EER (em %) no cenário com dedo e quatro referências e ajuste fino.

Falsificações Profissionais

A Tabela 4.19 apresenta os resultados do modelo proposto com ajuste fino em cada um dos subconjuntos de dados que compõem o DeepSign no cenário com falsificações profissionais e quatro referências. O destaque vai para o subconjunto eBioSignDS2_W6 em que o modelo foi capaz de encontrar uma separação perfeita para todos os usuários utilizando

limiar local. Em contrapartida, também é possível observar que no subconjunto eBioSignDS1_W5 o modelo não conseguiu encontrar boas separações nem mesmo utilizando limiar local.

Conjunto de dados	EER (%)	Média de EER (%) local
DeepSignDB	**5,980**	**3,958**
eBioSignDS1_W4	10,000	6,667
eBioSignDS1_W5	7,738	7,500
eBioSignDS2_W5	3,095	2,262
eBioSignDS2_W6	2,062	0,000

Tabela 4.19: Cenário 4vs1 com dedo e falsificações profissionais, modelo com ajuste fino.

Falsificações aleatórias

A Tabela 4.20 apresenta os resultados nos subconjuntos do DeepSign no cenário com quatro referências e falsificações aleatórias no modelo com ajuste fino. Observe que as médias apresentadas nesta tabela são maiores que as apresentadas pela Tabela 4.9 que se refere aos resultados neste mesmo cenário com modelo sem ajuste fino, o que comprova que o poder de discriminação do modelo foi prejudicado neste cenário durante o ajuste fino.

Conjunto de dados	EER	Média de EER local
DeepSignDB	**2,715**	**0,783**
eBioSignDS1_W4	1,450	0,168
eBioSignDS1_W5	3,655	0,126
eBioSignDS2_W5	2,416	0,735
eBioSignDS2_W6	2,038	0,084

Tabela 4.20: Cenário 4vs1 com dedo e falsificações aleatórias, modelo com ajuste fino

4.5.2 Cenário 1vs1

A Tabela 4.21 diz respeito ao cenário em que apenas uma referência está à disposição e mostra que o ajuste fino novamente foi capaz de apresentar melhores resultados no caso com falsificações profissionais, apresentando uma melhora global com relação ao modelo sem ajuste fino de 1,79%, que equivale a uma melhora relativa de 14,02%. Isso permitiu que o modelo com ajuste fino apresentasse uma melhora relativa de 7,26% com relação ao modelo de referência, que equivale a uma melhora global de 0,86%. Contudo, o ajuste fino prejudicou a performance do modelo no cenário com falsificações aleatórias, o que

resultou em uma piora global de 2,18%, que corresponde a uma piora relativa de 51,42% com relação ao modelo sem ajuste fino. Assim como aconteceu com o cenário com quatro referências e falsificações aleatórias, este é um indicativo de que o ajuste fino acabou por diminuir o poder de discriminação do modelo quando lidando com falsificações aleatórias.

Modelo	**Profissionais**	**Aleatórias**
DsDTW [10]	11,84	2,89
DsDTW Replicado	13,79	3,18
Proposto Sem Ajuste Fino	12,77	4,24
Proposto Com Ajuste Fino	10,98	6,42

Tabela 4.21: EER (em %) no cenário com dedo, uma referência e ajuste fino.

Falsificações Profissionais

A Tabela 4.22 apresenta o desempenho do modelo com ajuste fino em cada um dos subconjuntos que compõem o DeepSign no cenário com uma referência e falsificações profissionais. Assim como no caso com quatro referências, o destaque vai para o subconjunto eBioSignDS1_W5, que apresentou uma alta média de EER local, o que demonstra a dificuldade do modelo de realizar uma boa separação entre falsificações e originais neste conjunto.

Conjunto de dados	**EER (%)**	**Média de EER local (%)**
DeepSignDB	**10,982**	**7,091**
eBioSignDS1_W4	16,012	8,720
eBioSignDS1_W5	15,714	10,685
eBioSignDS2_W5	6,667	3,750
eBioSignDS2_W6	5,030	0,923

Tabela 4.22: Cenário 1vs1 com dedo e falsificações profissionais.

Falsificações Aleatórias

A Tabela 4.23 apresenta os resultados nos subconjuntos do DeepSign considerando apenas uma referência e falsificações aleatórias. Novamente, o conjunto eBioSignDS1_W5 é o maior vilão, apresentando EER muito superior aos demais conjuntos analisados.

Conjunto de dados	EER	Média de EER local
DeepSignDB	**6,423**	**2,166**
eBioSignDS1_W4	2,001	0,604
eBioSignDS1_W5	8,929	3,162
eBioSignDS2_W5	3,918	1,208
eBioSignDS2_W6	2,847	0,714

Tabela 4.23: Cenário 1vs1 com dedo e falsificações aleatórias.

4.6 Comportamento da Função de Perda e Reflexo na Validação

A Figura 4.1 apresenta o comportamento da função de perda e a variação do EER no cenário com caneta *stylus*, quatro assinaturas de referência e falsificações profissionais ao longo de 25 épocas no modelo replicado. A Figura 4.2 também apresenta o comportamento da função de perda e do mesmo cenário de avaliação, porém se refere ao modelo proposto e a apenas 15 épocas.

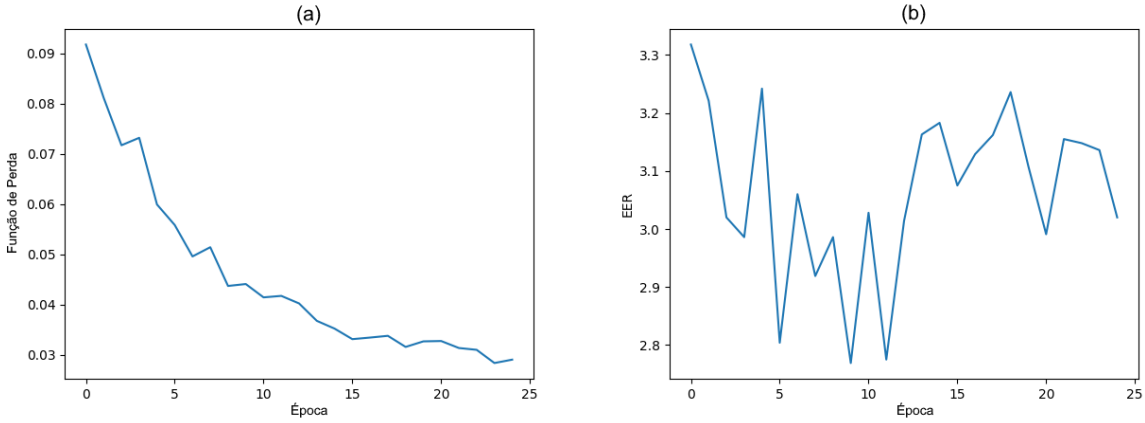

Figura 4.1: (a) variação da função de perda e (b) variação do EER ao longo das 25 épocas de treinamento do modelo replicado.

O comportamento de ambas as funções de perda é bastante semelhante, embora seja possível observar que a função proposta tende a diminuir de maneira mais lenta que a função de perda original, o que sugere que o modelo original tende a sofrer sobreajuste mais rapidamente.

Embora o EER após a primeira época no modelo replicado e proposto sejam semelhantes (cerca de $3,3\%$ e $3,1\%$, respectivamente), a queda no modelo proposto para a

próxima época é consideravelmente maior. Enquanto o modelo original atinge seu menor EER na décima época (2,77%), EER que é justamente o reportado na Tabela 4.3, após a segunda época de treinamento no modelo proposto o EER já é cerca de $2,7\%$, de modo a atingir seu menor valor após a oitava época. Assim, é possível concluir que o modelo proposto tende a atingir bons resultados com menos épocas de treinamento.

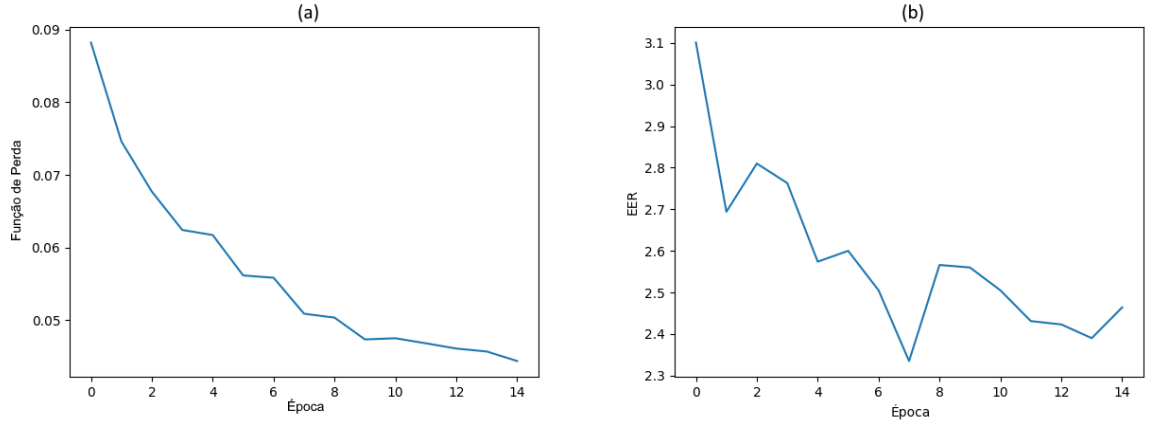

Figura 4.2: (a) variação da função de perda e (b) variação do EER ao longo das 15 épocas de treinamento do modelo proposto.

4.6.1 Comparação com o Estado da Arte

A Tabela 4.24 resume todas as tarefas do protocolo de avaliação do DeepSign e apresenta os resultados obtidos em outros trabalhos que também utilizaram esse conjunto de dados.

Entrada	Modelo	Profissionais		Aleatórias		Média
		4vs1	1vs1	4vs1	1vs1	
Stylus	DTW [10]	4,53	7,06	1,23	1,98	3,70
	TA-RNN [9]	3,30	4,20	**0,60**	**1,50**	2,40
	DsDTW [10]	2,54	**4,04**	0,97	1,69	**2,31**
	Proposto	**2,34**	4,22	1,27	2,19	2,51
Dedo	DTW [10]	10,66	14,74	1,02	**1,25**	6,92
	TA-RNN [9]	11,30	13,80	**1,00**	1,80	7,00
	DsDTW [10]	6,99	11,84	1,81	2,89	**5,88**
	Proposto **sem** ajuste fino	7,06	12,77	1,73	4,24	6,45
	Proposto **com** ajuste fino	**5,98**	**10,98**	2,72	6,42	6,53

Tabela 4.24: Comparação do EER (em %) com o estado da arte no conjunto de dados DeepSign.

A tabela mostra que as modificações propostas no modelo DsDTW apresentaram ganho significativo nos cenários que envolvem falsificações profissionais utilizando caneta *stylus* com quatro referências e também com dedo, no caso do modelo com ajuste fino, tanto quando utilizadas quatro ou apenas uma referência, o que faz com que o modelo proposto apresente melhor desempenho entre os comparados nesses casos. Além disso, o modelo proposto apresentou uma leve piora com relação ao DsDTW no cenário de falsificações profissionais e uma referência com assinaturas realizadas com caneta *stylus*, embora ainda tenha se demonstrado competitivo no cenário.

Já no que diz respeito ao cenário com falsificações aleatórias é possível observar que os resultados do próprio DsDTW não são tão bons comparados aos demais considerando assinaturas realizadas com caneta *stylus* e, embora o modelo proposto tenha melhorado os resultados em relação ao replicado, o mesmo ainda não conseguiu superar os demais resultados apresentados na tabela. No caso em que as assinaturas foram realizadas com dedo o modelo proposto sem ajuste fino foi capaz de superar o DsDTW no cenário com quatro referências, embora com pior resultado no cenário com uma referência. Por fim, apesar de o modelo com ajuste fino ter sido o que melhor performou com relação a falsificações profissionais, ele também foi o que apresentou piores resultados no caso com falsificações aleatórias.

Diante do exposto ao longo de todo este capítulo é possível concluir que o objetivo deste trabalho de diminuir a dispersão dos limiares ótimos locais dos usuários presentes no DeepSign foi alcançado e que essa diminuição foi acompanhada de uma melhora em seis das oito combinações de cenários presentes no protocolo de avaliação utilizado, incluindo todas as que envolviam assinaturas realizadas com caneta *stylus* em relação ao modelo que elas foram aplicadas. Além disso, se considerarmos apenas o cenário em que as assinaturas foram realizadas com caneta *stylus*, que é o cenário focado pela maioria dos sistemas que trabalham com verificação de assinaturas *online* tendo em vista os maiores conjuntos de dados disponíveis e o maior número de informações que podem ser coletadas, o modelo proposto foi o melhor entre os comparados no cenário com quatro referências e ainda se manteve competitivo no cenário com uma única referência quando consideradas falsificações profissionais.

Capítulo 5

Conclusão e Trabalhos Futuros

A biometria adiciona uma forte barreira de segurança nos sistemas de autenticação ao avaliar uma pessoa baseado não no que ela sabe, como acontece com senhas alfanuméricas, mas sim em quem ela é. Entre as diversas impressões biométricas utilizadas hoje em dia, assinaturas feitas à mão se demonstram uma ótima opção tendo em vista seu histórico como forma de autenticação e sua ampla aceitação pela sociedade, embora quando comparada com outras impressões biométricas ainda apresente resultados relativamente piores, o que evidencia que ainda há muito espaço para crescimento na área.

O Capítulo 2 apresentou conceitos básicos da área de biometria que devem ser levados em conta durante a criação de um sistema incluindo tanto questões relacionadas à interação entre o sistema e o usuário quanto as métricas de avaliação utilizadas na literatura. Além disso, o capítulo também apresentou detalhes acerca da verificação de assinaturas onde o principal objetivo é confirmar se duas assinaturas foram feitas pela mesma pessoa ou não.

O Capítulo 2 também apresentou conceitos relevantes para este trabalho no que se refere a área de aprendizagem de máquina e do uso de rede neuronais profundas para atacar o problema de verificação de assinaturas como um problema de aprendizagem de métricas ao explicar algumas das camadas mais utilizadas em sistemas deste tipo. Por fim, o capítulo também abordou alguns trabalhos relacionados e mostrou que mesmo hoje as abordagens que fazem uso de redes profundas ainda não são unanimidade entre os pesquisadores, embora certamente possuam bastante potencial, como mostrado pelo sistema DsDTW descrito com detalhes ainda no fim desse capítulo e que serviu de referência para este trabalho.

O Capítulo 3 apresentou o problema do limiar global que tem a função de perda *Triplet Loss* como um de seus principais causadores e propôs uma nova função de perda juntamente com a substituição de dois sinais de entrada a fim diminuir a dispersão dos limiares ótimos locais dos usuários presentes no conjunto de teste do DeepSign com a

expectativa de que essa redução impactasse positivamente na performance do modelo com relação à métrica EER.

O Capítulo 4 apresentou o conjunto de dados DeepSign, utilizado neste trabalho, e seu rígido protocolo de avaliação que se seguido corretamente permite uma comparação justa entre trabalhos de diferentes autores. Além disso, foram reportados os resultados obtidos a partir do protocolo de avaliação considerando todos os cenários existentes no DeepSign: assinaturas realizadas com dedo ou caneta *stylus*, uma ou quatro assinaturas de referência e a presença de falsificações profissionais ou aleatórias. Os resultados reportados revelaram que o objetivo de diminuir a dispersão dos limiares locais foi alcançado, uma vez que o modelo proposto apresentou melhoras com relação ao replicado entre 26.39% e 99.05% com relação à variância e entre 14.79% e 60.79% com relação à amplitude.

Além de uma análise exaustiva dos resultados referentes aos vários cenários e subconjuntos de dados presentes no DeepSign, o Capítulo 4 também revelou que as modificações propostas neste trabalho contribuíram para a melhoria do EER em seis dos oito cenários avaliados com relação ao modelo replicado, superando inclusive o modelo de referência, cujos resultados compõem o estado da arte, no cenário com quatro assinaturas de referências e falsificações profissionais apresentando uma melhora relativa de 7.87%. Esse capítulo também abordou o impacto de um ajuste fino utilizando assinaturas realizadas com dedo tendo como pesos iniciais os obtidos a partir de um modelo treinado com assinaturas capturadas com caneta *stylus*. Os modelos com ajuste fino revelaram uma melhora de 14.44% e 7.26% nos cenários com falsificações profissionais em relação ao modelo de referência com quatro e uma referência, respectivamente, embora tenha apresentado piora com relação aos resultados quando consideradas falsificações aleatórias.

Por fim, o Capítulo 4 também mostrou que a função de perda proposta permitiu que o modelo atingisse melhores resultados com menos épocas de treinamento quando comparado com o modelo replicado, além de apresentar uma comparação dos resultados aqui obtidos com o estado da arte e evidenciar os ganhos nos cenários que consideram falsificações profissionais, embora o modelo ainda apresente, assim como o modelo de referência, dificuldades para lidar com falsificações aleatórias.

A função de perda *Triplet Loss* é bastante tolerante a variações intraclasse, o que é desejável no problema de verificação de assinaturas *online*. Contudo, é justamente essa tolerância que gera o problema do limiar global. Este trabalho propôs alterações na função de perda a fim de mitigar o problema e embora tenha apresentado resultados satisfatórios, é possível que mudanças na arquitetura das camadas da rede em conjunto com a função de perda proposta apresentem resultados ainda melhores. Para o futuro, uma das abordagens que pode apresentar bons resultados se encontra no uso de Redes Generativas Adversárias, praticamente inexplorado na área de verificação de assinaturas *online*, em que a parte

geradora da rede atue como extratora de características e a discriminadora aprenda a identificar as diferenças entre as assinaturas originais a fim de ajudar a extratora a gerar características mais semelhantes entre elas, de modo semelhante à maneira como essas redes são empregadas na área de adaptação de domínio, diminuindo assim os efeitos do problema do limiar global.

A outra modificação proposta neste trabalho foi substituir as velocidades das coordenadas x e y das assinaturas, isto é, suas derivadas discretas, pelas coordenadas x e y normalizadas com relação ao centroide. Tendo em vista que essa mudança contribuiu positivamente para o sistema, outra possibilidade que surge para melhorar ainda mais os resultados é fazer uso de um sistema multimodal ao não mais utilizar as coordenadas x e y como vetores de uma dimensão, mas sim como uma imagem, o que permitiria tirar proveito de técnicas da área de verificação biométrica de assinaturas *offline* em conjunto com as técnicas de verificação de assinaturas *online*, como as propostas aqui, ao também fazer uso de sequências temporais, como a que indica o comportamento da pressão exercida pela caneta durante a construção da assinatura.

Referências

[1] Fierrez-Aguilar, Julian, Loris Nanni, Jaime Lopez-Peñalba, Javier Ortega-Garcia e Davide Maltoni: *An on-line signature verification system based on fusion of local and global information*. Em Kanade, Takeo, Anil Jain e Nalini K. Ratha (editores): *Audio- and Video-Based Biometric Person Authentication*, páginas 523–532, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg, ISBN 978-3-540-31638-1. 1

[2] Anil K. Jain, Arun A. Ross: *Handbook of Biometrics*, volume 01. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA, 2008. 1, 4, 6, 7

[3] Nemcek, Walter F. e Wen C. Lin: *Experimental investigation of automatic signature verification*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-4(1):121–126, Jan 1974, ISSN 2168-2909. 1

[4] Herbst, N. M. e C. N. Liu: *Automatic signature verification based on accelerometry*. IBM Journal of Research and Development, 21(3):245–253, May 1977, ISSN 0018-8646. 1

[5] EerNisse, E.P., C.E. Land e J.B. Snelling: *Piezoelectric sensor pen for dynamic signature verification*. Em *1977 International Electron Devices Meeting*, páginas 473–476, Dec 1977. 1

[6] Sundararajan, Kalaivani e Damon L. Woodard: *Deep learning for biometrics: A survey*. ACM Comput. Surv., 51(3), may 2018, ISSN 0360-0300. `https://doi.org/10.1145/3190618`. 1

[7] Lai, Songxuan e Lianwen Jin: *Recurrent adaptation networks for online signature verification*. IEEE Transactions on Information Forensics and Security, 14(6):1624–1637, June 2019, ISSN 1556-6021. 1, 2, 7, 13, 16, 19

[8] Okawa, Manabu: *Online signature verification using locally weighted dynamic time warping via multiple fusion strategies*. IEEE Access, 10:40806–40817, 2022, ISSN 2169-3536. 1, 6, 16

[9] Tolosana, Ruben, Ruben Vera-Rodriguez, Julian Fierrez e Javier Ortega-Garcia: *Deepsign: Deep on-line signature verification*. IEEE Transactions on Biometrics, Behavior, and Identity Science, 3(2):229–239, April 2021, ISSN 2637-6407. 1, 2, 16, 20, 33, 34, 50

[10] Jiang, Jiajia, Songxuan Lai, Lianwen Jin e Yecheng Zhu: *Dsdtw: Local representation learning with deep soft-dtw for dynamic signature verification*. IEEE Transactions on Information Forensics and Security, 17:2198–2212, 2022, ISSN 1556-6021. 1, 2, 6, 16, 17, 20, 21, 22, 23, 24, 25, 36, 37, 39, 42, 43, 46, 48, 50

[11] Okawa, Manabu: *Template matching using time-series averaging and dtw with dependent warping for online signature verification*. IEEE Access, 7:81010–81019, 2019, ISSN 2169-3536. 1, 6, 16, 18

[12] Prabhakar, S., S. Pankanti e A.K. Jain: *Biometric recognition: security and privacy concerns*. IEEE Security & Privacy, 1(2):33–42, 2003. 4, 5

[13] Maltoni, Davide, Dario Maio, Anil K. Jain e Salil Prabhakar: *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2nd edição, 2009, ISBN 1848822537. 4, 5

[14] Anil K. Jain, Arun A. Ross, Karthik Nandakumar: *Introduction to Biometrics*, volume 01. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA, 2011. 5

[15] Vacca, John R.: *Biometric Technologies and Verification Systems*. Butterworth-Heinemann, USA, 2007, ISBN 0750679670. 5

[16] Joshi, Mahesh, Bodhisatwa Mazumdar e Somnath Dey: *Security vulnerabilities against fingerprint biometric system*. ArXiv, abs/1805.07116, 2018. 6

[17] Saleem, Mohammad e Bence Kovari: *Optimized jk-nearest neighbor based online signature verification and evaluation of the main parameters*. Computer Science, 22(4), Nov. 2021. `https://journals.agh.edu.pl/csci/article/view/4102`. 6, 16, 18

[18] Saleem, Mohammad e Bence Kovari: *Online signature verification based on signer dependent sampling frequency and dynamic time warping*. Em *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, páginas 182–186, Nov 2020. 7

[19] Hafemann, Luiz G., Robert Sabourin e Luiz S. Oliveira: *Offline handwritten signature verification — literature review*. Em *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, nov 2017. `https://doi.org/10.1109%2Fipta.2017.8310112`. 7

[20] Mitchell, Tom M.: *Machine Learning*. McGraw-Hill, New York, 1997, ISBN 978-0-07-042807-2. 8

[21] Han, J., M. Kamber e J. Pei: *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011, ISBN 9780123814807. `https://books.google.com.br/books?id=pQws07tdpjoC`. 8

[22] Kaelbling, L. P., M. L. Littman e A. W. Moore: *Reinforcement learning: A survey*. Journal of Artificial Intelligence Research, 4:237–285, 1996. 8

[23] Janiesch, Christian, Patrick Zschech e Kai Heinrich: *Machine learning and deep learning*. Electronic Markets, 31(3):685–695, 2021. 9

[24] Haykin, Simon S.: *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edição, 2009. 9

[25] Goodfellow, Ian, Yoshua Bengio e Aaron Courville: *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. 9, 10, 14

[26] Yamashita, Rikiya, Mizuho Nishio, Richard K. G. Do e Kaori Togashi: *Convolutional neural networks: an overview and application in radiology*. Insights into Imaging, 9:611 – 629, 2018. 9

[27] Aggarwal, Charu C.: *Neural Networks and Deep Learning: A Textbook*. Springer Publishing Company, Incorporated, 1st edição, 2018, ISBN 3319944622. 10, 11, 12

[28] Krizhevsky, Alex, Ilya Sutskever e Geoffrey E Hinton: *Imagenet classification with deep convolutional neural networks*. Em Pereira, F., C.J. Burges, L. Bottou e K.Q. Weinberger (editores): *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`. 11

[29] Brownlee, J.: *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019. `https://books.google.com.br/books?id=DOamDwAAQBAJ`. 11, 15

[30] Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau e Yoshua Bengio: *On the properties of neural machine translation: Encoder–decoder approaches*. Em *SSST@EMNLP*, 2014. 12

[31] Ravanelli, Mirco, Philemon Brakel, Maurizio Omologo e Yoshua Bengio: *Improving speech recognition by revising gated recurrent units*, 2017. 13

[32] Ravanelli, Mirco, Philemon Brakel, Maurizio Omologo e Yoshua Bengio: *Light gated recurrent units for speech recognition*. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(2):92–102, apr 2018. `https://doi.org/10.1109%2Ftetci.2017.2762739`. 13

[33] Lai, Nan, Meina Kan, Chunrui Han, Xingguang Song e Shiguang Shan: *Learning to learn adaptive classifier–predictor for few-shot learning*. IEEE Transactions on Neural Networks and Learning Systems, 32(8):3458–3470, Aug 2021, ISSN 2162-2388. 13

[34] Mendez-Ruiz, Mauricio, Jorge Gonzalez-Zapata, Ivan Reyes-Amezcua, Daniel Flores-Araiza, Francisco Lopez-Tiro, Andres Mendez-Vazquez e Gilberto Ochoa-Ruiz: *Susana distancia is all you need: Enforcing class separability in metric learning via two novel distance-based loss functions for few-shot image classification*, 2023. 13

[35] Jung, Deunsol, Dahyun Kang, Suha Kwak e Minsu Cho: *Few-shot metric learning: Online adaptation of embedding for retrieval*. Em *Asian Conference on Computer Vision*, 2022. 14

[36] Koch, Gregory, Richard Zemel e Ruslan Salakhutdinov: *Siamese neural networks for one-shot image recognition*. 2015. 14

[37] Chopra, S., R. Hadsell e Y. LeCun: *Learning a similarity metric discriminatively, with application to face verification*. Em *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, páginas 539–546 vol. 1, 2005. 15

[38] Coria, Juan M., Hervé Bredin, Sahar Ghannay e Sophie Rosset: *A comparison of metric learning loss functions for end-to-end speaker verification*. Em Espinosa-Anke, Luis, Carlos Martín-Vide e Irena Spasić (editores): *Statistical Language and Speech Processing*, páginas 137–148, Cham, 2020. Springer International Publishing, ISBN 978-3-030-59430-5. 15

[39] Weinberger, Kilian Q. e Lawrence K. Saul: *Distance metric learning for large margin nearest neighbor classification*. Journal of Machine Learning Research, 10(9):207–244, 2009. `http://jmlr.org/papers/v10/weinberger09a.html`. 15

[40] Vintsyuk, T. K.: *Speech discrimination by dynamic programming*. Cybernetics, 4(1):52–57, 1968. Russian Kibernetika 4(1):81-88 (1968). 16

[41] Saleem, Mohammad e Bence Kovari: *K-nearest neighbour and dynamic time warping for online signature verification*, 2021. 16, 18

[42] Cuturi, Marco e Mathieu Blondel: *Soft-dtw: A differentiable loss function for time-series*. Em *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, página 894–903. JMLR.org, 2017. 17

[43] Singh, Abigail e Serestina Viriri: *Online signature verification using deep descriptors*. Em *2020 Conference on Information Communications Technology and Society (ICTAS)*, páginas 1–6, March 2020. 19

[44] Park, Chan Yong, Han Gyu Kim e Ho Jin Choi: *Robust online signature verification using long-term recurrent convolutional network*. Em *2019 IEEE International Conference on Consumer Electronics (ICCE)*, páginas 1–6, Jan 2019. 19

[45] Lai, Songxuan, Lianwen Jin, Yecheng Zhu, Zhe Li e Luojun Lin: *Synsig2vec: Forgery-free learning of dynamic signature representations by sigma lognormal-based synthesis and 1d cnn*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10):6472–6485, Oct 2022, ISSN 1939-3539. 20

[46] Young, Steve J., D. Kershaw, J. Odell, D. Ollason, V. Valtchev e P. Woodland: *The HTK Book Version 3.4*. Cambridge University Press, 2006. 22

[47] Barreto, Fabian, Jignesh Sarvaiya, Suprava Patnaik e Sushilkumar Yadav: *Unsupervised domain adaptation using maximum mean covariance discrepancy and variational autoencoder*. International Journal of Advanced Computer Science and Applications, 13(6), 2022. `http://dx.doi.org/10.14569/IJACSA.2022.01306104`. 29

[48] Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf e Alexander Smola: *A kernel two-sample test*. Journal of Machine Learning Research, 13(25):723–773, 2012. http://jmlr.org/papers/v13/gretton12a.html. 29, 30

[49] Yan, Hongliang, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu e Wangmeng Zuo: *Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation*. Em *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 945–954, July 2017. 29

[50] Wang, Wei, Haojie Li, Zhengming Ding, Feiping Nie, Junyang Chen, Xiao Dong e Zhihui Wang: *Rethinking maximum mean discrepancy for visual domain adaptation*. IEEE Transactions on Neural Networks and Learning Systems, 34(1):264–277, Jan 2023, ISSN 2162-2388. 29

[51] Weiss, Karl, Taghi M Khoshgoftaar e DingDing Wang: *A survey of transfer learning*. Journal of Big data, 3(1):9, 2016. 29

[52] Ortega-Garcia, J., J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez, V. Espinosa, A. Satue, I. Hernaez, J. J. Igarza, C. Vivaracho, D. Escudero e Q. I. Moro: *Mcyt baseline corpus: A bimodal biometric database*. IEE Proceedings Vision, Image and Signal Processing, Special Issue on Biometrics on the Internet, 150(6):395–401, December 2003. 33

[53] Fierrez, J., J. Galbally, J. Ortega-Garcia, M. R. Freire, F. Alonso-Fernandez, D. Ramos, D. T. Toledano, J. Gonzalez-Rodriguez, J. A. Siguenza, J. Garrido-Salas, E. Anguiano, G. Gonzalez de Rivera, R. Ribalda, M. Faundez-Zanuy, J. A. Ortega, V. Cardeñoso-Payo, A. Viloria, C. E. Vivaracho, Q. I. Moro, J. J. Igarza, J. Sanchez, I. Hernaez, C. Orrite-Uruñuela, F. Martinez-Contreras e J. J. Gracia-Roche: *Biosecurid: A multimodal biometric database*. Pattern Analysis and Applications, 13(2):235–246, May 2010. 33

[54] Ortega-Garcia, J., J. Fierrez, F. Alonso-Fernandez, J. Galbally, M. Freire, J. Gonzalez-Rodriguez, C.Garcia-Mateo, J.-L.Alba-Castro, E.Gonzalez-Agulla, E.Otero-Muras, S.Garcia-Salicetti, L.Allano, B.Ly-Van, B.Dorizzi, J.Kittler, T.Bourlai, N.Poh, F.Deravi, M.Ng, M.Fairhurst, J.Hennebert, A.Humm, M.Tistarelli, L.Brodo, J.Richiardi, A.Drygajlo, H.Ganster, F.M.Sukno, S.-K.Pavani, A.Frangi, L.Akarun e A.Savran: *The multi-scenario multi-environment biosecure multimodal database (bmdb)*. IEEE Trans. on Pattern Analysis and Machine Intelligence, 32(6):1097–1111, June 2010. 33

[55] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai e Soumith Chintala: *Pytorch: An imperative style, high-performance deep learning library*. Em *Advances in Neural Information Processing Systems 32*, páginas 8024–8035. Curran Associates, Inc., 2019. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. 37