UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO - CECAU

CAIO GOMES GONÇALVES

APLICAÇÃO DE TÉCNICAS DE DETECÇÃO DE ANOMALIAS PARA A VERIFICAÇÃO DA IDENTIDADE DE MOTORISTAS

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2021

CAIO GOMES GONÇALVES

APLICAÇÃO DE TÉCNICAS DE DETECÇÃO DE ANOMALIAS PARA A VERIFICAÇÃO DA IDENTIDADE DE MOTORISTAS

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Rodrigo César Pedrosa Silva, Ph.D.

Coorientador: Prof. Luciana Gomes Castanheira, Dra.

Ouro Preto
Escola de Minas – UFOP
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

G635a Gonçalves, Caio Gomes.
Aplicação de técnicas de detecção de anomalias para a verificação da identidade de motoristas. [manuscrito] / Caio Gomes Gonçalves. - 2021.
46 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Rodrigo César Pedrosa Silva.
Coorientadora: Profa. Dra. Luciana Gomes Castanheira.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia de Controle e Automação .

1. Aprendizado de máquina. 2. Motoristas - Comportamento. 3. Fraude - Detecção. I. Castanheira, Luciana Gomes. II. Silva, Rodrigo César Pedrosa. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:004.8

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Caio Gomes Gonçalves

Aplicação de Técnicas de Detecção de Anomalias para a Verificação da Identidade de Motoristas

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação.

Aprovada em 25 de agosto de 2021.

Membros da banca

DSc. - Rodrigo César Pedrosa Silva - Orientador (UFOP)

DSc. - Luciana Gomes Castanheira - Coorientadora (UFOP)

MSc. - Sofia Maria Amorim Falco Rodrigues - (CEFET-MG)

DSc. - Tamires Martins Rezende - (FITec)

Rodrigo César Pedrosa Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 01/09/2021.

Documento assinado eletronicamente por **Luciana Gomes Castanheira**, **COORDENADOR(A) DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMACAO**, em 01/09/2021, às 22:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0204209** e o código CRC **BC5C88AB**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: 3135591533 - www.ufop.br

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

Agradeço primeiramente a minha família por possibilitarem e me incentivarem a seguirem este caminho principalmente nos momentos difíceis. Ao professor Rodrigo Pedrosa que me acompanhou desde a metade da graduação me apresentando com afinco como aproveitar ao máximo as oportunidades oferecidas pela universidade, sendo além de orientador e um jovem dinâmico, um grande amigo. Agradeço a Equipe 42 e principalmente ao quarteto dos Pinguins de Madagascar que é formado por Antonio de Barros, Alan Santandrea, Hércules Corcini e eu. Sem estas pessoas a caminhada seria mais árdua e a chegada não seria tão satisfatória.

“Matéria é a parte acidental.” (Oliver Lodge)

RESUMO

Automóveis estão equipados com uma série de sensores que permitem o monitoramento tanto do funcionamento do carro quanto das ações executadas pelo condutor. Através da extração de padrões destes dados é possível identificar o motorista do automóvel. A vasta maioria dos trabalhos relacionados tratam a identificação de motoristas como um problema de classificação. Isso significa que num cenário de furto a aplicabilidade destes métodos é bastante limitada uma vez que o criminoso provavelmente não estará na base de dados.

Assim, neste projeto é proposto o estudo de técnicas de detecção de anomalia para este problema, analisando alguns modelos de detecção já existentes na literatura como *Elliptic Envelopes*, *Isolation Forest* e *One Class Support Vector Machine*. Na detecção de anomalia, o modelo aprende somente o que é o comportamento de direção do motorista designado para um veículo. Comportamentos que se diferenciam significativamente deste são considerados anômalos e denunciados como evidência de furto. Este tipo de método já tem sido utilizado com sucesso na detecção de fraudes em cartões de crédito e se apresenta como alternativa promissora na identificação de motoristas.

Palavras-chaves: detecção de anomalias; aprendizado de máquina; comportamento de motoristas; detecção de fraude

ABSTRACT

Cars are equipped with a series of sensors that allow the monitoring of both the car's functioning and the actions performed by the driver. By extracting patterns from these data, it is possible to identify the driver of the car. The vast majority of related works treat driver identification as a classification problem. This means that in a theft scenario the applicability of these methods is quite limited as the criminal will likely not be in the database.

Thus, in this project we propose the study of anomaly detection techniques for this problem, analyzing some existing detection models in the literature, such as Elliptic Envelope, Isolation Forest and One Class Support Vector Machine. In anomaly detection the model only learns what the designated driver's behavior for a vehicle is. Behaviors that differ significantly from this are considered anomalous and reported as evidence of theft. This type of method has already been used successfully in the detection of credit card fraud and presents itself as a promising alternative for identifying drivers.

Key-words: Machine Learning, Anomaly Detection, Fraud Detection, Driver Behavior.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Desde a criação dos automóveis, tecnologias são desenvolvidas visando o conforto e segurança de seus usuários. Diversos componentes mecânicos estão sendo substituídos por componentes elétricos com intuito de melhorar a forma de dirigir, trazendo segurança, reduzindo desgaste de peças mecânicas e melhorando o gerenciamento do carro como um todo.

Graças aos avanços tecnológicos, automóveis modernos estão, hoje, equipados com uma série de sensores que permitem o monitoramento tanto do funcionamento do carro quanto das ações executadas pelo condutor. Em carros com direção elétrica, é possível saber o ângulo do volante a cada instante. Em carros com frenagem eletrônica, existem sensores capazes de medir o acionamento dos freios, assim como a pressão exercida pelo condutor nos pedais. Além destes, ainda é possível ler informações como velocidade, injeção de combustível, posição do pedal do acelerador, consumo de combustível, marcha atual, entre outros.

Todos estes dados podem ser obtidos através da rede CAN (*Controller Area Network*) (LI; LIU; LUO, 2008; WIKIPEDIA, 2019 - Acessado em: 08-11-2019) de cada carro. Esta rede utiliza o protocolo OBD-II (*On-board Diagnostics*) (WIKIPEDIA, 2019) criado pela General Motors® justamente com intuito de disponibilizar informações importantes do veículo de maneira fácil e normatizada.

No Brasil uma grande preocupação é quanto a segurança dos automóveis. Em 2013, a Secretaria Nacional de Segurança Pública do Ministério da Justiça (Senasp/MJ) lançou o aplicativo Checkplaca® (JUSTICA.GOV, 2013) cuja única função é verificar se uma placa pertence a um veículo roubado. Anos depois são apresentados dados de (G1.GLOBO.COM, 2016) onde mostram que de janeiro a abril de 2016, mais de 27 mil carros foram roubados somente na cidade de São Paulo.

Como mostram os dados informados, o furto de automóveis afeta uma grande parte da população e não só no Brasil mas no mundo todo. Neste contexto se faz extremamente importante criar tecnologias capazes de identificar o mais rápido possível o condutor do automóvel. Assim, no caso de uma situação de furto o proprietário e as autoridades policiais podem ser acionados rapidamente limitando as ações dos infratores.

Os sistemas de segurança estão sempre sendo aprimorados, utilizando novos tipos de chaves, alarmes e até sistemas biométricos. Apesar destas técnicas, existe uma corrida armamentista entre sistemas de segurança e criminosos. Assim, o seu desenvolvimento é uma tarefa contínua e infindável.

É possível encontrar trabalhos na literatura que tem como propósito identificar o motorista que está conduzindo o veículo. Esta identificação pode ser feita de várias formas, como por exemplo utilizando um sensor biométrico como em (SADHUKHAN; ACHARYYA; PRA-

SAD, 2017), leitor de íris (PUNNOOSE; KUMAR, 2015) e até mesmo reconhecimento facial (RAHMAT et al., 2019). Para que estas tecnologias citadas sejam implementadas é necessária a instalação de sensores externos, por exemplo, para o reconhecimento facial, a instalação de uma câmera dentro do veículo.

Existem situações em que não é interessante alterar o veículo pois tais alterações podem demandar tempo, gastos extras e mão de obra especializada. Dessa forma, surgiram ideias de identificar os condutores de forma indireta, utilizando os sensores já embarcados nos na maioria dos veículos, partindo do princípio que cada condutor possui um padrão único de direção.

Na literatura é possível separarmos os trabalhos em duas classes, trabalhos que objetivam identificar qual o condutor está dirigindo o veículo e trabalhos que objetivam classificar o comportamento dos condutores em situações variadas durante a direção.

Os trabalhos citados a seguir estão relacionados a classificação não apenas da identidade do condutor mas também quanto aos tipos de comportamentos como por exemplo agressivo, desatendo, entre outros.

TOLEDO; MUSICANT; LOTAN em 2008 descrevem o potencial de um sistema gravador de dados *In-Vehicle Data Recorder* (IVDR) no veículo. Este sistema grava a localização, velocidade e aceleração do veículo utilizando GPS e acelerômetros e até mesmo câmeras de vídeo mostrando o interior e exterior do veículo além de sensores de radar e rastreador de faixa. Este sistema possibilita identificar vários tipos de manobras, principalmente manobras que podem resultar em acidentes e então utilizam estas informações para gerar índices de risco que indicam o nível de segurança na viagem. Estes indicadores gerados, posteriormente podem ser utilizados para classificar os motoristas quanto ao nível de segurança ao conduzir um veículo. Apesar do sistema proposto necessitar de instalações extras no veículo, o trabalho apresenta métricas de risco mensuráveis e bem definidas parecendo estar altamente correlacionados ao eventos de acidentes.

KEDAR-DONGARKAR; DAS 2012 propuseram utilizar a velocidade do veículo, acelerações, torque, ângulo do volante, pressão exercida no pedal de acelerador e freio. Além do objetivo de classificar os condutores, os autores pretendiam desenvolver um classificador que não precisasse de grande processamento para que ele fosse facilmente embarcado em um veículo. O método desenvolvido era capaz de classificar os motoristas em três classes diferentes: agressivo, moderado e conservador. Conseguiram obter uma acurácia geral de 77% sendo que este método era computacionalmente mais eficiente se comparado a HMM (*Hidden Markov Models*) (RABINER; JUANG, 1986) e ao KNN (*k-nearest neighbors*) (GUO et al., 2003) pois não precisava realizar cálculos de distâncias euclidianas em algumas circunstâncias.

CHOI et al., em 2007, tiveram como objetivo analisar e classificar o comportamento do condutor em cenários reais e diferenciar principalmente três situações: classificação da ação, detecção de distração e identificação do motorista. A base de dados utilizada foi construída

utilizando o UTDrive (ANGKITITRAKUL et al., 2007) e CAN-Bus (LI; LIU; LUO, 2008) para a coleta dos seguintes dados: ângulo do volante, posição do pedal do acelerador e do pedal de freio. Para a avaliação das características comportamentais e encontrar a correlação entre os comportamentos de condução utilizaram o MATLAB e na parte de classificação dois modelos, HMM e *Gaussian Mixture Model - GMM* (RASMUSSEN et al., 1999). Como resultado obtiveram 69% de acurácia para a classificação de ações e 25% de acurácia para a identificação do motorista e 35% quanto a distração do condutor. Este trabalho detalha bem quais dados foram escolhidos para serem analisados porém não obtém uma acurácia que seja suficiente para garantir um nível de segurança e assertividade em aplicações reais.

Os trabalhos citados a seguir são relacionados somente ao problema de identificar qual condutor que está dirigindo, ou seja, os próprios condutores são as classes de um problema de classificação.

Em 2006, WAKITA et al. propuseram a identificação de motoristas de acordo com seus padrões de direção enquanto estão seguindo outro veículo. O objetivo era, dada uma base de dados, identificar qual dos motoristas estava conduzindo o veículo. Os autores utilizaram uma parte de dados provenientes de um simulador de direção e outra parte de dados reais; dados esses como o uso do pedal do acelerador, pedal de freio, velocidade do veículo e distância do veículo à frente. Para os modelos de identificação, utilizaram o *Helly Model* (HELLY, 1959), *Optimal Velocity Model* (BANDO et al., 1995) e *GMM*. A partir dos experimentos os autores concluíram que o GMM identificou o motorista que estava conduzindo com uma melhor acurácia se comparado aos outros dois modelos, obtendo taxas de identificação de 81% para 12 motoristas no simulador de direção e 73% para 30 motoristas utilizando veículos reais.

MENG; LEE; XU, 2006 definiram como objetivo apresentar um sistema inteligente capaz de lidar com o problema de roubos de veículos. Este sistema consiste em analisar dados e associá-los a um condutor presente na base de dados. Além de utilizar dados reais, o sistema experimental foi testado também utilizando um simulador de direção. A base de dados continha informações do pedal do acelerador, freio e ângulo do volante. O método de aprendizado de máquina utilizado foi o *HMM* (RABINER; JUANG, 1986). Os autores concluíram que o método proposto é válido e útil contra o problema de furto de veículos já que o mesmo obteve uma taxa de sucesso em torno de 80%.

WAKITA et al., 2006, continuaram desenvolvendo o projeto e em 2007 publicaram MIYAJIMA et al.. O novo trabalho permanecia utilizando GMM como modelo de classificação e os mesmos tipos de dados coletados, porém dessa vez os dados foram submetidos a uma análise espectral o que resultou em uma redução relativa de erro de 55% em comparação a utilização de dados brutos.

Em 2009, WAHAB et al. realizaram a modelagem de comportamentos individuais durante a direção e a partir dos modelos GMM e na Transformada Wavelet (ARNEODO; GRASSEAU; HOLSCHNEIDER, 1988) identificaram quais dados são eficazes e necessários para distinguir

cada motorista. Estes dados são obtidos através dos sensores que medem a pressão do pedal do acelerador e do pedal de freio. Para determinar a identidade dos 30 motoristas foram usadas duas redes neurais difusas, a Rede Neural Difusa em Evolução (EFuNN) (MIIKKULAINEN et al., 2019) e o Sistema de Inferência Neuro Fuzzy Adaptável (ANFIS) (JANG, 1993). Estes sistemas foram comparados com uma implementação baseada em Rede Neural Artificial utilizando a Rede Perceptron Multicamadas MLP (PAL; MITRA, 1992) e o método estatístico GMSS (WILLIS; AUSTELL, 1983) baseado no GMM. A combinação dos sensores que apresentou as melhores acurácias foi utilizando a pressão no pedal do acelerador, do pedal do freio e a dinâmica de ambos os pedais. Os métodos ANFIS e GMSS foram os que conseguiram a melhor acurácia para identificar os motoristas, cerca de 94%. A EFuNN obteve algo próximo a 89% e em último lugar a MLP com 79%. Os autores ficaram satisfeitos com os resultados obtidos pelo ANFIS e GMSS.

Em 2016, ENEV et al. escolheram técnicas diferentes das já referenciadas neste trabalho. Realizaram dois experimentos com 15 condutores, no primeiro os condutores executaram uma série de manobras em um estacionamento isolado e no segundo dirigiram o veículo ao longo de uma volta definida em uma região metropolitana de Seattle. Os dados foram coletados da rede CAN do veículo e processados pelos seguintes métodos: SVM (*Support Vector Machine*), *Random Forest*, *Naive Bayes* e KNN. Obtiveram 87% de acurácia na distinção entre os 15 motoristas usando apenas dados da posição do pedal de freio e 99% de acurácia utilizando os 5 sensores que neste mesmo trabalho constataram serem os que mais caracterizam cada condutor.

A pesquisa desenvolvida por KWAK; WOO; KIM é uma das mais completas. Além dos sensores comumente utilizados nas pesquisadas já referenciadas aqui, os autores utilizaram outros sensores para testes e concluíram que dados como a temperatura do óleo de transmissão foram muito importantes na classificação de qual condutor estava dirigindo. Para 10 motoristas, foram coletados dados de 51 sensores do veículo na frequência de $1Hz$ resultando em um banco de dados com 94.401 linhas por 51 colunas. Estes dados foram coletados nas seguintes situações: estacionamento, condução em perímetro urbano e rodoviário. Como metodologia utilizaram *Decision Tree* (BREIMAN et al., 1984), *Random Forest* (BREIMAN, 2001), KNN e MLP (*Multilay Perceptron*). Realizaram vários experimentos alternando os dados e métodos utilizados, alcançaram acurácias entre 97% á até praticamente 100% na classificação de qual condutor estava dirigindo.

Em MARTINELLI et al. 2018a o objetivo também é de identificar o condutor que está dirigindo. Os autores utilizaram a base de dados descrita em KWAK; WOO; KIM 2016 combinada com os seguintes algoritmos de classificação: J48, J48graft, J48consolidated, RandomTree e RepTree (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Por fim obtiveram uma precisão média na classificação dos motoristas de 99,8%.

Todos os trabalhos e estudos relacionados a essa área tratam a identificação do motorista como um problema de classificação (MARTINELLI et al., 2018a; CHOI et al., 2007; WAKITA et al., 2006; KWAK; WOO; KIM, 2016). Isso dificulta a etapa de verificação de identidade pois

é improvável que um ladrão esteja na base de dados. Além disso, seria necessário que vários motoristas compartilhassem seus dados entre si ou com algum servidor criando problemas de privacidade.

Dentro do campo da Inteligência Artificial, existem técnicas de detecção de anomalias (ALLA; ADARI, 2019), como *Isolation Forest* em (LIU; TING; ZHOU, 2009) e *One Class Support Vector Machine* em (SCHÖLKOPF et al., 2000), que parecem ser mais adequadas para este tipo de problema. De forma geral, a detecção de anomalia consiste em encontrar objetos diferentes da maioria. A ideia é que objetos anômalos tem valores de atributos que se desviam significativamente dos valores esperados ou típicos. Assim, uma vez que identificamos o que é típico, podemos determinar o que é anômalo.

O mesmo raciocínio pode ser aplicado ao problema de verificação de motoristas. O proprietário ou condutor principal de um automóvel estabelece o que é o comportamento típico de direção. Qualquer anomalia observada pode então ser um sinal de furto. Desta forma, o algoritmo só precisa estabelecer como o condutor principal se comporta na direção sem a necessidade de uma base de dados com milhares de motoristas.

Apesar das vantagens citadas e do desempenho do método em outras aplicações (MARTÍ et al., 2015), (BUDALAKOTI; SRIVASTAVA; OTEY, 2009), (AKHILOMEN, 2013), (SCHMIDT et al., 2009), parece não existir na literatura um estudo profundo de análise destes métodos para a verificação de motoristas. Assim, a relevância deste trabalho se manifesta no preenchimento desta lacuna que irá contribuir para a validação de métodos mais apropriados para este problema que aflige tantos habitantes no Brasil e mundo.

1.1 Objetivos

O objetivo principal deste projeto é estudar a viabilidade de técnicas de detecção de anomalia para o problema de verificação de motoristas.

1.1.1 Objetivos específicos

- Estudar formas de representar os dados para o problema de detecção de anomalias;
- Estudar técnicas de detecção de anomalias;
- Implementar e adaptar modelos de detecção de anomalia para a verificação de motoristas.

2 VERIFICAÇÃO DE MOTORISTAS

2.1 Definição do problema

Neste trabalho avaliamos técnicas de detecção de anomalias para resolver o seguinte problema:

Dadas as leituras, r_i, de m sensores, i, do carro em uma janela de tempo w de n segundos começando no tempo t, i.e, $w_t = [r_{1_t}, r_{...t}, r_{m_t}, r_{1_{t+1}}, r_{...t+1}, r_{m_{t+1}}, ..., r_{1_{t+n-1}}, r_{...t+n-1}, r_{m_{t+n-1}}]$, determinar se os dados em w_t foram gerado pelo motorista designado, md, ou não. Para tal, o método verifica se w_t é uma anomalia em relação aos dados históricos gerados por md.

Anomalias, também chamadas de *outliers*, podem ser definidas como itens, eventos, observações ou padrões que não se enquadram em um determinando grupo ou fogem de um comportamento esperado (ALLA; ADARI, 2019; ZIMEK; SCHUBERT, 2017). Existem alguns tipos de anomalias e esses devem ser identificados para definir qual o método de detecção utilizar. As anomalias podem ser classificadas como Pontuais, Contextuais e Coletivas como mostra em (ALLA; ADARI, 2019):

- **Pontuais** se um tipo de dado individual pode ser considerado como anormal relativamente ao resto dos dados. Este tipo de anomalia é a mais simples e é o foco da maioria das pesquisas sobre detecção de anomalias.
- **Contextuais** se uma instância de dados é considerada anômala em um contexto e em outro contexto é considerada normal.
- **Coletiva** se uma coleção de instâncias de dados relacionados é anômala no que diz respeito a todo o conjunto de dados mas esses dados, individualmente, não são anômalos.

Neste trabalho consideraremos as anomalias contextuais. A ideia é que cada janela de dados, w_t, representa uma manobra do motorista. Assim, uma vez que w_t pertence a um tipo de manobras M, verifica-se se ela é anomalia apenas em relação apenas à manobras do tipo M. Assim, M é o contexto no qual a manobra definida pelos dados de w podem ser classificadas como anomalia ou não. Mais detalhes sobre a definição de manobra são dados na Seção 3.1.3.

A seguir serão definidos os modelos de detecção de anomalias utilizados no desenvolvimento desse trabalho, todos os modelos utilizados foram obtidos através da Scikit-Learn (PEDREGOSA et al., 2011).

2.2 Métodos de detecção de anomalias

Os métodos de detecção de anomalias são modelos de aprendizado não supervisionado que aprendem o que é normal através dos dados de treinamento. A partir disto, quaisquer outros dados que não sigam o padrão determinado por cada método é identificado como uma anomalia.

Neste trabalho foram estudados e testados três métodos bastante difundidos no campo de detecção de anomalias, *Elliptic Envelope*, *Isolation Forest* e a *One Class Support Vector Machine*. Esses métodos foram escolhidos por serem considerados simples soluções para o problema proposto e estarem prontamente disponíveis em bibliotecas de software públicas como a *Scikit-Learn* (PEDREGOSA et al., 2011).

2.2.1 Elliptic Envelope

O *Elliptic Envelope*, traduzido para Envelope Elíptico, é um método de detecção de anomalias proposto por ROUSSEEUW; DRIESSEN 1999. Ele utiliza a aproximação de uma distribuição gaussiana dos dados para inferir quais são as anomalias no conjunto.

O algoritmo tem um funcionamento simples, ele cria uma região elíptica estimando os parâmetros da distribuição gaussiana assim gerando a forma do envelope em torno de um determinado conjunto de dados normais.

Os parâmtros são estimados com o método *Minimum Covariance Determinant* - MCD. O MCD foi introduzido em 1984 por ROUSSEEUW e é um método resistente a observações distantes. Esta característica faz com que seja capaz de estimar o ponto médio e a matriz de covariância de forma a minimizar a influência de pontos distantes, o tornando muito útil para detecção de anomalias. Apesar de ter sido apresentado em 1984, seu uso se difundiu após o desenvolvimento do algoritmo *FastMCD* em 1999 por ROUSSEEUW; DRIESSEN já que este é computacionalmente mais eficiente que o MCD comum.

A Figura 1 mostra uma função de densidade de probabilidade gaussiana em 1 dimensão, onde o eixo x representa os possíveis valores e o eixo y a probabilidade relativa de um determinado valor para uma amostra aleatória dos dados.

Já a Figura 2 exibe a distribuição em 2 dimensões e mostra com mais clareza como o envelope é definido e os dados são rotulados.

O envelope é definido da forma que a região com maior volume de dados tenha uma maior elevação, formando o topo da montanha, e a medida que os dados começam a ter uma distribuição mais esparsa é formada a base desta montanha.

Na Figura 2 é possível notar três círculos na base, sendo que estes podem representar níveis de *contamination*. A *contamination* é um parâmetro do modelo que representa a probabilidade relativa a partir da qual uma amostra será considerada anomalia. Ou seja, uma vez que a função de densidade de probabilidade foi estimada, se a probabilidade relativa do valor amostra

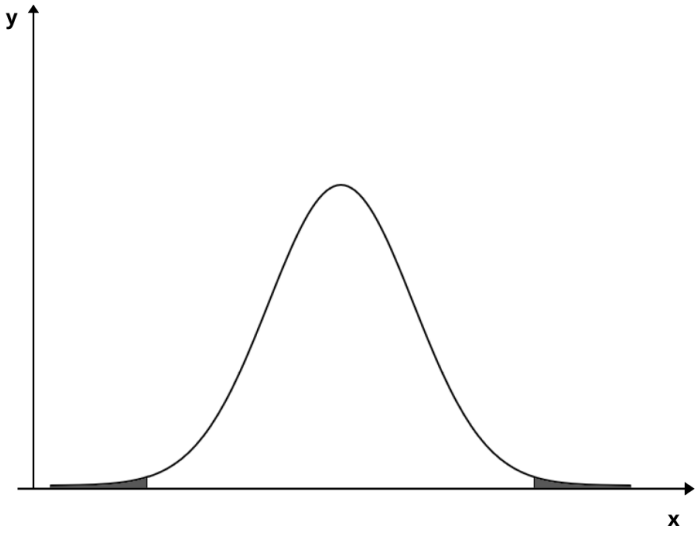

Figura 1 – Distribuição Gaussiana em 1D

em relação àquela função for menor do que *contamination*, ela será classificada como anomalia.

Este parâmetro é crucial para que o algoritmo consiga definir quais dados são anômalos ou normais. Tomando como exemplo que a *contamination* demarca o círculo verde ainda na Figura 2, todos os pontos externos a ele serão classificadas como anomalias e os pontos internos como normais.

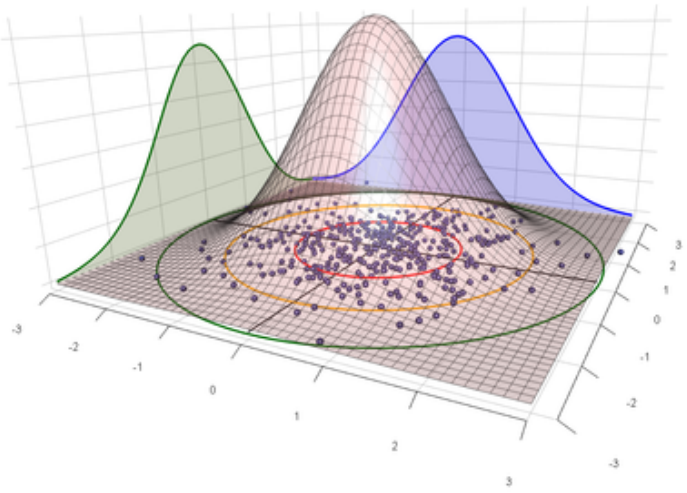

Figura 2 – Distribuição Gaussiana em 3D
Fonte - https://github.com/ashiq24/Copula

Para a aplicação do nosso problema o método será utilizado da seguinte forma:

1. O modelo será treinado apenas com os dados do motorista principal ajustando o parâmetro de *contamination*. Pelo fato do treinamento ser feito com um conjunto que não contém dados anômalos, este valor deve ser o menor possível em uma escala de 0 a 0.5.

2. Após o modelo ser ajustado, ele receberá os novos dados e então realizará a classificação dos mesmos.

2.2.2 Isolation Forest

Isolation Forest, traduzido para Floresta de Isolamento, é uma coleção de árvores de decisão individuais que particionam recursivamente um conjunto de dados (ALLA; ADARI, 2019). Este método foi desenvolvido a partir de outros dois métodos, a Árvore de Decisão (*Decision Tree*) e Floresta Aleatória (*Random Forest*) que serão explicados a seguir.

Por volta de 1984, a árvore de decisão foi desenvolvida por BREIMAN et al. e é um dispositivo de ajuda à tomada de decisão que utiliza um diagrama moldado em de árvore para apresentar visualmente as condições e as possibilidades para se chegar a um resultado ou classificação. A Figura 3 exemplifica a estrutura de uma árvore de decisão.

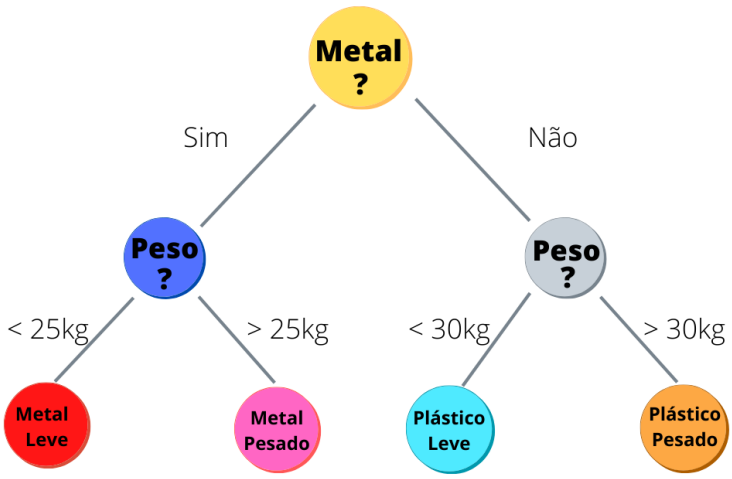

Figura 3 – Exemplo de uma árvore de decisão que classifica um material quanto ao seu tipo e peso.

Seguindo a linha de desenvolvimento, por volta de 2001, BREIMAN publicou o algoritmo *Random Forest*. Este algoritmo é um tipo de *ensemble learning*, ou seja, método que gera muitos classificadores e combina o seu resultado.

Neste caso, são geradas diversas árvores de decisão, sendo cada uma com suas particularidades e o resultado da classificação de todas elas é combinado. Esta combinação de modelos, torna o algoritmo muito mais poderoso que uma simples *Decision Tree*.

Em 2008, LIU; TING; ZHOU publicaram então a *Isolation Forest*. Um método de detecção de anomalias que, embasado nos dois métodos já descritos, tem como objetivo determinar limites de decisão afim de detectar anomalias em um conjunto de dados.

Na floresta de isolamento, as árvores são construídas de forma aleatória. O particionamento dos dados é repetido recursivamente, ou seja, em cada iteração do processo, um atributo é selecionado de forma aleatória e os dados são divididos com base em um valor escolhido aleatoriamente entre o valor mínimo e máximo daquele atributo. Esse processo é repetido até que todo o conjunto de dados seja particionado para formar uma árvore individual na floresta (ALLA; ADARI, 2019).

Para demonstrar a ideia de que as anomalias são mais suscetíveis ao isolamento sob particionamento aleatório é ilustrado, nas Figuras 4 (a) e (b) respectivamente, o particionamento de um ponto normal e um ponto anômalo.

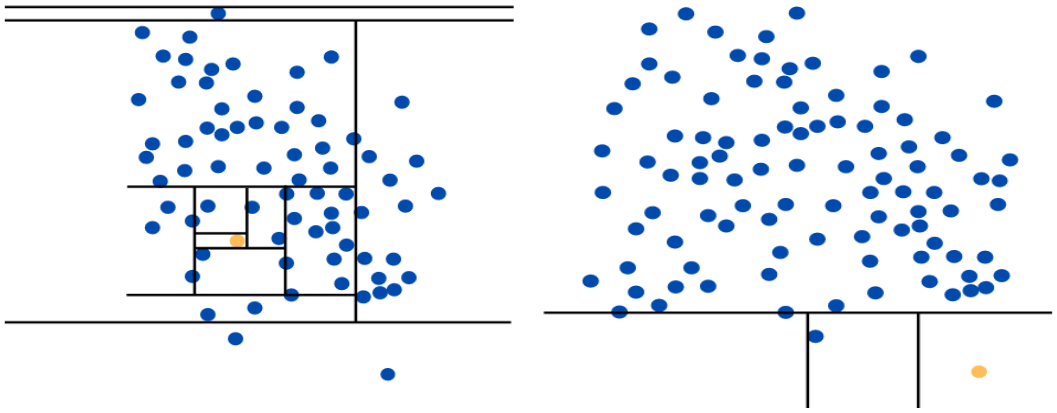

Figura 4 – (a) Mostra a dificuldade em isolar um ponto considerado normal.
(b) Mostra a facilidade em isolar um ponto anômalo.
Adaptado de (LIU; TING; ZHOU, 2009).

O particionamento é repetido recursivamente até que todas as instâncias sejam isoladas ou a altura máxima da árvore seja atingida.

Na Figura 5, os dois círculos em vermelho são dados anômalos e os círculos em rosa são dados normais.

O comprimento dos caminhos dos dados anômalos é claramente menor que o comprimento dos dados normais. Como cada partição é gerada de forma aleatória, as árvores individuais são geradas com diferentes conjuntos de atributos.

Para que uma amostra seja classificada como anomalia ou normal é necessária uma pontuação para essa tomada de decisão.

Após gerar as árvores e calcular o comprimento de cada caminho, é feito o cálculo do comprimento médio dos caminhos sendo esse o responsável por classificar um dado como

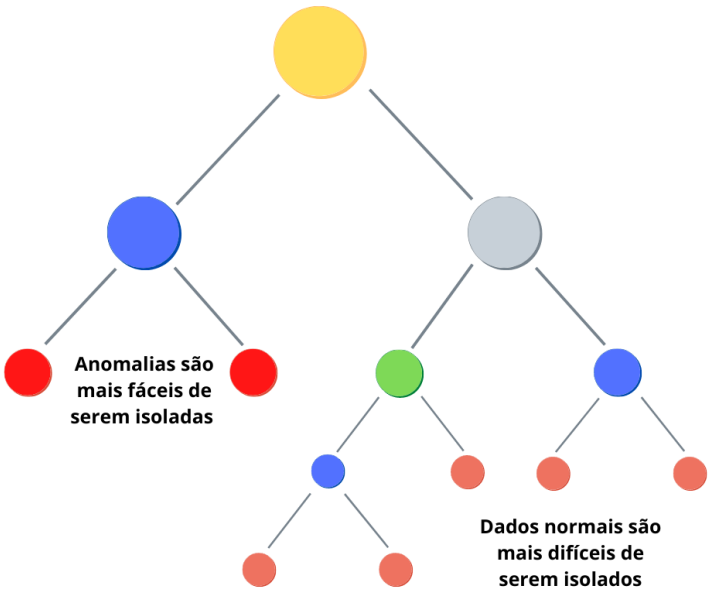

Figura 5 – *Isolation Forest* exemplificado com uma árvore de decisão.

anômalo ou normal, já que uma anomalia possui comprimento de caminho menor (LIU; TING; ZHOU, 2009). A equação 2.1 define um *score* de anomalia.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2.1}$$

Sendo que $E(h(x))$ é a média do comprimento do caminho da observação x, $c(n)$ é comprimento médio de um caminho que vai da raiz até algum nó externo e n o número de nós externos.

Caso, $s(x, n)$, seja próximo a 1 o modelo classifica a amostra como anomalia, muito menor que 0,5 como uma amostra normal e caso todas as pontuações forem próximas de 0,5 indica que não existem anomalias claramente distintas.

O algoritmo encontra-se disponível na biblioteca da *Scikit-Learn* (PEDREGOSA et al., 2011). Dentre os parâmetros disponíveis para ajuste, apenas o número de estimadores foi alterado, sendo este responsável pelo número de árvores de isolamento que serão utilizadas na construção da floresta.

É possível notar como a *Isolation Forest* utiliza uma metodologia de certa forma simples para funcionar. O método a seguir é um pouco mais complexo mas que também é muito utilizado na detecção de anomalias.

2.2.3 One Class Support Vector Machine

A *One Class Support Vector Machine* (OCSVM) é um método de detecção de anomalias adaptado da *Support Vector Machine*.

Por volta de 1992, BOSER; GUYON; VAPNIK propuseram um algoritmo supervisionado para classificação que evoluiu e atualmente é conhecido como *Support Vector Machine*.

Estas Máquinas Vetoriais de Suporte, tradução de *Support Vector Machine* (SVM), são um tipo de modelo de aprendizagem utilizado para classificação e regressão. Ela tem como principal objetivo determinar limites de decisão que produzam uma separação ótima entre as classes.

A SVM é gerada a partir dos dados de entrada durante o treinamento. Ela realiza um mapeamento dos dados em um espaço multidimensional e utiliza regressão para encontrar um hiperplano que possibilita separar o conjunto de dados em duas classes (PRESS et al., 2007).

Em um conjunto de dados, no momento da separação das classes podem existir diferentes possibilidades quanto ao posicionamento do hiperplano, portanto são necessárias métricas para definir o local ótimo.

A Figura 6 mostra os parâmetros necessários para que o local do hiperplano seja escolhido. Os pontos de cada classe mais próximos ao hiperplano são denominados Vetores de Suporte e também podem ser vistos nesta figura.

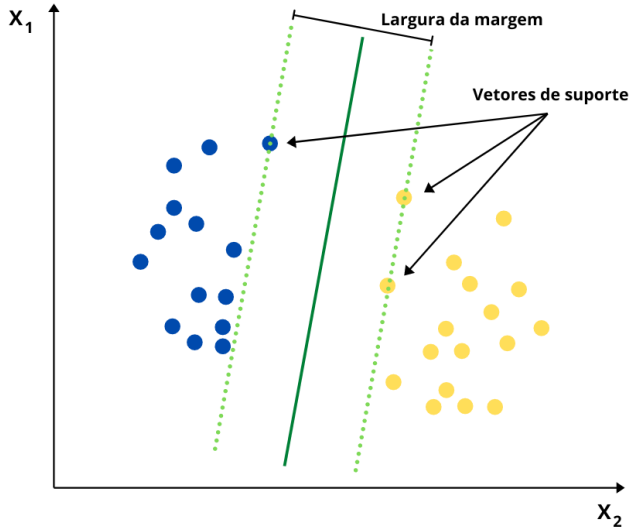

Figura 6 – Máquina de vetores de suporte bidimensional.

Existem infinitas combinações de margens possíveis mas apenas uma combinação é ótima para o problema. A combinação ótima, consiste em ter a maior largura de margem possível sem que existam pontos internos.

Para a determinação do limite de decisão, ou seja, o posicionamento do hiperplano, é

utilizada a função de otimização 2.2.

$$
\begin{aligned}
min_{\omega,b\ ,\xi\ i} \frac{||\omega||^2}{2} + C\sum_{i=1}^{n}\xi_i \\
\\
sujeito : y_i(\omega^T\phi(x_i) + b) \geq 1 - \xi_i \\
\xi_i \geq 0, \\
i = 1, ..., n
\end{aligned}
\tag{2.2}
$$

A Equação 2.2 pode ser entendida facilmente se dividida nos tópicos listados a seguir:

- Minimizando o primeiro termo, $\frac{||w||^2}{2}$, a margem é maximizada;
- O segundo termo, $C\sum_{i=1}^{n}\xi_i$, define o quanto de erro é aceitável dentro modelo (previne sobreajuste no modelo), sendo que o ξ_i é a quantidade de folga para cada ponto e C a prevenção quanto o modelo errar;
- O hiperplano é definido por $(\omega^T\phi(x_i) + b)$;
- A restrição, produto do hiperplano pelo ponto y_i, determina como o modelo deve acertar, sendo a classe negativa do lado negativo do hiperplano e a positiva do lado positivo do hiperplano;
- A variável n representa o número de amostras;
- A variável b representa o viés do modelo.

Logo, é possível concluir que a SVM é um método capaz de construir separadores de margem máxima.

Como mencionado em ALLA; ADARI 2019 a One-Class SVM é um modelo de Máquina Vetorial de Suporte modificado que é adequado para detecção de anomalias. Nesse caso o objetivo é que o modelo seja treinado com dados normais e que detecte anomalias quando novos dados forem apresentados a ele.

A SVM e OCSVM se diferem de acordo com a rotulação dos dados e aplicação do modelo. Na SVM todos os dados do conjunto são rotulados sendo um modelo voltado para classificação. Já a OCSVM é uma SVM adaptada para a detecção de anomalias, essa adaptação faz com que o modelo seja treinado apenas com os dados normais.

A OCSVM foi introduzida por SCHÖLKOPF et al. 2000 adaptando a metodologia da SVM para o problema de classificação de uma classe.

Esse problema será no fundo tratado como um problema de duas classes. Como visto em AMER; GOLDSTEIN; ABDENNADHER 2013 o algoritmo irá através de uma função de

transformação implícita $\phi(.)$, que é definida pelo *kernel*, projetar as amostras em um espaço dimensional superior. Neste espaço, os dados de entrada são considerados como dados normais e a origem é a anomalia. A partir dessas duas classes o algoritmo aprende e define o limite de decisão, ou seja, o hiperplano que separa a maioria dos dados da origem.

Podemos definir $g(.)$ como a Equação 2.3.

$$g(x) = \omega^T \phi(x) - \rho \tag{2.3}$$

Onde ω é o vetor perpendicular ao limite de decisão e ρ é o termo de polarização. Para a classificação de um ponto é utilizada a Equação 2.4.

$$f(x) = sinal(g(x)) \tag{2.4}$$

A Equação 2.4 retornará um valor positivo para pontos normais e negativo para anomalias.

O objetivo primário de uma SVM de classe única é dado pela Equação 2.5.

$$\begin{aligned} min_{\omega,\epsilon,\rho} \frac{||w||^2}{2} - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \\ sujeito : \omega^T \phi(x_i) \geq \rho - \xi_i \\ \xi_i \geq 0, \\ i = 1, ..., n \end{aligned} \tag{2.5}$$

Onde ξ é a variável de folga para a amostra, permitindo que ela permaneça do outro lado do limite de decisão, n é o tamanho do conjunto de dados de treinamento e ν é o parâmetro de regularização.

O limite de decisão é definido por 2.6:

$$g(x) = 0 \tag{2.6}$$

Sendo assim, a distância de qualquer amostra até o limite de decisão é dado por 2.7:

$$d(x) = \frac{|g(x)|}{||w||} \tag{2.7}$$

A distância a ser maximizada pelo algoritmo pode ser obtida ligando a origem na equação $\frac{\rho}{||w||}$. Também pode ser tratado como um problema de minimização de $\frac{||w^2||}{2} - \rho$.

A segunda parte do objetivo principal é a minimização das variáveis de folga ξ_i para todos os pontos. ν é o parâmetro de regularização e representa um limite superior na fração de

anomalias e um limite inferior no número de vetores de suporte. Este parâmetro é responsável por controlar o *trade-off* entre ξ e ρ.

A Figura 7 ilustra como seria uma definição de um hiperplano de separação dos dados normais e a origem.

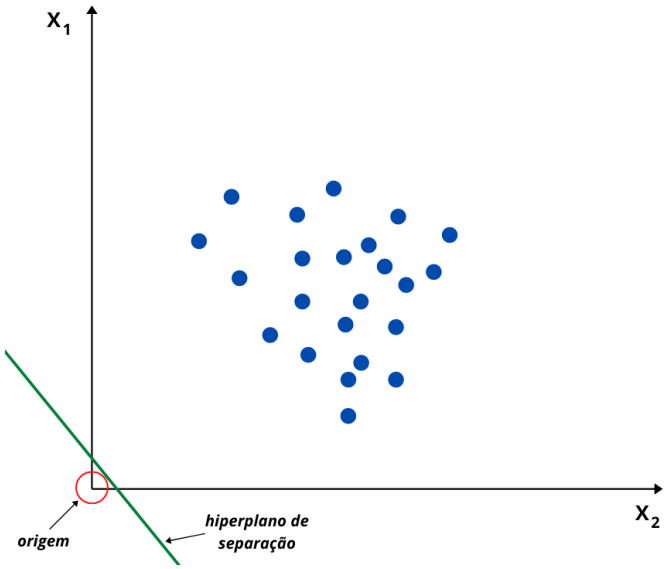

Figura 7 – Exemplo de um hiperplano separando os dados.

O modelo de OCSVM utilizado neste projeto encontra-se disponível na biblioteca da *Scikit-Learn* (PEDREGOSA et al., 2011). Dentre os parâmetros disponíveis para ajuste foram alterados os seguintes parâmetros.

- *kernel*: Especifica qual o tipo de função que irá mapear espaço de entrada no espaço de atributos.
- *nu*: Define a porcentagem do conjunto de dados que é discrepante auxiliando na criação de limites de decisão mais rígidos.

3 METODOLOGIA

Nesta seção será apresentada a base de dados e os tratamentos nela realizados além dos passos e parâmetros utilizados em cada método para a realização dos experimentos.

A Figura 8 facilita a visualização e entendimento dos processos e etapas utilizados na execução deste trabalho. O primeiro passo deste *pipeline* consiste em definir qual base de dados será utilizada no projeto. Esta base de dados está em uma forma bruta, então o segundo passo é aplicado e algumas manipulações serão realizadas gerando então dados tratados. A próxima etapa consiste em definir os modelos de detecção de anomalias e iniciar o treinamento destes; com os modelos treinados por fim teremos as análises que trarão os resultados finais, ou seja, se um dado é normal ou anômalo.

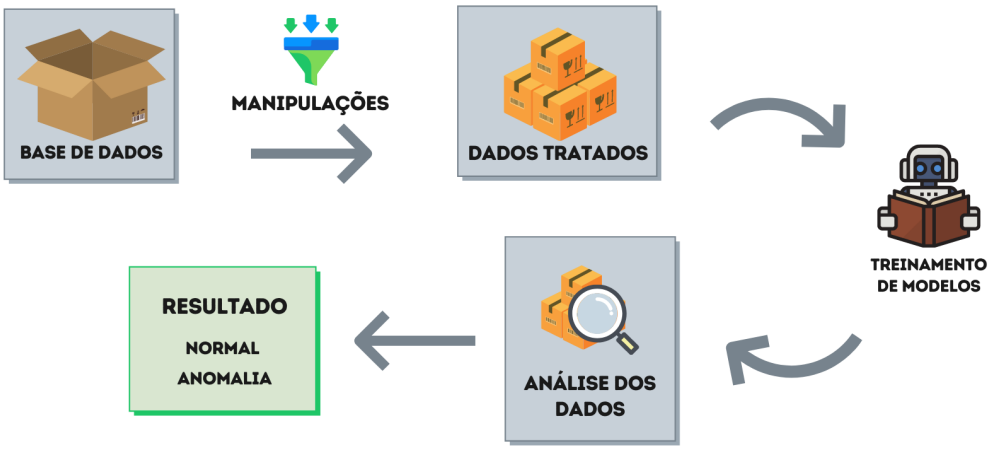

Figura 8 – Fluxograma de procedimentos e etapas executados neste trabalho.

3.1 Base de dados

Para o desenvolvimento da pesquisa, foi utilizada uma base de dados com dados reais obtidos pela rede CAN de um veículo através de um *OBD-II Scaner*. A base de dados foi construída por pesquisadores da Coreia do Sul KWAK; WOO; KIM 2016 com o objetivo de desenvolver um método antifurto baseado em perfis dos condutores.

O ambiente de execução do experimento foi totalmente real possibilitando coletar dados de situações que não ocorrem em um ambiente simulado. Os dados provenientes de 51 sensores embarcados são registrados a cada 1 segundo durante a condução em um modelo da *Kia Motors*

Corporation. Os 51 sensores permitem registrar dados da velocidade, RPM, pressão interna do tanque, temperatura do óleo de transmissão, ângulo do volante, posição do pedal do acelerador e outros como pode ser visto em (MARTINELLI et al., 2018b).

O mesmo veículo foi dirigido por 10 motoristas em 4 caminhos de ida e volta diferentes em Seul, capital da Coreia do Sul, sendo os caminhos entre a Universidade da Coreia e o estádio SANGAN World Cup. O percurso é constituído por via urbana, rodovia e local de estacionamento fazendo com que sejam coletados dados de locais com semáforos, faixas de pedestres, limite maior de velocidade e no caso do estacionamento manobras mais cautelosas. O caminho do início ao fim possui uma extensão de 46 km. O experimento foi iniciado em Julho de 2015 e era realizado sempre entre as 20h até as 23h em dias de semana.

A base de dados[1] é composta por 94401 registros de 51 sensores, ou seja, um total de 4.814.451 de amostras divididos entre 10 motoristas (rotulados de A-J). Para adequar a base de dados ao problema definido na Seção 2, algumas manipulações foram realizadas e serão detalhadas nas seções a seguir.

3.1.1 Descrição e procedimento de normalização

O primeiro procedimento realizado foi a normalização dos dados que consiste em padronizar para uma escala os valores mantendo as características. Na base de dados utilizada, é muito comum ter valores altamente discrepantes em sensores diferentes como por exemplo sensores que tem valores na faixa de 0 a 100 e sensores que tem valores na faixa de 0 a 1000.

Logo, esta normalização foi realizada nos grupos de dados provenientes de cada sensor. Esta tem como objetivo retirar algum viés que os modelos podem ter em relação a magnitude dos valores.

Com a base normalizada a próxima manipulação realizada foi a criação das Janelas Deslizantes que serão definidas a seguir.

3.1.2 Janelas deslizantes

Como dito anteriormente, a base de dados é constituída de dados amostrados uma vez por segundo para cada condutor. Devido a forma que foi construída e aos métodos que serão utilizados, é possível que uma amostra não forneça informação suficiente para que o modelo identifique o condutor corretamente. Com o intuito de construir padrões e identidades para cada condutor e testar a hipótese acima foi utilizada a técnica de janelas deslizantes.

O nome janela deslizante se deve ao fato de deslizar durante os segundos abrangendo todas as possibilidades de padrões.

Para construir as janelas é necessário definir dois parâmetros, o tamanho da janela, n,

[1] Disponível em https://sites.google.com/a/hksecurity.net/ocslab/ Datasets/driving-dataset

e os sensores, i, que serão utilizados. A Tabela 1 exemplifica uma janela com tamanho de 3 segundos e utilizando 2 sensores.

	Janela de Tempo					
Manobra 1	sensor1_s1	sensor2_s1	sensor1_s2	sensor2_s2	sensor1_s3	sensor2_s3
Manobra 2	sensor1_s2	sensor2_s2	sensor1_s3	sensor2_s3	sensor1_s4	sensor2_s4
Manobra 3	sensor1_s3	sensor2_s3	sensor1_s4	sensor2_s4	sensor1_s5	sensor2_s5

Tabela 1 – Exemplo de como as janelas são construídas.

Após esta manipulação, o resultado obtido é uma base de dados com diversas janelas deslizantes que mapeiam todos os dados antes separados. A seguir será feita uma relação destas janelas construídas com as manobras realizadas por cada condutor.

3.1.3 Manobras

Em uma situação de condução real, a manobra para fazer uma curva é realizada dentro de uma faixa de tempo e com uma sequência de movimentos, sendo que estes movimentos podem ser captados pelos sensores de forma direta ou indireta.

Esta definição também se aplica as janelas deslizantes, já que elas possuem uma faixa de tempo com dados obtidos pelos sensores.

Portanto, podemos relacionar também cada janela como sendo uma manobra do condutor, então a partir daqui adotaremos o nome de manobra. Após seguir os passos acima já temos os dados divididos por condutores e também manobras formadas.

Depois de relacionar cada janela deslizante como uma manobra, o passo seguinte é agrupar as manobras em classes.

3.1.4 Agrupamento de manobras

Dentro da base de dados, não é possível identificar quais dados representam quais manobras pelo simples fato dos criadores não apresentarem nenhuma classe neste quesito. Afim de contornar este problema, foi feito um agrupamento entre as manobras semelhantes através de um método chamado K-Means.

O K-Means foi utilizado pela primeira vez em MACQUEEN et al. 1967 e é um dos métodos de agrupamento mais populares. Este algoritmo tem como objetivo segregar em torno de K centros todos os dados, ou seja, particionar n observações para o seu K grupo mais próximo. Trazendo para este problema, o K é a quantidade de manobras existentes e assim serão formadas classes para cada K tipo de manobra.

A importância deste agrupamento é que a detecção de anomalia é feita por tipo de manobra. Pois uma manobra normal do condutor principal pode ser considerada anômala quando

comparada à outras manobras de tipos diferentes do mesmo condutor. Por isso, as anomalias são do tipo contextual.

Fazendo uma analogia com o objetivo de facilitar o entendimento, uma curva para a direita será um dado de entrada para um modelo que foi treinado para detectar anomalias em curvas para a direita. Portanto, para cada método de detecção existirão K modelos treinados.

3.2 Parâmetros dos modelos

Nesta seção serão detalhados os parâmetros utilizados para a execução dos experimentos.

Primeiro definimos a divisão da base de dados; esta foi embaralhada e dividida na proporção de 80 por 20, 80% destinado para o treinamento e 20% para os testes.

A Tabela 2 mostra os parâmetros utilizados em todos os métodos.

Nesta tabela estão os nomes dos sensores da mesma forma como estão na base de dados disponível.

selected_features	**window_size**	**n_clusters**
Intake_air_pressure	3	1
Engine_soacking_time	5	3
Long_Term_Fuel_Trim_Bank1	10	5
Torque_of_friction	30	10
Engine_coolant_temperature		30
Steering_wheel_speed		

Tabela 2 – Parâmetros utilizados em todos os métodos.

As Tabelas 3, 4 e 5 mostram, respectivamente, os parâmetros utilizados no métodos Ellipict Envelope, Isolation Forest e One Class Support Vector Machine.

O valor de contaminação foi definido como 0 devido ao modelo ser treinado apenas com dados do condutor principal e os demais parâmetros foram variados para entender os seus impactos nos modelos.

Parâmetro	**Valor**
contamination	0

Tabela 3 – Parâmetro utilizado no Elliptic Envelope.

Parâmetro	**Valor**			
n_estimators	5	15	25	50

Tabela 4 – Parâmetros utilizados na Isolation Forest.

Na próxima seção serão discutidas as métricas utilizadas para analisar os resultados que serão obtidos após a execução dos modelos.

Parâmetro	**Valor**					
kernel	rbf			sigmoid		
nu	0.1	0.01	0.001	0.0001	1e-05	1e-06

Tabela 5 – Parâmetros utilizados na One Class Support Vector Machine.

3.3 Métricas de avaliação

As duas primeiras métricas a serem definidas serão a *False Acceptance Rate* e a *False Rejection Rate*.

3.3.1 *False Acceptance Rate* - FAR

A *False Acceptance Rate* pode ser traduzida como Taxa de Falsas Aceitações. Esta métrica é a medida da probabilidade de o modelo aceitar incorretamente um condutor que não é o principal. Pode ser exemplificado na seguinte situação: Um meliante furta um automóvel e o modelo o classifica como o proprietário do veículo.

3.3.2 *False Rejection Rate* - FRR

A *False Rejection Rate* pode ser traduzida como Taxa de Falsas Rejeições. Esta é a medida da probabilidade de o modelo rejeitar o próprio condutor principal. O condutor principal está dirigindo mas o modelo o classifica como um meliante.

3.3.3 Tempo para detecção da anomalia

É o tempo necessário para que o método identifique uma anomalia. No caso de manobras de 5 segundos, quantas manobras serão necessárias para o modelo descubra que não é o motorista designado que está dirigindo.

O objetivo é rodar o experimento e salvar o todos os intervalos de tempo entre dados normais até que fosse apontada uma anomalia. No final serão calculados três tempos, o tempo mínimo até que uma anomalia fosse identificada, tempo máximo e tempo médio.

Essa métrica é importante pois é necessário saber quantos segundos dirigidos são utilizados para detectar se é o condutor principal ou um infrator. O modelo pode possuir uma baixa acurácia no período total, mas se estas falhas estiverem bem distribuídas durante o experimento em uma pequena faixa de tempo o modelo será capaz de identificar corretamente o condutor e isso faz com que seja considerado um resultado útil.

4 RESULTADOS E DISCUSSÃO

Para a análise dos resultados obtidos serão combinados dois tipos de gráficos; um deles é o *boxplot*, possibilitando a avaliação da distribuição empírica dos dados e o outro é um gráfico de pontos que possibilitará o detalhamento dos melhores resultados.

Dentre as métricas propostas para a avaliação dos modelos tem-se a FAR (False Acceptance Rate) e FRR (False Rejection Rate), que são julgadas como situações onde o modelo falhou, logo o objetivo do modelo é minimizar estas dua métricas.

Como mostrado na seção 3.2, dentro dos algoritmos foram iterados alguns parâmetros com intuito de ajustá-los da melhor forma. A partir daqui iremos nomear o número de *clusters* como tipos de manobra e a janela de tempo como tamanho das manobras para que os resultados fiquem mais simplificados.

Os experimentos são divididos em duas partes. Na Seção 4.1, avaliamos o comportamento de cada modelo em relação aos seus hiperparâmetros e outros parâmetros do problema. Na seção 4.2, fazemos uma comparação entre os diferentes modelos propostos.

4.1 Avaliação dos parâmetros

Para facilitar a análise dos resultados, foi então feita a média entre os FAR e FRR de cada experimento realizando, sendo que, quanto mais baixa a média, melhor o resultado. Essa média está em porcentagem e será padronizada como eixo das ordenadas nos gráficos a seguir.

Assim, nesta seção, para analisar o comportamento dos diferentes modelos, mostraremos os *boxplots* da média do FAR e FRR de cada experimento realizando. Sendo que, quanto mais baixa a média, melhor o resultado. Essa média está em porcentagem e será padronizada como eixo das ordenadas nos gráficos a seguir.

4.1.1 Elliptic Envelope

Neste experimento variou-se o número de tipos de manobra e o tamanho da manobra. O *Elliptic Envelope* foi o único método para o qual não foram iterados parâmetros internos. O parâmetro *contamination* foi fixado em 0 pelo fato de não conter contaminações no treinamento do modelo para aquele condutor principal.

A Figura 9 confirma que para o *Elliptic Envelope* o maior tamanho da manobra é a melhor opção e que o número de tipos de manobra maior também influenciou positivamente no resultado.

O maior tamanho da manobra e da quantidade de manobras influenciaram positivamente na maioria das situações. Manobras maiores dão aos algoritmos mais informação para determinar

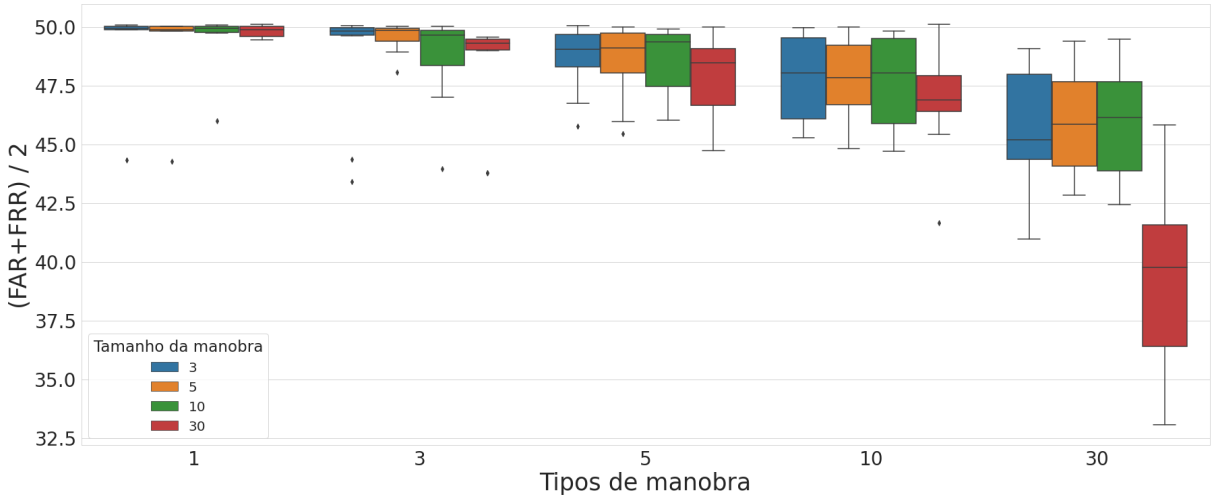

Figura 9 – Média FAR FRR do Elliptical Envelope variando o tamanho das manobras pelo número de tipos de manobra.

as anomalias. Permitir a definição de mais manobras dá ao algoritmo contextos mais específicos nos quais as anomalias devem ser identificadas.

4.1.2 Isolation Forest

Neste experimento variou-se o número de tipos de manobra, o tamanho da manobra e o *n_estimators* que é um parâmetro interno do algoritmo.

A Figura 10 mostra os resultados obtidos com a variação do *n_estimators*.

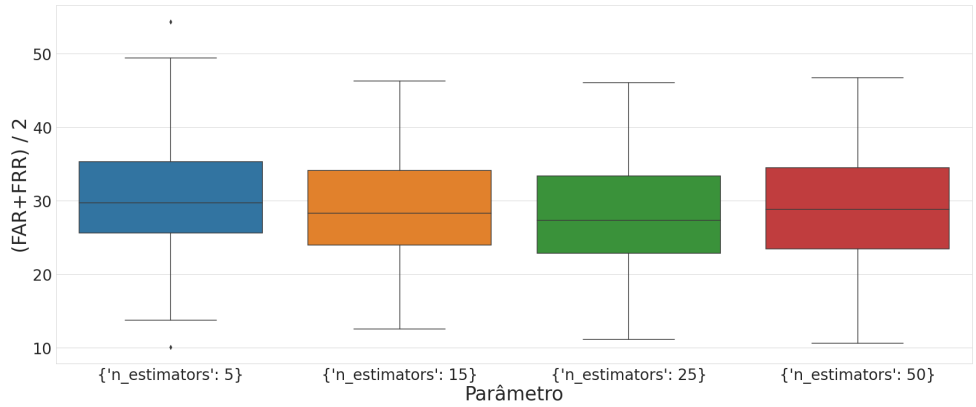

Figura 10 – Média FAR FRR da *Isolation Forest* variando o *n_estimators*.

Observa-se que não houve uma diferença significativa com a alteração do *n_estimators*. Ainda dentro da *Isolation Forest* podemos comparar como o tamanho da manobra e os tipos de manobra influenciam o seu desempenho. A Figura 11 ilustra esta comparação.

É perceptível um padrão decrescente ao longo do eixo das abscissas, confirmando que o maior número de tipos de manobra foi benéfico para o resultado do modelo. A construção das

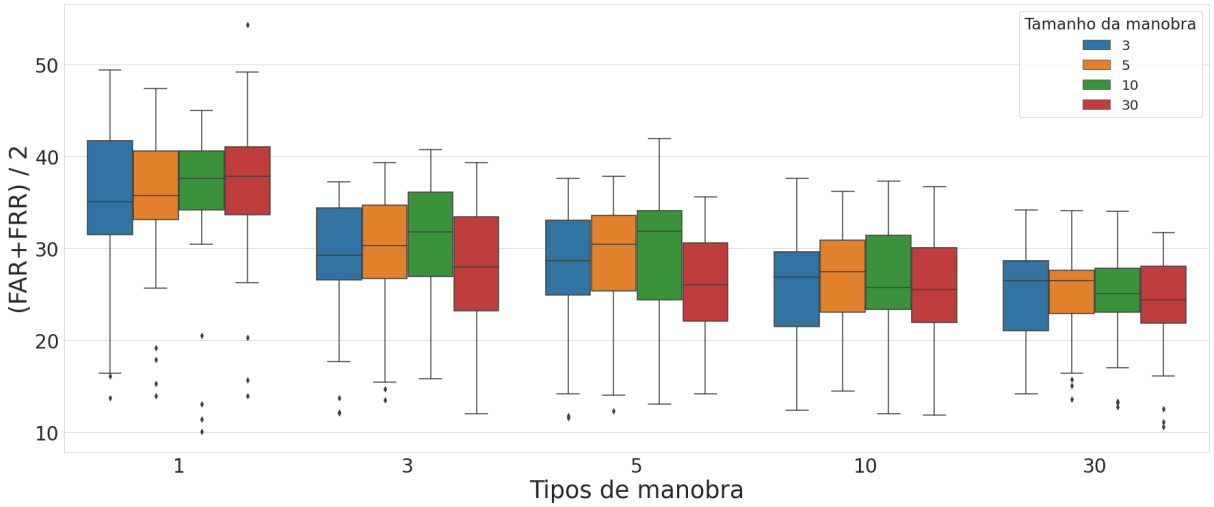

Figura 11 – Média FAR FRR para o modelo IF por tamanho de manobra variando o número de tipos de manobra.

classes de manobras faz com que as florestas sejam construídas para manobras mais específicas assemelhando-se aos resultados encontrados para os Ellipitic Envelopes.

Já o tamanho das manobras não influenciou significativamente o desempenho deste método.

4.1.3 One Class Support Vector Machine

Neste experimento variou-se o número de tipos de manobra, o tamanho da manobra e dois parâmetros internos do algoritmo, o *kernel* e o *nu*.

A Figura 12 mostra os resultados obtidos com a variação dos parâmetros internos do algoritmo. A melhor configuração encontrada foi utilizando como *kernel* a *rbf* e um *nu* de 0.1.

É observado que quanto menor o *nu* pior serão os resultados devido ao fato da variável de folga estar muito limitada e o modelo ficar mais sensível a ruídos. O *kernel rbf* produziu, em média, resultados melhores que o *sigmoid*.

Também na OCSVM, foram analisados os resultados obtidos através da variação do tamanho e tipo de manobras como mostra na Figura 13.

Diferentemente do outros métodos, não houve um padrão de melhora no desempenho que estivesse ligado aos tipos de manobra e aos tamanhos de manobra. Para as OCSVMs não houve diferença significativa de desempenho dentre as diferentes configurações.

4.2 Comparação entre modelos

A Figura 14 mostra a média da FAA e FRR obtidas em cada método. Como dito no parágrafo anterior, quanto menor essa média melhor o desempenho do método. Antes de explicar

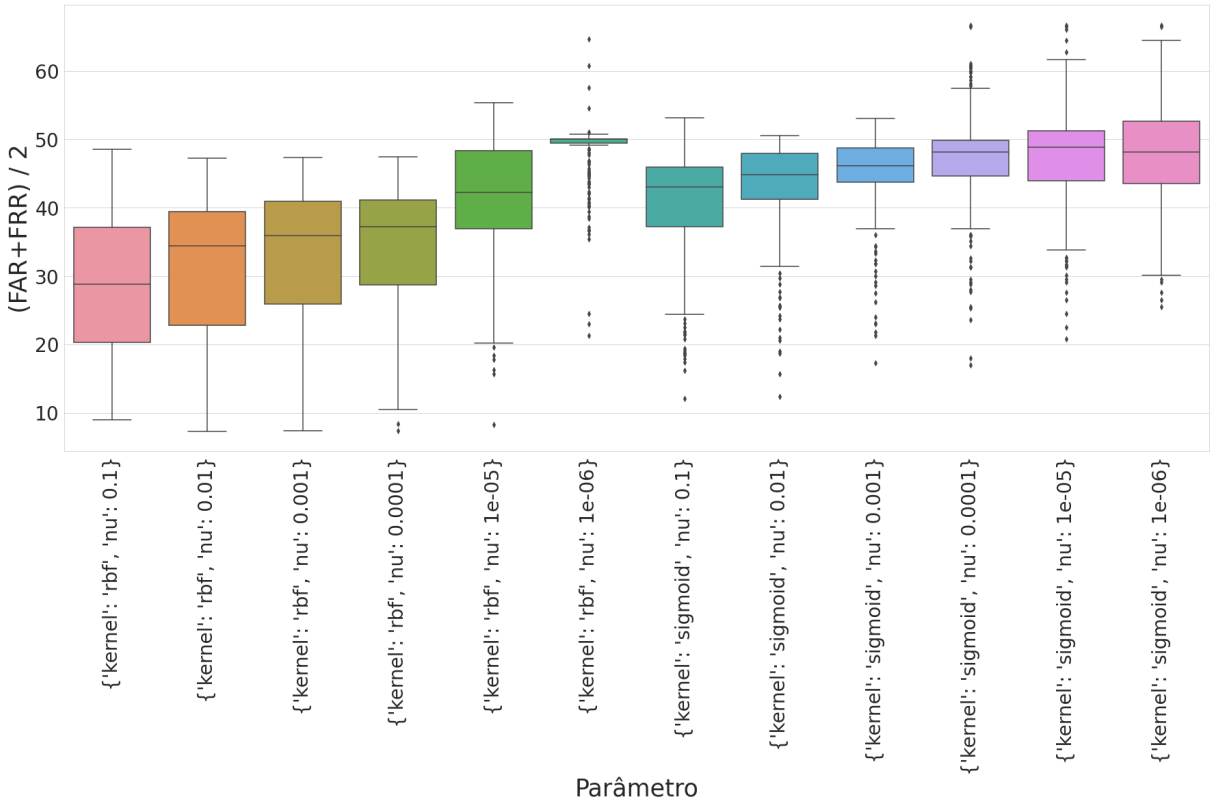

Figura 12 – Média FAR FRR da OCSVM variando os parâmetros.

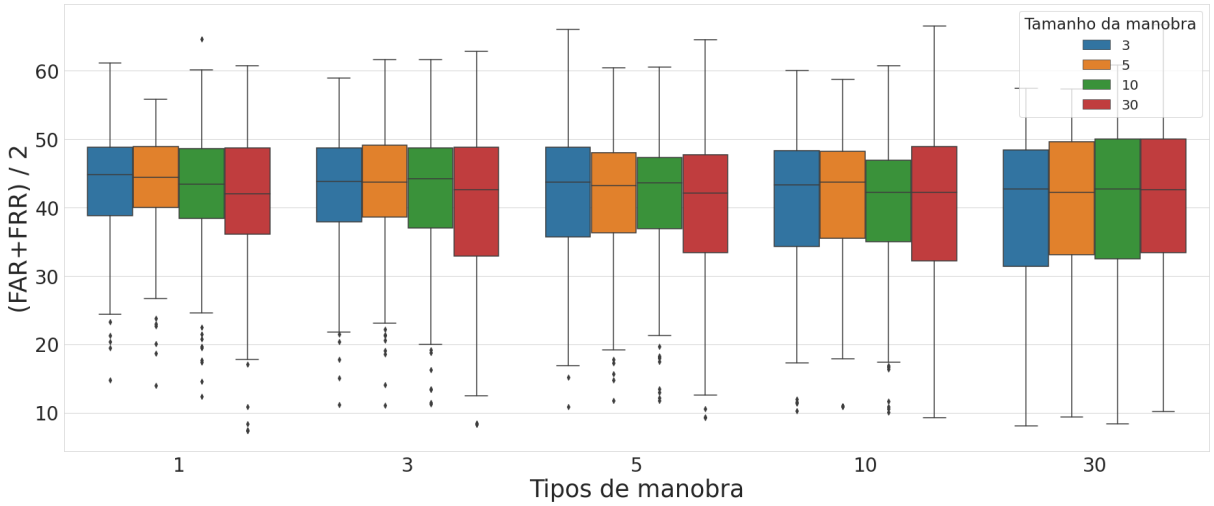

Figura 13 – Média FAR FRR para o modelo OCSVM por tamanho de manobra variando o número de tipos de manobra.

o gráfico, é necessário entender quais informações o mesmo está exibindo.

Na figura 14, temos uma comparação dos 3 tipos de modelos para diferentes tamanho das manobras.

Na figura, 15 comparamos os 3 modelos, variando-se o número de tipos de manobra.

Pode-se observar que em ambos os casos os modelos baseados em *Isolation Forests*

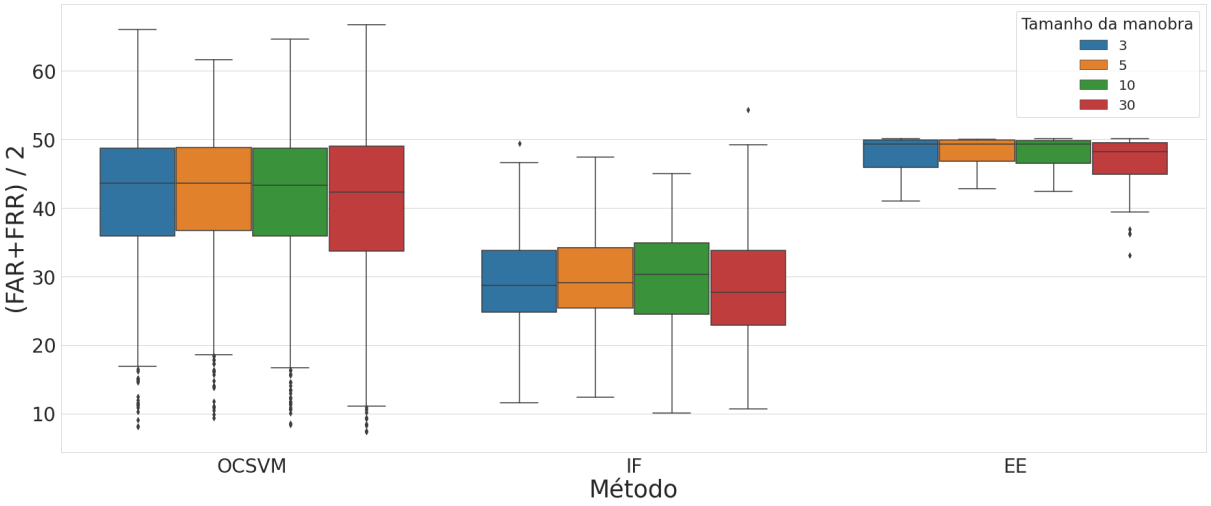

Figura 14 – Média FAR FRR dos métodos variando o tamanho das manobras.

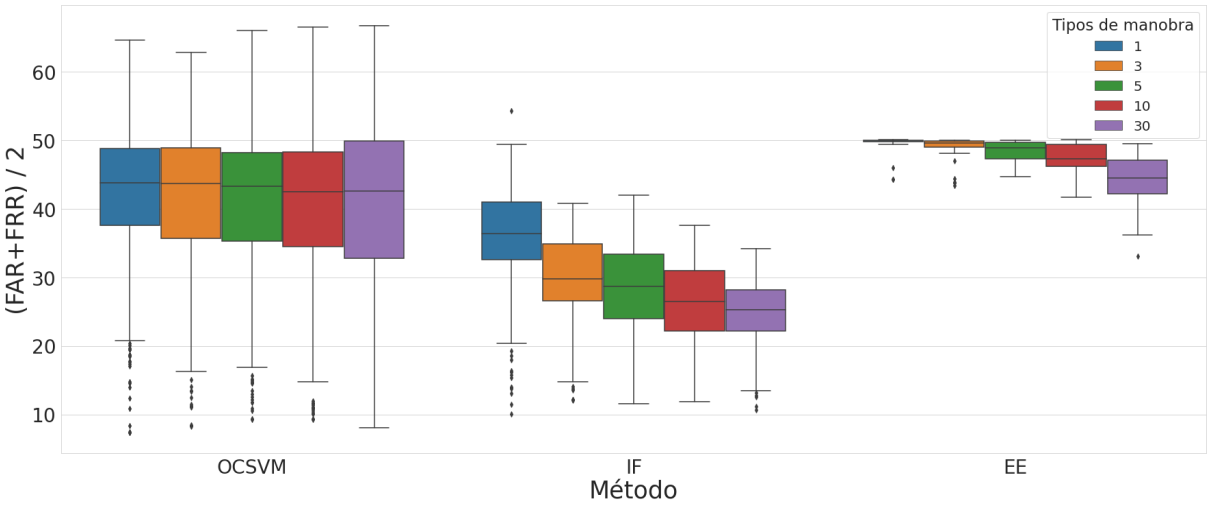

Figura 15 – Média FAR FRR dos métodos variando o número de tipos de manobras (clusters).

foram, na média, os mais bem sucedidos. Os outros dois modelos ficaram, na média, bem próximos de simples palpites aleatórios.

Durante a execução dos experimentos foi calculada também a média do tempo necessário para que o modelo identificasse o condutor como um impostor corretamente.

A Figura 16 mostra a média de tempo necessária para a detecção de um impostor pelo tamanho da manobra. O EE obteve o pior resultado dentre os métodos, em segundo lugar ficou o OCSVM e em primeiro o IF.

A Figura 17 mostra a média de tempo necessária para a detecção de um impostor pelos tipos de manobra. Os resultados se repetem, EE com o pior, em seguida o OCSVM e em primeiro o IF. Esta métrica é muito importante pois não é de valor um método com uma alta precisão dos resultados mas que necessite de uma grande faixa de tempo para entregar este resultado.

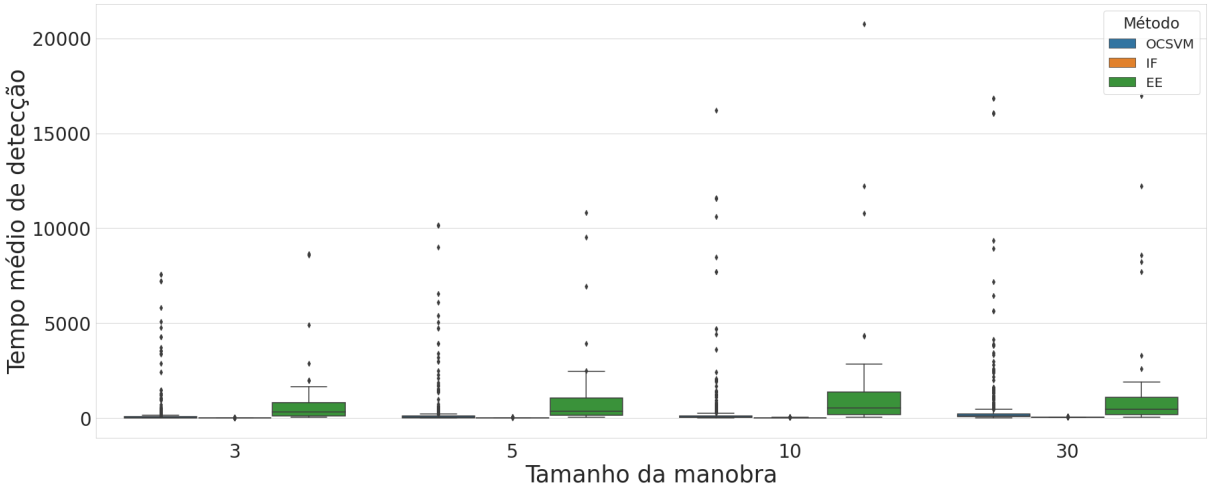

Figura 16 – Tempo médio de detecção dos algoritmos variando o tamanho das manobras.

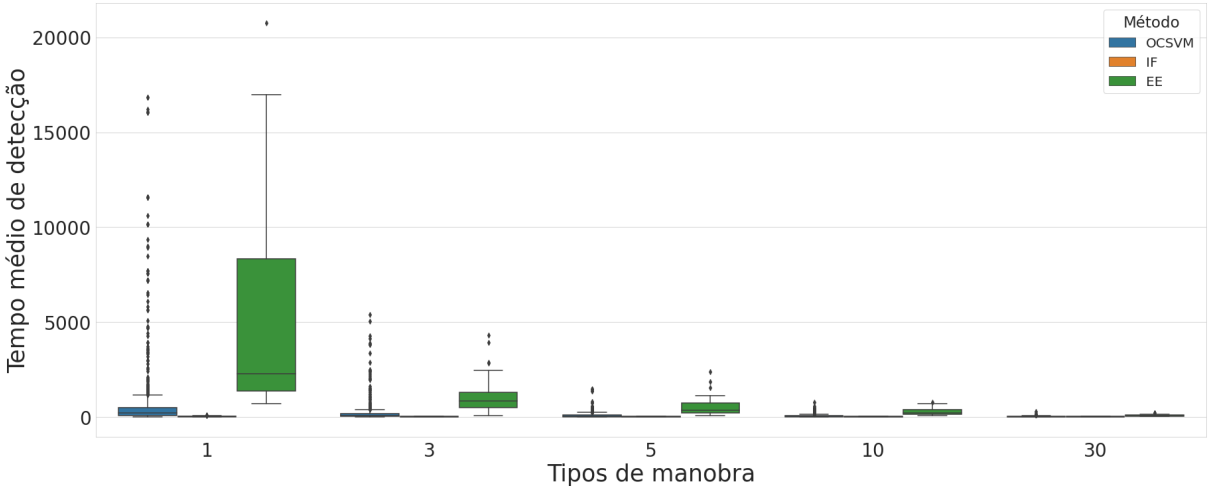

Figura 17 – Tempo médio de detecção dos algoritmos variando os tipos de manobra.

As análises realizadas acima possibilitam ver os resultados de uma perspectiva maior. Os *boxplots* representam de forma clara a distribuição de todos os dados, onde estão localizados a maior parte dos resultados e também valores extremos de máximos e mínimos.

Os gráficos a seguir tem como objetivo determinar de forma mais detalhada os resultados obtidos e então definir o melhor método para este problema.

As figuras 18 e 19 mostram a média do FAR e do FRR para cada tipo de modelo agrupados pelo $window\_size$ e $n\_cluster$.

A Figura 18 mostra a média alcançada por cada agrupamento mencionado no parágrafo anterior. O *Elliptic Envelope* apresenta os piores resultados pois possuem uma alta FAR, ou seja, aceitam praticamente todos motoristas. Esse fato reforça a hipótese que o envelope criado é tão grande que todos os pontos são internos, sendo assim classificados como dados normais.

Já *OCSVM* e *IF* possuem resultados melhores e mais próximos da origem do gráfico,

lembrando que o resultado ótimo é quando as duas taxas são iguais a 0.

Figura 18 – Média FAR X FRR entre todos os modelos.

É possível ver que o FAR e FRR são conflitantes, com intuito de visualizar os melhores resultados, foram filtradas as soluções não-dominadas, no sentido de dominância Pareto. A Figura 19 ilustra o resultado obtido.

Nota-se que os dois modelos mais próximos da origem foram um *Isolation Forest* e uma *OCSVM*. Os parâmetros destes dois melhores métodos são apresentados na Tabela 6. A partir destes parâmetros podemos concluir que o agrupamento das manobras em classes foi decisão benéfica ao resultado final e em relação ao tamanho da janela deslizante, a *Isolation Forest* precisou de uma janela de tempo bem maior que a *OCSVM*.

A partir destes parâmetros os resultados obtidos estão detalhados na Tabela 7. Esta tabela é composta pela porcentagem, tempo médio em segundos para correta identificação e tempo máximo tanto na FAR e FRR.

Para classificar qual o melhor método é necessário relembrar que o ideal é que a FRR e a

Figura 19 – Fronteira de Pareto da média FAR X FRR entre todos os modelos.

method	time_window	method_parameters	number_clusters
IF	30	n_estimators: 25	30
OCSVM	3	kernel:"rbf"; nu: 0.1	30

Tabela 6 – Configuração dos dois modelos que obtiveram os melhores resultados no experimento.

FAR sejam 0%, pois assim o modelo não rejeitou o condutor principal e também não aceitou um infrator como condutor principal.

Porém, a situação em que o condutor principal está dirigindo e o sistema emite um sinal como se o veículo estivesse sendo roubado compromete a viabilidade do sistema, ou seja, um menor valor do FRR resultando no modelo da OCSVM.

A situação da falsa aceitação, quando infrator é aceito como motorista em primeiro momento é ruim porém deve ser analisada com mais cautela. No experimento, o valor máximo de tempo até que o modelo identificasse um infrator foi coletado. No caso da OCSVM, este tempo foi de apenas 276,6 segundos ou 4,61 minutos se convertido. Na média, o modelo demorou apena $8s$ para detectar um infrator.

A OCSVM obteve um maior valor de FAR, mas em contrapartida o seu tempo máximo para detectar um infrator é a metade do tempo gasto pela IF.

	FAR			FRR		
method	**%**	**average_time_s**	**max_time_s**	**%**	**average_time_s**	**max_time_s**
IF	30.93	36.43	485.3	15.03	35.88	66.1
OCSVM	32.78	8.43	276.6	11.58	10.61	45.9

Tabela 7 – Resultados obtidos nos dois melhores experimentos.

Após estas análises, considerando todos os experimentos feitos, a das versão testada da OCSVM com os parâmetros mostrados na Tabela 6 foi a que apresentou os melhores resultados para esta aplicação.

5 CONCLUSÃO

A busca por segurança e inovação foram os combustíveis que motivaram o desenvolvimento deste trabalho. Na literatura vários pesquisadores atacam um problema semelhante tratando-o, no entanto, como um problema de classificação. Isso faz com que seja necessário que todos os motoristas estejam listados na base de dados. Em uma situação em que um infrator furta um veículo, é provável que, o mesmo não esteja listado na base de dados. Daí a ideia de tratar esse problema como um problema de detecção de anomalias.

Foram utilizados três métodos de detecção de anomalias populares e de fácil acesso em bibliotecas de aprendizado de máquina. Dentre eles o que trouxe o melhor resultado foi a *One Class Support Vector Machine*. O método obteve um bom resultado, mas que ainda não é suficiente para ser embarcado em um sistema de segurança antifurto por exemplo, pelo fato de ser eficiente em detectar um infrator mas identificar o condutor principal como infrator em algumas situações. Para ser um dispositivo de segurança efetivo alarmes falsos devem ser inexistentes ou quase nulos.

Resolver este problema como um problema de detecção de anomalias se mostrou bem mais difícil do que resolvê-lo como um problema de classificação. Contudo, por mais que o modelo não acerte 100% das situações em que o condutor é um infrator, em uma faixa de tempo menor que 5 minutos, pior caso, e 8 segundos, na média, ele é capaz de identificar e isso já é suficiente para que o sistema emita um alerta. Algo que deve ser evitado são as situações que o condutor principal está dirigindo e o modelo o classifique como infrator, pois assim o alerta emitido será falso.

Até aqui, concluiu-se que o método apresenta resultados promissores, mas ainda não suficientes para que a ferramenta seja utilizada na prática. É importante salientar que todos os motoristas da base de dados dirigiram o mesmo carro em apenas 4 percursos diferentes. Assim, boa parte da variação nas amostras é explicada somente pela variação dos motoristas. Num sistema prático, este cenário pode ser bem diferente.

De qualquer forma, este trabalho abre uma nova perspectiva de pesquisa na verificação de motoristas e é apenas o início de um vasto campo de métodos de detecção de anomalias que ainda pode ser explorado.

5.1 Trabalhos Futuros

Para trabalhos futuros existem alguns caminhos a serem seguidos de imediato. Primeiramente, poderia-se explorar mais cada um dos três métodos, no momento de treinamento adicionar ao conjunto de dados alguma contaminação e ver como os modelos se comportam, principalmente no método da OCSVM.

A troca de sensores utilizados também é um ótimo ponto a se discutir. Talvez o desenvolvimento de um algoritmo que seja capaz de selecionar dentre os 51 atributos disponíveis os melhores.

No campo de detecção de anomalias existem diversos outros métodos como Local Factor Outlier e até mesmo alguns métodos de agrupamento como o DBSCAN que podem ser aplicados a este problema.

REFERÊNCIAS

AKHILOMEN, J. Data mining application for cyber credit-card fraud detection system. In: SPRINGER. *Industrial Conference on Data Mining*. [S.l.], 2013. p. 218–228. Citado na página 16.

ALLA, S.; ADARI, S. K. *Beginning Anomaly Detection Using Python-Based Deep Learning*. [S.l.]: Springer, 2019. Citado 5 vezes nas páginas 16, 17, 20, 21 e 24.

AMER, M.; GOLDSTEIN, M.; ABDENNADHER, S. Enhancing one-class support vector machines for unsupervised anomaly detection. In: *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. New York, NY, USA: Association for Computing Machinery, 2013. (ODD '13), p. 8–15. ISBN 9781450323352. Disponível em: <https://doi.org/10.1145/2500853.2500857>. Citado na página 24.

ANGKITITRAKUL, P. et al. Utdrive: Driver behavior and speech interactive systems for in-vehicle environments. In: IEEE. *2007 IEEE Intelligent Vehicles Symposium*. [S.l.], 2007. p. 566–569. Citado na página 14.

ARNEODO, A.; GRASSEAU, G.; HOLSCHNEIDER, M. Wavelet transform of multifractals. *Physical review letters*, 1988. APS, v. 61, n. 20, p. 2281, 1988. Citado na página 14.

BANDO, M. et al. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E*, 1995. American Physical Society, v. 51, p. 1035–1042, Feb 1995. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.51.1035>. Citado na página 14.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.: s.n.], 1992. p. 144–152. Citado na página 23.

BREIMAN, L. Random forests. *Machine Learning*, 2001. Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023-/A%3A1010933404324>. Citado 2 vezes nas páginas 15 e 20.

BREIMAN, L. et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984. Citado 2 vezes nas páginas 15 e 20.

BUDALAKOTI, S.; SRIVASTAVA, A.; OTEY, M. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2009. v. 39, n. 1, p. 101–113, 2009. Citado na página 16.

CHOI, S. et al. Analysis and classification of driver behavior using in-vehicle can-bus information. In: *Biennial workshop on DSP for in-vehicle and mobile systems*. [S.l.: s.n.], 2007. p. 17–19. Citado 2 vezes nas páginas 13 e 15.

ENEV, M. et al. Automobile driver fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2016. v. 2016, 01 2016. Citado na página 15.

G1.GLOBO.COM. *Nove carros foram roubados por hora em 2016 em SP, aponta levantamento*. 2016. Acessado em: 05-11-2019. Disponível em: <http://g1.globo.com/sao-paulo/noticia/2016-/06/nove-carros-foram-roubados-por-hora-em-2016-em-sp-aponta-levantamento.html>. Citado na página 12.

GUO, G. et al. Knn model-based approach in classification. In: SPRINGER. *OTM Confederated International Conferences"On the Move to Meaningful Internet Systems"*. [S.l.], 2003. p. 986–996. Citado na página 13.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics). Citado na página 15.

HELLY, W. Simulation of bottlenecks in single-lane traffic flow. *Proc. Symposium on Theory of Traffic Flow*, 1959. v. 322, n. 10, p. 207–238, 1959. Citado na página 14.

JANG, J.-S. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 1993. IEEE, v. 23, n. 3, p. 665–685, 1993. Citado na página 15.

JUSTICA.GOV. *Aplicativo que identifica carros roubados já tem 740 mil downloads*. 2013. Acessado em: 05-11-2019. Disponível em: <https://www.justica.gov.br/news-/aplicativo-que-identifica-carros-roubados-ja-tem-740-mil-downloads>. Citado na página 12.

KEDAR-DONGARKAR, G.; DAS, M. Driver classification for optimization of energy usage in a vehicle. *Procedia Computer Science*, 2012. v. 8, p. 388–393, 12 2012. Citado na página 13.

KWAK, B. I.; WOO, J.; KIM, H. K. Know your master: Driver profiling-based anti-theft method. In: IEEE. *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. [S.l.], 2016. p. 211–218. Citado 2 vezes nas páginas 15 e 27.

LI, R.; LIU, C.; LUO, F. A design for automotive can bus monitoring system. In: *2008 IEEE Vehicle Power and Propulsion Conference*. [S.l.: s.n.], 2008. p. 1–5. Citado 2 vezes nas páginas 12 e 14.

LIU, F. T.; TING, K.; ZHOU, Z.-H. Isolation forest. In: *Isolation Forest*. [S.l.: s.n.], 2009. p. 413 – 422. Citado 4 vezes nas páginas 9, 16, 21 e 22.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. *2008 Eighth IEEE International Conference on Data Mining*. [S.l.], 2008. p. 413–422. Citado na página 21.

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. Citado na página 29.

MARTÍ, L. et al. Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 2015. Multidisciplinary Digital Publishing Institute, v. 15, n. 2, p. 2774–2797, 2015. Citado na página 16.

MARTINELLI, F. et al. Who's driving my car? a machine learning based approach to driver identification. In: *ICISSP*. [S.l.: s.n.], 2018. p. 367–372. Citado na página 15.

MARTINELLI, F. et al. Human behavior characterization for driving style recognition in vehicle system. *Computers Electrical Engineering*, 2018. v. 83, 01 2018. Citado na página 28.

MENG, X.; LEE, K. K.; XU, Y. Human driving behavior recognition based on hidden markov models. In: *2006 IEEE International Conference on Robotics and Biomimetics*. [S.l.: s.n.], 2006. p. 274–279. Citado na página 14.

MIIKKULAINEN, R. et al. Evolving deep neural networks. In: *Artificial intelligence in the age of neural networks and brain computing*. [S.l.]: Elsevier, 2019. p. 293–312. Citado na página 15.

MIYAJIMA, C. et al. Driver modeling based on driving behavior and its evaluation in driver identification. *Proceedings of the IEEE*, 2007. v. 95, n. 2, p. 427–437, 2007. Citado na página 14.

PAL, S. K.; MITRA, S. Multilayer perceptron, fuzzy sets, classifiaction. 1992. 1992. Citado na página 15.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 2011. v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <http://jmlr.org/papers/v12/pedregosa11a.html>. Citado 4 vezes nas páginas 17, 18, 22 e 26.

PRESS, W. et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN 9780521880688. Disponível em: <https://books.google.com.br/books?id=1aAOdzK3FegC>. Citado na página 23.

PUNNOOSE, S.; KUMAR, J. S. J. Iris recognition for security & safety of automobiles. *International Journal of Innovative Science. Engineering & Technology*, 2015. v. 2, n. 4, 2015. Citado na página 13.

RABINER, L.; JUANG, B. An introduction to hidden markov models. *IEEE ASSP Magazine*, 1986. v. 3, n. 1, p. 4–16, 1986. Citado 2 vezes nas páginas 13 e 14.

RAHMAT, R. et al. Facial recognition for car security system using fisherface method. *Journal of Physics: Conference Series*, 2019. v. 1235, p. 012119, 06 2019. Citado na página 13.

RASMUSSEN, C. E. et al. The infinite gaussian mixture model. In: CITESEER. *NIPS*. [S.l.], 1999. v. 12, p. 554–560. Citado na página 14.

ROUSSEEUW, P.; DRIESSEN, K. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 1999. v. 41, p. 212–223, 08 1999. Citado na página 18.

ROUSSEEUW, P. J. Least median of squares regression. *Journal of the American statistical association*, 1984. Taylor & Francis, v. 79, n. 388, p. 871–880, 1984. Citado na página 18.

ROUSSEEUW, P. J.; DRIESSEN, K. V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 1999. Taylor & Francis, v. 41, n. 3, p. 212–223, 1999. Citado na página 18.

SADHUKHAN, S.; ACHARYYA, A.; PRASAD, R. Car security system using fingerprint scanner and iot. *Indian Journal of Science and Technology*, 2017. v. 10, p. 1–4, 12 2017. Citado na página 13.

SCHMIDT, A.-D. et al. Monitoring smartphones for anomaly detection. *Mobile Networks and Applications*, 2009. Springer, v. 14, n. 1, p. 92–106, 2009. Citado na página 16.

SCHÖLKOPF, B. et al. Support vector method for novelty detection. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2000. p. 582–588. Citado 2 vezes nas páginas 16 e 24.

TOLEDO, T.; MUSICANT, O.; LOTAN, T. In-vehicle data recorders for monitoring and feedback on drivers' behavior. *Transportation Research Part C: Emerging Technologies*, 2008. v. 16, p. 320–331, 06 2008. Citado na página 13.

WAHAB, A. et al. Driving profile modeling and recognition based on soft computing approach. *IEEE Transactions on Neural Networks*, 2009. v. 20, n. 4, p. 563–582, 2009. Citado na página 14.

WAKITA, T. et al. Driver identification using driving behavior signals. *IEICE TRANSACTIONS on Information and Systems*, 2006. The Institute of Electronics, Information and Communication Engineers, v. 89, n. 3, p. 1188–1194, 2006. Citado 2 vezes nas páginas 14 e 15.

WIKIPEDIA. *OBD-II PIDs*. 2019. https://en.wikipedia.org/wiki/OBD-II_PIDs. Disponível em: <https://en.wikipedia.org/wiki/OBD-II_PIDs>. Acesso em: 05 nov. 2019. Citado na página 12.

WIKIPEDIA. *CAN BUS*. 2019 – Acessado em: 08–11–2019. https://en.wikipedia.org/wiki/CAN_bus. Disponível em: <https://en.wikipedia.org/wiki/CAN_bus>. Acesso em: 08 nov. 2019. Citado na página 12.

WILLIS, R. R.; AUSTELL, W. P. Gmss graphic modelling and simulation system. In: *Proceedings of the 16th Annual Symposium on Simulation*. Washington, DC, USA: IEEE Computer Society Press, 1983. (ANSS '83), p. 137–160. Citado na página 15.

ZIMEK, A.; SCHUBERT, E. Outlier detection. In: ____. *Encyclopedia of Database Systems*. New York, NY: Springer New York, 2017. p. 1–5. ISBN 978-1-4899-7993-3. Disponível em: <https://doi.org/10.1007/978-1-4899-7993-3_80719-1>. Citado na página 17.