ANDREY GUSTAVO DE SOUZA

INTELIGÊNCIA ARTIFICIAL PARA A AUTENTICAÇÃO DE CONDUTORES:

UMA ABORDAGEM UTILIZANDO REDES NEURAIS SIAMESAS

LAVRAS – MG

2019

ANDREY GUSTAVO DE SOUZA

INTELIGÊNCIA ARTIFICIAL PARA A AUTENTICAÇÃO DE CONDUTORES: UMA ABORDAGEM UTILIZANDO REDES NEURAIS SIAMESAS

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação para a obtenção do título de Mestre.

Prof. DSc. Wilian Soares Lacerda
Orientador

Prof. DSc. Danilo Alves de Lima
Coorientador

LAVRAS – MG

2019

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo próprio autor

de Souza, Andrey Gustavo
Inteligência Artificial para a Autenticação de Condutores : Uma abordagem utilizando Redes Neurais Siamesas / Andrey Gustavo de Souza. – Lavras : UFLA, 2019.
75 p. : il.

Dissertação (mestrado acadêmico)–Universidade Federal de Lavras, 2019.
Orientador: Prof. DSc. Wilian Soares Lacerda.
Coorientador: Prof. DSc. Danilo Alves de Lima.
Bibliografia.

1. Redes Neurais Artificiais. 2. Autenticação de Condutores. 3. Redes Siamesas. I. Lacerda, Wilian Soares. II. de Lima, Danilo Alves. III. Título.

ANDREY GUSTAVO DE SOUZA

INTELIGÊNCIA ARTIFICIAL PARA A AUTENTICAÇÃO DE CONDUTORES: UMA ABORDAGEM UTILIZANDO REDES NEURAIS SIAMESAS

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação para a obtenção do título de Mestre.

APROVADA em 12 de Julho de 2019.

Prof. DSc. Wilian Soares Lacerda	UFLA
Prof. DSc. Danton Diego Ferreira	UFLA
Prof. DSc. Gustavo Lobato Campos	IFMG

Prof. DSc. Wilian Soares Lacerda
Orientador

Prof. DSc. Danilo Alves de Lima
Co-Orientador

LAVRAS – MG
2019

À minha família, à glória de Deus
e a todos mencionados nos agradecimentos,
dedico.

AGRADECIMENTOS

Primeiramente agradeço a minha família, em especial aos meus pais Gilson e Ivone, que mesmo diante de todas as adversidades foram como escudos que me permitiram seguir sonhando esse sonho, que também é deles. Aos meus irmãos André e Arlênio, que junto comigo descobriram os estudos como a melhor maneira de crescer. Aos meus avós, bisavó, tios e primos que acreditaram, torceram e rezaram por mim. À Camila, por todo amor, carinho e amparo durante todas as incertezas e inseguranças que me cercaram durante todo esse tempo, sendo meu porto seguro meio a tudo isso.

Agradeço ao meu orientador, Wilian, por todo o conhecimento e conselhos transmitidos nestes dois anos de mestrado e sobretudo a compreensão e amizade que tive em todas as boas conversas que tivemos.

Ao meu coorientador, Danilo, que aceitou compartilhar esse desafio conosco e que de maneira efetiva contribuiu não só com seus conhecimentos, mas com apoio e conselhos em todas as decisões tomadas.

À todos os professores do PPGESISA em especial Sílvia, Danton, Bruno e Daniel que por meio do ensinamentos transmitidos, não só em aulas, mas também boas conversas me fizeram crescer e chegar até aqui.

Aos meus colegas do PPGESISA da turma de 2017/01, que eram colegas e se tornaram irmãos que levarei comigo durante a vida. Também aos membros do LABSINE, amigos com os quais nesse ambiente pude crescer, não só tecnicamente, mas como ser humano.

Aos colegas e mentores que tive durante o programa de formação ITAú Analytics, em especial meu orientador do período de especialização, Carlos Forster, e meu tutor, Tiago Nazaré, que contribuíram muito com novas ideias que fizeram este trabalho possível. Aos companheiros de trabalho no Itaú Unibanco, que me ajudaram com incentivos constantes e toda a compreensão.

Aos meus antigos professores do IFMG Campus Formiga que sempre me incentivaram sempre a buscar mais, em especial ao professor Gustavo, que sempre me acompanhou e me aconselhou, mesmo durante o mestrado, se tornando um bom amigo.

À Universidade Federal de Lavras, por toda a estrutura disponibilizada para que esse sonho fosse possível. E também ao povo brasileiro que, mesmo em momento difícil, contribui com seus impostos para que exista a educação pública, gratuita e de qualidade. Espero retribuir à sociedade esse investimento de maneira efetiva.

Por fim e mais importante, agradeço à Deus por me conceder viver tudo isso, me inspirar e me dar forças para chegar até aqui.

“É muito melhor lançar-se em busca de conquistas grandiosas, mesmo expondo-se ao fracasso, do que alinhar-se com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem numa penumbra cinzenta, onde não conhecem nem vitória, nem derrota.”
(Theodore Roosevelt)

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota”.
(Madre Teresa de Calcutá)

RESUMO

O problema crônico de roubos e furtos de veículos em todo mundo, e especialmente no Brasil, tem crescido consideravelmente nos últimos anos. Em paralelo à esse problema, cada vez mais o uso de dados tem revolucionado diversos segmentos do mercado por meio de aplicações de técnicas de inteligência computacional para tarefas antes difíceis de serem solucionadas por meio de algoritmos tradicionais. Ciente desta realidade, este trabalho visa o desenvolvimento de um sistema de autenticação de condutores baseado em inteligência artificial, que faz uso de dados proprioceptivos do veículo, obtidos por meio da porta OBDII e de sensores inerciais de *smartphones*. Diferentes de outras abordagens que adotam essa temática na literatura, o presente trabalho foca na autenticação de condutores que não foram usados no treinamento do modelo em questão. Para tal, o uso de redes neurais siamesas é explorado para a tarefa de autenticação de condutores diante da limitação imposta. Redes neurais siamesas são conhecidas pelo seu desempenho em aplicações que envolvem identificação de indivíduos, como em reconhecimento facial, mesmo em situações em que se tenha somente poucos dados do indivíduo o qual se queira autenticar. A metodologia adotada explora a capacidade dessas redes de criar *embeddings* dos dados de indivíduos para efetuar sua posterior autenticação com técnicas baseadas em distância, formando uma função de decisão. Também é explorado o uso de técnicas de filtragem e extração de características, nesse caso o uso de janelas deslizantes que fomentam o desempenho dos resultados da rede neural siamesa. Essa combinação de técnicas de processamento de dados e técnicas de inteligência computacional obteve bons resultados na tarefa de autenticação de condutores, mesmo para os dados que não foram utilizados no treinamento da rede neural siamesa. Obteve-se uma ROC-AUC superior à 99% nos experimentos executados, o que indica boa aptidão das redes neurais siamesas para a tarefa de autenticação de condutores.

Palavras-chave: Autenticação de Condutores. Dados Veiculares. Redes Neurais Artificiais. Redes Siamesas. Identificação Comportamental de Condutores.

ABSTRACT

The chronic problem of vehicle theft and robbery worldwide, and especially in Brazil, has grown considerably in recent years. In parallel with this problem, the increasingly abundant use of data has revolutionized various segments of the market through applications of computational intelligence techniques for tasks previously difficult to solve using traditional algorithms. Aware of this reality, this work aims to develop a system based on an artificial intelligence model of driver authentication, which makes use of vehicle's proprioceptive data, obtained through the on-board diagnostics interface (OBDII) and inertial sensors present in smartphones. Different from other approaches that adopt this theme in the literature, the present work aims the authentication of drivers that were not used during the training step of the current model. For this, we used siamese neural networks for the driver's authentication task to deal with this imposed limitation. Siamese neural networks are known for their performance in applications involving people identification, such as face recognition, even in situations where only few data are available for authentication. The adopted methodology exploits the ability of these networks to create embeddings from individuals' data to carry out their later authentication through techniques based on distance, forming a decision function. It is also explored filtering techniques and features extraction, in this case, the use of sliding windows, which improves the performance of the siamese neural network. This combination of data processing and computational intelligence techniques has well performed the driver authentication task, even when the data have not been used for the Siamese neural network training. A ROC-AUC greater than 99 percent was obtained in real experiments, which indicates a good suitability of the siamese neural networks for the drivers' authentication task.

Keywords: Drivers' Authentication. Vehicle Data. Artificial Neural Networks. Siamese Networks. Drivers' Behavior Identification.

LISTA DE FIGURAS

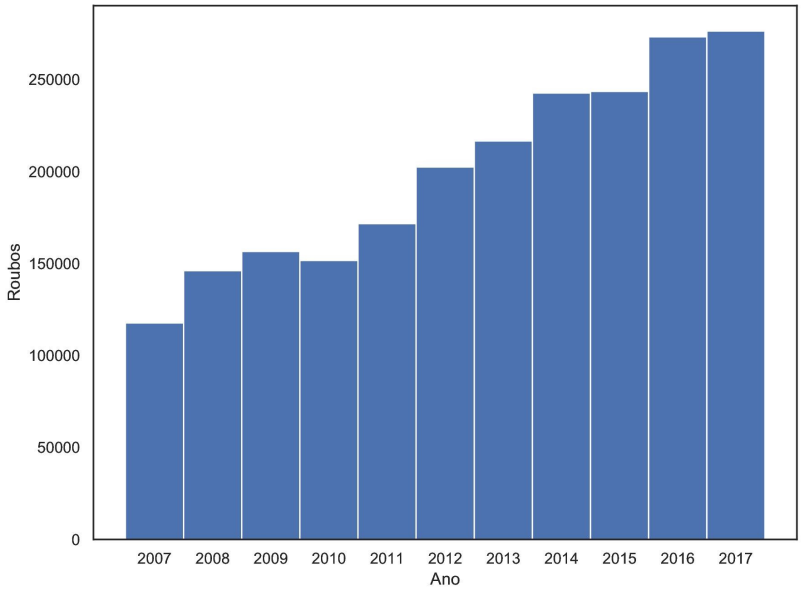
LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Um dos grandes problemas relacionados à segurança pública no Brasil é o número crescente de roubos e furtos de veículos, quando, somente em 2017, foram furtados ou roubados mais de 538 mil veículos em todo país (Fórum Brasileiro de Segurança Pública, 2018). A Figura 1.1 apresenta a série histórica do número de veículos roubados no Brasil desde 2007, sem considerar furtos, que passaram a ser contabilizados somente a partir de 2013 pelo Fórum Brasileiro de Segurança Pública. Tais números corroboram para a discussão de soluções alternativas e imediatas que visam mitigar esse problema, visto que elas não beneficiariam somente proprietários de veículos, mas também companhias de seguro e forças de segurança pública, que diretamente sofrem com este problema.

Figura 1.1 – Série histórica de roubos de veículos no Brasil por ano.



Fonte: Adaptado do Fórum Brasileiro de Segurança Pública (2018).

Ciente dessa realidade, o desenvolvimento e aplicação de tecnologias de segurança são vistas com bons olhos, principalmente quando funcionam de forma rápida e de tal modo a deter a ação criminosa a tempo de se evitar perdas maiores. Grande parte dos dispositivos antifurto veicular empregados atualmente são restritos a sistemas físicos instalados no veículo, tais como travas elétricas e sistemas de alarme, porém sem nenhum sistema de comunicação externo que alerte o proprietário ou as autoridades de segurança. Estes sistemas também são facilmente burlados e, consequentemente, não impedem o criminoso de cometer o delito. Em paralelo a este problema, aplicações envolvendo a análise do comportamento de condutores mediante

dados de direção têm sido exploradas com diversos objetivos, como detecção de agressividade na direção, sonolência, desatenção, entre outros. Aplicações também podem ser utilizadas para a autenticação e/ou identificação desses condutores em um determinado veículo. Isso é possível graças ao estilo de direção único que cada condutor tem e que pode ser identificado por meio da aplicação de técnicas de aprendizado de máquina.

Na literatura, autenticação e identificação têm sido objeto recente de estudo e têm ganhado interesse não só de acadêmicos, mas também de fabricantes de veículos estimulados pelo impacto que a aplicação de tais funcionalidades teria no mercado automotivo. Porém, o estado da arte desta temática ainda passa por diversas limitações que impedem a implementação real deste tipo de sistema. Dentre essas limitações, pode-se destacar a necessidade que algoritmos de aprendizado de máquina têm de serem retreinados a cada novo condutor introduzido no modelo de identificação ou autenticação de um veículo (WANG et al., 2017; BURTON et al., 2017; MARTINEZ; ECHANOBE; CAMPO, 2016; EZZINI; BERRADA; GHOGHO, 2018). Esta situação pode ser restritiva devido à capacidade de processamento limitado das centrais de eletrônicas do veículo. Diante de tal cenário, este trabalho tem como objetivo estudar alternativas para a autenticação de condutores que não demandem novo treinamento a cada novo condutor ou veículo em que o modelo venha a ser aplicado.

1.1 Objetivos

O presente trabalho tem como objetivo o desenvolvimento de um modelo de autenticação de condutores de veículos. São considerados os seguintes objetivos específicos:

- Avaliar o estado da arte referente à temática de modelagem e identificação comportamental de condutores e em especial o ramo de autenticação de condutores e estudar o funcionamento de redes neurais siamesas e como aplicá-las ao problema de autenticação de condutores;
- Avaliar e implementar técnicas de processamento de dados, como seleção e extração de características, que fomentem o desempenho do modelo;
- Implementar uma rede neural siamesa, treinada com parte dos condutores presentes no *dataset* disponibilizado por Vasconcelos (2017);
- Implementar uma função de decisão baseada em distância euclidiana que efetue a autenticação do restante dos condutores presentes no *dataset* utilizado, por meio da aplicação da

rede neural base da rede siamesa treinada para gerar *embeddings* desses dados, em uma aplicação de *few shot learning*.

1.2 Contribuições

Ciente das limitações das técnicas hoje empregadas para a identificação e autenticação de condutores, a principal contribuição do presente trabalho reside na implementação de um modelo de autenticação de condutores que seja capaz de efetuar sua tarefa mesmo em situações em que novos condutores são submetidos à esse, demandando somente da coleta de uma certa quantidade de dados de novos condutores para esta execução. Para tanto, é utilizado para esta tarefa uma topologia especial de redes neurais artificiais conhecidas como redes neurais siamesas. Essa topologia é amplamente empregada em tarefas que demandam identificação, como reconhecimento facial e identificação de autoria de textos, contudo sua aplicação em identificação de condutores ainda é inédita na literatura.

1.3 Estrutura do Trabalho

Este trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta a revisão da literatura acerca dos temas que foram abordados ao longo da pesquisa, tais como modelagem e identificação comportamental de condutores e a esfera de identificação e autenticação de condutores;
- No Capítulo 3 são apresentados conceitos referentes à redes neurais artificiais e a topologia siamesa. Também são apresentadas técnicas secundárias no desenvolvimento do projeto, bem como métricas de validação do modelo;
- No Capítulo 4 são apresentadas as etapas de desenvolvimento do sistema proposto de forma detalhada, bem como os insumos utilizados;
- No Capítulo 5 estão os resultados obtidos nos experimentos executados, tal como uma análise sobre os mesmos;
- Por fim, no Capítulo 6 é feito um apanhado geral do desenvolvimento e resultados obtidos, as conclusões acerca destes e também sugestões de continuidade do presente trabalho;

2 REVISÃO DA LITERATURA

Esse capítulo apresenta o estado da arte relacionado à modelagem e identificação comportamental de condutores por meio de dados veiculares naturalísticos oriundos de redes veiculares obtidos por meio da porta OBDII e unidades de medição inercial (IMU – do inglês *Inertial Measurement Unit*) de *smartphones*. Uma ênfase especial é dada na aplicação destes conceitos na tarefa de autenticação e identificação de condutores, temática tratada pelo presente trabalho.

2.1 Modelagem e Identificação Comportamental de Condutores

A revolução tecnológica que diversos setores têm experimentado se aplica também ao ramo automobilístico. Desde quando o primeiro veículo automotor ganhou as ruas no século XIX, novas tecnologias vem sendo empregadas de forma contínua de modo a proporcionar conforto, segurança e economia aos seus usuários. Atualmente, a maioria dos veículos em circulação possuem algumas destas tecnologias, como os freios ABS (*Anti-Lock Braking System*) e Programas de Estabilidade Eletrônica (ESP, *Electronic Stability Program*), que de modo contundente proporcionam segurança aos condutores e passageiros por meio do uso de sistemas eletrônicos. Seguindo a evolução e popularização dos microprocessadores, surgiram assim as Unidades de Controle Eletrônico (ECU). Na indústria automotiva, uma ECU é um dispositivo eletrônico embarcado que realiza a leitura de sinais oriundos de sensores localizados em diversas partes e componentes do veículo e dependendo destas informações colhidas controla várias partes importantes do mesmo, como o motor e outras opções automatizadas (EBERT; JONES, 2009).

A grande quantidade de informação que passou a circular entre ECUs fez com que surgisse a necessidade de se desenvolver redes de comunicação multiplexadas entre as ECUs, de tal forma a simplificar o cabeamento e reduzir custos de implementação. Assim na década de 80, foi desenvolvido pela Bosch a *Controller Area Network* (CAN), que é um barramento intraveicular de comunicação entre ECUs, sensores e atuadores. A CAN é conhecida por sua robustez na transmissão de dados, sendo resistente à interferências eletromagnéticas e operação em tempo real (TUOHY et al., 2015). Dados provenientes da CAN trazem em si informações relevantes sobre a dinâmica de direção do veículo que podem ser coletados por meio da interface de diagnóstico do veículo OBDII (*On-Board Diagnostic* segunda geração), que a partir de 2010 passou a ser adotado por todos os veículos e estabeleceu uma padronização e versatilidade no acesso à esses dados. O OBDII (PAN; YU; CHENG, 2017) permite a conexão do sistema do

veículo por meio de um equipamento com um *software* de coleta de dados instalado em um computador ou *smartphone*.

Outra fonte de informações referentes à dinâmica de direção são as unidades de medição inercial (IMU) (KAPLAN et al., 2015), que vêm embarcados nos próprios *smartphones*. Estes sensores utilizados em conjunto com os dados do OBDII, têm sido explorados por diversos trabalhos para a modelagem e identificação comportamental de condutores de veículos por meio aplicação de técnicas de aprendizado de máquina, que vem tornando possível a implementação de algoritmos que aprendem com informações passadas a generalizar tarefas específicas, antes difíceis de serem resolvidas por meio de técnicas de programação tradicionais. Por exemplo, Andria et al. (2016) desenvolveram uma plataforma de aquisição de dados do OBDII e IMU's de *smartphones* para análise comportamental e modelagem de perfis de direção dos condutores estudados. Por sua vez, Zhang et al. (2017) desenvolveram uma aplicação chamada *SafeDrive*, que faz uso das mesmas fontes de dados para a detecção de anomalias na direção do veículo em tempo real.

Apesar da atual facilidade de se obter dados de direção com fins de modelagem e identificação comportamental de condutores, os primeiros trabalhos com a temática faziam uso somente poucas variáveis de direção, muitas vezes simulados e tampouco usavam técnicas de aprendizado de máquina para tal. Como exemplo, Summala (2000) propõe um modelo de análise comportamental de condutores fazendo uso somente de informações de tempo de reação de frenagem em dados simulados. Além disso, a técnica usada é um sistema baseado em regras obtidas por meio de estatísticas que usam certos limiares no tempo de atuação do freio para determinar o que os autores chamam de situações inesperadas. De maneira semelhante, Sathyanarayana, Boyraz e Hansen (2008) propõem a aplicação de métodos probabilísticos baseados em modelos ocultos de Markov para modelar comportamento de condutores, como detecção de distração, por meio de dados coletados pela interface OBDII. O modelo proposto faz uso somente de três variáveis OBDII sendo essas velocidade, ângulo do volante e leituras do pedal de aceleração e freio. Os autores enaltecem também o potencial das aplicações de modelagem comportamental de condutores e sua aplicação em sistemas de segurança veicular.

Com a padronização do uso do OBDII em todos os veículos a partir de 2010, a coleta de dados veiculares para estudos comportamentais se tornou mais acessível e propiciou a utilização de novas variáveis, que também foi somado ao uso de IMUs de *smartphones*, tal qual mencionado anteriormente. Como exemplo, Blaszczyk, Turek e Cetnarowicz (2014) realiza ex-

perimentos controlados, onde dois condutores são submetidos a um teste em circuito fechado. Os dados são coletados por meio da interface OBD-II e da plataforma embarcada Raspberry Pi, além do uso de IMUs embarcados em um *smartphone*, tendo o objetivo modelar o estilo de direção de cada condutor. Foram aplicados modelos probabilísticos de classificação com erro obtido relativamente baixo. Contudo, é importante ressaltar que somente dois condutores são comparados e em condições controladas de captura dos dados, o que distancia os experimentos de situações reais. Já fazendo uso de técnicas de aprendizado de máquina, Kumtepe, Akar e Yuncu (2016) propõem um modelo que faz a fusão de dados provenientes do OBDII e câmeras alocadas no interior do veículo com o objetivo de se decidir se o condutor apresenta sinais de agressividade ou desatenção. Essas informações são usadas para formar o vetor de características que representam o comportamento do condutor e então são submetidos a uma máquina de vetores de suporte (SVM) de modo a classificar se o condutor testado apresenta os sinais comportamentais estudados.

Os exemplos de aplicação supracitados indicam a aptidão da aplicação de técnicas de aprendizado de máquina em dados veiculares para modelagem e identificação comportamental de condutores. De modo condensado, a revisão bibliográfica elaborada por Meiring e Myburgh (2015) aborda quais algoritmos de aprendizado de máquina são mais adequados para análise de estilos de direção e comportamento de condutores. São elencados tipos de comportamentos de direção, bem como as causas e consequências de cada um deles e aponta-se o potencial de tais algoritmos em detectar tais condições e prevenir casualidades proporcionadas pelas mesmas, baseado em diversos trabalhos na literatura. É também realizada uma enumeração de técnicas de aprendizado de máquina e suas aplicações mais comuns, as quais são apresentadas na Tabela 2.1. A aplicação de técnicas de aprendizado de máquina em dados veiculares se mostram promissores para criação de aplicações práticas em modelagem e identificação comportamental de condutores. Entretanto, grande parte das aplicações de modelagem e identificação comportamental de condutores são voltadas para detecção de situações de agressividade, que podem acarretar em situações de insegurança aos usuários do veículo e à terceiros. O treinamento de técnicas de aprendizado de máquina para este tipo de aplicação pode ser considerado de simples replicação, uma vez que os padrões de direção que regem um estilo agressivo ou sonolento são comuns entre diversos condutores. Outra aplicação de modelagem comportamental de condutores é a autenticação e identificação de condutores, vertente explorada por este trabalho. Este tipo de aplicação é mais complexa, visto que não existe um padrão único a ser detectado, mas

sim o padrão de direção inerente de cada condutor, o que dificulta sua aplicação em escala. Na próxima seção são apresentadas as definições e exemplos relacionados à autenticação de condutores e sua dificuldades de implementação.

Tabela 2.1 – Técnicas de inteligência computacional e suas aplicações em veículos inteligentes.

Técnica	Aplicações
Redes Neurais Artificiais	Detecção de sonolência e distração, predição do comportamento do volante, visão computacioal.
Fast Fourier Transform	Detecção de sonolência.
Clusterização	Distinção de estilo de direção e rotulação de condutor.
Clusterização K-means	Identificação individual de condutor e monitoramento de condições de rota.
Máquina de estados	Reconhecimento de manobras.
Máquina de estados finitos	Modelagem de tomadas de decisão do condutor.
Máquina de estados híbridos	Veículos Autônomos.
Lógica Fuzzy	Detecção de fadiga, identificação de distração, métodos de pontuação e métodos de reconhecimento de estilo de direção.
Modelos Ocultos de Markov (HMM)	Estimação de comportamento do condutor, reconhecimento de manobras, análise de performance de direção e identificação de distração.
Técnicas Bayesianas	Estimação de comportamento do condutor em situações de dados faltantes.
Árvores de decisão	Estimação de confiança de resultados em fusão de dados para detecção de sonolência.
Modelo de misturas de gaussianas	Identificação de distração, reconhecimento de manobras e monitoramento de condições de rota.
Dynamic time warping (DTW)	Classificação de perfil de risco do condutor e assistentes de direção ou alerta de segurança.
Filtros de Kalman	Predição de processos e modelagem de comportamento humano.
Máquinas de vetor de suporte (SVM)	Métodos de reconhecimento de estilo de direção, detecção de sonolência e estimação de estado do veículo.
Algoritmos Genéticos	Calibração de processos de modelagem de detecção de veículos próximos.

Fonte: Adaptado de Meiring e Myburgh (2015).

2.2 Autenticação e Identificação de Condutores

Autenticação de condutores é a funcionalidade que um sistema tem em definir se uma determinada pessoa pertence ao grupo dos autorizados a operar um veículo. Por sua vez, identificação de condutores é definida como a funcionalidade que um sistema tem de identificar um condutor específico entre um grupo de condutores que estão aptos a operar um determinado

veículo. Apesar de ser uma área com interesse recente, em 2005 foi implementado por Wakita et al. (2005) um sistema de identificação de condutores por meio de métodos estatísticos para um grupo de trinta condutores com taxas de identificação de 73% e em Miyajima et al. (2006) abordagem semelhante foi adotada, com a inclusão da análise cepstral dos dados de direção de modo a melhorar o desempenho para 89,6%. No entanto, ambos os casos foram testados com dados simulados.

A evolução dos meios de coleta de dados veiculares permitiu o uso de dados de direção reais para implementação de modelos de identificação de condutores. Como exemplo, dados coletados por meio da porta OBDII são utilizados por Campo et al. (2014) para a tarefa de identificação do condutor, onde os dados foram submetidos a análise cepstral e em seguida submetidos a uma rede neural artificial com taxa de identificação para um grupo de três condutores de 84,6%. Por sua vez, Burton et al. (2017) realiza a autenticação de condutores por meio de análise onde um modelo *one class* SVM (máquina de vetores de suporte, do inglês: *support vector machine*) para identificar intrusos no veículo em uma população de dez condutores de teste com dados de direção simulados. Usando somente dados provenientes do OBDII de seções de direção de trinta condutores distintos, Wang et al. (2017) fez uso de Florestas Aleatórias para a identificação de condutores, onde, com seis minutos de dados de cada condutor para treinamento do modelo, chegam a 100% de acurácia na identificação em alguns experimentos. A metodologia adotada por Kwak, Woo e Kim (2016b) envolve o teste de pré-processamento dos dados de direção no desempenho de diversos modelos de identificação, tais como o tamanho da janela de extração de características ideal para minimizar o efeito de flutuação da leitura dos sensores, atingindo uma acurácia superior a 99% na identificação em um grupo de dez condutores.

Trabalhos recentes exploram outros panoramas relacionados à identificação e autenticação de condutores. Em uma análise de eventos como o ato de abrir e fechar portas, uso do cinto de segurança, seleção de marcha, entre outros, Kar et al. (2017) realiza a identificação do condutor mesmo antes da movimentação do veículo, atingindo uma acurácia de 91% de identificação nos primeiros 20 segundos de acionamento do veículo em um conjunto de vinte e quatro condutores. Já Jafarnejad, Castignani e Engel (2017) explora o impacto no número de condutores do desempenho dos modelos de identificação, mostrando que quanto maior a população de condutores conhecidos pelo modelo, menor a capacidade desse discernir entre os condutores. Um modelo baseado em pré-processamento de dados e o algoritmo *Extremely*

Randomized Trees (Extra-Trees), formado por etapas de autenticação e subsequente identificação foi implementado por Rettore et al. (2018), com objetivo de identificar intrusos e também gerar configurações personalizadas de funções do veículo para cada usuário identificado. Esse trabalho também utiliza sensores virtuais, como forma de contornar a situação onde veículos pouco sofisticados são providos de menos sensores.

Contudo, todos os trabalhos supracitados têm um fator limitante para aplicações práticas de um sistema de autenticação/identificação de condutores: a necessidade de treinamento do modelo a cada novo condutor que possa a vir conduzir o veículo. Tem-se o fato de que o poder de processamento das centrais do veículo é limitado, o que poderia demandar a implementação de um *hardware* mais robusto, que elevaria os custos de tal sistema. Sendo assim, este trabalho propõe o uso de um modelo treinado que não necessite de posterior adaptação mesmo em casos onde o novo condutor é desconhecido pelo modelo. Este modelo é feito por meio do uso de redes neurais siamesas, conhecidas por sua aplicação em reconhecimento facial e em casos onde o número de dados por indivíduo é limitado ou desconhecido, tais como aplicações de *zero/one/few shot learning*. No próximo capítulo serão apresentados os conceitos gerais acerca deste trabalho, com uma breve enfase às redes neurais siamesas, onde é demonstrado seu princípio de funcionamento e como essa técnica pode ser aplicada no problema de autenticação de condutores.

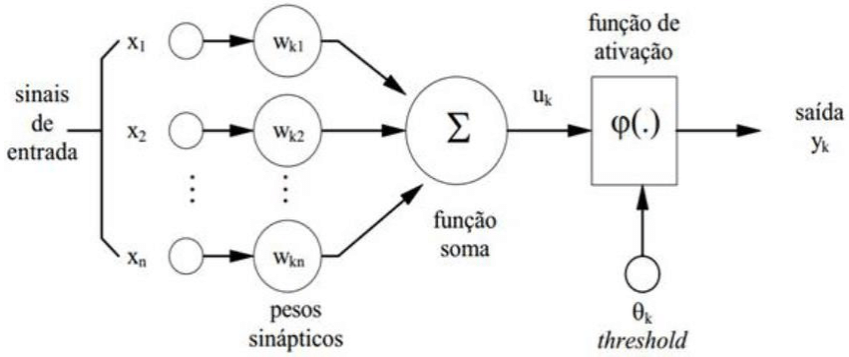
3 CONCEITOS GERAIS

Neste capítulo são apresentados conceitos referentes às redes neurais artificiais e à topologia siamesa e como esta pode ser aplicada ao problema de autenticação de condutores endereçado. Também são brevemente apresentados conceitos de técnicas secundárias, isto é, que contribuíram para o desenvolvimento e avaliação do modelo, mas não contribuem diretamente para a resolução do problema. Por fim, é feita uma breve explanação referente às métricas que foram utilizadas para a avaliação e validação do modelo de autenticação de condutores proposto.

3.1 Redes Neurais Artificiais

Umas das mais difundidas técnicas de aprendizado de máquina, as redes neurais artificiais (RNAs) são amplamente empregadas na solução de diversos problemas de regressão e classificação, atribuindo seu sucesso à sua flexibilidade de síntese de mapeamento multidimensional não-linear de variáveis dependentes e independentes. Isso, devido a sua capacidade de aproximação universal de qualquer função f^*. RNAs tradicionais são formadas por conjuntos de diversos neurônios artificiais, propostos por McCulloch e Pitts em 1943. Este modelo de neurônio consiste em receber sinais de entrada X nos dendritos do neurônio para retornar um único sinal de saída y no axônio do mesmo. Este modelo é apresentado na Figura 3.1.

Figura 3.1 – Representação de um neurônio artificial não linear.



Fonte: Adaptado de Haykin et al. (2009).

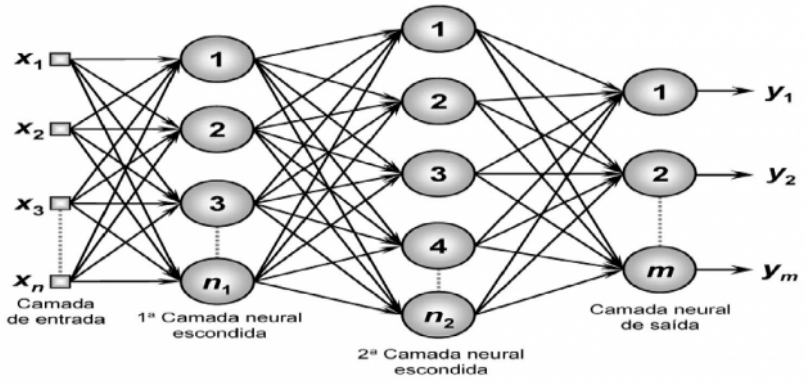
A expressão que representa este neurônio, com função de ativação unitária, é apresentada pela Equação 3.1:

$$y_k = \varphi(u_k) = \varphi(\sum_{i=1}^{n} w_i x_i + b), \tag{3.1}$$

onde x_i são as entradas dos neurônios, w_i são os pesos das entradas, b é o *bias* e y é a saída para uma função de ativação unitária φ. Observa-se que, sem a função de ativação um perceptron, seria apenas um modelo linear, sendo assim, a função de ativação introduz não-linearidade ao perceptron além de saturar a saída da função soma em certos limiares que vão de acordo com a função de ativação escolhida. Dentre as funções mais utilizadas para esta operação estão as funções ativação linear retificada (ReLu), sigmoidal, linear, tangente hiperbólica, logarítmica e senoidal. Outro ponto importante das funções de ativação é que estas permitem a combinação de vários perceptrons em forma de rede, sem que seja possível reduzir a rede em um único modelo linear.

Tal qual mencionado anteriormente, combinações destes perceptrons formam uma Rede Neural Artificial. Dentre as muitas arquiteturas utilizadas, a mais comum é a rede com múltiplas camadas. Haykin et al. (2009) definiu estas redes como sendo um conjunto de unidades sensoriais que constituem a camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Estas redes são conhecidas como Perceptron de Múltiplas Camadas (*Multilayer Perceptron* - MLP) (ROSENBLATT, 1962). Na Figura 3.2, é apresentado um modelo de uma MLP com duas camadas escondidas (*hidden layers*), camadas essas que determinam a profundidade da rede. A camada final é conhecida como camada neural de saída (*output layer*) . Este tipo de modelo também é conhecido como redes neurais *feedforward*, porque a informação segue um caminho direto, onde a saída de cada neurônio alimenta os neurônios seguintes até que se obtenha a saída y. Não existe nenhum tipo de retroalimentação na topologia *feedforward*, sendo que, quando essa retroalimentação existe, estas redes passam a se chamar redes neurais recorrentes, que são úteis para determinados tipos de problemas, como séries temporais e aplicações em sistemas de controle.

Figura 3.2 – Representação de uma MLP com duas camadas escondidas.



Fonte: Adaptado de Haykin et al. (2009).

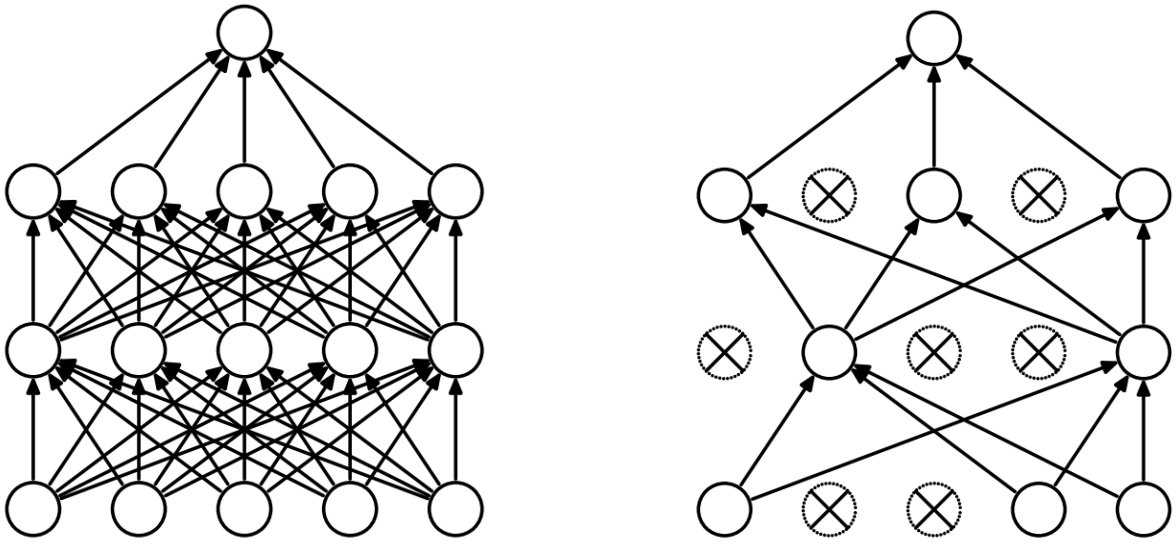
O aprendizado de uma rede neural artificial se dá por meio do algoritmo de treinamento *backpropagation*, introduzido por Rumelhart et al. (1988). Em cada época de treinamento, *backpropagation* calcula o erro proveniente da seguinte função custo:

$$C = \frac{1}{2n} \sum_{x} \|y(x) - \hat{y}(x)\|^2, \tag{3.2}$$

onde n é o número de exemplos de treinamento e o somatório se dá sobre cada exemplo de treinamento x; $y(x)$ é a saída correspondente desejada de x; $\hat{y}(x)$ é o valor predito pela rede após todos os cálculo de ativação. Como deseja-se diminuir o erro a cada época de treinamento é aplicado um algoritmo de otimização do tipo gradiente descendente e suas variantes (GOODFELLOW; BENGIO; COURVILLE, 2016). Para tal é necessário que a função de ativação seja derivável. O cálculo dos gradientes locais δ pela função custo proporciona o ajuste dos pesos por meio da chamada Regra Delta Generalizada que é ponderada por alguns parâmetros, como a taxa de aprendizado, que podem influenciar de maneira efetiva no desempenho da rede.

Existem outras maneiras de otimizar o desempenho do treinamento de redes neurais artificiais de modo que se garanta a convergência do treinamento. Uma dessas técnicas é o *dropout*, que consiste, durante o treinamento, no desligamento aleatório de alguns neurônios nas etapas *forward* e *backward*. Este mecanismo, tal qual exemplificado na Figura 3.3, previne sobre-ajuste da rede, fazendo com que a rede aprenda características específicas dos dados, que se tornam mais úteis para a tarefa em questão, uma vez que em cada época, cada subconjunto de neurônios treinado observa de maneira diferente os dados (SRIVASTAVA et al., 2014). Em Nielsen (2015) e Goodfellow, Bengio e Courville (2016) é possível obter uma explanação mais detalhada do funcionamento dos conceitos supracitados.

Figura 3.3 – Exemplo de aplicação da técnica *dropout*.



(a) Rede neural padrão. (b) Após aplicação do *dropout*

Fonte: Adaptado de Srivastava et al. (2014).

Nos últimos anos, sua ampla aplicação tem sido na forma de redes neurais profundas (*deep learning*). Por convenção, *deep learning* se refere à redes neurais com mais de duas camadas de neurônios, sem considerar a camada de saída da rede. Até um passado recente, existia certas dificuldades de se treinar redes com topologias mais profundas, não somente relacionadas ao poder de processamento dos computadores, mas também à problemas oriundos do treinamento, como o gradiente desvanecendo ou explodindo. O primeiro problema foi contornado a medida em que o poder de processamento de CPUs foi aumentando de acordo com a Lei de Moore, e também com a descoberta da capacidade de realização de cálculos complexos por meio de unidades de processamento gráficos (GPUs) (GPUs - do inglês *graphic processing unit*) (NETO; LACERDA; SIMÃO, 2015). O problema do gradiente desvanecente é oriundo da regra da cadeia da diferenciação, que é usada no treinamento da rede. Cada camada de neurônios introduzida na rede, resulta em mais uma parcela na multiplicação da regra. Se essas derivadas forem maiores que zero, o produto explode para um número muito grande (menos comum), e se menores que zero, levam a um resultado muito próximo de zero (mais comum). Isso torna o acréscimo de camadas às redes neurais arriscado, o que, apesar de trazer mais eficiência ao desempenho da tarefa dedicada, torna mais difícil seu treinamento. Este problema do gradiente ainda é um desafio a ser contornado para a garantia de sucesso do treinamento de redes neurais profundas.

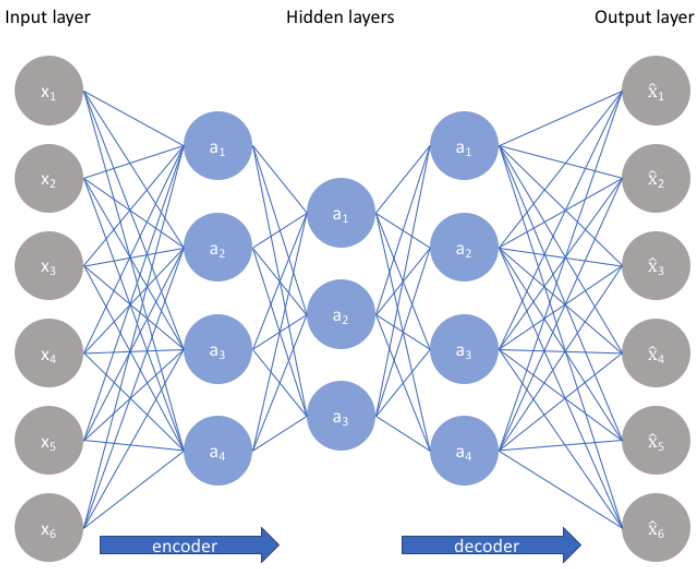
Aplicações de redes neurais profundas atualmente são o estado da arte em diversos campos de estudos. Uma dessas aplicações em evidência é no processamento de imagens, no qual o uso de redes neurais convolucionais, que são redes que aplicam em uma ou mais camadas a operação de convolução no lugar da multiplicação de matrizes tradicional, tem potencializado a detecção de objetos, reconhecimento facial, entre outras aplicações (LECUN; BENGIO et al., 1995). Este tipo de aplicação resultou num novo conceito chamado de *transfer learning* (transferência de aprendizado), que foi resultante a partir de redes convolucionais treinadas em uma grande massa de imagens, como a ImageNet que possui aproximadamente 3,2 milhões de imagens rotuladas (DENG et al., 2009). Estas redes treinadas, como a ResNet, VGG-16, AlexNet, Inception, entre outras, têm camadas convolucionais ocultas que são capazes de realizar uma extração de características (*embeddings*) e entregam às camadas de saída uma transformação das imagens de entrada, facilitando a tarefa de classificação especificada (PAK; KIM, 2017). Sendo assim, *transfer learning* consiste no uso de camadas intermediárias de redes profundas já treinadas em outras aplicações, diferentes das originais, de modo que durante o treinamento, estas camadas são travadas e somente os pesos das camadas de saída são ajustado no treinamento de acordo com os alvos desejados.

Em paralelo ao uso de redes convolucionais profundas surgiram diversas topologias que são treinadas com o propósito de serem empregadas na extração de características em um campo de aprendizado de máquina conhecido como aprendizado por representação (NARGESIAN et al., 2017). Aprendizado por representação consiste no uso de técnicas supervisionadas ou não com a finalidade de se obter representações úteis dos dados de entrada que facilitem a construção de classificadores ou outros preditores. Esse tipo de técnica tem sido aplicada em tarefas que envolvem dados não estruturados, como reconhecimento de voz e assinaturas e processamento de linguagem natural (NLP). Nelas, os dados originais são transformados em *embeddings*[1] que permitem o uso desses dados em técnicas convencionais de aprendizado de máquina. Dentre essas topologias pode-se destacar a aplicação de *auto-encoders*, tal qual representado na Figura 3.4, que consistem em redes neurais artificiais que buscam durante seu treinamento criar representações aprimoradas dos dados de entrada (*encoder*) tal que a camada de saída consiga recriar os dados de entrada (*decoder*). O *encoder* desta rede pode ser aplicado como uma

[1] No presente texto, o termo *embedding* refere-se a um mapeamento de dados para vetores de números reais. As dimensões individuais nestes vetores normalmente não possuem significado inerente. Em vez disso, são os padrões gerais de localização e distância entre os vetores dos quais técnicas de aprendizado de máquina tiram proveito.

camada de extração de características que podem ser usadas por outros classificadores (TSCHANNEN; BACHEM; LUCIC, 2018).

Figura 3.4 – Topologia de um *Auto-Encoder*.



Fonte: Jeremy Jordan (2018).

Ciente desse potencial que redes neurais artificias têm de aprender representações de dados que simplificam uma dada tarefa, existem algumas topologias de redes neurais que objetivam desempenhar algumas tarefas específicas, como as de aprendizado por similaridade. Esse tipo de aplicação busca discriminar o quão parecidas são duas instâncias relativas a uma função de similaridade angular ou baseada em distância. A técnica mais empregada para esse tipo de problema é conhecida como redes neurais siamesas, que consiste em duas redes neurais que compartilham os mesmos parâmetros e uma função de similaridade como camada de saída. Essa rede recebe como entrada pares de dados de modo a determinar se eles pertencem a uma mesma classe. O problema abordado por este trabalho, de autenticação de condutores de veículos, se adéqua bem à proposta das redes neurais siamesas de aprender a similaridade entre classes. Sendo assim, na próxima seção serão detalhados os conceitos relacionados às redes neurais siamesas e como estas podem ser aplicadas para o problema de autenticação de condutores.

3.1.1 Redes Neurais Siamesas

Redes neurais siamesas têm sido aplicadas na resolução de diversos problemas que envolvem a determinação do grau de similaridade entre dois objetos. Grande parte destas aplica-

ções envolvem o reconhecimento de similaridade entre imagens. Do ponto de vista de aplicação, sua implementação original foi para verificação de assinaturas e determinar se pertencem a mesma pessoa (BROMLEY et al., 1994). A partir daí seu uso se popularizou para aplicações de reconhecimento e/ou verificação de faces de pessoas ou objetos, reconhecimento de voz, entre outras aplicações listadas por Harandi, Kumar e Nock (2017). Da perspectiva do aprendizado, redes neurais siamesas foram empregadas com sucesso como medida de similaridade (NECULOIU; VERSTEEGH; ROTARU, 2016; MUELLER; THYAGARAJAN, 2014; SHAHAM; LEDERMAN, 2015), *hashing*[2] (MASCI et al., 2012) e *zero/one/few shot learning*, que no contexto deste trabalho, é a que melhor se aplica à realidade de autenticação de condutores.

Zero/one/few shot learning consiste na aplicação de redes neurais siamesas treinadas com um determinado conjunto de pares de dados que naturalmente mensura o grau de similaridade entre as entradas. Uma vez treinada, a rede pode ser usada para criar *embeddings* que geram um espaço de características com grande poder de generalização, não somente de novos dados, mas também de novas classes, que, no caso deste trabalho, são novos condutores que não são utilizados na fase de treinamento da rede neural siamesa (KOCH; ZEMEL; SALAKHUTDINOV, 2015). Um exemplo de aplicação de *zero/one/few shot learning* é no caso de identificação de autoria de um determinado texto (PUSHP; SRIVASTAVA, 2017). Em (QIAN; HE; ZHANG, 2016), a abordagem para a resolução deste problema consiste no treinamento de uma rede siamesa com um conjunto de textos e quando deseja-se identificar se um determinado texto pertence a um mesmo autor é submetido à rede neural siamesa um texto que sabidamente pertence ao autor em questão e o outro que se deseja autenticar, obtendo acurácia média de 99,8%. *One shot learning* é quando se tem somente uma amostra rotulada da classe que se deseja autenticar e quando se possui mais de uma amostra rotulada se trata de *few shot learning*. *Zero shot learning* é um caso mais extremo, onde não se tem nenhuma amostra rotulada do que se deseja identificar e o modelo busca dentre suas classes conhecidas a que mais se aproxima da amostra desconhecida.

Redes neurais siamesas, introduzidas por Bromley et al. (1994), é uma arquitetura, tal qual mostrada na Figura 3.5, que consiste em duas redes neurais que compartilham de pesos idênticos ligadas por uma ou mais camadas. Na maioria dos casos, uma rede neural siamesa executa uma codificação não linear dos dados de entrada com o objetivo de atingir um espaço semanticamente significativo onde padrões relacionados sejam próximos uns dos outros (tais

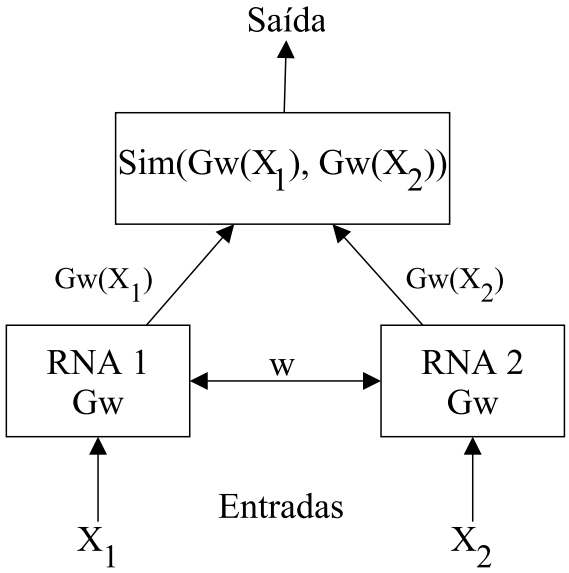
[2] Uma função *hash* é um algoritmo que mapeia dados de comprimento variável para fixo

como faces de pessoas, assinaturas, entre outros) e os não relacionados sejam distantes uns dos outros (HARANDI; KUMAR; NOCK, 2017). Uma rede neural siamesa recebe como entrada um par de leituras, tanto no treinamento quanto no teste, com o objetivo de desenvolver similaridade entre pares de uma mesma classe Y, a classe genuína, e de distanciar pares de dados de classes diferentes, a classe impostora. A partir do treinamento, a rede cria um espaço multidimensional (*embeddings*) $G_w(X)$, de dimensão igual ao número de neurônios de saída da rede neural base, rede essa que é compartilhada entre os pares de entrada. A representação espacial de cada entrada então é submetida à uma função de similaridade $Sim(G_w(X_1), G_w(X_2))$, normalmente sendo uma medida de distância Euclidiana, obtida pela Equação 3.3:

$$Sim(G_w(X_1), G_w(X_2)) = \sqrt{[G_w(X_1) - G_w(X_2)]^2}, \tag{3.3}$$

onde $G_w(X_1)$ e $G_w(X_2)$ são dois pontos nos *embeddings* criados pelos parâmetros compartilhados W quando mapeiam as entradas X_1 e X_2.

Figura 3.5 – Arquitetura de uma Rede Neural Siamesa.



Fonte: Adaptado de Chopra, Hadsell e LeCun (2005).

Uma vez obtida a medida de similaridade, é definido se os dados de entrada pertencem ou não à uma mesma classe, dado um limiar de distância definido. Como função de custo ou perda (*loss function*), redes neurais siamesas fazem uso da Perda Contrastiva (*Contrastive Loss)* para o treinamento. Essa função perda, introduzida por (CHOPRA; HADSELL; LECUN, 2005) e que possibilitou um treinamento aprimorado dessa topologia, é calculada por meio do

somatório das perdas individuais para pares genuínos e impostores. Quando pares genuínos estão muito distantes, esses são penalizados por L_G. Por sua vez, pares impostores que estão dentro do valor limiar de distância, são penalizados por L_I. As redes irmãs têm seus pesos atualizados via *backpropagation*. Então a cada época de treinamento, pares genuínos são atraídos a um sub-espaço próximo, enquanto pares impostores são mantidos a uma distância acima da margem definida. A função de perda contrastiva é definida pelas Equações 3.4, 3.5 e 3.6:

$$L_G = (1 - Y_A)Y_P^2, \tag{3.4}$$

$$L_I = Y_A(max(M - Y_P, 0))^2, \tag{3.5}$$

$$L = L_G + L_I, \tag{3.6}$$

onde Y_A e Y_P são, respectivamente, o valor real dos pares e o retornado pela rede neural siamesa. Y_A e Y_P são valores binários que são iguais a 1 para pares genuínos e 0 para impostores. M é o valor de margem da distância que define quais pares são genuínos ou impostores (MARTIN et al., 2017).

Para que o treinamento de uma rede neural siamesa seja adequado é necessário que a construção seja feita de tal modo que as classes (iguais e diferentes) tenham proporções balanceadas e que pares diferentes sejam formados pela a combinação de todas as outras classes (HARANDI; KUMAR; NOCK, 2017). Outro ponto de cuidado é com a magnitude dos valores dos *embeddings* criados em cada rede antes de serem submetidos à função de similaridade. Isto se deve ao fato de nas ativações do tipo ReLu ou Leaky ReLu não existe saturação dos valores (porém não é conveniente que exista essa saturação dos *embeddings*, tal como ocorreria um funções de ativação do tipo sigmoidal, uma vez que essa saturação pode reduzir muito o espaço no qual os *embeddings* podem habitar). Para contornar tal situação, Schroff, Kalenichenko e Philbin (2015a) utiliza entre as saídas de cada rede irmã e a função de similaridade uma normalização do tipo L2 ou norma Euclidiana, que fazem com que a margem M possa ter valor constante durante todo o treinamento e em posteriores aplicações.

Uma vez apresentados os conceitos básicos de uma rede neural siamesa e sua aptidão para problemas que envolvem identificação se determinados pares de dados pertencem a uma mesma classe ou não, analisa-se agora sua aptidão para desempenhar a tarefa de autenticação

de condutores. Tal como elucidado anteriormente, as soluções para autenticação de condutores por meio de dados veiculares baseadas em técnicas de aprendizado de máquina se mostraram viáveis para a execução da tarefa em questão. Porém, os modelos utilizados foram treinados e testados em um mesmo grupo de condutores, ou seja, para aplicações destes modelos em novos condutores existe a necessidade de retreinamento do modelo, o que pode dificultar sua aplicação prática, por limitações diversas, como poder computacional limitado e necessidade de coleta de um volume de dados considerável para tal retreinamento. Diante de tal cenário, a aplicação de redes neurais siamesas se mostra promissor, uma vez que sua construção é voltada para identificação de pares de dados. A aplicação das redes neurais siamesas em forma de *one/few shot learning* se mostra uma opção interessante para a identificação de condutores que não foram usados no treinamento da rede neural siamesa, uma vez que esta aplicação não necessita que uma classe que se deseja autenticar esteja presente no treinamento. Sendo assim, o presente trabalho tem como objetivo aplicar redes neurais siamesas para a autenticação de condutores que não foram usados no treinamento da mesma, abordagem até então inédita para a temática de autenticação de condutores por meio de dados veiculares naturalísticos.

3.2 Técnicas Secundárias

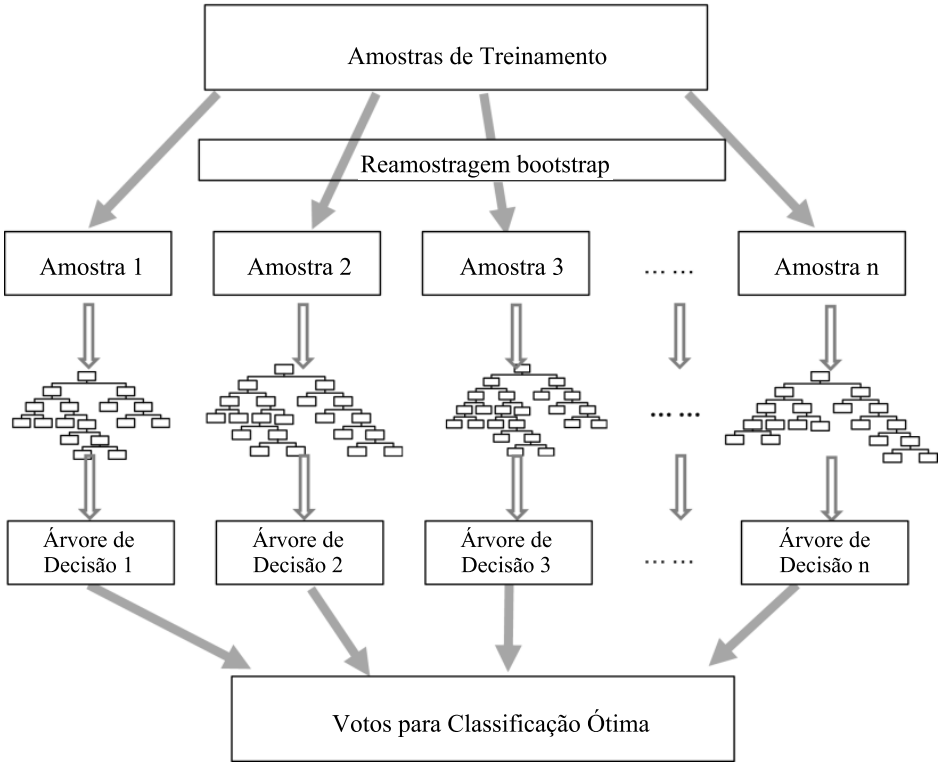
Nesta seção são apresentas técnicas que foram utilizadas para o desenvolvimento ou avaliação do mesmo do sistema proposto. Estas técnicas foram importantes em etapas preliminares, como seleção de características, e como forma de testar o desempenho do modelo, seja para visualização dos dados ou na análise dos mesmos.

3.2.1 Florestas Aleatórias

Florestas Aleatórias (tradução literal de *Random Forest*) é uma técnica de inteligência computacional para classificação e regressão introduzida por Breiman (2001) e que consiste no agrupamento de diversas árvores de decisão (SAFAVIAN; LANDGREBE, 1991) de maneira que sua estrutura seja composta de forma aleatória (ver Figura 3.6). Em árvores de decisão comuns, cada nó é dividido de modo a se obter a melhor divisão entre as variáveis do problema em questão. Por sua vez, nas florestas aleatórias cada nó é dividido usando o melhor entre os subconjuntos de preditores escolhidos aleatoriamente no nó em questão. A técnica em si requer somente a configuração de dois parâmetros de entrada para a geração do modelo de predição:

o número de árvores de decisão desejadas (n_{tree}) e o número de variáveis de predição (m_{try}) usados em cada nó para o crescimento da árvore.

Figura 3.6 – Representação de uma Floresta Aleatória com *n* Árvores de Decisão.



Fonte: Adaptado de Zhang et al. (2017).

Essa estratégia produz resultados satisfatórios se comparados a outros classificadores, incluindo análise discriminante, máquinas de vetor de suporte e redes neurais artificiais, além de ser robusto contra o problema de *overfitting*[3] (BREIMAN, 1999) e possuir boa imunidade ao ruído gaussiano (ZHANG et al., 2017). Porém, apesar dessa imunidade ao *overfitting*, as florestas aleatórias tendem à saturação do erro final, isto é, o erro se estagna em um ponto mesmo com a inserção de novas árvores à floresta. Um dos produtos gerados pelas florestas aleatórias é uma ordenação de quais variáveis são mais importantes para a classificação de um determinado alvo, que é mensurado por meio do ganho ou importância que cada variável tem para a classificação de uma amostra. Nesse trabalho, esta técnica foi utilizada com o intuito de elencar a importâncias das variáveis disponíveis no *dataset* utilizado e uma possível remoção daquelas com pouco ganho. A metodologia para essa seleção será detalhada na Seção 4.3.

[3] Ou sobre-ajuste: quando um modelo se ajusta satisfatoriamente ao conjunto de dados treinado, mas se mostra ineficaz para prever novos resultados.

3.2.2 Análise de Componentes Principais

Análise de Componentes Principais (PCA) é um procedimento matemático que usa uma transformação ortogonal para converter um conjunto de variáveis possivelmente correlatadas em um conjunto de variáveis não correlacionadas chamadas de componentes principais (WOLD; ESBENSEN; GELADI, 1987). Sejam $x_t (t = 1, \ldots, l$ e $\sum_{t=1}^{l} x_t = 0)$ um conjunto de vetores de entrada de m dimensões $x_t = (x_t(1), x_t(2), \ldots, x_t(m))^T$, então PCA transforma linearmente x_t em um novo vetor s_t pela seguinte expressão:

$$s_t = U^T x_t, \tag{3.7}$$

onde U é uma matriz ortogonal de $m \times m$ dimensões no qual a i-ésima coluna u_i é o i-ésimo autovetor da matriz de covariância C. Portanto, PCA primeiro soluciona o problema dos autovalores,

$$\lambda_i u_i = C u_i, \qquad i = 1, \ldots, m, \tag{3.8}$$

onde λ_i é o i-ésimo autovalor de C e u_i é seu o autovetor relativo. Então, os componentes de s_t, baseado no obtido em u_i, são calculados como a transformação ortogonal de x_t por

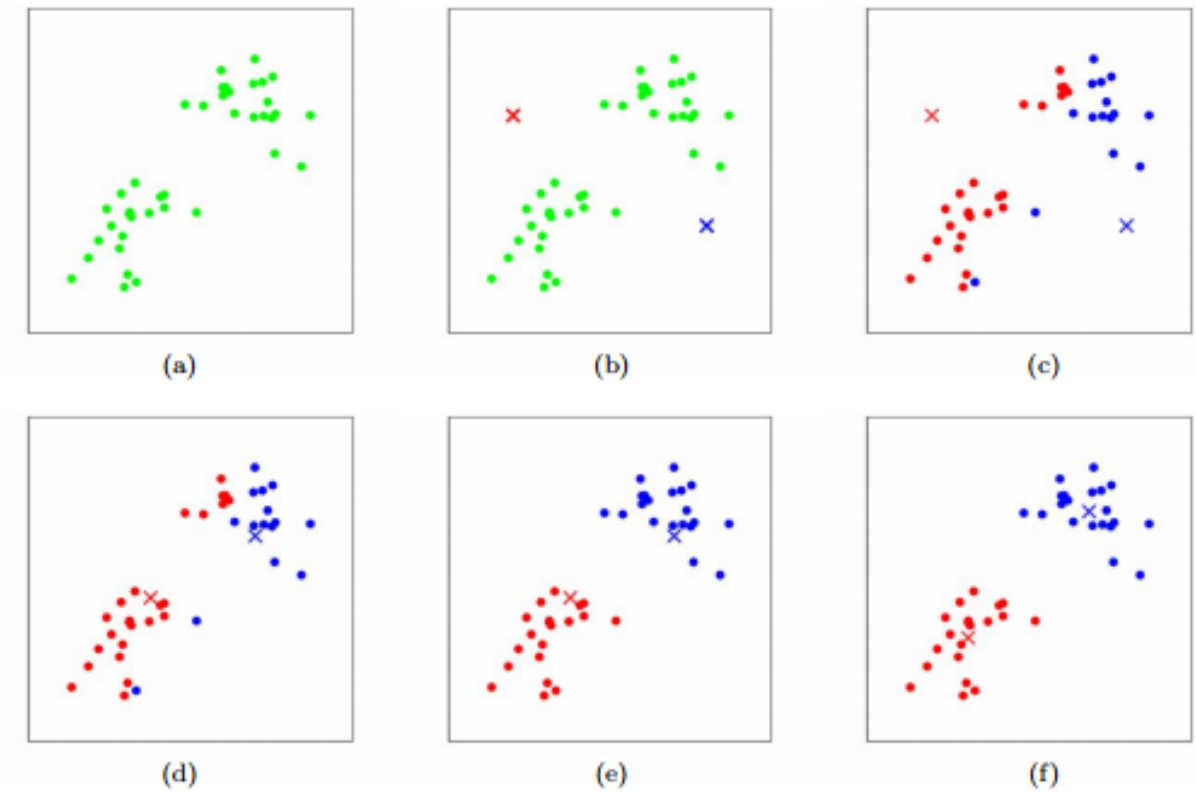
$$s_t(i) = {u_i}^T x_t, \qquad i = 1, \ldots, m, \tag{3.9}$$

Estes novos componente $s_t(i)$ são os componentes principais. Tal qual a Equação 3.9 demonstra, o número de componentes principais de s_t pode ser reduzido pelo uso de somente alguns dos primeiros autovetores ordenados na ordem descendente dos autovalores. Assim, PCA tem a característica de reduzir a dimensão de um conjunto de dados que pode ser aplicado na visualização de dados de dimensões elevadas, utilizando-se as duas primeiras componentes deste conjunto de dados originais (CAO et al., 2003). Neste trabalho, PCA é explorado na análise dos dados derivados da rede neural siamesa, que executa uma transformação não linear dos dados de entrada em um espaço derivado simplificado, que possibilita a visualização dos dados de alta dimensionalidade em uma representação bidimensional.

3.2.3 Agrupamento *K-means*

K-means (MACQUEEN et al., 1967) é uma das mais utilizadas técnicas de agrupamento (*clustering*) de dados não rotulados. Esta técnica busca retornar o agrupamento ideal dos dados baseado na similaridade entre os mesmos dados com um número de grupos desejados baseados em centroides. O algoritmo por detrás do *K-means* se dá de forma iterativa e começa pela inicialização aleatória ou heurística dos K centroides que serão o centro de cada um dos grupos. É calculada a distância de cada ponto a cada um dos centroides e cada ponto é associado ao centroide mais próximo. Na iteração seguinte é calculado o novo centroide de cada grupo e um novo cálculo das distâncias até que o nível de convergência seja satisfatório de acordo com o método de otimização utilizado (número máximo de iterações ou pouca variação dos valores dos centroides). Na Figura 3.7 é apresentado um exemplo simples do funcionamento do algoritmo *K-means*, onde os pontos são os dados e os centroides são marcados em x. Em (a) tem-se os dados originais e em (b) é feita a inicialização aleatória dos centroides. De (c) até (f) representam-se duas iterações do algoritmo *K-means*, com o cálculo das distâncias e dos novos centroides até sua convergência.

Figura 3.7 – Exemplo de funcionamento do agrupamento *K-means* .



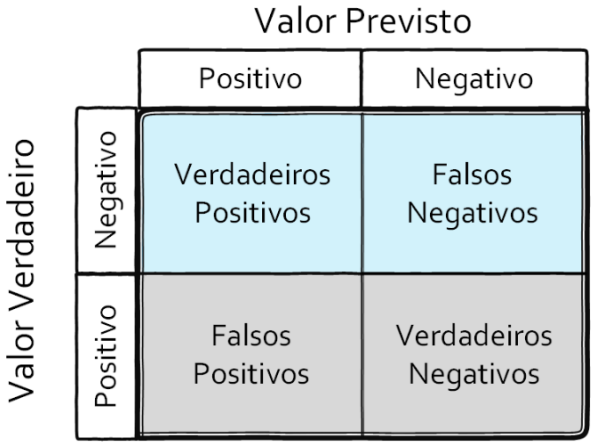
Fonte: Piech (2013).

A quantidade ideal de grupos para um determinado conjunto de dados pode ser feito executando o algoritmo diversas vezes e variado o número K de grupos e utiliza-se uma métrica de qualidade de um agrupamento. Usualmente aplica-se o escore de silhueta, que varia de -1 a 1 de acordo com a qualidade dos grupos encontrados. Um valor de silhueta maior que 0,71 indica que uma estrutura forte foi encontrada, entre 0,51 e 0,70 indica uma estrutura razoável, entre 0,26 e 0,50 que a estrutura é fraca e pode indicar que há necessidade de alterações nos hiper-parâmetros do *K-means* (iterações ou K) e menor ou igual a 0,25 indica que nenhuma estrutura efetiva foi encontrada. Uma explanação mais completa destes conceitos pode ser encontrada em Kaufman e Rousseeuw (2009). No contexto do presente trabalho, o agrupamento *K-means* é usado como uma forma de se avaliar a qualidade do espaço de dados gerado pela rede neural siamesa. Como os dados de um mesmo condutor deveriam estar num espaço próximo e separado dos demais, o agrupamento *K-means* e o escore de silhueta deveriam indicar que o número ideal de grupos deve ser igual ao número de condutores do conjunto de dados. Esta avaliação é robusta, uma vez que se trata de uma maneira não supervisionada de mensurar o desempenho da rede neural siamesa para a tarefa de autenticação de condutores.

3.3 Métricas de Validação do Modelo

Uma das partes mais importantes no desenvolvimento de um modelo é a avaliação e validação do desempenho do mesmo. É nesta etapa que é feita a comprovação de que os resultados obtidos são satisfatórios ou não, e possibilita a comparação de resultados obtidos por outros trabalhos. Uma das ferramentas mais utilizadas para avaliação de modelos é a matriz de confusão. Nela estão contidas todas as predições do modelo e a comparação destes com os alvos reais. A matriz de confusão opera da seguinte forma: considera-se os valores realmente positivos que o classificador previu como verdadeiros positivos (VP) e valores positivos em que o mesmo previu como negativos como falso negativo (FN), onde o mesmo se aplica à classe negativa, verdadeiro negativo (VN) e falso positivo (FP). Assim a matriz de confusão, tal qual mostrada na Figura 3.8, apresenta em termos numéricos ou percentuais esses valores de saída do classificador (SOUZA, 2009).

Figura 3.8 – Representação de elementos da Matriz de Confusão.



Fonte: Vaz (2018).

A partir da matriz de confusão é possível calcular diversas métricas que avaliam o desempenho do modelo. A primeira delas é a acurácia (*Accuracy*), demonstrada na Equação 3.10, que mensura o quanto dos valores previstos VP e VN foram preditos corretamente sobre o número total de predições. A acurácia é uma métrica interessante se as classes do problema forem balanceadas, uma vez que esta não leva em consideração a desproporção entre classes. Por sua vez, a precisão (*Precision*), demonstrada na Equação 3.11, é a métrica que leva em consideração o quanto o classificador acertou as predições classe positiva, isto é, a taxa de VP sobre todas as predições positivas. Esta métrica é útil quando a prevalência de um das classes é muito baixa e pode ser útil para detectar situações em que o modelo prediz a maioria das amostras como sendo da classe majoritária. Já a sensibilidade ou revocação (*Recall*), demonstrada na Equação 3.12, leva em consideração o quanto de VP são classificados como FN, que em certas aplicações podem ser altamente prejudiciais, como no caso do presente trabalho um possível impostor seja tratado como autêntico. Uma forma de sumarizar a precisão e a sensibilidade em uma única métrica é pelo índice F1-*Score*, que nada mais é que a média harmônica entre essas duas métricas, tal qual mostrado na Equação 3.13. Um F1-*Score* alto indica bom desempenho de ambas as métricas, enquanto um valor baixo indica uma baixa concordância nas predições do classificador, mesmo em situações de desbalanceamento entre as classes. F1-*Score* é uma métrica que complementa a acurácia, uma vez que é possível que um modelo obtenha predições com uma acurácia alta, mas com um F1-*Score* baixo, principalmente em casos de classes desbalanceadas. Quando as duas métricas tem um valor alto indica que o modelo tem um bom poder de generalização do problema para o qual foi treinado.

$$Accuracy = \frac{VP \times VN}{Total}, \tag{3.10}$$

$$Precision = \frac{VP}{VP + FP}, \tag{3.11}$$

$$Recall = \frac{VP}{VP + FN}, \tag{3.12}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{3.13}$$

Outra métrica observada é o coeficiente de concordância Kappa (*Cohen's Kappa Score*), que é utilizada para descrever a concordância entre os valores previstos e os verdadeiros, sendo também baseada na matriz de confusão. O coeficiente de concordância Kappa κ é calculado pela Equação 3.14:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}, \tag{3.14}$$

onde p_o é a concordância relativa observada entre os valores previstos e verdadeiros, sendo este valor idêntico à acurácia. Por sua vez, p_e é a probabilidade hipotética concordância, usando os valores da matriz de confusão para calcular as probabilidades de cada observador (que nesse caso são os valores previstos e verdadeiros) avaliar aleatoriamente cada categoria. p_e é definido pela Equação 3.15. seguinte maneira:

$$p_e = (VP + FN)^2 + (FP + VN)^2, \tag{3.15}$$

onde os valores de VP, FN, FP e VN são percentuais (divididos pelo total geral). Os valores de concordância podem ser interpretados conforme a Tabela 3.1. Uma explanação mais completa dessas métricas pode ser encontrada em (BANERJEE et al., 1999).

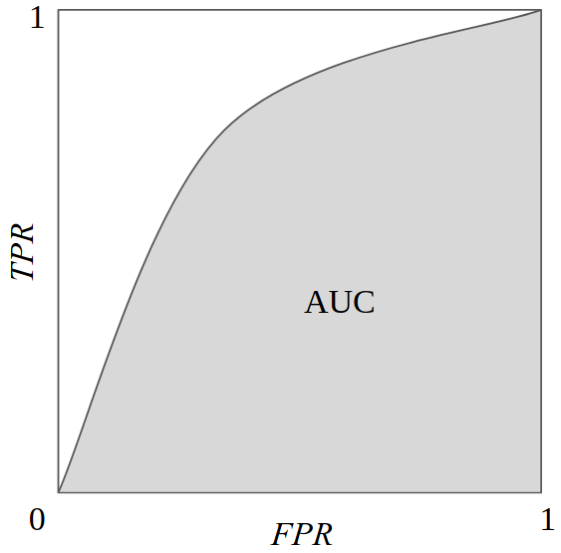
Tabela 3.1 – Interpretação dos valores de Kappa

Kappa	Interpretação
<0	Sem concordância
0-0,19	Concordância baixa
0,20-0,39	Concordância razoável
0,40-0,59	Concordância moderada
0,60-0,79	Concordância substancial
0,80-1,00	Concordância quase perfeita

Fonte: Adaptado de Banerjee et al. (1999).

Todas as métricas descritas acimas dependem que exista um limiar de separação entre as classes, que normalmente é fixado em 0,5 em uma escala de 0 a 1. Porém, em alguns casos o limiar que maximiza o desempenho do modelo pode ser diferente deste valor. Uma maneira eficiente de se avaliar o modelo independentemente do limiar é por meio da Curva Característica de Operação do Receptor (ROC). A curva ROC permite avaliar a variação da sensibilidade (taxa de verdadeiros positivos) e especificidade (taxa de falsos positivos) para diferentes valores de corte. A sensibilidade é a proporção dos verdadeiros positivos, isto é, a capacidade que o sistema tem em predizer as condições para os casos que realmente são verdadeiros. A especificidade é a proporção dos verdadeiros negativos, que é a capacidade do sistema predizer a ausência das condições para os casos que realmente não as têm. Na Figura 3.9, o ponto (0,0) representa a tática de classificar todas as entradas como negativo, o ponto (1,1) representa a estratégia de classificar todas entradas como positivas. O ponto (0,1) seria o modelo perfeito, onde todos os negativos e todos os positivos são classificados corretamente. A linha diagonal ligando os pontos (0,0) e (1,1) representa um modelo totalmente estocástico em que cada ponto (p , p) pode ser obtido pela previsão da classe positiva com probabilidade p e da classe negativa 100% – p. Qualquer ponto pertencente à parte superior esquerda à essa linha são modelos melhores que uma predição aleatória (SOUZA, 2009).

Figura 3.9 – Curva Característica de Operação do Receptor (ROC).



Fonte: Paulino (2018).

Para medir o quanto o modelo consegue distinguir entre condutores autênticos e impostores, usa-se a AUC-ROC. AUC (*Area Under the Curve*) é a área abaixo da curva ROC. Enquanto a ROC é uma curva de probabilidade, a AUC é uma medida de separabilidade entre classes. Uma AUC próximo de 1 indica que o modelo consegue separar bem as classes e próximo de 0,5 indica que o modelo não tem capacidade discriminativa de distinguir entre as classes positivas e negativas. Uma AUC próxima de 0 indica que o modelo está comutando entre classes ou seja, predizendo a classe negativa como positiva e vice-versa. Um vantagem de se usar a AUC-ROC nesse caso é a funcionalidade de determinação do limiar (*threshold*) que realiza a separação entre classes autêntico e impostor. É possível determinar o limiar observando o *trade-off* entre a taxa de falsos positivos e verdadeiros positivos.

4 METODOLOGIA

Esse capítulo descreve a metodologia e as técnicas que serão empregadas na pesquisa e desenvolvimento do sistema de autenticação de condutores e destaca-se as tecnologias e dispositivos necessários.

4.1 Visão geral do modelo proposto

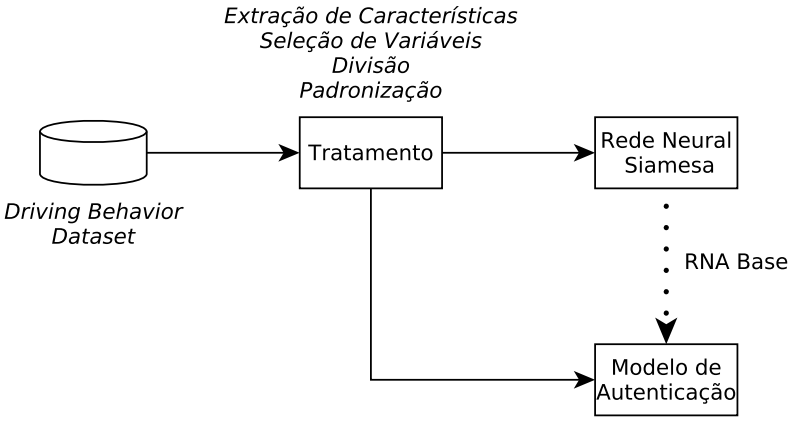
O modelo de autenticação de condutores terá a função de efetuar a autenticação do condutor que dirige o veículo e detectar possíveis situações de roubo ou furto do mesmo. Para tal, a autenticação será feita por meio do reconhecimento de certos padrões de direção característicos de cada condutor. Isso é possível por meio da coleta de dados provenientes da rede de comunicação do veículo (CAN), por meio da interface OBD-II com o dispositivo de leitura ELM327, que transmite via *bluetooth* para um *smartphone* a ele conectado, que também fornece dados referentes à dinâmica de direção provenientes de unidades de medição inercial (IMU) e de geolocalização (GPS). No *dataset* utilizado, o *Driving Behavior Dataset*, estão presentes todos esses sensores, que são descritos na Seção 4.2.

A primeira etapa na implementação do modelo de autenticação de condutores é o tratamento dos dados. Nessa etapa é efetuada a extração de características estatísticas (mediana, média e desvio padrão) das variáveis do *dataset*. Em seguida, é feita a seleção de variáveis que serão utilizadas como entradas no modelo por meio da aplicação da correlação de Pearson e do algoritmo de Florestas Aleatórias que permite identificar as variáveis mais importantes para a tarefa de autenticação de condutores. Por fim, nessa etapa, é realizada a divisão dos dados entre treinamento e teste do modelo de autenticação de condutores, além da padronização da magnitude dos dados de entrada da rede neural siamesa.

A etapa seguinte da implementação do sistema de autenticação de condutores é a construção do modelo baseado em aprendizado por representação, a rede neural siamesa. Esse modelo recebe como entrada pares de dados e tem como alvo identificar se os pares pertencem ao mesmo condutor. A rede neural siamesa é composta por um modelo base, formado por uma rede neural perceptron de múltiplas camadas (MLP), que tem pesos compartilhados entre as entradas e a saída de cada uma das redes irmãs, e é submetida a uma função de similaridade baseada, nesse caso, na distância euclidiana entre as saídas. O objetivo é que pares de entrada pertencentes a um mesmo condutor tenham uma distância curta entre si e pares de condutores distintos tenham uma maior distância.

A etapa final consiste na utilização do modelo base da rede neural siamesa treinada para a formação de *embeddings*. Como o modelo base realiza o mapeamento dos dados de direção em um espaço que aglomere numa região próxima dados de um mesmo condutor e afasta dados de condutores distintos, dados de condutores que não foram utilizados no treinamento da rede neural siamesa são submetidos ao modelo base da mesma. Os *embeddings* gerados por esse modelo base é usado para criação de um conjunto de dados desses novos condutores para fins de comparação com novos dados desconhecidos. Sendo assim, cada nova leitura, que se queira verificar autenticidade, é submetida, também, ao modelo base e comparada a distância dessa leitura para com os *embeddings* salvos. Se a menor distância dentre a amostra testada e uma amostra salva for maior que um limiar pré-determinado, esse condutor é considerado impostor e em caso contrário legítimo. O fluxo geral de desenvolvimento do sistema de autenticação de condutores é mostrado na Figura 4.1 e detalhes dessa implementação são mostrados nas próximas seções do presente capítulo.

Figura 4.1 – Visão geral do fluxo de desenvolvimento do modelo de autenticação de condutores.



Fonte: Elaborado pelo Autor (2019).

4.2 O *Dataset* utilizado

Para a implementação desse trabalho, foi utilizado o *Driving Behavior Dataset* disponibilizado pelo *Laboratory of Advanced Collaboration* da PUC Rio. Esses dados foram coletados por Oliveira Vasconcelos (2017) para a detecção de anomalias durante a condução do veículo. Esses dados foram obtidos pela leitura da CAN do veículo, por meio da interface OBDII, em comunicação com um *smartphone*, que também fornece dados oriundos de unidade de medição inerciais (IMU) e de geolocalização (GPS) embarcados no dispositivo.

Cada um dos vinte e cinco motoristas voluntários percorreu um mesmo trajeto de 14,5 km uma vez entre as 09:00 e 20:00 em dias úteis. Os motoristas diferem entre si quanto a experiência como motorista (2 à 42 anos), idade (20 a 60 anos) e gênero (dezesseis do sexo masculino e nove do sexo feminino). O *dataset* contém no total 12,5 horas de *logs* de condução com um percurso total de 362,5 km, com frequência de leitura de 1 Hz (uma leitura por segundo). A Tabela 4.1 mostra a descrição de cada variável coletada, bem como sua fonte e uma breve descrição.

Tabela 4.1 – Variáveis disponíveis no *Driving Behavior Dataset*.

Variável	**Fonte**	**Descrição**
gpsSpeed	GPS	Velocidade lida pelo GPS
pitchInDegrees	GPS	Arfagem (ângulo ao redor do eixo X)
rollInDegrees	GPS	Rolagem (ângulo ao redor do eixo Y)
azimuthInDegrees	GPS	Azimute (ângulo ao redor do eixo Z)
gyro_x	Giroscópio	Aceleração no eixo X (incluindo gravidade)
gyro_y	Giroscópio	Aceleração no eixo Y (incluindo gravidade)
gyro_z	Giroscópio	Aceleração no eixo Z (incluindo gravidade)
acc_x	Acelerômetro	Aceleração no eixo X (incluindo gravidade)
acc_y	Acelerômetro	Aceleração no eixo Y (incluindo gravidade)
acc_z	Acelerômetro	Aceleração no eixo Z (incluindo gravidade)
acc_isolatedX	Acelerômetro	Aceleração no eixo X (excluindo gravidade)
acc_isolatedY	Acelerômetro	Aceleração no eixo Y (excluindo gravidade)
acc_isolatedZ	Acelerômetro	Aceleração no eixo Z (excluindo gravidade)
rpm	OBD-II	Rotações por minuto do motor
speed	OBD-II	Velocidade atual em km/h
throttlePosition	OBD-II	Posição do acelerador em porcentagem
driverId	Condutor	Código de identificação do motorista

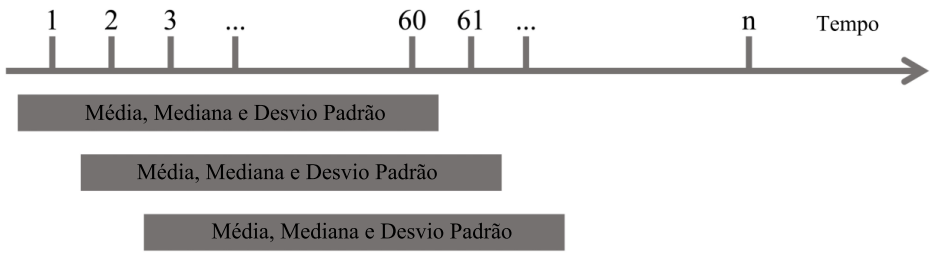
Fonte: Elaborado pelo Autor (2019).

4.3 Análise e tratamento dos dados

Antes de iniciar-se de fato a construção do modelo de autenticação de condutores, os dados são submetidos a uma análise descritiva, tal que possa-se identificar detalhes específicos nos mesmos e um posterior tratamento. Ela tem por objetivo eliminar quaisquer *outliers* presentes nesses e a extração de características estatísticas básicas que propiciem uma aprimorada capacidade de classificação do modelo proposto. Também é efetuada uma seleção de variáveis que melhor discriminem o perfil dos condutores. Por fim, como o modelo proposto é baseado em uma rede neural artificial, é importante que os dados sejam padronizados dentro de uma mesma faixa de valores, tal que todas as variáveis tenham mesmo peso com relação à dimensão dessas.

Cada variável disponível no *dataset* utilizado é submetida à uma extração de características estatísticas de tendência (média e mediana) e dispersão (desvio padrão). Esta extração se dá da seguinte maneira: é definido um tamanho de janela temporal no qual é feito o cálculo de cada uma das estatísticas citadas. Esta janela temporal em seguida é movida, para que os mesmos cálculos sejam efetuados considerando dados mais recentes na série temporal e excluído os mais antigos (SOUZA et al., 2018). A Figura 4.2 exemplifica como é efetuado este tipo de extração para um tamanho de janela de 60 segundos. Nos experimentos são testados cinco janelas temporais de tamanhos diferentes (15, 30, 45, 60 e 90 segundos).

Figura 4.2 – Funcionamento da extração de características para uma janela temporal de 60 segundos.



Fonte: Adaptado de Kwak, Woo e Kim (2016a).

Uma vez efetuada a extração de características estatísticas é feita uma seleção dessas, de modo que somente variáveis que tenham importância discriminatória na identificação dos condutores sejam utilizadas como entradas do modelo. Para isso, é feito primeiramente a análise de correlação de Pearson entre as variáveis originais, de modo que possa-se identificar quais delas tenham uma alta correlação entre si, de modo a eliminar variáveis redundantes. Além da análise de correlação, é também implementado um modelo de Florestas Aleatórias (BREIMAN, 2001) tendo como entrada todas as variáveis já com suas características estatísticas extraídas e como alvo a identidade dos condutores. Como a técnica de Florestas Aleatórias é baseada em um conjunto de várias árvores de decisão, essas árvores conseguem capturar, por meio das quebras em cada nível de cada árvore, quais variáveis tem maior poder discriminatório para a identificação dos condutores (MENZE et al., 2009).

Após a análise, filtragem e seleção dos dados, esses são divididos aleatoriamente em treinamento e teste. Essa divisão é feita considerando um grupo de condutores cujos dados serão utilizados para o treinamento do modelo de autenticação de condutores e o restante é utilizado para a validação do modelo. A proporção usada em treinamento e teste foi de 13 e 12 condutores, respectivamente, com o objetivo de se garantir o maior número de testes de

desempenho do modelo de autenticação de condutores. Essa divisão foi feita aleatoriamente, porém mantidas para todos os experimentos executados, de modo que as condições de teste das diferentes topologias da rede neural siamesa sejam as mesmas. A Tabela 4.2 apresenta quais condutores foram selecionados para treinamento e teste do modelo.

Tabela 4.2 – Divisão dos condutores em treinamento e teste.

Grupo	**Condutores**
Treinamento	8, 16, 0, 23, 11, 9, 13, 1, 22, 5, 2, 12, 15
Teste	3, 4, 6, 7, 10, 14, 17, 18, 19, 20, 21, 24

Fonte: Elaborado pelo Autor (2019).

Os doze condutores selecionados para teste são divididos em quatro grupos, onde cada grupo simula um veículo. São efetuados experimentos para cada veículo de teste, onde os condutores de cada um desses são confrontados com condutores de outros veículos, que serão para o veículo em questão impostores, isto é, quando se testa o modelo em um dos veículos tem-se 3 condutores autênticos e 9 impostores, o que analisa a capacidade de identificação do modelo em condições com muitos condutores impostores. A Tabela 4.3 apresenta como os condutores estão divididos em grupos. Cada um destes grupos é definido, por convenção, como sendo um veículo, em que seus membros são os condutores autorizados a conduzi-lo. É válido ressaltar que mesmo sendo efetuada esta divisão em veículos, os dados dos condutores em questão foram coletados no mesmo veículo, sob mesmas condições. Dados oriundos de modelos de veículos distintos poderiam influenciar diretamente nos resultados obtidos, visto que os sensores empregados em cada modelo de veículo podem ser distintos e, portanto, os dados coletados poderiam ter comportamentos diferentes.

Tabela 4.3 – Divisão dos condutores de teste em veículos.

Veículo	**Condutores**
1	3, 4, 6
2	7, 10, 14
3	17, 18, 19
4	20, 21, 24

Fonte: Elaborado pelo Autor (2019).

Os dados antes de utilizados no treinamento do modelo de autenticação de condutores são submetidos a uma padronização de seus valores entre 0 e 1. Esse passo é importante, visto que os dados tem magnitudes de valores diferentes, o que é prejudicial para o treinamento de

uma rede neural artificial, onde essa pode dar mais importância a variáveis cujos valores tem maior amplitude. A padronização dos dados é representado pela seguinte equação:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{4.1}$$

onde X é o valor a ser padronizado, X_{min} e X_{max} é o mínimo e o máximo valores encontrados de uma varável a ser padronizada, respectivamente e X_{sc} é o valor entre 0 e 1 correspondente de X padronizado. É importante ressaltar que o valor mínimo e máximo usados para transformar o conjunto de treinamento são os mesmo usados para o conjunto de teste.

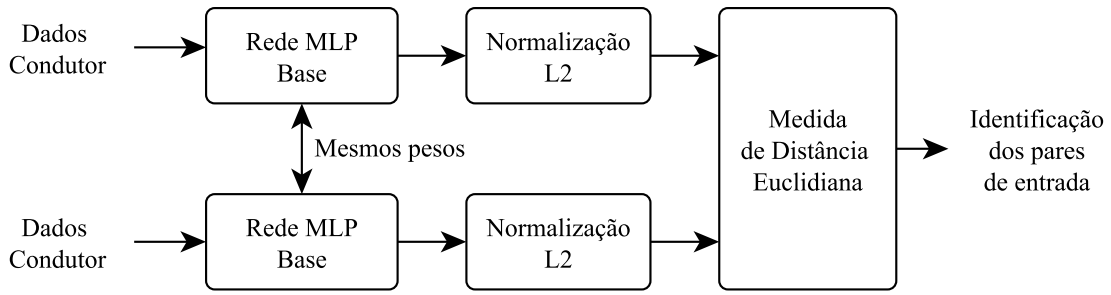
4.4 Implementação da Rede Neural Siamesa

Para a implementação da rede neural siamesa, primeiramente são formados pares entre leituras dos dados em pares genuínos (classe 1) e pares impostores (classe 0), com respeito à identidade dos condutores alocados na base de treinamento. A proporção entre as classes foi de 50% e também garantindo que se tenham pares impostores formados por todas as combinações de condutores da bases. Para o teste de desempenho da rede neural siamesa, pares também são formados por condutores da base de teste. O próximo passo é definir a topologia do modelo base da rede neural siamesa. O modelo base é uma rede neural que é configurada de acordo com os tipos de dados de entrada, sendo geralmente um modelo sequencial, baseado em camadas convolucionais (para imagens), recorrentes como a LSTM (processamento de linguagem natural ou séries temporais) ou em perceptron de múltiplas camadas (MLP) para dados estruturados, sendo essa última topologia a base do modelo implementado nesse trabalho.

O modelo MLP base é composto por um número variável de camadas (duas e três) e o número de neurônios em cada uma dessas camadas é um hiperparâmetro que é avaliado durante os testes de desempenho dos modelos. A ativação usada na saída de cada um dos neurônios é a *Leaky ReLu*, que, diferentemente da *ReLu*, não zera saídas negativas dos neurônios, o coeficiente angular para ativações negativas foi fixado em 0,3. Uma camada *Dropout* de 10% é adicionada nas camadas densas intermediárias. Antes das saídas do modelo base (*embeddings*) serem submetidas à perda contrastiva, essas são submetidas à uma normalização L2 (Euclidiana) para que a distância entre as características geradas sejam normalizadas tal que a margem de distância entre classes possa ser constante e independente da escala da saída da rede base

(SCHROFF; KALENICHENKO; PHILBIN, 2015b). A Figura 4.3 mostra a topologia de uma rede neural siamesa desde a entrada até a saída que será submetida à função de similaridade.

Figura 4.3 – Configuração do modelo.



Fonte: Elaborado pelo Autor (2019).

São testadas quatro topologias diferentes para o modelo MLP base. A primeira é uma MLP simples com duas camadas de 64 e 32 neurônios respectivamente (Rede 1). A segunda rede é uma MLP com um número maior de neurônios por camada (128 e 64), de modo a se comparar o impacto desse incremento no desempenho da rede neural siamesa (Rede 2). A terceira topologia é uma MLP de três camadas com 64, 64 e 32 neurônios em cada uma dessas camadas (Rede 3) de modo a comparar o impacto de um número diferente de camadas no desempenho do sistema. A quarta topologia é uma MLP também de três camadas de 128, 128 e 64 neurônios (Rede 4), testando uma topologia com mais camadas e neurônios em cada uma delas. A Tabela 4.4 apresenta de forma resumida cada uma das redes testadas e suas respectivas configurações. É válido ressaltar que a dimensão dos *embeddings* gerados é a quantidade de neurônios da camada de saída do modelo MLP base da rede neural siamesa. Outros hiperparâmetros devem ser definidos para a implementação da rede neural siamesa, como a margem de distância M, fixada em 0,2. Para o treinamento dessa rede foi considerado o algoritmo de otimização RMSprop (HINTON; SRIVASTAVA; SWERSKY, 2012), com taxa de aprendizado de 0,001, 40 épocas de treinamento, *batch size* de 64 e acurácia como métrica de treinamento, que foram determinados em experimentos preliminares.

Tabela 4.4 – Topologias MLP bases do modelo neural siamês para identificação de dados de condutores.

MLP Base	Número de Camadas Escondidas	Neurônios por Camada
Rede 1	2	[64, 32]
Rede 2	2	[128, 64]
Rede 3	3	[64, 64, 32]
Rede 4	3	[128, 128, 64]

Fonte: Elaborado pelo Autor (2019).

4.5 Modelo de autenticação de condutores

A hipótese central desse trabalho é o uso de uma rede neural artificial treinada em uma topologia siamesa que possa mapear características de condutores em *embeddings* em um subespaço próximo, mesmo que dados referentes à direção desses condutores não sejam usados para o treinamento dessa rede. Para tanto, é utilizado o modelo base da rede neural siamesa já treinado para gerar esses *embeddings* quando dados de novos condutores, já submetidos aos tratamentos supracitado, são submetidos à essa rede. Sendo assim, um conjunto de dados derivados de cada novo condutor é formado. Partindo da premissa que o modelo MLP base tende a aproximar as saídas de dados de um mesmo condutor, a formação desse conjunto de dados permite que novas leituras dos sensores sejam aplicadas nesse modelo e comparadas com os dados salvos. Se os dados submetidos tiverem uma distância próxima das de algum condutor autorizado existe uma alta probabilidade desse condutor ser genuíno. Em contrapartida, se uma leitura dos sensores de algum condutor for distante dos dados presentes no conjunto de suporte existe chance desse condutor ser um impostor.

Para a execução desses testes, os condutores do grupo de teste são divididos em quatro veículos com três condutores autorizados em cada um desses como usuários do mesmo veículo, tal qual mostrado na Tabela 4.3. Então, os dados tratados são submetidos ao modelo base e seus *embeddings* serão utilizadas para a tarefa de autenticação de condutores. Para tal, foi efetuada um divisão dos dados derivados de cada condutor para a formação do conjunto de treinamento (40%) e para testes de autenticidade (60%).

Para a avaliação de como os *embeddings* dos novos condutores de comportam no espaço, foi aplicado a técnica de análise de componentes principais (PCA) para a visualização dos dados em duas dimensões. PCA realiza a redução de dimensionalidade dos *embeddings* (que nos experimentos podem ter 64 ou 128 dimensões) sem que ocorra muitas perdas da distribuição original dos dados. Assim, pode-se observar se os *embeddings* estão aglomerados para

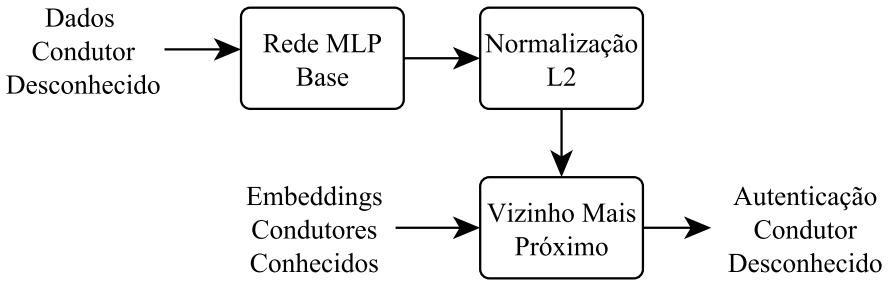
os dados de um mesmo condutor e espaçados para condutores distintos. Outra abordagem para tal avaliação é a aplicação da técnica de agrupamento baseado em distância K-*means*. Por meio desta técnica, é avaliado se uma técnica não-supervisionada, que não tem conhecimento a qual condutor pertence cada ponto no espaço, consegue determinar se o número ideal de grupos é igual ao número de condutores, o que indica boa distribuição no espaço dos *embeddings* dos condutores.

A função de decisão que determinará a autenticidade de um condutor em um veículo é baseada na menor distância entre o ponto testado e um outro pertencente ao suporte. Se essa distância for menor que um limiar pré-determinado, o condutor é considerado autêntico e em caso contrário impostor. Considerando x_i uma leitura de um condutor que se deseja autenticar e X_S o suporte de dados do conjunto de condutores autorizados S temos a seguinte função de decisão booleana $F(x_i)$:

$$F(x_i) = \min_{X_S \in S}(D(x_i, X_S)) < T, \tag{4.2}$$

onde D é a distância Euclidiana do ponto testado para todos dos pontos do conjunto de dados armazenado dos condutores autorizados e T é o limiar de distância. Para a implementação é utilizado o algoritmo não supervisionado do Vizinho mais Próximo (*Nearest-Neighbour*), que calcula e retorna a menor distância entre os *embeddings* de um condutor desconhecidos daqueles *embeddings* salvos condutores de condutores desconhecidos. O fluxo do modelo de autenticação de condutores é apresentado na Figura 4.4.

Figura 4.4 – Fluxo do modelo de autenticação de condutores.



Fonte: Elaborado pelo Autor (2019).

4.6 Configuração dos experimentos

Para a implementação do modelo proposto, foram efetuados diversos experimentos levando em consideração o tamanho da janela de extração de características estatísticas e a topologia da rede neural siamesa. Como foram definidas cinco janelas distintas para extração de características e quatro topologias MLP bases, são efetuados vinte experimentos diferentes, que são repetidos cinco vezes cada, de modo a diminuir os efeitos da inicialização e otimização estocástica dos pesos da rede neural siamesa e, consequentemente, nos resultados obtidos. Cada um dos cem modelos de MLP base são usados também na segunda etapa dos experimentos, que consiste no sua utilização para a extração de características de condutores que não foram usados para o treinamento da rede neural siamesa para cada um dos quatro grupos de condutores (veículos). Para cada veículo é considerada a AUC em cada uma das cem topologias MLP base e janela de extração de caraterísticas, totalizando quatrocentos testes diferentes para essa etapa.

Os experimentos foram implementados utilizando a linguagem Python por meio de diversos pacotes de uso livre. Para leitura e manipulação dos dados foi utilizado o Pandas (MCKINNEY, 2011), uma biblioteca de alto nível e desempenho para manipulação e análise de grandes volumes de dados. Para o pré-processamento, seleção de características com Florestas Aleatórias, padronização, análise de componente principais, agrupamento K-*means* e algoritmno de *Nearest-Neighbours*, foi utilizado o pacote *Scikit-learn* (PEDREGOSA et al., 2011), uma dos pacotes de pré-processamento, mineração de dados e aprendizado de máquina mais difundidos atualmente. Para a implementação das redes neurais foi utilizado o pacote Keras (CHOLLET et al., 2015) que utiliza como *backend* o pacote desenvolvido pelo Google para implementação de redes neurais convencionais e profundas (*deep learning*) em diversas topologias *Tensorflow* (ABADI et al., 2015). Os experimentos foram executados utilizando um *notebook* Sony Vaio® com processador Intel© Core™ i7-3537U de 2,00GHz com 12Gb de memória RAM e sistema operacional Manjaro Linux 18.0.4. Cada experimento levou em média 5 minutos para ser executado, considerando o tempo de treinamento de cada topologia testada.

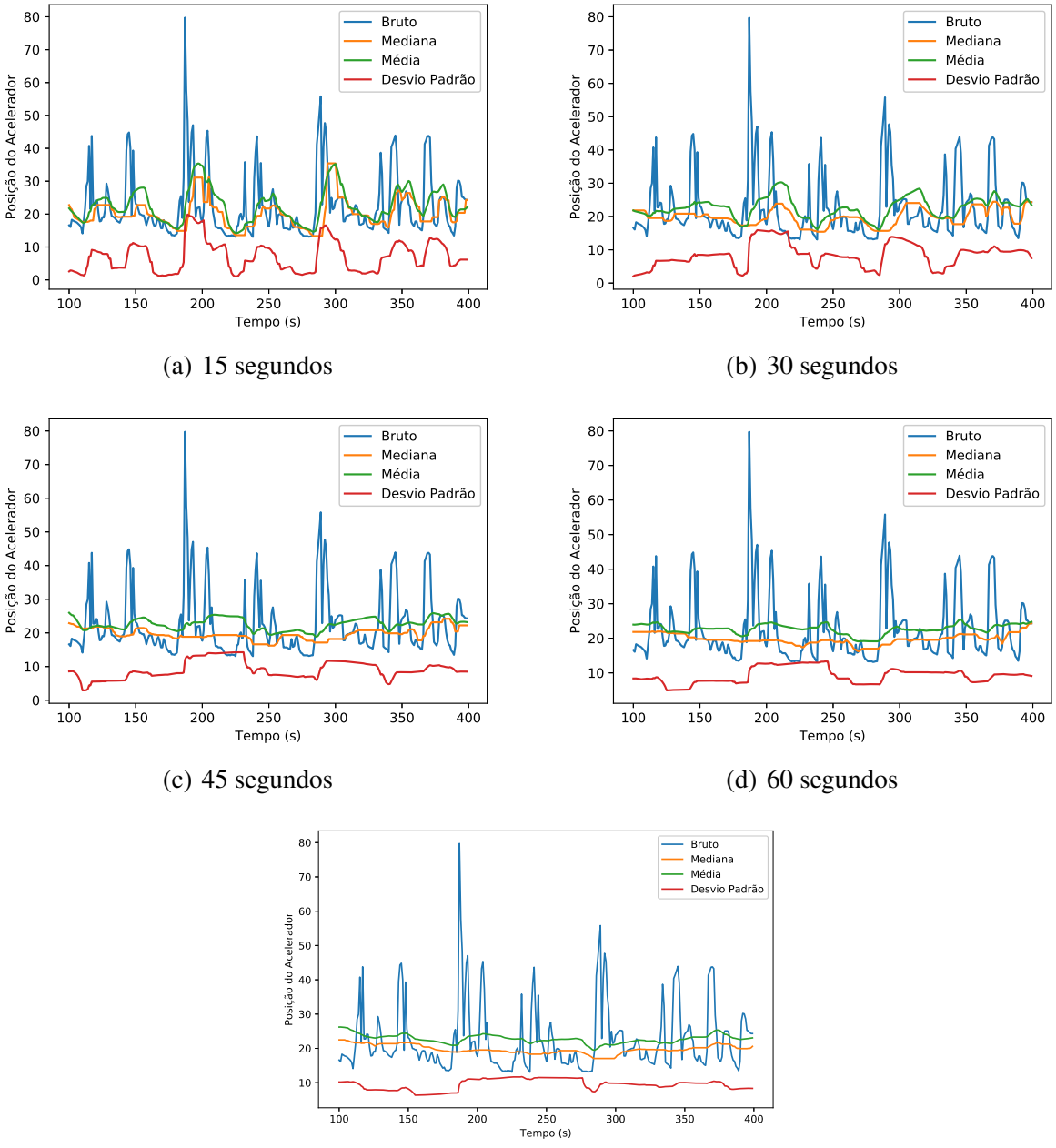
5 RESULTADOS E DISCUSSÕES

Os resultados do presente projeto estão divididos em três partes. A primeira delas consiste no tratamento dos dados, onde o impacto de cada uma das etapas de pré-processamento dos dados é avaliado. A segunda parte concentra-se na análise de desempenho das diferentes topologias de rede neural siamesa. Nessa etapa, o objetivo é que pares de dados de um mesmo condutor sejam classificados como legítimos e que condutores distintos sejam identificados. A terceira parte envolve o teste das redes neurais bases que foram treinadas na primeira etapa para gerar *embeddings* representativos dos dados de condutores que não foram usados no treinamento e sua aplicação para autenticação destes condutores.

5.1 Análise e tratamento dos dados

Primeiramente, é efetuada a extração de características estatísticas, no qual são testados cinco janelas temporais de tamanhos diferentes (15, 30, 45, 60 e 90 segundos) tal que sejam analisados o impacto desses tamanhos no desempenho do modelo. A Figura 5.1 mostra a comparação das janelas de extração de características para a variável de posição relativa do acelerador, uma das mais importante para a autenticação do condutor, de acordo com Meiring e Myburgh (2015). É importante ressaltar que a extração de características é aplicada individualmente para cada condutor, tal que não ocorra vazamento de informações entre cada um desses. Pode-se observar que quanto maior o tamanho da janela, maior a atenuação do sinal. Contudo, essa maior atenuação pode trazer perdas com relação às variações de comportamento intrínsecas de cada condutor.

Figura 5.1 – Posição relativa do acelerador em diferentes janelas de extração de características estatísticas para o condutor 1.



(a) 15 segundos

(b) 30 segundos

(c) 45 segundos

(d) 60 segundos

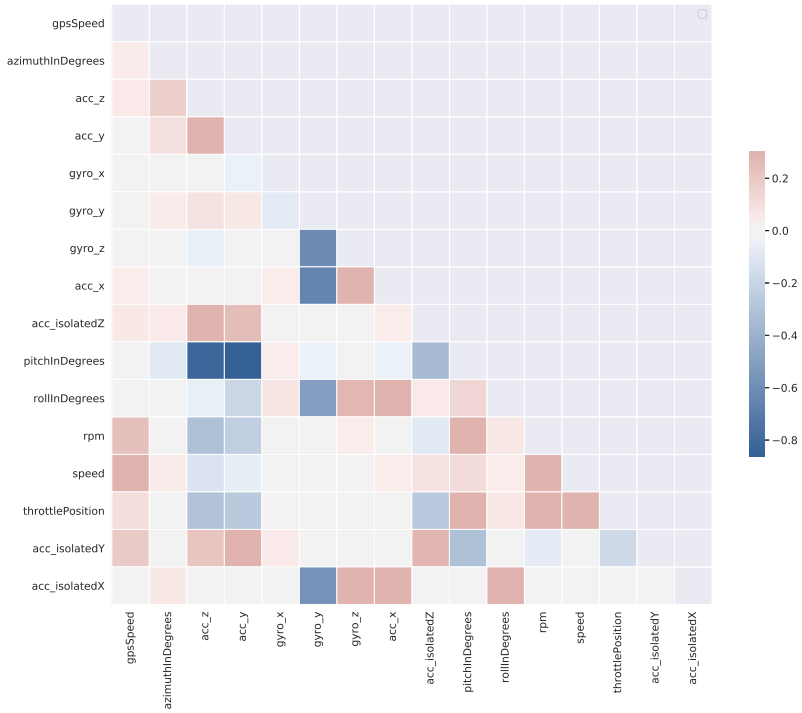
(e) 90 segundos

Fonte: Elaborado pelo Autor (2019).

O passo seguinte é a seleção de variáveis do *dataset* utilizado. Uma avaliação preliminar para seleção de variáveis é a análise de correlação de Pearson entre as variáveis do *dataset*. Uma alta correlação, positiva ou negativa, entre pares de de variáveis, pode indicar que exista a relação linear entre essas, isto pode ocasionar em redundância na entrada do modelo. o que resulta em aumento no esforço computacional com pouco ganho discriminativo. A Figura 5.2 apresenta a correlação entre as variáveis, onde pode-se notar que existe uma baixa correlação

entre as mesmas. Portanto, antes de uma análise mais aprofundada, é válida a manutenção de todas essas variáveis para a implementação do modelo de autenticação de condutores.

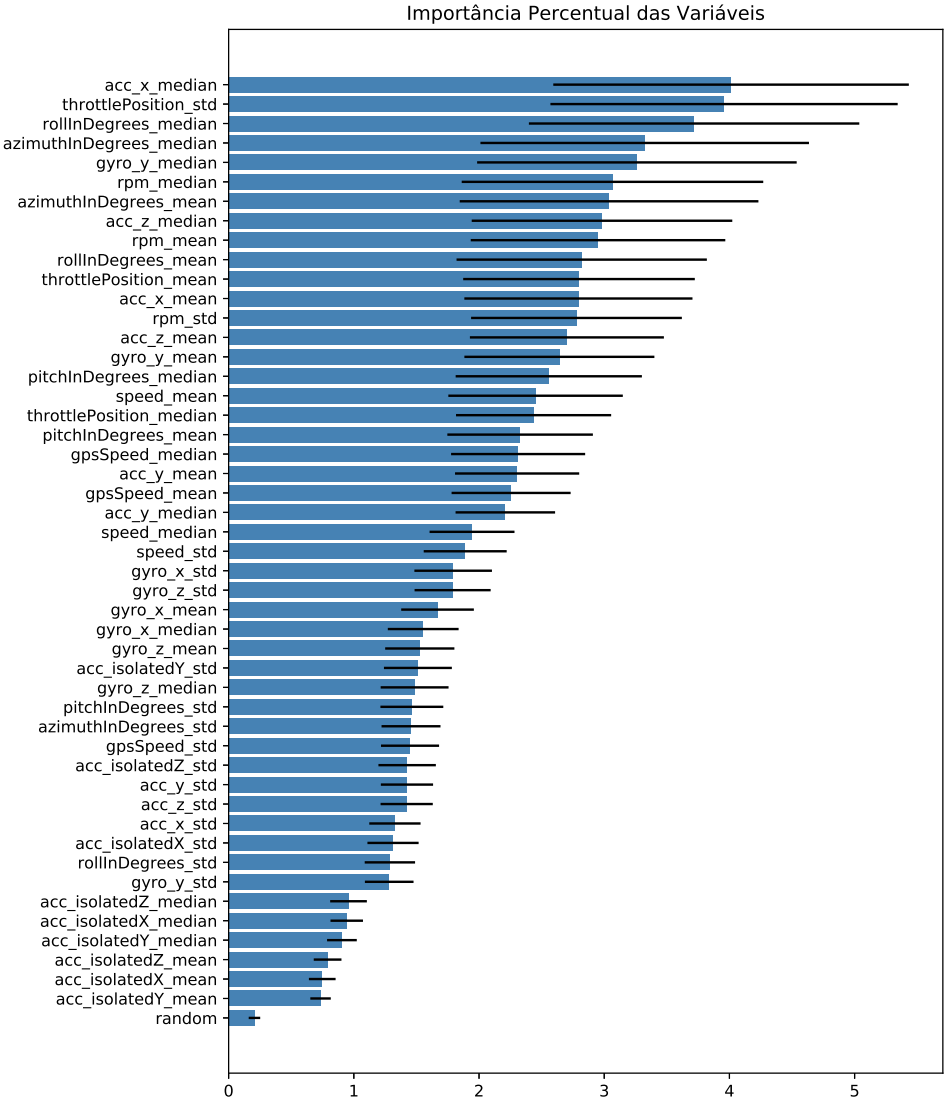
Figura 5.2 – Correlação entre variáveis do *Driving Behavior Dataset*.



Fonte: Elaborado pelo Autor (2019).

Outra abordagem adotada para a seleção de variáveis é a aplicação de uma técnica baseada em árvores de decisão, no caso um modelo de Florestas Aleatórias. Com um modelo de trinta árvores, onde cada uma retorna um nível de importância das variáveis individualmente, é feita a listagem da importância relativa das mesmas. Com a inserção de uma variável composta de ruído gaussiano, tal que se alguma variável tenha importância menor que esse ruído, essa é removida do conjunto. A Figura 5.3 mostra a importância percentual média retornada pelo modelo de Florestas Aleatórias, bem como seu desvio, de cada uma das variáveis. Todas as variáveis tiveram poder discriminante superior àquela variável aleatória inserida no conjunto. Sendo assim, todas as variáveis são mantidas como entrada do modelo.

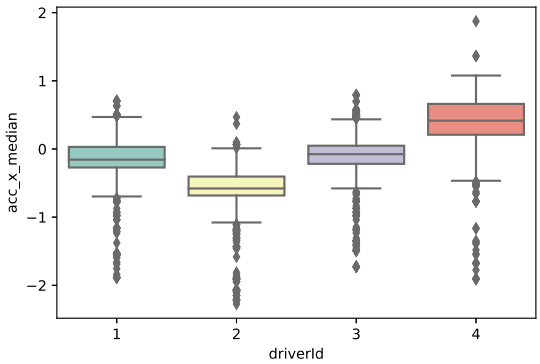
Figura 5.3 – Importância das Variáveis do *Driving Behavior Dataset* para a discriminação dos condutores para janela de extração de 15 segundos.



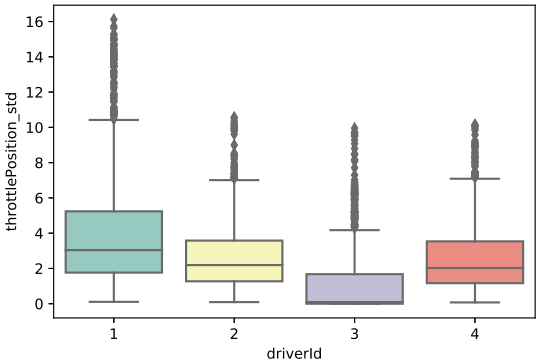
Fonte: Elaborado pelo Autor (2019).

Uma maneira de observar a capacidade de identificação dos condutores das variáveis mais importantes é analisar a distribuição dessas variáveis para diferentes condutores. A Figura 5.4 mostra essa distribuição para os condutores 1, 2, 3 e 4 para as quatro variáveis com maior importância. É possível observar que a distribuição dos dados tem diferenciação entre os condutores, o que indica a possibilidade da utilização de dados de direção para a autenticação de condutores.

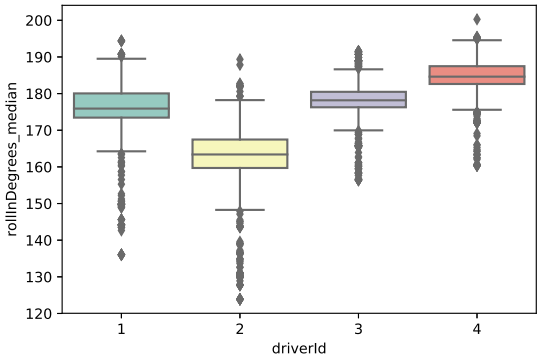
Figura 5.4 – Distribuição das quatro variáveis mais discriminativas para os condutores 1 a 4 para janela de 15 segundos.



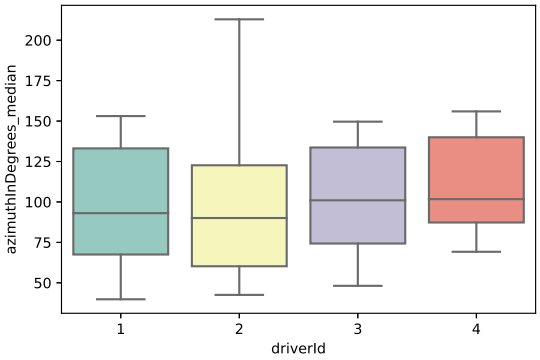
(a) Mediana da aceleração no eixo x.



(b) Desvio padrão da posição relativa do acelerador.



(c) Mediana da taxa oscilação em graus.

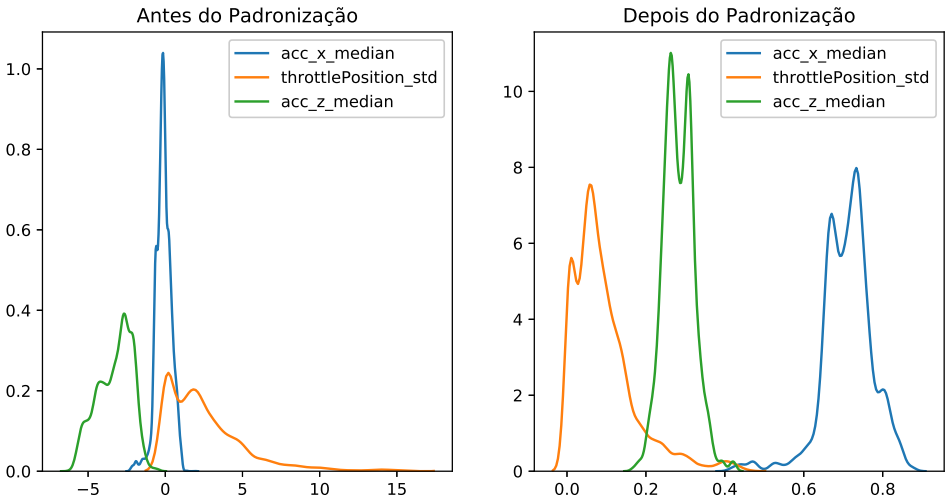


(d) Mediana do azimute em graus.

Fonte: Elaborado pelo Autor (2019).

Na etapa de padronização dos dados, é averiguado o quanto a distribuição original de cada variável é afetada pela técnica. Na Figura 5.5 é possível observar o impacto de tal tratamento na distribuição dos dados. A distribuição dos dados antes da padronização tem uma magnitude de valores de cada variável discrepantes uns dos outros. Após a padronização os valores, as distribuições das variáveis estão dentro da mesma faixa de valores, sem que ocorra perdas significativas na distribuição original de cada uma dessas.

Figura 5.5 – Comparação da distribuição dos dados de três variáveis do *Driving Behavior Dataset* antes e depois de submetidos ao padronização.

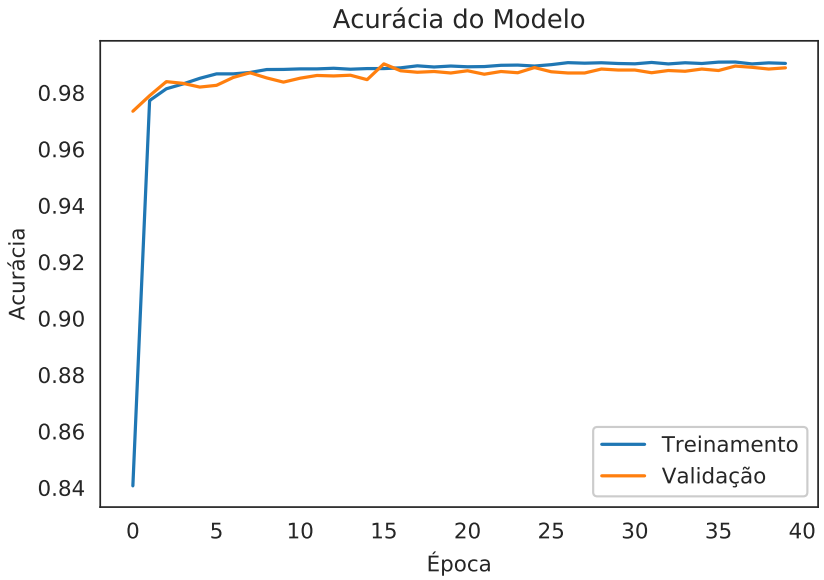


Fonte: Elaborado pelo Autor (2019).

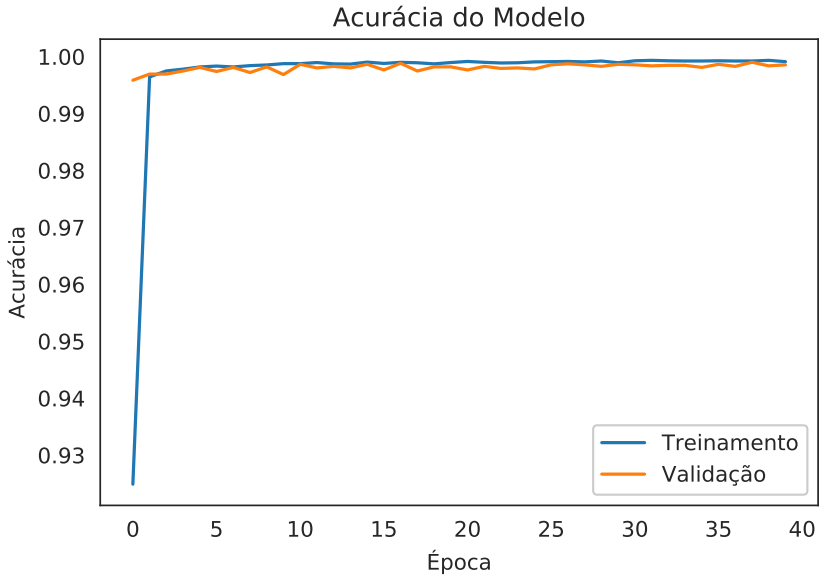
5.2 Rede Neural Siamesa

Nesta fase dos experimentos é testada a capacidade de diversas topologias MLP que formam a rede neural siamesa com diferentes janelas de extração de características, com relação à identificação maximizada de pares de dados de condutores. Para tanto, todas as combinações descritas no Capítulo 4 são avaliadas ante diversas métricas. Primeiramente, é avaliado o desempenho de treinamento da rede neural siamesa por época de treinamento para o conjunto de treinamento e validação. A Figura 5.6 mostra a acurácia por época de treinamento para a pior e melhor rede em termos de acurácia no conjunto de condutores de teste. Em ambos os casos, pode-se observar um bom comportamento da curva de aprendizado, porém com menor variação para a acurácia do conjunto de validação, o que indica uma melhor capacidade de identificação de pares de dados de condutores e também a não ocorrência de sobre ajuste do modelo.

Figura 5.6 – Comparação da acurácia por época para conjunto de treinamento e teste da pior (a) e melhor topologia (b).



(a) MLP base 3 e janela de 15 segundos



(b) MLP base 2 e janela de 60 segundos
Fonte: Elaborado pelo Autor (2019).

Os modelos Neurais Siameses treinados têm como função objetivo a minimização da função de perda contrastiva e a maximização da acurácia no que diz respeito à identificação de pares de dados de condutores, sejam eles pertencentes a um mesmo condutor ou a condutores distintos. Dentre as diferentes combinações de hiperparâmetros testados, a Tabela 5.1 apresenta a acurácia média e seu desvio para o conjunto de validação (uma parte do conjunto de treinamento que é desmembrada desse com objetivo de monitorar o desempenho dessa comparado ao

conjunto usado no treinamento). A Figura 5.7 retrata graficamente o desempenho da acurácia para o conjunto de validação, onde observa-se que a rede MLP base 2 tem melhor desempenho em todas as janelas de extração de características, sempre superior à 99,5%.

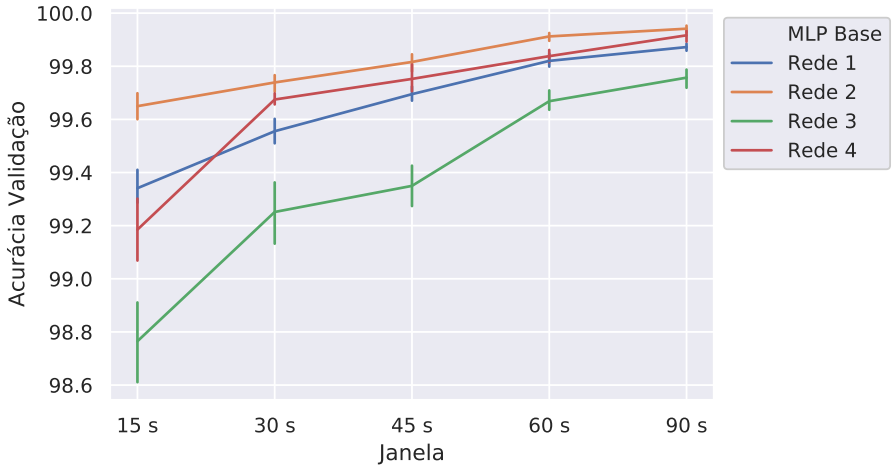
Por sua vez, a rede MLP base 3 apresentou o pior desempenho dentre as topologias testadas, porém, sempre com acurácia percentual superior à 98,6%. Para essa métrica, quanto maior a janela de extração de características, melhor o desempenho das redes neurais siamesas testadas, o que indica que um sinal de entrada mais suavizado fomenta melhores resultados. Contudo, esses testes para o conjunto de validação servem apenas como análise da eficiência do treinamento das redes neurais siamesas e, consequentemente, gera um modelo MLP base que tenha melhor capacidade de produzir um espaço de dados significativos para a tarefa de autenticação de condutores. Portanto, para o conjunto de testes os resultados serão analisados mais profundamente, em diferentes métricas, de modo a se definir o melhor modelo MLP base para a tarefa mencionada.

Tabela 5.1 – Acurácia percentual para conjunto de dados de validação.

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	99,34% ± 0,08%	99,65% ± 0,06%	98,77% ± 0,20%	99,19% ± 0,16%
30 segundos	99,56% ± 0,06%	99,74% ± 0,04%	99,25% ± 0,15%	99,67% ± 0,03%
45 segundos	99,69% ± 0,04%	99,82% ± 0,04%	99,35% ± 0,10%	99,75% ± 0,06%
60 segundos	99,82% ± 0,03%	99,91% ± 0,02%	99,67% ± 0,05%	99,84% ± 0,03%
90 segundos	99,87% ± 0,02%	**99,94% ± 0,02%**	99,76% ± 0,05%	99,92% ± 0,03%

Fonte: Elaborado pelo Autor (2019).

Figura 5.7 – Acurácia percentual dos experimentos para o conjunto de validação.



Fonte: Elaborado pelo Autor (2019).

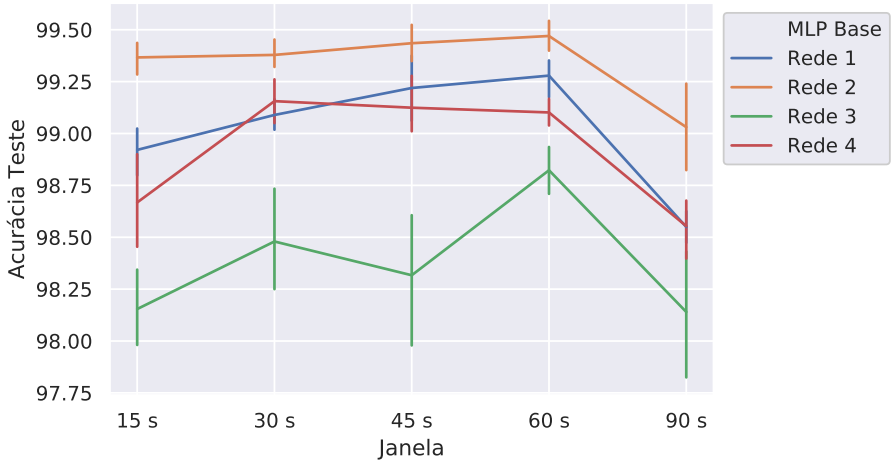
Os resultados apresentados a seguir são para os dados do conjunto de condutores teste, que não foram utilizados para o treinamento da rede neural siamesa. Esses resultados servem para avaliar a capacidade que essa topologia tem em autenticar condutores mesmo que esses não tenham sido usados no treinamento, de modo que para sua aplicação em veículos cujos condutores sejam desconhecidos pela rede sejam também identificados como autênticos ou impostores. O primeiro teste para esse conjunto é a acurácia na identificação de pares de dados de condutores, isto é, o quanto a rede neural siamesa consegue identificar corretamente esses pares. A Tabela 5.2 apresenta a acurácia percentual média e seu desvio para os diferentes experimentos executados, enquanto a Figura 5.8 mostra graficamente o desempenho desses. Assim como ocorre para o conjunto de validação, a rede MLP base 2 tem o melhor desempenho em todas as janelas de extração de características analisadas, enquanto que também a rede MLP base 3 tem o pior desempenho relativo, porém sempre com acurácia média superior à 98%. Um ponto interessante é que para o conjunto de testes a janela de 60 segundos é a que maximiza a capacidade de identificação de pares de condutores do conjunto de teste, o que indica que a janela de 90 segundos, que teve melhor desempenho para o conjunto de validação, pode causar um início de sobre-ajuste da rede.

Tabela 5.2 – Acurácia percentual para conjunto de dados de teste.

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	98,97% ± 0,15%	99,39% ± 0,10%	98,15% ± 0,24%	98,47% ± 0,29%
30 segundos	99,08% ± 0,10%	99,36% ± 0,08%	98,55% ± 0,33%	99,23% ± 0,14%
45 segundos	99,32% ± 0,22%	**99,44% ± 0,12%**	98,49% ± 0,42%	99,08% ± 0,17%
60 segundos	99,32% ± 0,12%	**99,43% ± 0,10%**	98,85% ± 0,15%	99,10% ± 0,08%
90 segundos	98,52% ± 0,10%	99,01% ± 0,26%	98,12% ± 0,38%	98,61% ± 0,18%

Fonte: Elaborado pelo Autor (2019).

Figura 5.8 – Acurácia percentual dos experimentos para o conjunto de teste.



Fonte: Elaborado pelo Autor (2019).

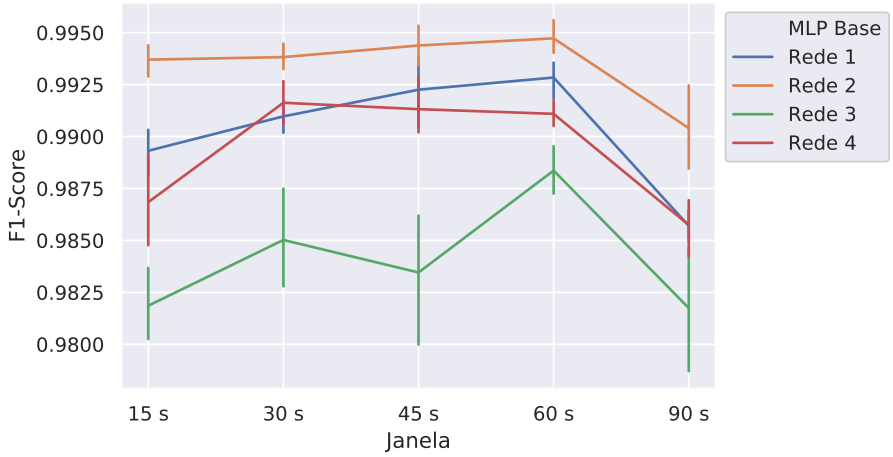
Um comportamento semelhante ocorre quando a métrica avaliada é o F1-*Score*, onde a rede MLP base 2 e a janela de 60 segundos obtém o melhor desempenho dentre os experimentos executados. O F1-Score é uma métrica que leva em consideração a precisão e revocação, sendo que o F1-Score é a média harmônica dessas duas métricas (ver Seção 3.3). A precisão refere-se à quantidade de pares autênticos identificados sobre a quantidade de pares de condutores autênticos totais. Já a revocação consiste na quantidade de pares autênticos identificados corretamente sobre a quantidade total de condutores autênticos retornados pela rede. É importante avaliar essa métrica do ponto de vista do quanto a rede identifica corretamente pares de dados de condutores que deveriam ser identificados tais quais. Diante dos resultados apresentados, observa-se que, mesmo na pior topologia testada, a rede neural siamesa obteve bom desempenho considerando tais ponderações impostas por essas métricas.

Tabela 5.3 – F1-*Score* percentual para conjunto de dados de teste.

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	98,93% ± 0,14%	99,37% ± 0,10%	98,19% ± 0,23%	98,68% ± 0,29%
30 segundos	99,10% ± 0,10%	99,38% ± 0,08%	98,50% ± 0,32%	99,16% ± 0,14%
45 segundos	99,23% ± 0,21%	99,44% ± 0,12%	98,35% ± 0,40%	99,13% ± 0,17%
60 segundos	99,28% ± 0,11%	**99,47% ± 0,10%**	98,84% ± 0,15%	99,11% ± 0,08%
90 segundos	98,57% ± 0,10%	99,04% ± 0,26%	98,17% ± 0,36%	98,57% ± 0,18%

Fonte: Elaborado pelo Autor (2019).

Figura 5.9 – F1-*Score* percentual para conjunto de dados de teste.



Fonte: Elaborado pelo Autor (2019).

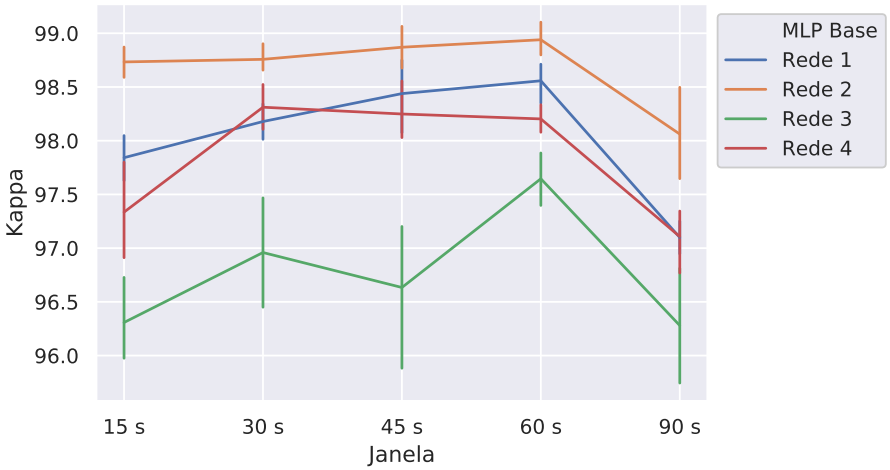
Para esta etapa de treinamento da rede neural siamesa, que envolve uma classificação binária de pares de entrada, a última métrica testada é o Índice *Kappa*, que mensura a concordância entre os resultados obtidos com os verdadeiros. Essa medida, baseada na matriz de confusão, avalia o quanto os resultados do modelo se afastam daqueles reais. Sendo assim, ela tende a ser mais conservadora, apresentando valores inferiores aos de outras métricas como o F1-*Score* se o nível de concordância for considerado baixo, como exemplo se o modelo tem uma diferença razoável entre a taxa de erro de falsos pares positivos e falsos pares negativos. Os valores apresentados na Tabela 5.4 e sumarizados na Figura 5.10 mostram que mesmo diante de uma métrica mais exigente, os resultados obtidos são, dentro dos níveis de concordância da métrica, considerados excelentes. Com relação ao desempenho individual de cada combinação de hiperparâmetros, o padrão se mantém semelhante ao encontrado nas métrica anteriores, onde a rede MLP base 2 com a janela de 60 segundos obteve o melhor desempenho e a rede MLP base 3 o pior desempenho relativo e a maior variância dentre os resultados, mas em termos gerais também considerados satisfatórios.

Tabela 5.4 – Índice *Kappa* percentual para conjunto de dados de teste.

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	97,84% ± 0,29%	98,73% ± 0,19%	96,31% ± 0,47%	97,34% ± 0,59%
30 segundos	98,18% ± 0,20%	98,76% ± 0,17%	96,96% ± 0,66%	98,31% ± 0,28%
45 segundos	98,44% ± 0,43%	98,87% ± 0,25%	96,63% ± 0,84%	98,25% ± 0,35%
60 segundos	98,56% ± 0,23%	**98,94% ± 0,20%**	97,65% ± 0,30%	98,20% ± 0,16%
90 segundos	97,10% ± 0,21%	98,06% ± 0,52%	96,28% ± 0,76%	97,11% ± 0,37%

Fonte: Elaborado pelo Autor (2019).

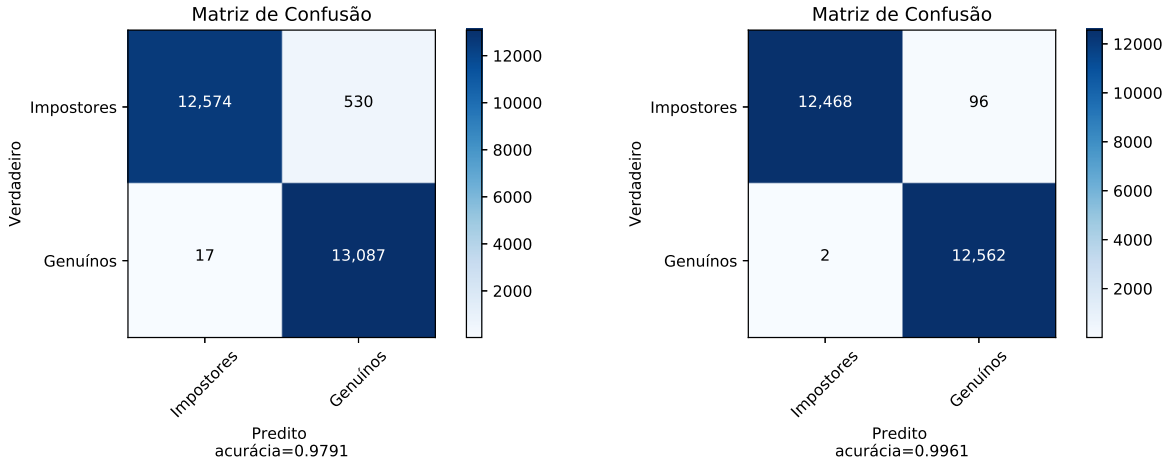
Figura 5.10 – Índice *Kappa* percentual para conjunto de dados de teste.



Fonte: Elaborado pelo Autor (2019).

Por fim, para se ter uma noção quantitativa do comportamento do modelo neural siamês, é apresentado na Figura 5.11 a matriz de confusão para os modelos que obtiveram, respectivamente, o melhor e o pior resultado relativo. Observa-se que existem um certo desequilíbrio entre o número de falsos genuínos, superior ao número de falsos impostores, embora, se comparados ao número de predições corretas, esses números são relativamente baixos. Outro ponto a ser ressaltado é o baixo número de falsos impostores, que no caso da rede MLP base 2, somente duas das entradas foram classificadas como tais. É importante ressaltar que para essa etapa o limiar de distância que determina se um par de dados pertence ao mesmo condutor é constante. Isto significa que o *trade-off* entre o número de falsos positivos e negativos pode ser ajustado de acordo com o impacto que cada tipo de erro tem em implementações práticas.

Figura 5.11 – Comparação da matriz de confusão para o conjunto de teste da pior (a) e melhor topologia (b).



(a) MLP base 3 e janela de 15 segundos (b) MLP base 2 e janela de 60 segundos

Fonte: Elaborado pelo Autor (2019).

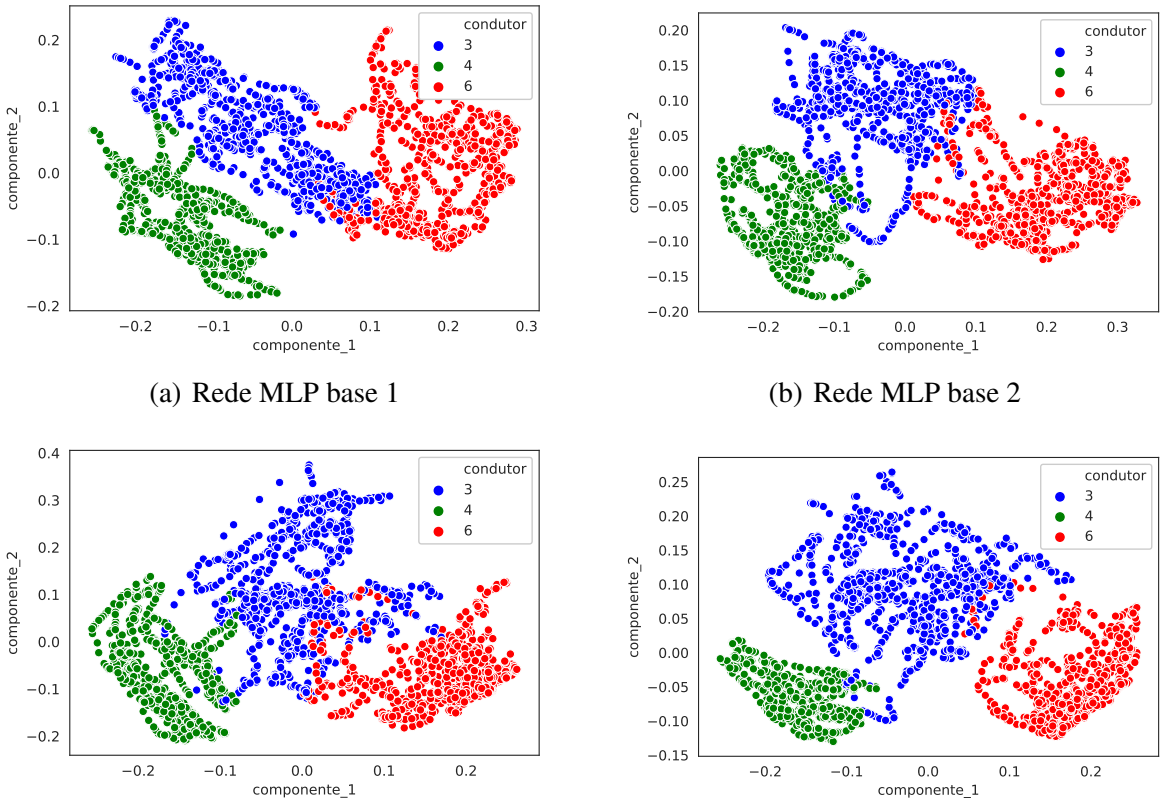
5.3 Autenticação de Condutores por Veículos

Uma vez avaliado o desempenho das diferentes redes neurais siamesas e das janelas de extração de características para a identificação de pares de dados de direção de diferentes condutores, é avaliado agora o desempenho das redes MLP base para um conjunto de condutores que não foram usados para o treinamento da mesma. Essa etapa consiste em submeter dados desses novos condutores na rede MLP base de modo que ocorra uma transformação desses dados pouco discriminatórios para a autenticação desses em um espaço de dados derivados chamados de *embeddings*, onde técnicas simples, que não demandem treinamento, consigam determinar a autenticidade desse condutores. A premissa base das redes neurais siamesas é que dados pertencentes a um mesmo indivíduo quando submetidos à essa sejam agrupados em um espaço próximo em termos de distância e que dados pertencentes a indivíduos distintos se mantenham distantes nesse espaço retornado pela rede. Sendo assim, os doze condutores do grupo de teste, divididos em quatro veículos, são submetidos a cada uma das redes MLP base, formando assim um conjunto de dados derivados para cada veículo.

Uma maneira de observar como os *embeddings* das redes MLP base se comportam é utilizando a Análise de Componentes Principais (PCA), técnica de redução de dimensionalidade descrita na Seção 3.2.2. A dimensão dos *embeddings* é igual ao número de neurônios da camada de saída da rede, que no caso da rede MLP base 2 tem 64 dimensões, e após a ser submetido ao PCA é reduzido a apenas duas dimensões, de modo a ser possível a análise grá-

fica da distribuição dos dados. A Figura 5.12 apresenta a projeção dos *embeddings* dos dados dos condutores do Veículo 1 para todas as redes MLP base com a janela de 60 segundos. Tal qual o esperado, os *embeddings* de cada um dos três condutores do Veículo 1, foram agrupados em regiões distintas, porém observando que em todas topologias os *embeddings* do condutor 3 ficaram um pouco mais espaçados. Contudo, considerando que se trata de uma projeção PCA de duas dimensões, não pode-se afirmar que esses *embeddings* em suas dimensões originais (32 para as redes MLP base 1 e 3 e 64 para as redes MLP base 2 e 4) estão mais ou menos agrupados tal qual projetado pelo PCA. Essa análise indica que o uso do modelo MLP base da rede siamesa demonstra aptidão para a transformação dos dados de condutores em *embeddings* que permitam que uma técnica baseada em distância (que não demanda de treinamento) consiga efetuar a autenticação de condutores.

Figura 5.12 – Projeção PCA dos dados do Veículo 1 para as diferentes redes MLP base testadas para janela de 60 segundos.



(a) Rede MLP base 1

(b) Rede MLP base 2

(c) Rede MLP base 3

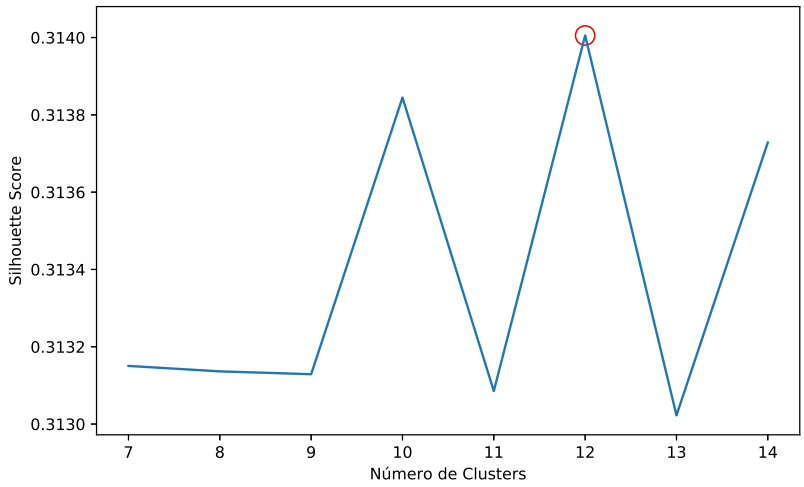
(d) Rede MLP base 4

Fonte: Elaborado pelo Autor (2019).

Uma segunda análise para testar a capacidade da rede MLP base em separar os *embeddings* de condutores distintos é o método de agrupamento k-*means*, descrito no Seção 3.2.3. Esse método, baseado em distância, tem como objetivo encontrar o número adequado de *clus-*

ters para um conjunto de dados. Sendo assim são submetidos à técnica os *embeddings* dos doze condutores de teste e analisa-se de 7 à 14 clusters. Como métrica para determinar o número ideal de clusters é usado o escore de silhueta, que quanto mais alto, mais adequado é o número de grupos. Como são doze condutores de teste é espera-se que o número ideal de clusters seja doze, tal qual o número de condutores. A Figura 5.13 mostra o escore de silhueta para cada quantidade de cluters avaliado. Tal como esperado, o número adequado de clusters foi de doze. Ressalta-se que o k-*means* é uma técnica não-supervisionada, ou seja, não se tem nenhuma informação a priori sobre a distribuição dos dados. Portando, essas análises preliminares sobre a capacidade das redes MLP base em agrupar *embeddings* de um mesmo condutor indicam a aptidão dessa técnica para a autenticação de condutores.

Figura 5.13 – Gráfico de silhueta para agrupamento k-*means*.



Fonte: Elaborado pelo Autor (2019).

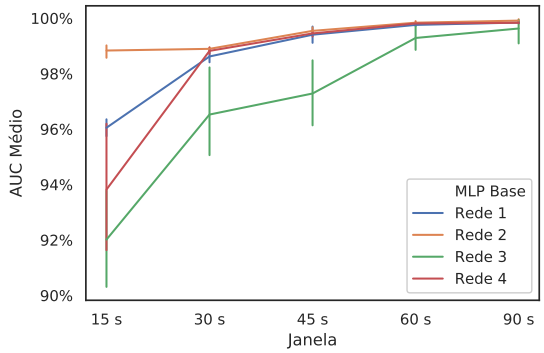
Uma vez analisado o comportamento espacial dos *embeddings* dos dados de entrada dos condutores do grupo de teste e atestado que os mesmos são agrupados em espaços distintos, passa-se agora a análise de autenticação de condutores divididos em veículos de teste. Os experimentos executados, tal qual detalhado na Seção 4.6, consistem na divisão dos doze condutores de teste em quatro veículos de três condutores autorizados a conduzí-lo, simulando membros de uma mesma família ou empresa que fazem uso regular desse veículo. Parte dos dados de cada condutor (40%) é usado para formar o suporte de *embeddings* do veículo que é usado para determinar se *embeddings* de condutores não identificados pertencem ao grupo de condutores autênticos.

Para o teste de capacidade de autenticação de cada veículo testado, são usados dados dos condutores de outros veículos como impostores e o restante dos dados dos condutores do grupo em questão como autênticos. Essa abordagem provoca um desbalanceamento considerável entre a classe positiva e negativa (60% dos dados dos três condutores autênticos e dados de nove condutores impostores), porém esse tratamento radical tem como objetivo submeter a situações onde somente uma pequena parte dos condutores são autênticos. Em aplicações reais, essa situação dificilmente ocorrerá, visto que em condições normais um veículo somente é conduzido por indivíduos autênticos e casos de condutores impostores é tido como evento raro. Logo se o sistema obtiver desempenho satisfatório em tais experimentos extremos, espera-se que esse tenha um bom desempenho em aplicações práticas.

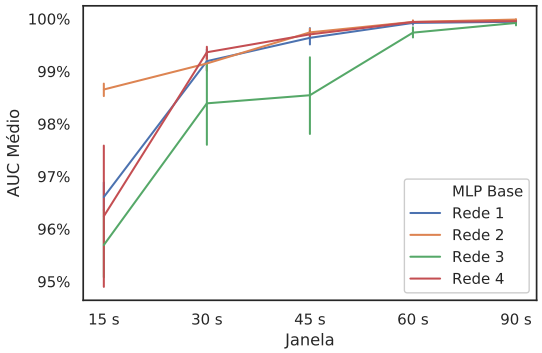
O objetivo desses experimentos supracitados não é avaliar explicitamente a classificação dos condutores em autênticos ou impostores, mas sim avaliar a capacidade da função de decisão de discernir entre esses independentemente do limiar de distância T. Portanto, para essa etapa é observado a curva ROC e a AUC dos experimentos visto que essas métricas observam como o sistema se comporta em diferentes níveis de limiar, propiciando avaliar o *trade-off* entre verdadeiros e falsos condutores autênticos. Sendo assim, para cada combinação de rede MLP base e janela de extração de características, treinados cinco vezes cada, é avaliado a capacidade de autenticação de condutores de cada combinação desses hiperparâmetros.

A Figura 5.14 apresenta graficamente os resultados obtidos para cada um dos veículos definidos. Tal qual como esperado, obteve-se resultados satisfatórios no que diz respeito à capacidade da função de decisão em discernir entre condutores autênticos e impostores representada por meio da AUC. Como exposto anteriormente, ela indica o quanto é possível aferir o quão separável são as duas distribuições de distâncias, de autênticos e impostores, retornados pela função de decisão, independentemente do ponto de corte. Os resultados de cada veículo são semelhantes, visto que em cada um desses as redes MLP bases 1, 2 e 4 obtiveram resultados semelhantes para as janelas de 60 e 90 segundos, com a rede MLP base 3 com resultados um pouco inferiores e maior desvio destes. Um padrão que se manteve com relação aos experimentos anteriores é que as topologias com mais neurônios na camada de saída (redes MLP base 2 e 4) conseguem gerar *embeddings* mais representativos, o que indica que quanto maior a dimensão dos dados derivados, melhor a capacidade da função de decisão de executar sua tarefa de autenticação de condutores.

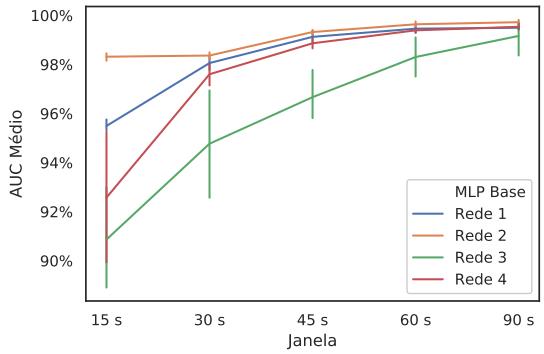
Figura 5.14 – AUC média e seu desvio para cada um dos quatro veículos testados quando confrontado com condutores de outros veículos.



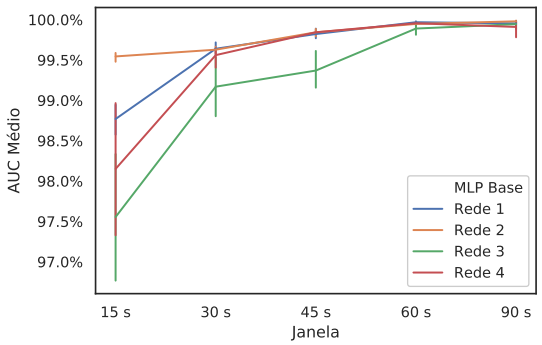
(a) AUC Veículo 1



(b) AUC Veículo 2



(c) AUC Veículo 3



(d) AUC Veículo 4

Fonte: Elaborado pelo Autor (2019).

As Tabelas 5.5, 5.6, 5.7 e 5.8 apresentam as AUC médias e seus respectivos desvios para cada um dos experimentos executados. Os resultados mostram que quanto maior a janela de filtragem e extração de características, maior a eficiência na tarefa de autenticação de condutores, o que mostra a importância dessa etapa, que realiza a filtragem dos dados dos sensores do veículo e do *smartphone*, sabidamente ruidosos. As diferentes estatísticas extraídas nestas janelas inserem no modelo informações de tendência e dispersão dos dados pertinentes para o modelo, que contribuem para a tarefa de autenticação de condutores. A janela de 15 segundos apresentou resultados inferiores às demais, o que mostra que este tamanho de janela ainda não filtra de forma efetiva os ruídos presentes nos dados. Este tamanho de janela, em aplicações práticas, é também o tempo necessário para que a tarefa de autenticação de condutores se inicie pelo sistema, uma vez que a primeira análise pelo modelo só seria efetuada após uma janela ser completada.

Tabela 5.5 – AUC percentual média para condutores do Veículo 1 (classe positiva) e condutores de outros veículos (classe negativa),

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	96,05% ± 0,40%	98,83% ± 0,29%	92,01% ± 2,28%	93,82% ± 3,00%
30 segundos	98,62% ± 0,29%	98,89% ± 0,07%	96,52% ± 2,15%	98,82% ± 0,15%
45 segundos	99,40% ± 0,35%	99,54% ± 0,17%	97,28% ± 1,52%	99,45% ± 0,14%
60 segundos	99,76% ± 0,06%	99,84% ± 0,08%	99,29% ± 0,57%	99,81% ± 0,06%
90 segundos	99,84% ± 0,03%	**99,91% ± 0,07%**	99,63% ± 0,59%	99,83% ± 0,03%

Fonte: Elaborado pelo Autor (2019).

Tabela 5.6 – AUC percentual média para condutores do Veículo 2 (classe positiva) e condutores de outros Veículos (classe negativa),

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	96,61% ± 0,35%	98,65% ± 0,15%	95,69% ± 0,84%	96,24% ± 1,73%
30 segundos	99,19% ± 0,24%	99,15% ± 0,17%	98,39% ± 1,06%	99,36% ± 0,15%
45 segundos	99,64% ± 0,20%	99,74% ± 0,09%	98,54% ± 0,93%	99,70% ± 0,07%
60 segundos	99,92% ± 0,04%	99,94% ± 0,04%	99,74% ± 0,13%	99,94% ± 0,02%
90 segundos	99,94% ± 0,02%	**99,99% ± 0,01%**	99,92% ± 0,06%	99,96% ± 0,03%

Fonte: Elaborado pelo Autor (2019).

Tabela 5.7 – AUC percentual média para condutores do Veículo 3 (classe positiva) e condutores de outros Veículos (classe negativa),

Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	95,49% ± 0,31%	98,31% ± 0,19%	90,86% ± 2,63%	92,57% ± 3,52%
30 segundos	98,06% ± 0,37%	98,36% ± 0,18%	94,77% ± 2,82%	97,60% ± 0,62%
45 segundos	99,12% ± 0,31%	99,32% ± 0,12%	96,67% ± 1,31%	98,86% ± 0,25%
60 segundos	99,46% ± 0,22%	99,64% ± 0,16%	98,30% ± 1,03%	99,39% ± 0,12%
90 segundos	99,50% ± 0,09%	**99,72% ± 0,13%**	99,16% ± 0,87%	99,53% ± 0,15%

Fonte: Elaborado pelo Autor (2019).

Tabela 5.8 – AUC percentual média para condutores do Veículo 4 (classe positiva) e condutores de outros Veículos (classe negativa),

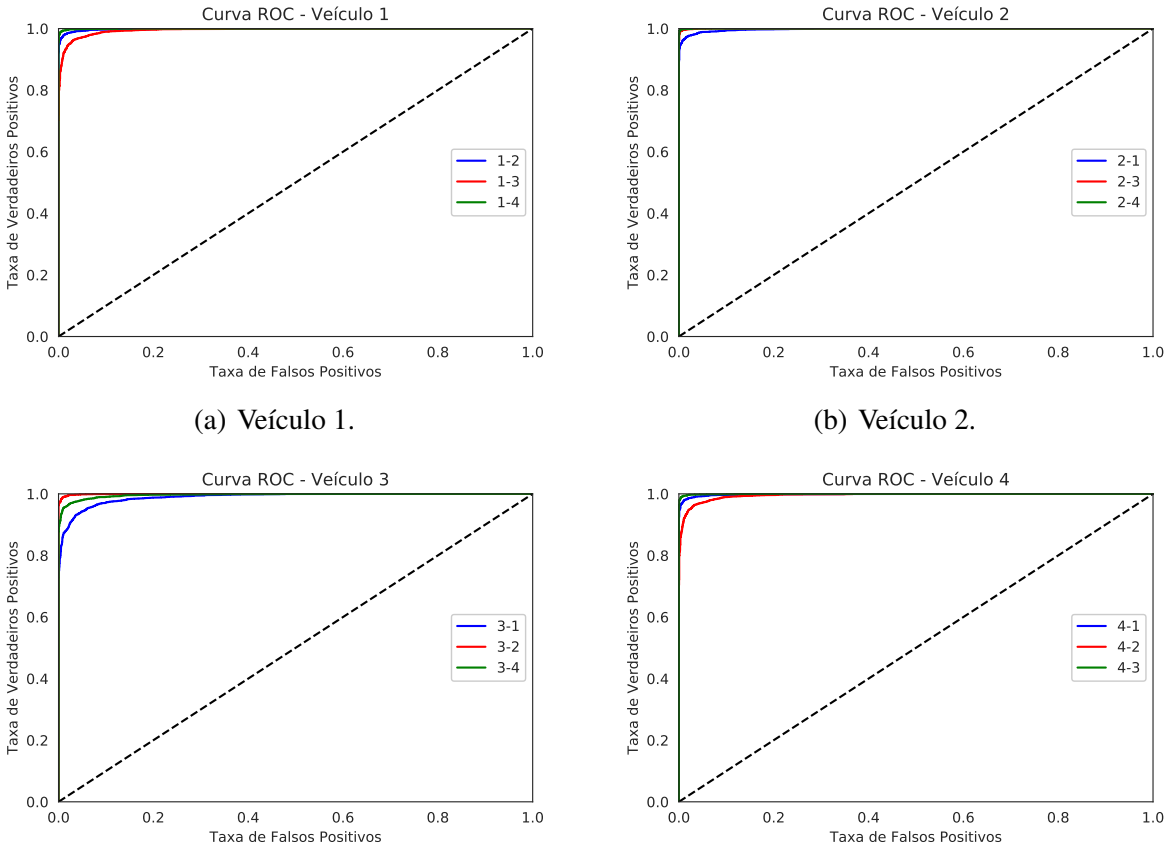
Janela	Topologia MLP base			
	Rede 1	Rede 2	Rede 3	Rede 4
15 segundos	98,77% ± 0,24%	99,54% ± 0,07%	97,55% ± 1,01%	98,15% ± 1,10%
30 segundos	99,64% ± 0,10%	99,63% ± 0,05%	99,17% ± 0,48%	99,56% ± 0,17%
45 segundos	99,82% ± 0,08%	99,84% ± 0,06%	99,37% ± 0,30%	99,84% ± 0,01%
60 segundos	99,97% ± 0,02%	99,95% ± 0,04%	99,89% ± 0,10%	99,96% ± 0,02%
90 segundos	99,95% ± 0,02%	**99,98% ± 0,01%**	99,94% ± 0,07%	99,91% ± 0,14%

Fonte: Elaborado pelo Autor (2019).

Nessa próxima etapa é analisada a curva ROC que, conforme elucidado na Seção 3.3, mostra o *trade-off* entre a taxa de verdadeiros positivos, que nesse caso é a taxa de condutores autênticos identificados como tais, e a taxa de falsos positivos, que se trata dos condutores impostores identificados como autênticos. Com a curva ROC é possível ponderar o limiar de distância que em aplicações práticas deve ser definido, respeitando os níveis aceitáveis de falsos positivos, que deve ser evitado de modo a não permitir que uma possível ação impostora possa ser identificada como sendo autêntica erroneamente. De forma semelhante, é necessário ter o maior nível de certeza ao se reportar uma situação de condutor não autorizado, de tal modo que um condutor autêntico não passe pela incômoda situação de ser tratado como impostor.

A Figura 5.15 mostra a curva ROC para cada um dos veículos testados para a rede MLP base 2 e janela de 60 segundos, combinação essa que obteve melhor desempenho e estabilidade, conforme resultados previamente apresentados. Como considerou-se os condutores de outros veículos como impostores para cada veículo testado, avaliou-se a curva ROC de cada um das combinações individualmente. Tal abordagem permite encontrar possíveis condutores com perfil semelhante de condução, que possam prejudicar o desempenho do sistema. Os resultados apresentados apontam que em alguns casos pode haver semelhanças nos perfis de direção entre os condutores. Nos veículos 1 e 3 é possível observar um desempenho inferior quando esses são confrontados (ressalta-se que avalia-se a capacidade de discriminação entre os condutores dos dois veículos), porém sem perdas de capacidade de autenticação significativas.

Figura 5.15 – Curva ROC para a rede MLP base 2 e janela de 60 segundos de cada veículo testado.



(a) Veículo 1.

(b) Veículo 2.

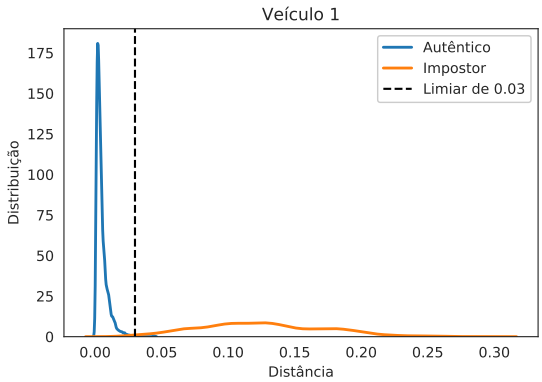
(c) Veículo 3.

(d) Veículo 4.

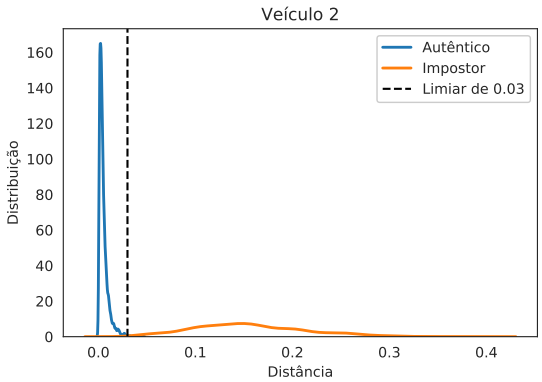
Fonte: Elaborado pelo Autor (2019).

Após analisadas as AUC's e as curvas ROC de cada veículo, é interessante observar o comportamento das distâncias retornadas pelo modelo de vizinhos mais próximos, que baseadas num limiar, separam os condutores autênticos dos impostores. Conforme ressaltado anteriormente, redes neurais siamesas têm por característica aproximar no espaço *embeddings* que pertencem a um mesmo indivíduo e separa àqueles que pertencem à indivíduos diferentes. A função de decisão parte da premissa que uma leitura desconhecida, se pertencente a um condutor autêntico, terá uma distância do *embedding* mais próximo dentro do limiar estabelecido. Sendo assim, a Figura 5.16 apresenta a distribuição das distâncias de *embeddings* de condutores autênticos e impostores e uma fronteira de decisão sendo um limiar de distância fixado em 0,03. É observado que em todos os veículos as distâncias correspondente a condutores autênticos são concentrados abaixo do limiar definido e com baixa dispersão desses. A distribuição das distâncias de condutores impostores possui uma dispersão superior dos valores obtidos, porém com baixa concentração de valores abaixo do limiar. A baixa interseção entre as distribuições indica uma capacidade de discriminação entre os tipos de condutores.

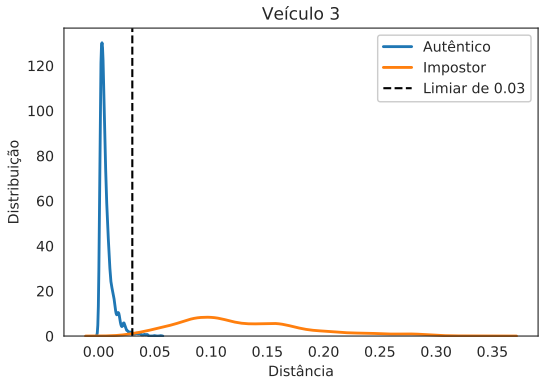
Figura 5.16 – Distribuição de distâncias entre condutores impostores e autênticos para os veículos definidos, para rede MLP base 2 e janela de 60 segundos.



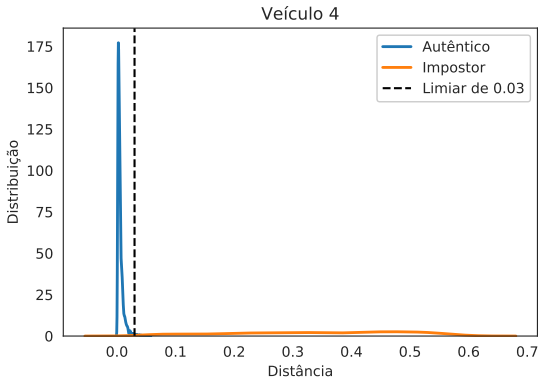
(a) Distribuição de Distâncias Veículo 1



(b) Distribuição de Distâncias Veículo 2



(c) Distribuição de Distâncias Veículo 3



(d) Distribuição de Distâncias Veículo 4

Fonte: Elaborado pelo Autor (2019).

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um sistema de autenticação de condutores de veículos por meio de dados veiculares oriundos da interface OBDII do mesmo e de uma unidade de medições inerciais embarcados em um *smartphone*, que fornecem informações relevantes quanto à dinâmica de direção. Diferente de outros trabalhos da literatura, focou-se no problema de autenticação de novos condutores, ou seja, condutores que não estiveram presentes no treinamento do modelo de autenticação. Para tanto, explorou-se a técnica conhecida como redes neurais siamesas, que na literatura demonstra aptidão para tarefas que envolvem determinar a autenticidade de dados, como reconhecimento facial. Sob essa perspectiva, aplicou-se a metodologia *few shot learning*, que consiste no uso da topologia MLP base da rede siamesa para gerar *embeddings* dos dados de entrada, tal que se necessite somente de algumas amostras conhecidas de um novo condutor para que sejam comparadas com novas leituras que se desejem autenticar, retornando assim o grau de similaridade entre estes *embeddings* gerados. A função de decisão, que realiza a comparação entre estes *embeddings*, é baseada em uma função de vizinho mais próximo, de modo que se a distância do vizinho mais próximo for maior que um limiar definido este é considerado impostor.

Os resultados apresentados indicam a aptidão da aplicação de redes neurais siamesas para a tarefa de autenticação de novos condutores. Na abordagem adotada, os condutores de teste foram agrupados em veículos hipotéticos com três condutores em cada, de modo que os condutores de outros veículos sejam os impostores do veículo em questão. Também não existe a necessidade de recalibração do modelo para a autenticação de novos condutores submetidos ao modelo, sejam eles autênticos ou impostores. Alcançou-se uma AUC-ROC superior à 99% para todos os veículos testados, o que indica uma boa separabilidade dos dados de condutores autênticos e impostores. No entanto, testes mais robustos necessitariam de um número maior de condutores de teste além dos 25 que o conjunto de dados possui, para que o treinamento da rede siamesa tenha a maior variabilidade de perfis de direção possível. Outro ponto importante é que existe a necessidade de testes em mais de uma seção de direção de cada condutor, para que seja avaliado o desempenho do modelo de autenticação em diferentes rotas de direção e condições de tráfego. Entretanto, existe uma escassez de dados semelhantes disponibilizados para esse tipo de estudo e o conjunto de dados utilizado, mesmo com tais limitações, é um dos mais adequados para a abordagem de autenticação de condutores, dado seu número de condutores disponíveis, superior a inúmeros outros trabalhos, sua metodologia de coleta robusta e por essa

coleta ter sido executada em vias brasileiras, o que aproxima os experimentos executados de casos reais.

Ciente dos bons resultados obtidos e do potencial que tal sistema de autenticação de condutores tem de mitigar o problema crônico de roubos e furtos de veículos e também para aplicações que envolvem personalização de configurações para cada usuário do veículo, elenca-se aqui algumas sugestões de aprimoramentos do sistema proposto. A primeira sugestão envolve o objetivo do trabalho, que nesse caso é autenticação dos condutores, o que poderia ser desenvolvido com a intenção de realizar a autenticação e a posterior identificação específica do condutor autorizado, que demandaria aprimoramentos na função de decisão para que esta indique qual o condutor está operando o veículo. Outro fator que pode ser levado em consideração é o caráter temporal dos dados. Uma vez que os dados são produzidos sequencialmente e que uma determinada leitura tem correlação com as anteriores, esta característica pode ser exploradas por topologias de redes neurais que exploram esse caráter temporal. Um exemplo de topologia neural a ser aplicada é a chamada rede neural recorrente, que pode ser utilizada como base para o modelo siamês utilizado por esse trabalho.

REFERÊNCIAS

ABADI, M. et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>. Acesso em: 04/05/2019.

ANDRIA, G. et al. Development of an automotive data acquisition platform for analysis of driving behavior. **Measurement: Journal of the International Measurement Confederation**, v. 93, p. 278–287, 2016. ISSN 02632241.

BANERJEE, M. et al. Beyond kappa: A review of interrater agreement measures. **Canadian journal of statistics**, Wiley Online Library, v. 27, n. 1, p. 3–23, 1999.

BLASZCZYK, P.; TUREK, W.; CETNAROWICZ, K. Extensible platform for studying the behavior of drivers in urban traffic. In: **2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014**. [S.l.]: IEEE, 2014. p. 1359–1362. ISBN 9781479960781.

BREIMAN, L. Random forests. **UC Berkeley TR567**, 1999.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BROMLEY, J. et al. Signature verification using a"siamese"time delay neural network. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1994. p. 737–744.

BURTON, A. et al. Driver identification and authentication with active behavior modeling. In: **2016 12th International Conference on Network and Service Management, CNSM 2016 and Workshops, 3rd International Workshop on Management of SDN and NFV, ManSDN/NFV 2016, and International Workshop on Green ICT and Smart Networking, GISN 2016**. [S.l.]: IEEE, 2017. p. 388–393. ISBN 9783901882852. ISSN 2165-9605.

CAMPO, I. D. et al. A real-time driver identification system based on artificial neural networks and cepstral analysis. In: **Proceedings of the International Joint Conference on Neural Networks**. [S.l.]: IEEE, 2014. p. 1848–1855. ISSN 2161-4393.

CAO, L. et al. A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. **Neurocomputing**, Elsevier, v. 55, n. 1, p. 321–336, 2003.

CHOLLET, F. et al. **Keras**. 2015. <https://keras.io>. Acesso em: 04/05/2019.

CHOPRA, S.; HADSELL, R.; LECUN, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**, IEEE, v. 1, p. 539–546, 2005.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

EBERT, C.; JONES, C. Embedded software: Facts, figures, and future. **Computer**, v. 42, n. 4, p. 42–52, apr 2009. ISSN 00189162.

EZZINI, S.; BERRADA, I.; GHOGHO, M. Who is behind the wheel? Driver identification and fingerprinting. **Journal of Big Data**, Springer International Publishing, v. 5, n. 1, p. 9, dec 2018. ISSN 21961115.

Fórum Brasileiro de Segurança Pública. **12º Anuário Brasileiro de Segurança Pública**. 2018. Disponível em: <http://www.forumseguranca.org.br/atividades/anuario/>. Acesso em: 05/02/2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Acesso em: 04/05/2019.

HARANDI, M.; KUMAR, S. R.; NOCK, R. **Siamese Networks: A Thing or Two to Know**. [S.l.], 2017. Acesso em: 15/03/2019.

HAYKIN, S. S. et al. **Neural networks and learning machines**. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009. v. 3.

HINTON, G.; SRIVASTAVA, N.; SWERSKY, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. **Cited on**, p. 14, 2012.

JAFARNEJAD, S.; CASTIGNANI, G.; ENGEL, T. Towards a Real-Time Driver Identification Mechanism Based on Driving Sensing Data. In: **20th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2017. p. 7.

Jeremy Jordan. **Introduction to autoencoders.** 2018. Disponível em: <https://www.jeremyjordan.me/autoencoders/>. Acesso em: 19/05/2019.

KAPLAN, S. et al. Driver Behavior Analysis for Safe Driving: A Survey. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 6, p. 3017–3032, dec 2015. ISSN 15249050.

KAR, G. et al. PreDriveID: Pre-Trip Driver Identiication from In-Vehicle Data. In: **Proceedings of the Second ACM/IEEE Symposium on Edge Computing - SEC '17**. [S.l.: s.n.], 2017. v. 13, p. 1–12. ISBN 9781450350877.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding groups in data: an introduction to cluster analysis**. [S.l.]: John Wiley & Sons, 2009. v. 344.

KOCH, G.; ZEMEL, R.; SALAKHUTDINOV, R. Siamese Neural Networks for One-shot Image Recognition. **International Conference on Machine Learning**, p. 1–8, 2015. ISSN 19454589.

KUMTEPE, O.; AKAR, G. B.; YUNCU, E. Driver aggressiveness detection via multisensory data fusion. **EURASIP Journal on Image and Video Processing**, v. 2016, n. 1, p. 5, 2016. ISSN 1687-5281.

KWAK, B. I.; WOO, J.; KIM, H. K. Know your master: Driver profiling-based anti-theft method. In: **PST 2016**. [S.l.: s.n.], 2016.

KWAK, B. I.; WOO, J. Y.; KIM, H. K. Know your master: Driver profiling-based anti-theft method. In: **2016 14th Annual Conference on Privacy, Security and Trust, PST 2016**. [S.l.]: IEEE, 2016. p. 211–218. ISBN 9781509043798.

LECUN, Y.; BENGIO, Y. et al. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. 10, p. 1995, 1995.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MARTIN, K. et al. **A Convolutional Siamese Network for Developing Similarity Knowledge in the SelfBACK Dataset**. [S.l.], 2017. Disponível em: <https://github.com/selfback/activity-recognition>. Acesso em: 04/03/2019.

MARTINEZ, M.; ECHANOBE, J.; CAMPO, I. del. Driver identification and impostor detection based on driving behavior signals. In: **2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.]: IEEE, 2016. p. 372–378. ISBN 978-1-5090-1889-5.

MASCI, J. et al. Multimodal similarity-preserving hashing. jul 2012. Disponível em: <http://arxiv.org/abs/1207.1522>. Acesso em: 18/05/2019.

MCKINNEY, W. pandas: a foundational python library for data analysis and statistics. **Python for High Performance and Scientific Computing**, v. 14, 2011.

MEIRING, G. A. M.; MYBURGH, H. C. A review of intelligent driving style analysis systems and related artificial intelligence algorithms. **Sensors (Switzerland)**, Multidisciplinary Digital Publishing Institute, v. 15, n. 12, p. 30653–30682, dec 2015. ISSN 14248220.

MENZE, B. H. et al. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. **BMC bioinformatics**, BioMed Central, v. 10, n. 1, p. 213, 2009.

MIYAJIMA, C. et al. Cepstral Analysis of Driving Behavioral Signals for Driver Identification. In: **2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings**. [S.l.]: IEEE, 2006. v. 5, p. V–921–V–924. ISBN 1-4244-0469-X. ISSN 1520-6149.

MUELLER, J.; THYAGARAJAN, A. Siamese Recurrent Architectures for Learning Sentence Similarity. **Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)**, n. 2012, p. 1386–1393, 2014. ISSN 10636919. Disponível em: <www.aaai.org>. Acesso em: 27/05/2019.

NARGESIAN, F. et al. Learning feature engineering for classification. **IJCAI International Joint Conference on Artificial Intelligence**, p. 2529–2535, jun 2017. ISSN 10450823.

NECULOIU, P.; VERSTEEGH, M.; ROTARU, M. **Learning Text Similarity with Siamese Recurrent Networks**. [S.l.], 2016. 148–157 p. Disponível em: <https://www.aclweb.org/anthology/W16-1617>. Acesso em: 27/04/2019.

NETO, F. M. M.; LACERDA, W. S.; SIMÃO, V. O. Implementação de Redes Neurais em Plataforma GPU. In: **Anais do 12. Congresso Brasileiro de Inteligência Computacional**. [S.l.]: ABRICOM, 2015. p. 1–6. ISBN 9788569972006.

NIELSEN, M. A. **Neural networks and deep learning**. [S.l.]: Determination press San Francisco, CA, USA:, 2015. v. 25.

Oliveira Vasconcelos, I. **Detecção móvel e online de anomalia em múltiplos fluxos de dados: uma abordagem baseada em processamento de eventos complexos para detecção de comportamento de condução**. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil, mar 2017.

PAK, M.; KIM, S. A review of deep learning in image recognition. In: IEEE. **2017 4th international conference on computer applications and information processing technology (CAIPT)**. [S.l.], 2017. p. 1–3.

PAN, Y. J.; YU, T. C.; CHENG, R. S. Using OBD-II data to explore driving behavior model. In: **Proceedings of the 2017 IEEE International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, ICASI 2017**. [S.l.]: IEEE, 2017. p. 1816–1818. ISBN 9781509048977.

PAULINO, A. **Área Abaixo da Curva ROC**. 2018. Disponível em: <https://medium.com/ensina-ai/{á}rea-abaixo-da-curva-roc-15d2ae>. Acesso em: 19/05/2019.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIECH, C. **K Means**. 2013. Disponível em: <https://stanford.edu/{~}cpiech/cs221/handouts/kmeans.ht>. Acesso em: 19/04/2019.

PUSHP, P. K.; SRIVASTAVA, M. M. Train Once, Test Anywhere: Zero-Shot Learning for Text Classification. dec 2017. Disponível em: <http://arxiv.org/abs/1712.05972>. Acesso em: 04/05/2019.

QIAN, C.; HE, T.; ZHANG, R. **Deep Learning based Authorship Identification**. [S.l.], 2016. 1–9 p. Disponível em: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760185.pdf>. Acesso em: 05/05/2019.

RETTORE, P. H. L. et al. Autenticação Comportamental de Motoristas em Redes Veiculares. **Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**, v. 36, 2018. ISSN 2177-9384.

ROSENBLATT, F. Principles of neurodynamics. Spartan Book, 1962.

RUMELHART, D. E. et al. Learning representations by back-propagating errors. **Cognitive modeling**, v. 5, n. 3, p. 1, 1988.

SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 21, n. 3, p. 660–674, 1991.

Sathyanarayana, A.; Boyraz, P.; Hansen, J. H. L. Driver behavior analysis and route recognition by hidden markov models. In: **2008 IEEE International Conference on Vehicular Electronics and Safety**. [S.l.: s.n.], 2008. p. 276–281.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. mar 2015. Disponível em: <http://arxiv.org/abs/1503.03832http://dx.doi.org/10.1109/CVPR.2015.7298682>. Acesso em: 28/03/2019.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. **CoRR**, abs/1503.03832, 2015. Disponível em: <http://arxiv.org/abs/1503.03832>. Acesso em: 12/05/2019.

SHAHAM, U.; LEDERMAN, R. Common Variable Learning and Invariant Representation Learning using Siamese Neural Networks. dec 2015. Disponível em: <http://arxiv.org/abs/1512.08806>. Acesso em: 05/04/2019.

SOUZA, A. de et al. Sistema de identificação de condutores baseado em métodos de extração de características estatísticas e técnicas de redução de dimensionalidade. In: **XXII Congresso Brasileiro de Automática**. [S.l.]: SBA, 2018.

SOUZA, C. R. de. **Análise de Poder Discriminativo Através de Curvas ROC – César Souza**. 2009. Disponível em: <http://crsouza.com/2009/07/13/analise-de-poder-discriminativo-atraves-de-curvas-roc/>. Acesso em: 12/05/2019.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SUMMALA, H. Brake Reaction Times and Driver Behavior Analysis. **Transportation Human Factors**, Routledge, v. 2, n. 3, p. 217–226, 2000.

TSCHANNEN, M.; BACHEM, O.; LUCIC, M. Recent Advances in Autoencoder-Based Representation Learning. 2018. Disponível em: <https://arxiv.org/pdf/1812.05069.pdfhttp://arxiv.org/abs/1812.05069>. Acesso em: 20/05/2019.

TUOHY, S. et al. Intra-Vehicle Networks: A Review. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 2, p. 534–545, apr 2015. ISSN 1524-9050.

VASCONCELOS, I. O. **DETECÇÃO MÓVEL E ONLINE DE ANOMALIA EM MÚLTIPLOS FLUXOS DE DADOS: UMA ABORDAGEM BASEADA EM PROCESSAMENTO DE EVENTOS COMPLEXOS PARA DETECÇÃO DE COMPORTAMENTO DE CONDUÇÃO**. Tese (Doutorado) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, Rio de Janeiro, Brazil, mar 2017.

VAZ, A. L. **Model Validation — Data Science**. 2018. Disponível em: <https://medium.com/@arthurlambletvaz/model-validation-data-science-3084bb3a4ff8>. Acesso em: 04/05/2019.

WAKITA, T. et al. Driver identification using driving behavior signals. In: **Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.** [S.l.]: IEEE, 2005. v. 2005, n. 3, p. 907–912. ISBN 0-7803-9215-9. ISSN 09168532.

WANG, B. et al. Driver Identification Using Vehicle Telematics Data. **WCX™ 17: SAE World Congress Experience**, mar 2017. ISSN 01487191.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. **Chemometrics and intelligent laboratory systems**, Elsevier, v. 2, n. 1-3, p. 37–52, 1987.

ZHANG, H. et al. Image classification using rapideye data: Integration of spectral and textual features in a random forest classifier. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 10, n. 12, p. 5334–5349, Dec 2017. ISSN 1939-1404.