

Análise de comportamento de direção de motoristas por meio de um simulador preditivo

Driving behavior analysis of drivers using a predictive simulator

Análisis de conducta de manejo de conductores por medio de simulador predictivo

Resumo

Os acidentes e as fatalidades no trânsito brasileiro ocorrem especialmente por falha humana, normalmente associada ao comportamento e atitudes indevidas do condutor do veículo, como abuso de velocidade, consumo de álcool e outras infrações à legislação de trânsito. A análise de comportamento de direção pode ser aplicada para auxiliar o motorista em averiguar a qualidade de sua condução. Este trabalho desenvolveu um protótipo de *software* para analisar a direção de motoristas fazendo uso de *machine learning* e uma base de dados pública para geração do seu modelo preditivo. Para isso, foi realizada uma revisão sistemática da literatura para identificar os métodos de *machine learning* mais utilizados. Baseado nos resultados dessa revisão, foi proposto uma solução que considera o algoritmo de correspondência de padrão, baseado na técnica DTW e o uso de acelerômetros do *smartphone* para capturar os dados de movimento. O modelo gerado com DTW e a base de dados obtida encontraram uma acurácia de 84,21%. Esse modelo foi utilizado para compor o protótipo *Canary Driver Alert* que lê um arquivo contendo dados de condução de motoristas coletados por aplicativo de *smartphone*, bem como identifica o evento de direção realizado. O *Canary Driver Alert* foi avaliado por meio de 12 experimentos de viagens com diferentes eventos de direção e foi observado que em vias com terreno plano identifica adequadamente os eventos de direção, em particular a aceleração e frenagem agressivas.

Palavras-chave: Comportamento de direção; Mineração de dados; Aprendizado de máquina; DTW; Eventos de direção; Protótipo de *Software*.

Abstract

Accidents and fatalities in Brazilian traffic occur mainly due to human error, usually associated with inappropriate behavior such as excessive speed, alcohol consumption, and other traffic law violations. Driving behavior analysis can be applied to assist the driver in ascertaining the quality of his driving. This work developed a software prototype to analyze drivers using machine learning and a public database to generate the predictive model. For this, a systematic literature review was carried out to identify the most used machine learning methods. Based on the results of this review, a solution was proposed that considers the pattern matching algorithm based on the DTW technique and the use of smartphone accelerometers to capture the motion data. This model was generated with DTW, and the obtained database achieved an accuracy of 84.21%. This model was used to compose a prototype named Canary Driver Alert. It reads driving data files obtained by smartphone applications and identifies driving events. The Canary Driver Alert was evaluated through twelve travel experiments with different driving events, and it was observed that on roads on flat ground, it identifies properly driving events, aggressive acceleration and braking, in particular.

Keywords: *Driving behavior; Data mining; Machine learning; DTW; Driving event; Software prototype.*

Eduardo Tamanaha  
eduardo.massao.tamanaha@gmail.com

Ivan Carlos Alcântara de Oliveira  
ivan.oliveira@mackenzie.br

Resumen

Los accidentes y las fatalidades en el tránsito brasileños ocurren especialmente por fallo humano, normalmente en conjunto con la conducta y actitudes indebidas del conductor del vehículo, como exceso de abuso de velocidad, consumo de alcohol y otras infracciones a la legislación de tránsito. El análisis de conducta de manejo puede ser aplicado para ayudar al conductor a evaluar la calidad de su manejo. Este trabajo desarrolló un prototipo de software para analizar el manejo de conductores utilizando machine learning y una base pública de datos para generación de su modelo predictivo. Para eso, se realizó una revisión sistemática de la literatura para identificar los métodos de machine learning más utilizados. Basado en los resultados de esta revisión, fue propuesta una solución que considera el algoritmo de correspondencia estándar, basado en la técnica DTW y el uso de acelerómetros del Smartphone para capturar los datos de movimiento. El modelo generado con DTW y la base de datos obtenida encontró una precisión de 84,21%. Este modelo fue utilizado para componer el prototipo Canary Driver Alert que lee un archivo conteniendo datos de manejo de conductores colectados por aplicación de Smartphone e identifica el evento de conducción realizado. El Canary Driver Alert fue evaluado por medio de doce experimentos de viajes con diferentes eventos de dirección y fue observado que en vías con suelo plano identifica adecuadamente los eventos de dirección, en particular la aceleración y frenado agresivos.

Palabras-clave: Conducta de manejo; Minería de datos; Aprendizaje de máquina; DTW; Eventos de dirección; Prototipo de Software.

1 Introdução

Em 2018, 90% dos acidentes ocorridos nas rodovias no Brasil foram devido a fatores humanos, conforme ilustrado no Quadro 1 (POLÍCIA RODOVIÁRIA FEDERAL, 2019). Observa-se que mais de 50% dos acidentes estão relacionados a: i) falta de atenção à condução; ii) desobediência às normas de trânsito e iii) velocidade incompatível. Fundamentado nessas informações, nota-se que o comportamento de direção do motorista afeta a segurança no trânsito e é o principal motivo para acidentes nas rodovias brasileiras.

Segundo Hong, Margines, Dey (2014), diferentes abordagens de avaliação de comportamento de direção podem ser investigadas, como: a) questionários de comportamento de direção; b) simuladores; c) exame supervisionado por perito e d) detecção de eventos de direção utilizando tecnologia de sensores de veículos. Os autores destacam que estudos relacionados às tecnologias mostram que aplicativos em *smartphones* podem coletar informações úteis para caracterizar estilos de direção.

O tema comportamento de direção teve um crescente aumento de interesse em diferentes contextos de aplicações e podem ser colocados em prática em diversos setores, como: na gestão de frotas de veículos (transportadoras) para monitoramento dos veículos em tempo real; no ramo de seguros (seguradoras de veículos), com o conceito *Usage-Based Insurance* (UBI) ou *Pay As You Drive* (PAYD), em que o cliente (conductor) paga seu seguro conforme o uso do veículo (Castignani *et al.*, 2015).

Este artigo tem como objetivo apresentar o desenvolvimento do protótipo de *software Canary Driver Alert*, capaz de analisar a direção de motoristas, bem como expor o processo para geração do seu modelo preditivo, fazendo uso de uma base de dados pública e da técnica *Dynamic Time Warping* (DTW), encontrando uma acurácia de 84,21%. Além disso, serão destacadas a revisão sistemática da literatura, que permitiu encontrar algumas técnicas de *machine learning*, incluindo a DTW, utilizadas para detectar eventos de direção de motoristas, e os resultados de doze experimentos reais avaliados como protótipo.

Quadro 1 – Números e causas de acidentes nas rodovias brasileiras em 2018.

ACIDENTES POR CAUSA HUMANA	Quantidade	Percentual
Falta de atenção à condução	25.799	37,28%
Desobediência às normas de trânsito pelo condutor	7.178	10,37%
Velocidade incompatível	6.848	9,90%
Ingestão de álcool	5.196	7,51%
Não guardar distância de segurança	4.255	6,15%
Defeito mecânico no veículo	3.714	5,37%
Condutor dormindo	2.632	3,80%
Falta de atenção do pedestre	2.071	2,99%
Ultrapassagem indevida	1.332	1,92%
Avarias e/ou desgaste excessivo no pneu	1.009	1,46%
Mal súbito	677	0,98%
Carga excessiva e/ou mal acondicionada	366	0,53%
Agressão externa	353	0,51%
Ingestão de álcool e/ou substâncias psicoativas pelo pedestre	297	0,43%
Desobediência às normas de trânsito pelo pedestre	273	0,39%
Defeito ou não acionamento do sistema de iluminação	186	0,27%
Ingestão de substâncias psicoativas	73	0,11%
Total	62.259	89,96%
ACIDENTES POR CAUSA NÃO HUMANA		
Pista escorregadia	2.545	3,68%
Animais na pista	1.609	2,32%
Defeito na via	1.117	1,61%
Restrição de visibilidade	539	0,78%
Objeto estático sobre o leito carroçável	538	0,78%
Fenômenos da natureza	309	0,45%
Sinalização da via insuficiente ou inadequada	290	0,42%
Total	6.947	10,04%
TOTAL GERAL	69.206	

Fonte: Polícia Rodoviária Federal, 2019.

2 Metodologia

Neste trabalho, foi realizada uma revisão sistemática da literatura, como detalhada em Felizardo *et al.* (2017). Nessa revisão, foi feita a definição do problema, a elaboração de questões de pesquisa, a seleção de estudos e o desenvolvimento de um protocolo, conforme apresentado na subseção 2.1. Com os resultados dessa revisão, foram selecionados alguns textos em bases de trabalhos científicos. Baseado neles, foi escolhido o algoritmo DTW *Dynamic Time Warping*, por realizar avaliação de séries temporais e fornecer resultados melhores em trabalhos similares. Esse algoritmo está demonstrado na subseção 2.2. A geração do modelo preditivo, para realização de experimentos posteriores, é descrito na subseção 2.3, e o desenvolvimento de um protótipo de *software* é exposto na subseção 2.4.

2.1 Revisão sistemática

Com base na questão de pesquisa –“Quais métodos podem ser utilizados para desenvolver um modelo preditivo capaz de fornecer resultados que permitam avaliar a viagem de um motorista?” –, realizou-se uma revisão sistemática da literatura, conforme Felizardo *et al.* (2017). A revisão sistemática utilizou as bases científicas: IEEE *Digital Library*, ACM *Digital Libray*, SCOPUS *Digital Library*, *Web of Science*, *Springere Plos One*. As palavras chaves utilizadas na busca foram: *driving behavior*, *driver behavior*, *machine learning*, *data mining*, *algorithm*, *classifier model*, *classification model*, *smartphone sensor* e *Obd*. De um total de 218 artigos identificados, 9 artigos foram selecionados pelos critérios: uso de sensores do *smartphone*, detecção de eventos de direção e uso de técnicas de *machine learning*, conforme ilustrado no Quadro 2.

Quadro 2 – Trabalhos relacionados contendo técnicas de detecção de eventos de direção e sensores baseados em *smartphone*

Referência	Técnica de detecção	Sensores utilizados	
Carvalho et al. (2017)	<i>Exploiting the use of recurrent neural networks for driver behavior profiling</i>	<i>Recurrent Neural Networks</i> (RNN)	Acelerômetro
Castignani et al. (2015)	<i>Driver behavior profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring</i>	Lógica Fuzzy	GPS, acelerômetro, magnetômetro, tempo e hora do dia
Castignani et al. (2017)	<i>Smartphone-Based Adaptive Driving Maneuver Detection: A Large-Scale Evaluation Study</i>	<i>Multivariate Normal distribution</i> (MVN)	Acelerômetro e GPS
Engelbrecht et al. (2015)	<i>Performance Comparison of Dynamic Time Warping (DTW) and a Maximum Likelihood (ML) Classifier in Measuring Driver Behavior with Smartphones</i>	Classificador <i>Maximum Likelihood</i> (ML) e DTW	GPS, acelerômetro e giroscópio
Eren et al. (2012)	<i>Estimating Driving Behavior by a Smartphone</i>	Detecção de ponto final, DTW e classificador Bayesian	Acelerômetro, giroscópio, magnetômetro
Fazeen et al. (2012)	<i>Safe Driving Using Mobile Phones</i>	Reconhecimento de padrão	Acelerômetro e GPS
Ferreira et al. (2017)	<i>Driver behavior profiling: An investigation with different smartphone sensors and machine learning</i>	<i>Bayesian Network</i> (BN), <i>Multilayer Perceptron</i> (MLP), <i>Random Forest</i> (RF) e <i>Support Vector Machine</i> (SVM)	Acelerômetro, giroscópio, magnetômetro
Johnson e Trivedi (2011)	<i>Driving style recognition using a smartphone as a sensor platform</i>	Detecção de ponto final e DTW	Acelerômetro, giroscópio, magnetômetro e GPS
Saiprasert, Pholprasit e Thajchayapong (2015)	<i>Detection of Driving Events using Sensory Data on Smartphone</i>	Algoritmo baseado em regra, algoritmo reconhecimento de padrão com DTW e algoritmo de correspondência padrão autoativada	Acelerômetro e GPS

Fonte: Elaborado pelo autor.

Os trabalhos selecionados possuem três aspectos fundamentais:

- A identificação e classificação de eventos de direção, como aceleração, frenagem, trocas de faixas ou curvas e a categorização de comportamento do evento como agressivo ou não agressivo.
- A utilização de diferentes técnicas e algoritmos de *machine learning*.
- A proliferação de aparelhos *mobile* do tipo celular e *tablet*, que proporcionam avanço na utilização da plataforma de sensores e conectividade, e o uso de sensores de *smartphones* como acelerômetro, giroscópio, magnetômetro e GPS, individualmente ou combinados.

Dentre os trabalhos selecionados, a utilização de sensores de *smartphone* em conjunto com o algoritmo DTW demonstraram a capacidade de comparar séries temporais com eficiência e acurácia. Por este motivo, foi o escolhido para ser utilizado neste trabalho. O algoritmo DTW é simples, não requer modelos matemáticos complexos e seu único requisito é uma base de dados para treinamento (POLI *et al.*, 2007).

2.2 Descrição do algoritmo *Dynamic Time Warping*

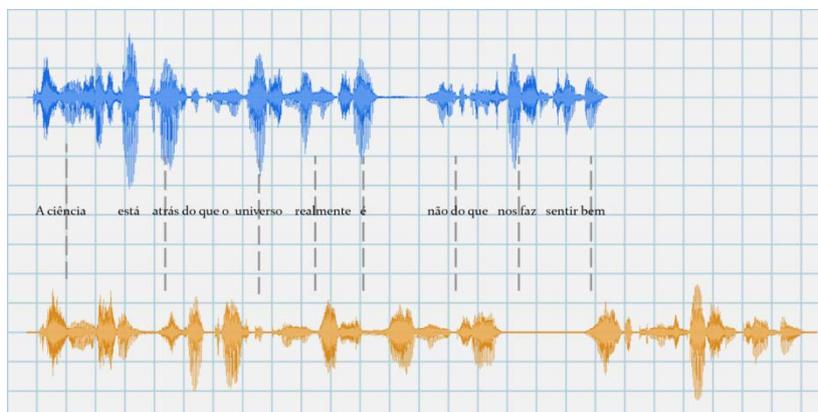
O DTW é um algoritmo que foi utilizado originalmente para comparar diferentes padrões em reconhecimento de fala (SAKOE; CHIBA, 1978). É uma técnica para medir a similaridade entre duas séries temporais que podem variar em tempo e velocidade. Por exemplo, o caminhar de duas pessoas em diferentes velocidades, o reconhecimento de fala em diferentes velocidades e diferentes taxas de pronúncia.

Comparar os dados entre séries temporais diretamente (pareamento Euclidiano) é restritivo e não identifica as similaridades entre as séries, porém, a técnica de pareamento do DTW permite que duas séries correspondam uniformemente, mesmo que os eixos de tempo não estejam sincronizados. Como exemplo, a frase “A ciência está atrás do que o universo realmente é, não do que nos faz sentir bem.” de Carl Sagan, foi gravada e é visualizada no gráfico de áudio em dois exemplos com entonações e velocidades diferentes, como

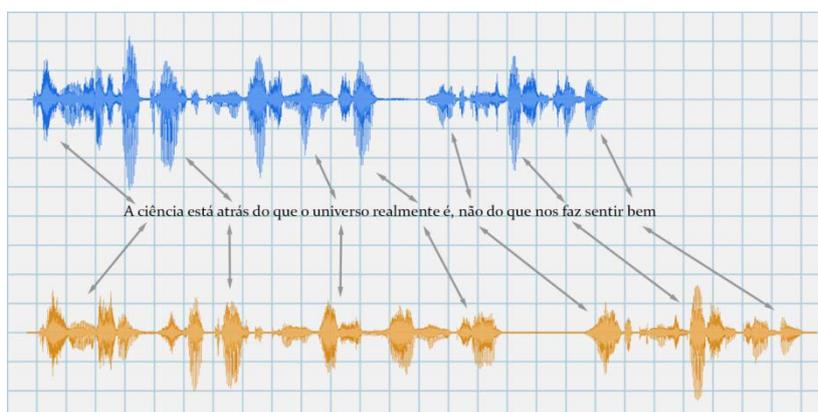
destacado na Figura 1. Se a abordagem utilizada é a euclidiana tradicional, os tempos entre o clipe azul e o clipe laranja não coincidirão e resultará em uma medição inadequada. Com a técnica DTW, o tempo é alinhado e permite uma medição mais próxima e adequada entre os dois clipes de áudio, comparando os padrões de som.

Figura 1 – Áudio de Correspondência Euclidiana versus DTW

Correspondência Euclidiana:



Correspondência por DTW:



Fonte: Elaborado pelo autor.

O conjunto de dados a ser analisado pode não possuir o mesmo número de elementos, porém, o algoritmo possibilita calcular a medição da melhor similaridade entre os exemplares. A técnica converte os dados em vetores, calcula o caminho otimizado e obtém a distância entre esses pontos. Quanto menor o valor encontrado da distância, maior será a similaridade das duas séries temporais. Na Figura 2 é ilustrado um exemplo de caminho otimizado de uma matriz DTW. Os procedimentos e as respectivas fórmulas para calcular a matriz DTW são descritas a seguir:

- a) A linha 1, coluna 1 inicia sempre com o valor zero.
- b) O cálculo da linha $i = 1$, coluna $j = 2$ e demais colunas dessa linha é destacado na Equação 1.

$$|A_i - B_j| + \text{Matriz_dtw}[i, j - 1] \quad (1)$$

- c) O cálculo da coluna $j = 1$ e demais linhas da coluna 1 é ilustrado na Equação 2.

$$|A_i - B_j| + \text{Matrix_dtw}[i - 1, j] \quad (2)$$

- d) O cálculo para as demais células da matriz (todos outros elementos) é apresentado na Equação 3.

$$|A_i - B_j| + \min(\text{Matriz_dtw}[i - 1, j - 1], \text{Matriz_dtw}[i - 1, j], \text{Matriz_dtw}[i, j - 1]) \quad (3)$$

Onde:

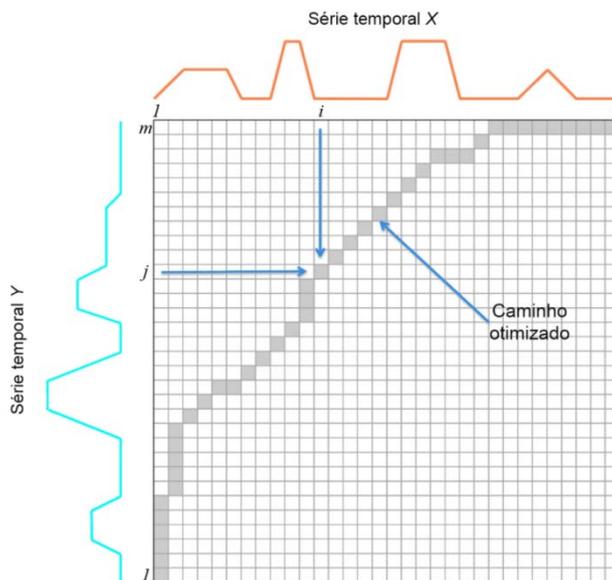
A = Série temporal do eixo X

B = Série temporal do eixo Y

i = Índice do eixo X

j = Índice do eixo Y

Figura 2 – Exemplo de DTW



Fonte: Adaptado de Saiprasert, Pholprasit e Thajchayapong (2015).

Após encontrar o caminho otimizado, calcula-se a função C – Custo Acumulado da Matriz, descrita na Equação 4.

$$C(X, Y) = \text{Somatório } w_k(x_{nk}, y_{mk}) \tag{4}$$

Onde:

C = Custo acumulado

X = Série temporal A

Y = Série temporal B

W_k = Caminho otimizado

x_{nk} = Elemento da matriz linha i

y_{mk} = Elemento da matriz coluna j

O cálculo para se obter o valor da distância entre duas séries temporais é dado pela Equação 5.

Onde:

Serie_A = Série temporal do eixo X

Serie_B = Série temporal do eixo Y

i = Índice do eixo X

j = Índice do eixo Y

Abaixo é demonstrado o exemplo de cálculo da coordenada 3,2 (linha 3 e coluna 2) da Matriz DTW e obtenção do resultado de valor 10, conforme ilustrado na Figura 3.

Exemplo de cálculo: $\text{Matriz_dtw}[3,2] = (2 - 3)^2 + \min(9,25,26) = 1 + 9 = 10$.

Deve-se repetir o cálculo para todos os elementos da matriz. O valor da Distância entre as duas séries temporais será o último elemento da matriz encontrado na coordenada 10,9 (linha 10 e coluna 9) no valor de 52 (cor laranja). Quanto menor o valor da distância, maior será a similaridade entre as duas séries temporais.

Figura 3 – Exemplo de Matriz DTW

		COLUNA									
		1	2	3	4	5	6	7	8	9	
LINHA	SÉRIE A	0	inf								
1	1	inf	0	4	13	77	126	127	127	143	179
2	6	inf	25	9	8	17	21	37	62	63	64
3	2	inf	26	10	12	57	53	21	22	31	56
4	3	inf	30	10	11	47	72	22	25	26	42
5	0	inf	31	19	26	92	111	26	23	48	75
6	9	inf	95	55	44	26	27	75	87	39	43
7	4	inf	104	56	44	51	42	31	40	40	48
8	3	inf	108	56	45	80	67	32	35	39	55
9	6	inf	133	65	49	54	58	48	57	36	37
10	3	inf	137	65	50	85	79	49	52	40	52

SÉRIE B	1	3	4	9	8	2	1	5	7
---------	---	---	---	---	---	---	---	---	---

Último elemento da Matriz

2.3 Geração do modelo preditivo

Para a geração do modelo preditivo foi utilizada a linguagem de programação Python, a biblioteca DTW, a plataforma Anaconda e o *Integrated Development Environment* (IDE) Spyder para edição de *scripts*, testes e *debugging*. A base de dados pública utilizada foi a *Driver Behavior Dataset* disponibilizada por Ferreira *et al.* (2017). Essa base contém um conjunto de dados de medições dos sensores do *smartphone* e para este trabalho foi utilizado o sensor acelerômetro e os eixos x e y que indicam respectivamente movimentos laterais como curvas à direita e à esquerda e mudanças de faixa à esquerda e à direita, além de movimentos longitudinais como aceleração e frenagem. Esse conjunto de dados contém dados de quatro viagens que totalizam 156.516 registros e 69 exemplos de tipos de eventos de direção rotulados de 1 a 7, conforme Quadro 3.

Na geração do modelo preditivo, foi realizado o pré-processamento dos dados da base pública, como rotulagem dos eventos, conversão de valores de unidades e tratamento dos eixos x e y do acelerômetro e tipos de eventos conforme o sentido de direção correspondente, e, por fim, foi gerado um arquivo contendo os dados preparados para o formato de lista, adequado para o uso da função do algoritmo DTW.

A aplicação do algoritmo DTW para a geração do modelo preditivo teve como base o uso do DTW em um programa classificador de movimentos de séries temporais de López (2017) para identificar movimentos do controle do videogame Nintendo Wii, que traduz os dados dos sensores de movimento em comandos nos jogos. Efetuou-se a divisão aleatória dos tipos de eventos dos exemplares em 70% para treino (50 eventos) e 30% para testes (19 eventos).

O funcionamento do modelo preditivo tem os seguintes passos: ler o registro da base de teste; ler todos os registros da base de treino; submeter as duas séries temporais para o algoritmo DTW; obter o cálculo da distância; e armazenar o resultado da distância de cada leitura. Ao término da leitura de todos os registros da base de treino, o modelo verifica qual das séries possui o menor valor da Distância obtida. O registro com menor valor da distância é o mais semelhante da base de treino. O modelo realiza a leitura do próximo registro da base de teste e são repetidos os procedimentos anteriores até o término da leitura da base de teste.

Quadro 3 – Tipos de eventos de direção rotulados

Tipos de evento de direção	Número do rótulo	Quantidade de exemplos
Aceleração agressiva	1	12
Frenagem agressiva	2	12
Curva agressiva à direita	3	11
Curva agressiva à esquerda	4	11
Mudança agressiva de faixa à direita	5	5
Mudança agressiva de faixa à esquerda	6	4
Evento não agressivo	7	14
Total		69

Fonte: Adaptado de Ferreira *et al.* (2017)

Para obter a estatística de acurácia, foi comparado o rótulo da base de treino e teste, se forem iguais, o modelo acertou, se não forem iguais, o modelo errou. O resultado obtido na classificação pelo modelo preditivo foi de 84,21% de acurácia. Dos 19 eventos de teste, 16 eventos foram identificados corretamente.

2.4 Desenvolvimento do protótipo *Canary Driver Alert*

A análise de comportamento de direção tem como proposta obter os resultados de identificação dos eventos que o motorista realiza nos tipos: aceleração agressiva, frenagem agressiva, curva e mudança agressiva de faixa à esquerda e direita e evento não agressivo. O protótipo exibe os tipos de eventos identificados, a quantidade de eventos detectados, a data e hora dos eventos e data e hora do início e final da viagem do percurso realizado do motorista. Os dados da entrada para o *Canary Driver Alert*, na forma de um arquivo *Comma Separated Value* (CSV), podem ser capturados por um *software* que coleta os dados da viagem na frequência de 10Hz, como o Torque Pro, escolhido devido seu baixo custo. No entanto, qualquer outro *software* pode ser utilizado, desde que seja capaz de capturar os dados dos sensores de acelerômetro do *smartphone* e transcrevê-los num arquivo texto.

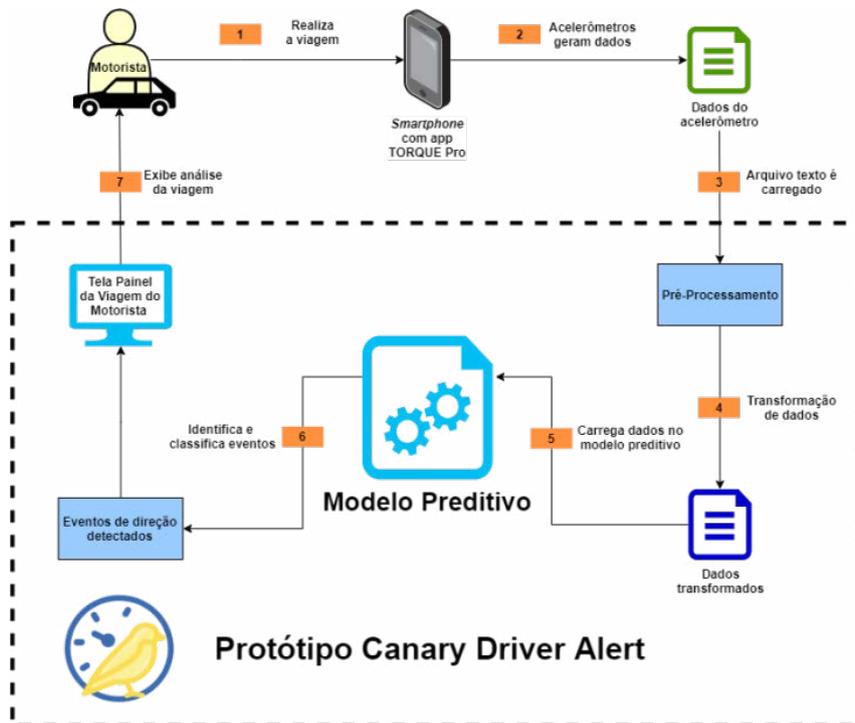
A arquitetura da solução implementada pode ser visualizada na Figura 4. A sequência das etapas apresentadas nessa figura são:

- 1) O motorista realiza a viagem;
- 2) O aplicativo Torque Pro captura os dados acelerômetro. Ao final da viagem, gera o arquivo no formato do tipo CSV e transfere para o servidor;
- 3) O arquivo com os dados da viagem é copiado do servidor para a pasta “Entrada” do computador no qual está instalado o protótipo *Canary Driver Alert*;
- 4) É realizado o pré-processamento e transformação dos dados por um *script* adequadamente elaborado;
- 5) Os dados tratados são processados pelo modelo preditivo;
- 6) O modelo preditivo identifica e classifica os eventos de direção;
- 7) Os eventos de direção identificados são exibidos na tela painel do protótipo.

Na elaboração do *script* da etapa 4, foi utilizado o conceito da janela móvel de Ferreira *et al.* (2017). O motivo foi que nos testes notou-se que uma série temporal de uma viagem completa, por exemplo, de 15 minutos (900 seg) é muito longa para ser comparado com um evento de 6 segundos da base de treino, ou seja, a comparação é inadequada (900 seg x 6 seg). Nesse cenário, é conveniente realizar comparações com pedaços de tempo de tamanho compatível. Então, como solução, foi implementado a janela móvel, em que uma viagem é dividida em pedaços de 6 segundos e, cada um deles, é comparado com a base de treino de mesmo tamanho. Os autores identificaram em seus experimentos que o tempo médio do evento de direção mais longo (evento não agressivo) é de 6 segundos e o evento mais curto é de 2 segundos (evento agressivo). Por isso, o tamanho da janela móvel do protótipo é de 6 segundos. A janela móvel desliza em segmentos de 1 segundo, deste modo a janela captura completamente qualquer evento de direção. O processo é repetido até

o fim da leitura de todos os dados da viagem e ao final gera a base com os dados tratados, conforme ilustrado na Figura 5.

Figura 4 – Arquitetura da solução do protótipo de software



Fonte: Elaborado pelo autor.

Figura 5 – Exemplo de janela móvel.



Fonte: Elaborado pelo autor.

3 Resultados e discussão

Nesta seção, apresenta-se a interface do software *Canary Driver Alert* (subseção 3.1), além dos resultados obtidos com o uso do protótipo e experimentos realizados na condução de veículos nas ruas em cenário real (subseção 3.2).

3.1 Interface da solução

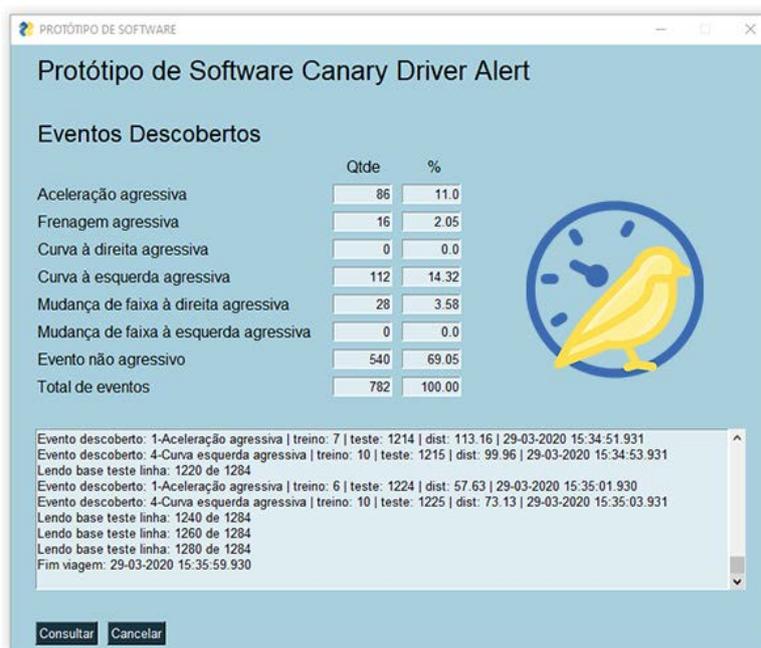
Nesta subseção, é apresentada a interface criada para o software *Canary Driver Alert*, ilustrada na Figura 6. A tela foi desenvolvida com a biblioteca *PySimpleGUI* do Python. O motorista transfere o arquivo dos dados da viagem do servidor para a pasta “Entrada” do computador no qual está instalado o protótipo, executa-o e clica no botão “Consultar”. O *Canary Driver Alert* realiza o processamento dos dados, aplica o modelo preditivo e ao final do processo apresenta o resumo dos eventos identificados. Nesse resumo são exibidos: a quantidade e percentual de eventos identificados de cada tipo de evento, como aceleração agressiva, frenagem agressiva, curva agressiva à esquerda, curva agressiva à direita, mudança agressiva de faixa à esquerda, mudança agressiva de faixa à direita e evento não agressivo. Na caixa inferior da tela, são exibidas a lista dos eventos identificados e a data e hora das ocorrências.

3.2 Experimentos e resultados

Doze experimentos reais foram realizados para a identificação de eventos de direção de motoristas, avaliação e validação do protótipo *Canary Driver Alert*. Os experimentos foram realizados em dois veículos, um Honda Fit 2012 de câmbio automático e um Corsa Sedan 1998 de câmbio manual. Foram utilizados um *smartphone* moto G5 com sistema operacional Android, aplicativo de captura de dados dos sensores do *smartphone* Torque Pro, câmera veicular Fhd 1080, cartão de memória de 32 Gb e suporte de ventosa para celular. O *smartphone* foi fixado internamente nos veículos na posição horizontal com a tela posicionada para cima. A câmera veicular foi instalada no para-brisa do lado interno para gravação das imagens. O motorista é um dos autores do trabalho, possui 37 anos de experiência de direção, dirigiu os veículos em cenário real de vias e locais coletoras da capital do estado de São Paulo (Brasil), em condição regular e em dias de sol e pouco trânsito. Experimentos com eventos de curvas e mudanças de faixa agressivas não foram realizados devido ao risco de acidentes. Para experimentos em sua totalidade é necessário pista exclusiva de testes em ambiente controlado e seguro.

Os doze experimentos são apresentados no Quadro 4, sendo 10 eventos identificados corretamente (cor verde) e 2 eventos identificados incorretamente (cor laranja). Esses 2 eventos identificados incorretamente podem indicar problemas na captura dos dados, devido a posição do *smartphone* em subidas ou descidas, que interfere no ângulo de inclinação e altera os valores dos eixos x ou y, devido a força da gravidade influenciando o acelerômetro. Nos experimentos realizados, foi obtido um percentual de 83,3% de acertos.

Figura6 – Tela do Protótipo de Software *Canary Driver Alert*



Fonte: Elaborado pelo autor.

Quadro 4 – Síntese dos 12 experimentos

Número do experimento	Evento realizado	Identificação do protótipo	Data e hora do experimento
1	Frenagem agressiva	Frenagem agressiva	15-03-2020 09:40:39
2	Aceleração agressiva	Aceleração agressiva	15-03-2020 09:43:41
3	Aceleração agressiva	Aceleração agressiva	15-03-2020 09:40:27
4	Aceleração agressiva	Frenagem agressiva	15-03-2020 09:40:23
5	Frenagem agressiva	Frenagem agressiva	21-06-2020 17:21:33
6	Frenagem agressiva	Frenagem agressiva	21-06-2020 17:23:03
7	Aceleração agressiva	Aceleração agressiva	21-06-2020 17:21:39
8	Aceleração agressiva	Aceleração agressiva	21-06-2020 17:21:59
9	Aceleração agressiva	Aceleração agressiva	21-06-2020 17:22:59
10	Aceleração agressiva	Aceleração agressiva	21-06-2020 17:23:59
11	Aceleração agressiva	Aceleração agressiva	21-06-2020 17:24:47
12	Aceleração agressiva	Frenagem agressiva	21-06-2020 17:28:13

Fonte: Elaborado pelo autor.

4 Conclusão

O objetivo deste trabalho foi desenvolver um protótipo de *software* capaz de analisar a viagem de um motorista fazendo uso de técnicas de *machine learning* de dados de uma base pública para geração de um modelo preditivo. Para alcançar este objetivo foi realizada a revisão sistemática da literatura sobre o tema da análise comportamental da direção de motoristas e análise quantitativa da técnica mais apropriada para a classificação da viagem. Dos nove trabalhos selecionados, a utilização de sensores de *smartphone* em conjunto com o algoritmo DTW demonstraram eficiência e acurácia na capacidade de comparar séries temporais e na identificação de eventos de direção, por este motivo, foi a selecionada para ser utilizada neste trabalho.

O modelo preditivo utilizou a base de dados pública de Ferreira *et al.* (2017) com seus 69 exemplares de tipos de eventos de direção. Uma vez estabelecido o padrão de eventos pela base de treino, é possível calcular a similaridade de dois diferentes eventos por meio do algoritmo DTW. Com o DTW, o modelo foi gerado com acurácia de 84,21% e foi utilizado para o desenvolvimento do protótipo *Canary Driver Alert*. O algoritmo DTW empregado foi adaptado de Saiprasert, Pholprasit e Thajchayapong (2015) e baseado na aplicação de classificação de movimentos do videogame "Wii, de López (2017).

A arquitetura do sistema envolvendo o *Canary Driver Alert* consiste na captura de dados do sensor do *smartphone* por um aplicativo de terceiro, denominado Torque Pro, que transmite os dados da viagem para um servidor ao término da viagem. Realiza-se, manualmente, a cópia dos dados para o computador no qual está instalado o protótipo, efetua-se o pré-processamento, aplicação do modelo preditivo, identificação dos eventos e, ao final, é exibido o resumo dos tipos de eventos encontrados, quantidades e percentual em relação ao total de eventos da viagem do motorista.

Realizaram-se 12 experimentos com diferentes eventos de direção utilizando o *Canary Driver Alert*, dos quais 10 foram identificados corretamente. Eventos de curvas e mudanças de faixas agressivas não foram realizadas por motivo de segurança. Em terreno plano, o protótipo identifica adequadamente os eventos de direção, em particular a aceleração e frenagem agressivas. Em vias com inclinação, como subidas, o protótipo gera falsos positivos devido ao ângulo de inclinação e à interferência da força da gravidade aplicada no acelerômetro.

O modelo preditivo gerado neste trabalho com o DTW tem potencial para ser utilizado em um aplicativo capaz de auxiliar na segurança e qualidade da direção do motorista.

Alguns trabalhos futuros foram identificados: a) desenvolver mecanismo capaz de considerar a aceleração originada pela força da gravidade e corrigir os dados fornecidos de acordo com o sensor giroscópio; b) adaptar o modelo preditivo de modo que seja capaz de realizar a análise do evento de direção em tempo real; c) realizar testes em ambiente controlado com pista segura e própria para tal finalidade; d) desenvolver a captura de dados dos sensores sem depender de aplicativos de terceiros, uma vez que a biblioteca de comunicação com sensores está disponível; e) desenvolver um aplicativo fazendo uso do modelo preditivo obtido integrado ao

computador de bordo e plataforma multimídia do veículo; f) otimizar a performance do algoritmo DTW, seja pelo uso da técnica de banda de Sakoe e Chiba (1978) ou do DTW compilado em C; e g) avaliar outras técnicas para identificação e análise de eventos, como *Support Vector Machine* (SVM) e lógica *fuzzy* comparando com os resultados obtidos neste trabalho.

Referências

- CARVALHO, E. *et al.* Exploiting the use of recurrent neural networks for driver behavior profiling. *In: International Joint Conference On Neural Networks*, 2017, Anchorage. **Anais [...]**. Anchorage: IEEE, 2017. p. 3016-3021. DOI: <http://dx.doi.org/10.1109/ijcnn.2017.7966230>.
- CASTIGNANI, G. *et al.* Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring. **IEEE Intelligent Transportation Systems Magazine**, [s. l.], v. 7, n. 1, p. 91-102, 2015. DOI: <http://dx.doi.org/10.1109/mits.2014.2328673>.
- CASTIGNANI, G. *et al.* Smartphone-Based Adaptive Driving Maneuver Detection: A Large-Scale Evaluation Study. **IEEE Transactions On Intelligent Transportation Systems**, [s. l.], v. 18, n. 9, p. 2330-2339, set. 2017. DOI: <http://dx.doi.org/10.1109/tits.2016.2646760>.
- ENGELBRECHT, J. *et al.* Performance Comparison of Dynamic Time Warping (DTW) and a Maximum Likelihood (ML) Classifier in Measuring Driver Behavior with Smartphones. *In: IEEE Symposium Series On Computational Intelligence*, 2015, Cape Town. **Anais [...]**. Cape Town: IEEE, 2015. p. 427-433. DOI: <http://dx.doi.org/10.1109/ssci.2015.70>.
- EREN, H. *et al.* Estimating driving behavior by a smartphone. *In: IEEE Intelligent Vehicles Symposium*, 2012, Madrid. **Anais [...]**. Madrid: IEEE, 2012. p. 234-239. DOI: <http://dx.doi.org/10.1109/ivs.2012.6232298>.
- FAZEEN, M. *et al.* Safe Driving Using Mobile Phones. **IEEE Transactions On Intelligent Transportation Systems**, [s. l.], v. 13, n. 3, p. 1462-1468, set. 2012. DOI: <http://dx.doi.org/10.1109/tits.2012.2187640>.
- FELIZARDO, K. R. *et al.* **Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática**. Rio de Janeiro: Elsevier Editora LTDA, 2017.
- FERREIRA, J. *et al.* Driver behavior profiling: An investigation with different smartphone sensors and machine learning. **Plos One**, [s. l.], v. 12, n. 4, p. 1-16, 10 abr. 2017. DOI: <http://dx.doi.org/10.1371/journal.pone.0174959>.
- HONG, J.; MARGINES, B.; DEY, A. K. A smartphone-based sensing platform to model aggressive driving behaviors. *In: Annual ACM Conference On Human Factors In Computing Systems*, 32., 2014, New York. **Anais [...]**. New York: ACM Press, 2014. p. 4047-4056. DOI: <http://dx.doi.org/10.1145/2556288.2557321>.
- JOHNSON, D. A.; TRIVEDI, M. M. Driving style recognition using a smartphone as a sensor platform. *In: International IEEE Conference On Intelligent Transportation Systems*, 14., 2011, Washington. **Anais [...]**. Washington: IEEE, 2011. p. 1609-1615. DOI: <http://dx.doi.org/10.1109/itsc.2011.6083078>.
- LÓPEZ, Roque. Classifier using Dynamic Time Warping Distance. **GitHub**, [s. l.], 2017. Disponível em: https://github.com/roquelopez/classifier_using_dynamic_time_warping. Acesso em: 20 jan. 2020.
- POLI *et al.* Voice Command Recognition with Dynamic Time Warping (DTW) using Graphics Processing Units (GPU) with Compute Unified Device Architecture (CUDA). *In: International Symposium on Computer Architecture and High Performance Computing*, 19., 2007, Gramado. **Anais [...]**. Gramado: IEEE, 2007. DOI: <http://dx.doi.org/10.1109/SBAC-PAD.2007.21>.
- POLÍCIA RODOVIÁRIA FEDERAL (Brasil). **Acidentes por Ocorrência**. [Brasília, DF]: Polícia Rodoviária Federal, 2019. Disponível em: <https://portal.pr.f.gov.br/portal/dados-abertos/acidentes/acidentes>. Acesso em: 02 set. 2019.
- SAIPRASERT, C.; PHOLPRASIT, T.; THAJCHAYAPONG, S. Detection of Driving Events using Sensory Data on Smartphone. **International Journal Of Intelligent Transportation Systems Research**, [s. l.], v. 15, n. 1,

p. 17-28, 23 jul. 2015. DOI: <http://dx.doi.org/10.1007/s13177-015-0116-5>.

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions On Acoustics, Speech, And Signal Processing**, [s. l.], v. 26, n. 1, p. 43-49, 1978. DOI: <http://dx.doi.org/10.1109/tassp.1978.1163055>.

Sobre os autores

Eduardo Tamanaha

Graduado em Processamento de Dados pela Universidade Cruzeiro do Sul (1992) e Pós-Graduado MBA em Engenharia de *Software* pela Faculdade de Informática e Administração Paulista 9FIAP) e Mestre em Engenharia da Computação pelo Instituto de Pesquisas Tecnológicas de São Paulo (IPT). Atualmente trabalha na empresa Portoseg S/A Crédito e Financiamento e atua como analista de soluções. Possui 31 anos de experiência em desenvolvimento de sistemas atuando nos ramos de comércio, seguradoras, financeiras e cartão de crédito. Experiência em plataformas *mainframe* Unisys, *client-server* e WebSphere Application Server IBM. Linguagens Linc II, Cobol, Clipper, PowerBuilder, 4GL e Java. Bancos de dados Informix, Sybase e Oracle. Sistemas operacionais Unix, Linux e Windows.

Ivan Carlos Alcântara de Oliveira

Graduado em Ciência da Computação (USP-SC), Mestre em Engenharia da Computação (IPT-SP) e Doutor em Ciências (POLI-USP-SP). Estuda, pesquisa e trabalha com Computação desde 1998, quando iniciou o mestrado, direcionado para ambientes colaborativos (CSCL) e, no doutorado, entre 2008 e 2013, voltado para Sistemas de Aprendizagem Adaptativos e Inteligentes. Atuou como professor nas instituições Faculdade São Francisco da Paraíba (FASP), Universidade Cruzeiro do Sul (UNICSUL), Serviço Nacional de Aprendizagem Comercial (SENAC), Universidade de São Paulo (USP), Universidade Municipal de São Caetano do Sul (USCS) e Centro Universitário de Adamantina (UniFAI) em variadas disciplinas dos cursos de Computação, além de ter sido docente colaborador no Mestrado Profissional em Engenharia de Computação do IPT/SP, no qual atuou na função de orientador de 2015 a 2020, orientando diversos trabalhos. Atualmente, é Professor Assistente Doutor I da Universidade Presbiteriana Mackenzie (UPM) atuando nos cursos de Computação e Informática da Faculdade de Computação e Informática (FCI). Tem interesse de pesquisa nas áreas Informática na Educação, Recursos Educacionais Digitais, Teoria dos Grafos, Inteligência Artificial, Ciência de Dados, Aprendizado de Máquina, Processamento de Linguagem Natural e *Deep-Learning*.

Recebido em: 27.07.2020

Avaliado em: 14.06.2022

Aceito em: 13.09.2022