

# Seminário de Sistemas Embarcados e IoT 2020



Eronides da Silva Neto

Palestra

*Sistema de detecção de  
anomalias em sensor veicular  
baseado em sistema de telemetria*

Apoio



## print("Hello World!");



- Engenheiro no CESAR, Recife - PE.  
Atuo com projetos envolvendo sistemas embarcados e IoT desde 2015.
- Entusiasta do movimento open-source hardware.
- 
- Mestre em Ciência da Computação (CIn/UFPE)
- Engenheiro Eletrônico (UFPE)



## AGENDA

- Por que coletar informações veiculares?
- O sistema de detecção de anomalias veiculares
- Experimentos realizados a partir de base de dados carOBD
- Resultados



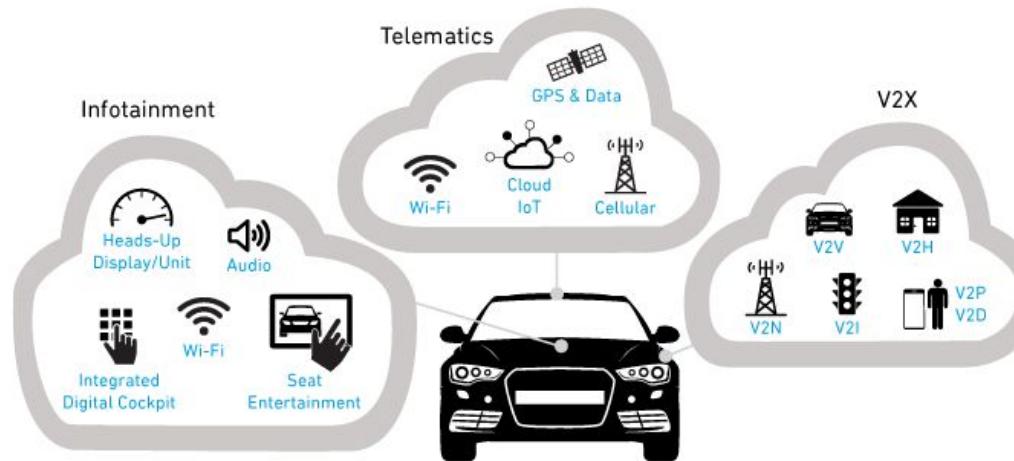
## O carro conectado





## Introdução

- Tendência dos veículos estabelecerem algum tipo de conexão a fim de fornecer novas experiências ao condutor.



qorvo.

©2018 Qorvo, Inc.



## Quantos MB de dados produzimos diariamente?





## Níveis de Automação do veículo x Dados produzidos

**1Mb/s**

Sem Automação



Veículo comum

**10 Mb/s**

Assistência ao Condutor



Execução de tarefas básicas

**Até 30Mb/s**

Sem Automação



Mais de um sistema de automação

**1GB/s**

Automação Condicionada



Executa maior parte das funções autônomas, mas necessita de motorista

**> 1GB/s**

Alta ou Total automação

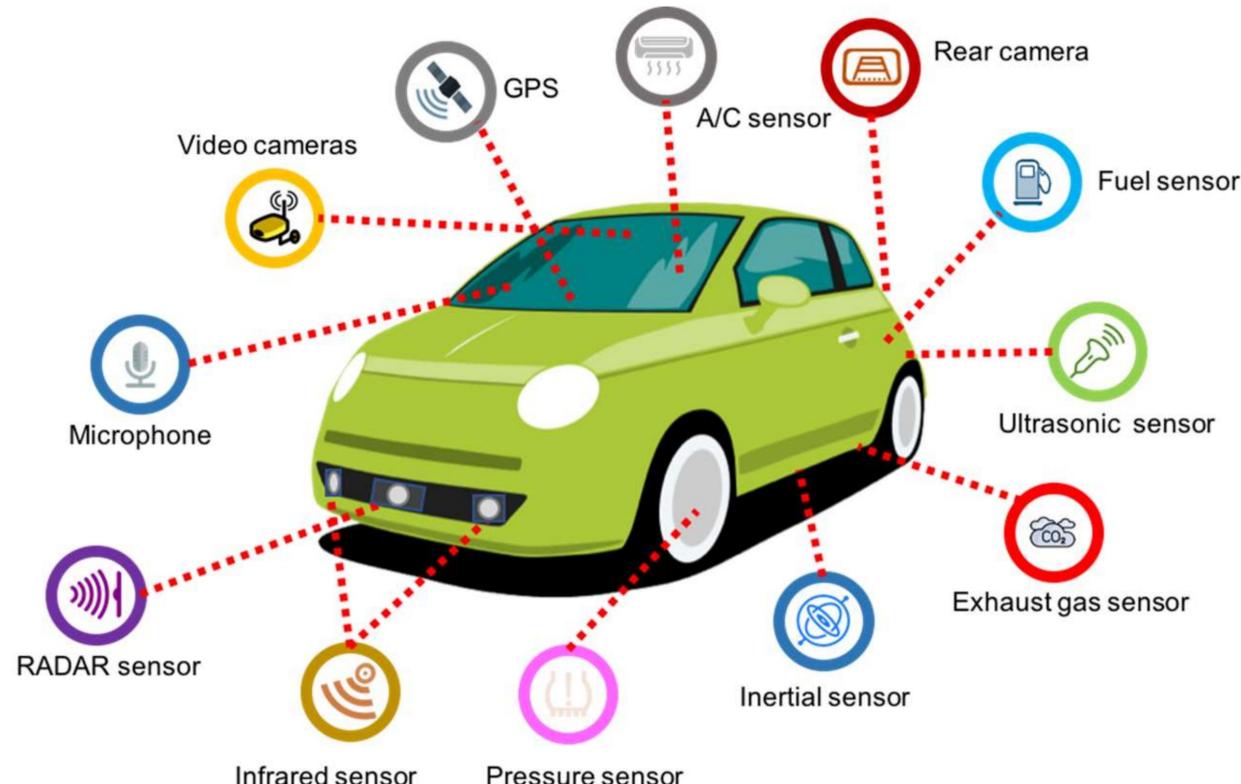


Veículo Autônomo

De onde vem essa quantidade de dados toda?



## Os sensores...



Fonte: Sensor Technologies for Intelligent Transportation Systems

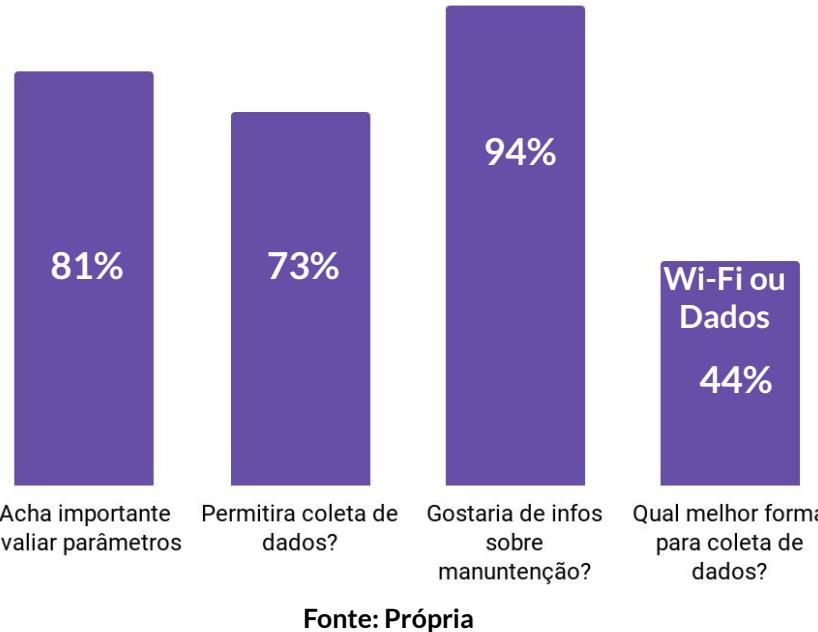


# O que fazer com os dados de um veículo comum?



## O carro como um produto conectado

Experiência do consumidor ao usar carro conectado



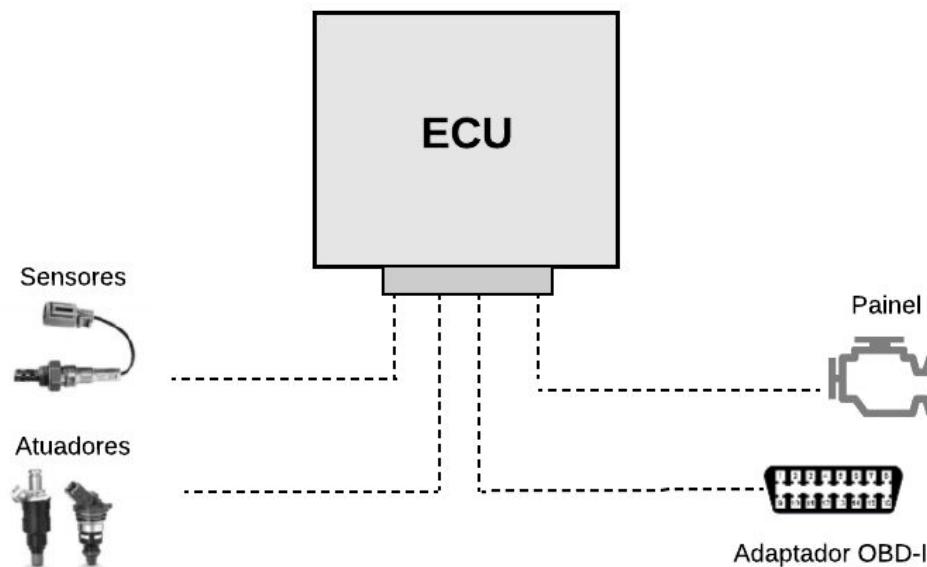


# **Sistema de Telemetria para captura de dados veiculares**



## ECUs

- Uma *Electronic Control Unit (ECU)*, é definida como o dispositivo de controle de um grupo de funções específicas do veículo.





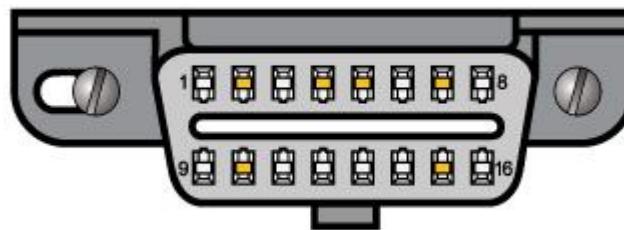
## Redes Intraveiculares

- As principais funções do veículo são distribuídas em diferentes unidades de controle (ECUs). Inicialmente a **interconexão** entre as diversas ECUs era realizada por meio de conexões ponto-a-ponto.
- Um **barramento intraveicular** específico para a comunicação das ECUs é proposto como solução, a *Controller Area Network (CAN)*.



## O padrão de diagnóstico OBD-II

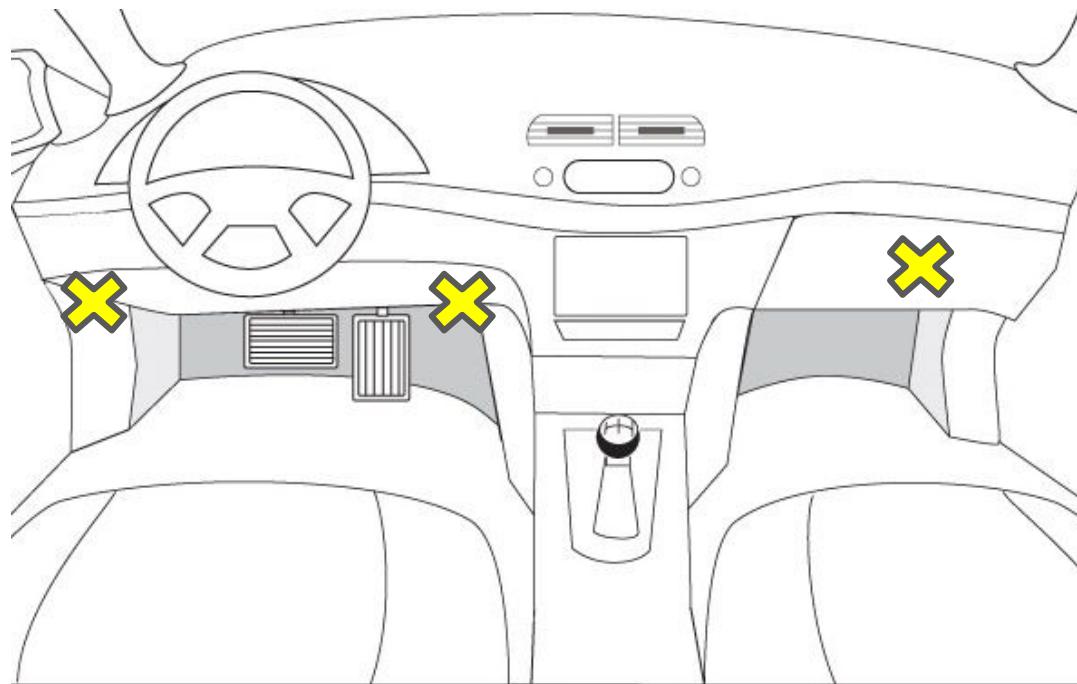
- A partir da rede CAN foi definido um padrão de diagnóstico veicular, *On-Board Diagnosis* (OBD).
- Estabelecimento do padrão *On-Board Diagnosis II* (OBD-II) no início dos anos 2000.
- Captura das informações veiculares a partir do envio de um código PID (Parameter IDs).



Fonte: ScanTool.Net

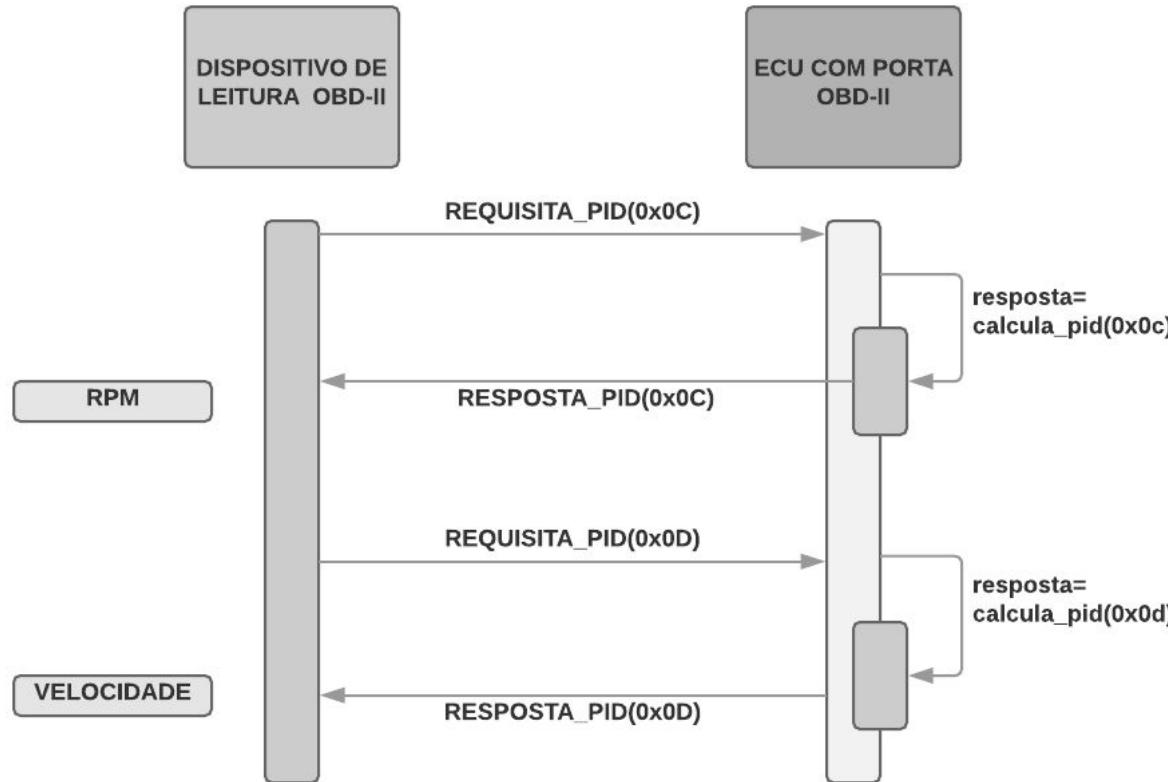


## Como localizar a OBD-II





## Como obter dados da OBD-II





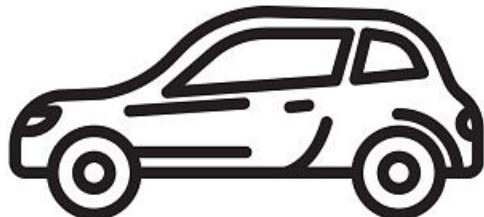
## Como obter dados da OBD-II





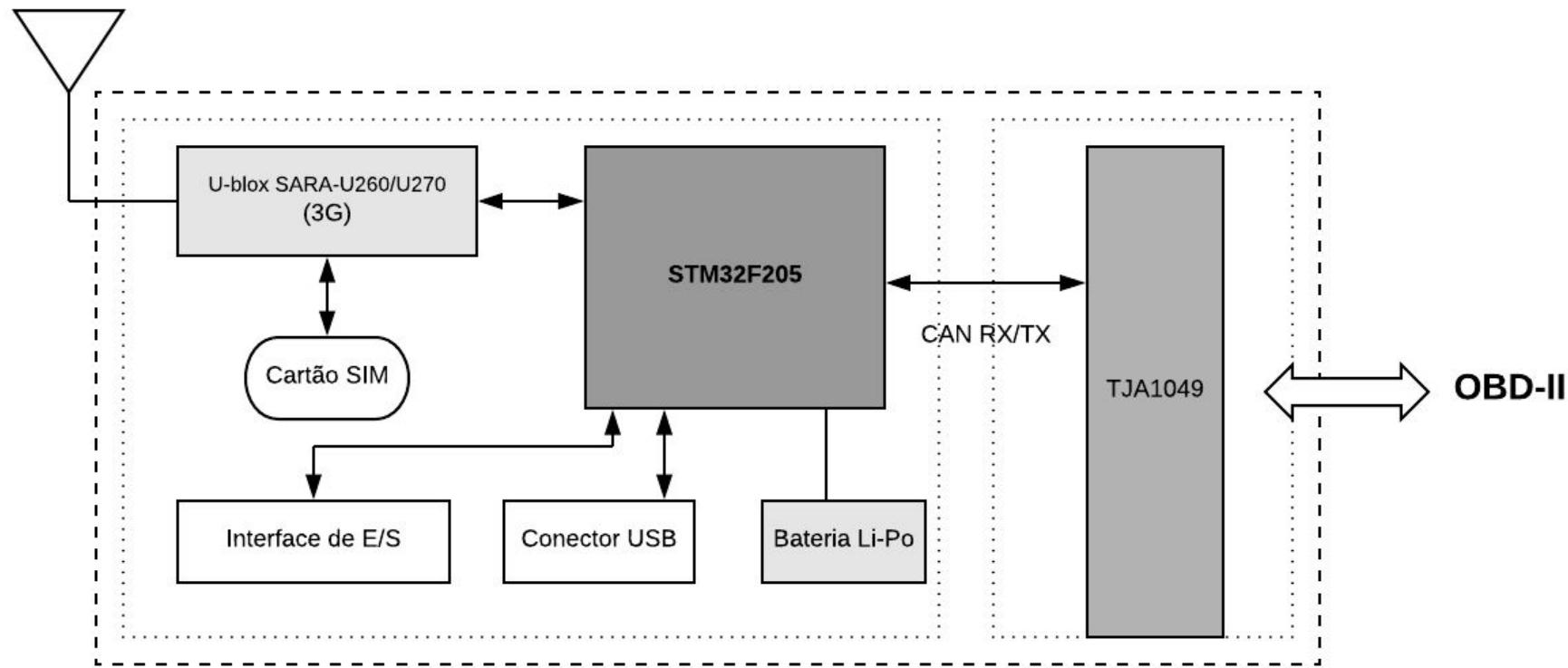
## Um carro, uma plaquinha, uma ideia!

- Aquisição a partir da interface OBD-II do veículo (**Toyota Etios 2014 1.5 16v Flex**).
- Uso da plataforma de desenvolvimento **Carloop**.
- Além da possibilidade da criação de base de dados local, a plataforma possui um módulo 3G.





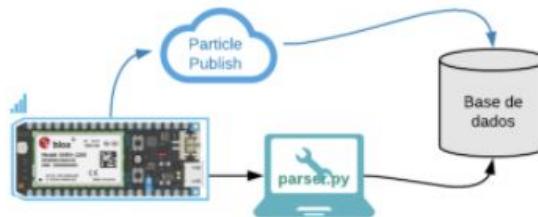
## Arquitetura de Hardware detalhada





## Criação da base de dados veiculares

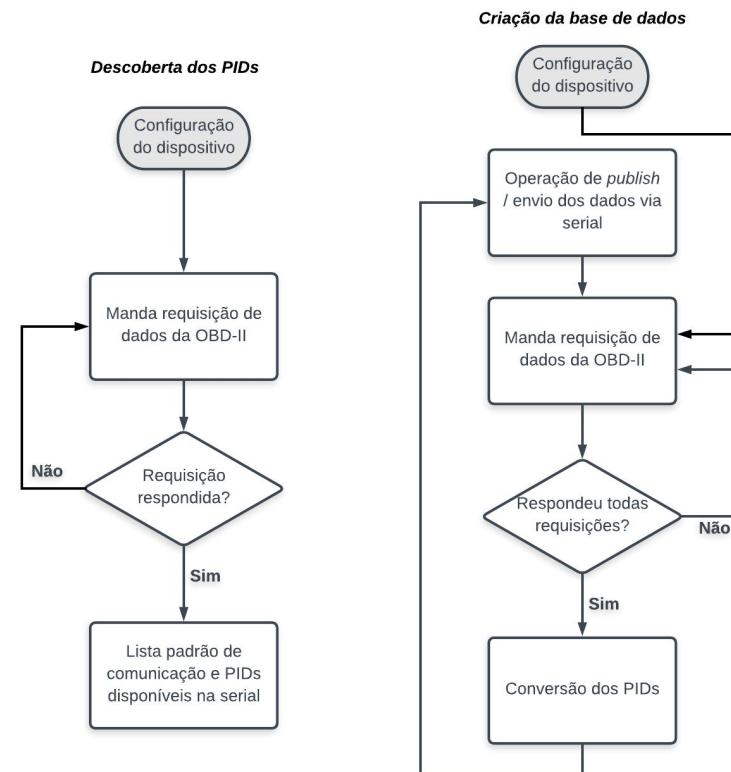
- Veículo com rede CAN com identificador de 11-bit e velocidade de transmissão de 500 kbit/s.
- São definidos dois modos de operação do veículo: modo em movimento e modo veículo parado (IDLE);
- Captura de **27 diferentes parâmetros** veiculares utilizando taxa de amostragem de **1Hz**.



Fonte: adaptada de (Particle, 2019)



## Criação da base de dados veiculares





## Quais dados estão disponíveis?

- Criação de uma base de dados com dois modos veiculares. Modo em movimento possui **quatro cenários distintos**, considerando diferentes tipos de trajetória e o modo IDLE representa o veículo parado.
- Ciclo veicular (*trip*): uma trajetória completa, para ambos modos veiculares.
- Trajeto em Áreas Urbanas
- Trajeto em Vias Expressas
- Trajeto em Universidade
- Trajeto Fixo



## Quais dados foram obtidos?

- Carga do Motor
- Velocidade
- Rotações por Minuto (RPM)
- Posição do pedal
- Nível do tanque de combustível

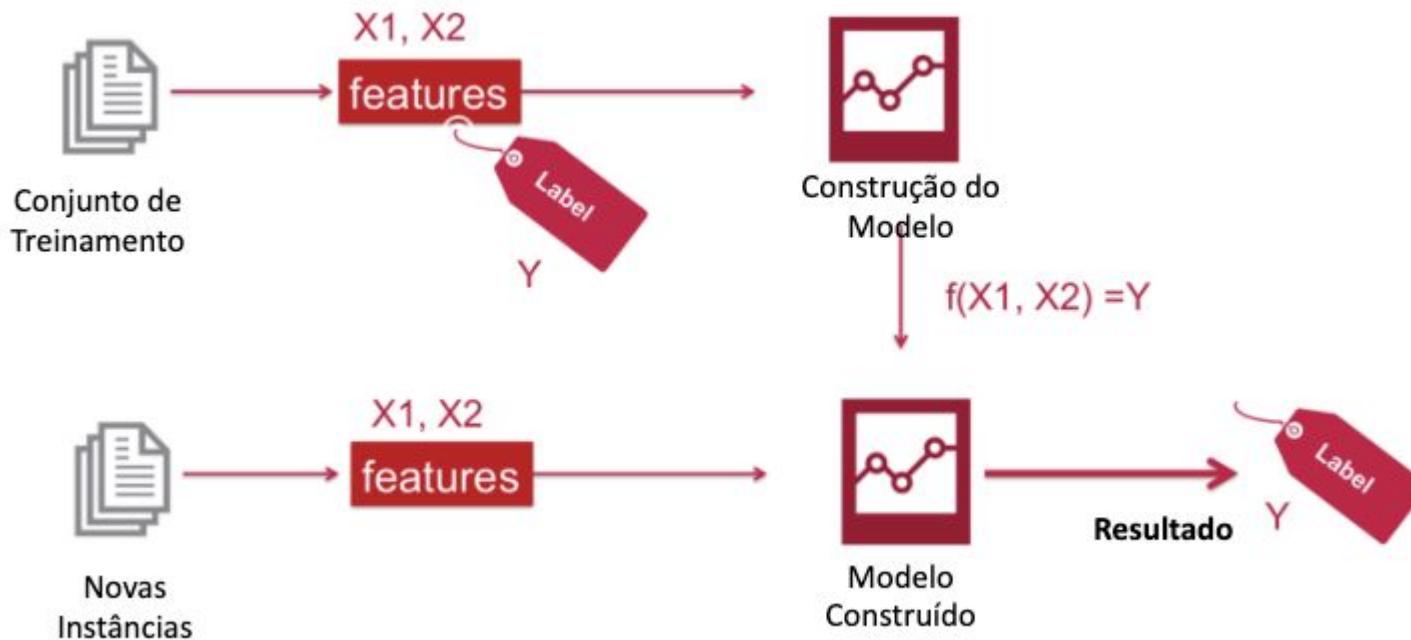
A lista completa está disponível em [github.com/eron93br/carOBD](https://github.com/eron93br/carOBD)



# Como detectar uma anomalia?



## Como então detectar uma anomalia?



Fonte: adaptado de (MAPR Technologies, 2019)



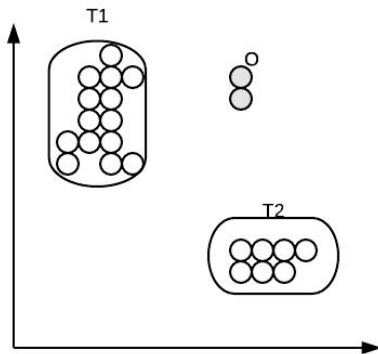
## Alguns conceitos básicos de Aprendizagem de Máquina

Podemos utilizar alguns fundamentos de aprendizado de máquina para a análise de anomalias em sinais veiculares.

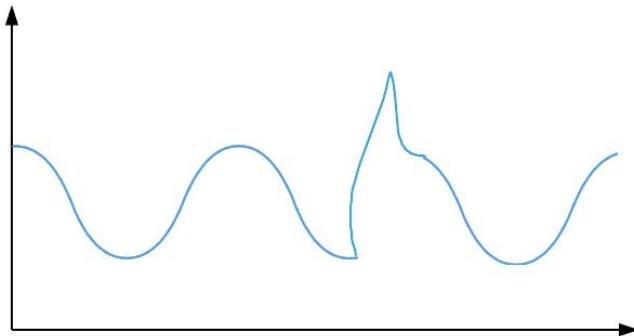
- **Medida de Desempenho P:** porcentagem de instâncias classificadas de maneira correta;
- **Tarefa T:** reconhecer a operação em condição normal de um componente veicular;
- **Experiência de treinamento E:** o processo de aprendizado é realizado a partir de base de dados com informações da rede veicular.



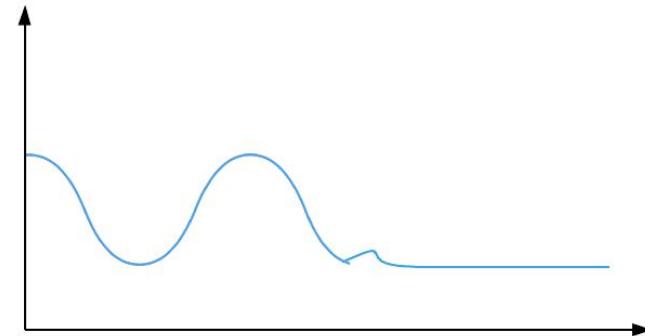
## Tipos de Anomalia



Anomalia Pontual



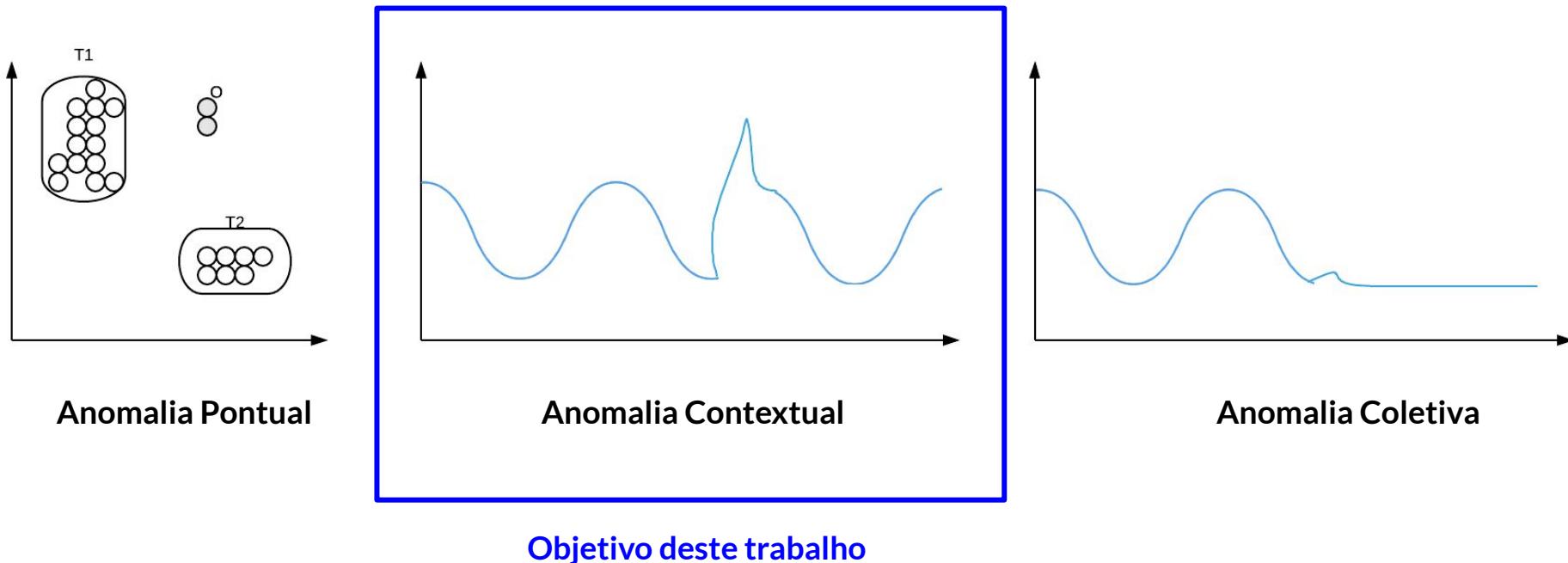
Anomalia Contextual



Anomalia Coletiva

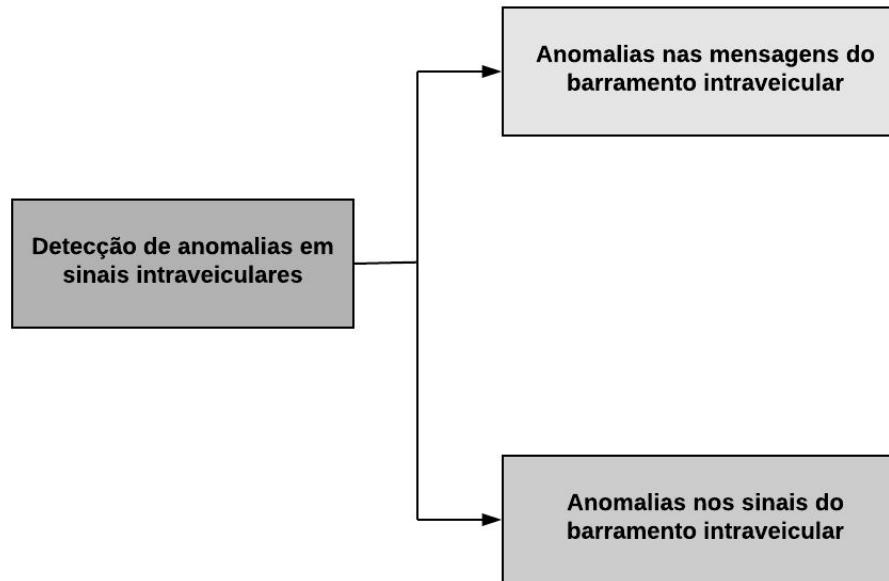


## Tipos de Anomalia



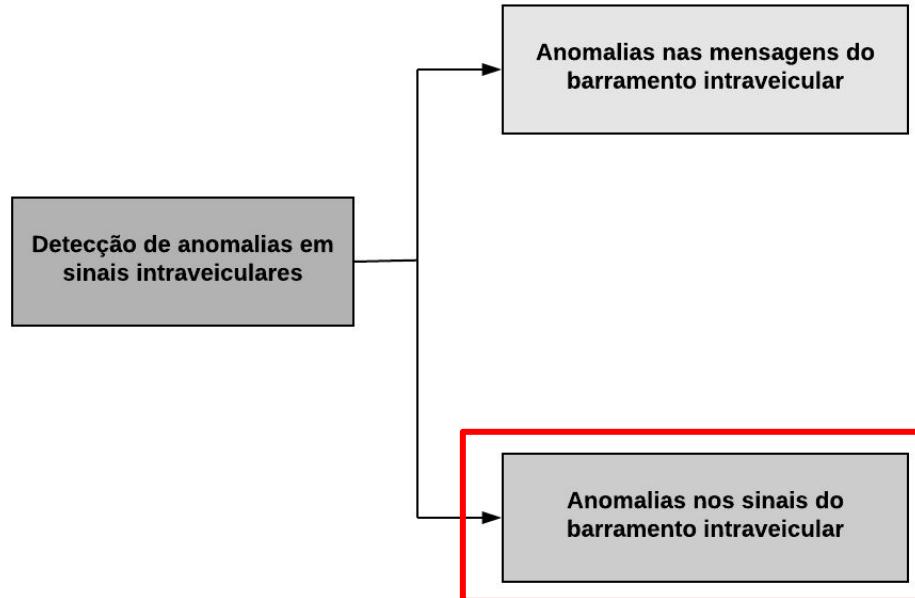


## Como então detectar uma anomalia?





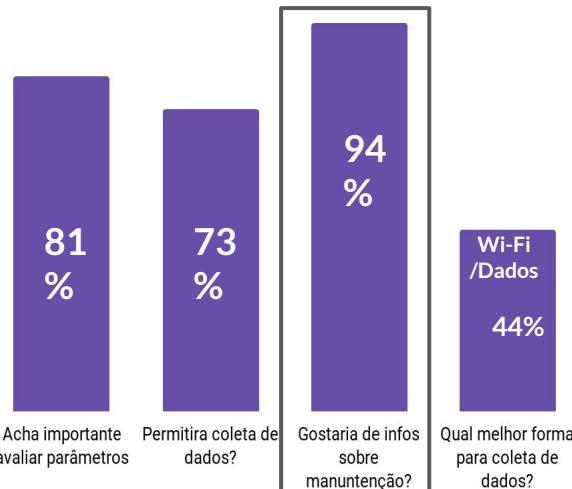
## Como então detectar uma anomalia?





## Qual sensor veicular pode ser monitorado?

Experiência do consumidor ao usar carro conectado

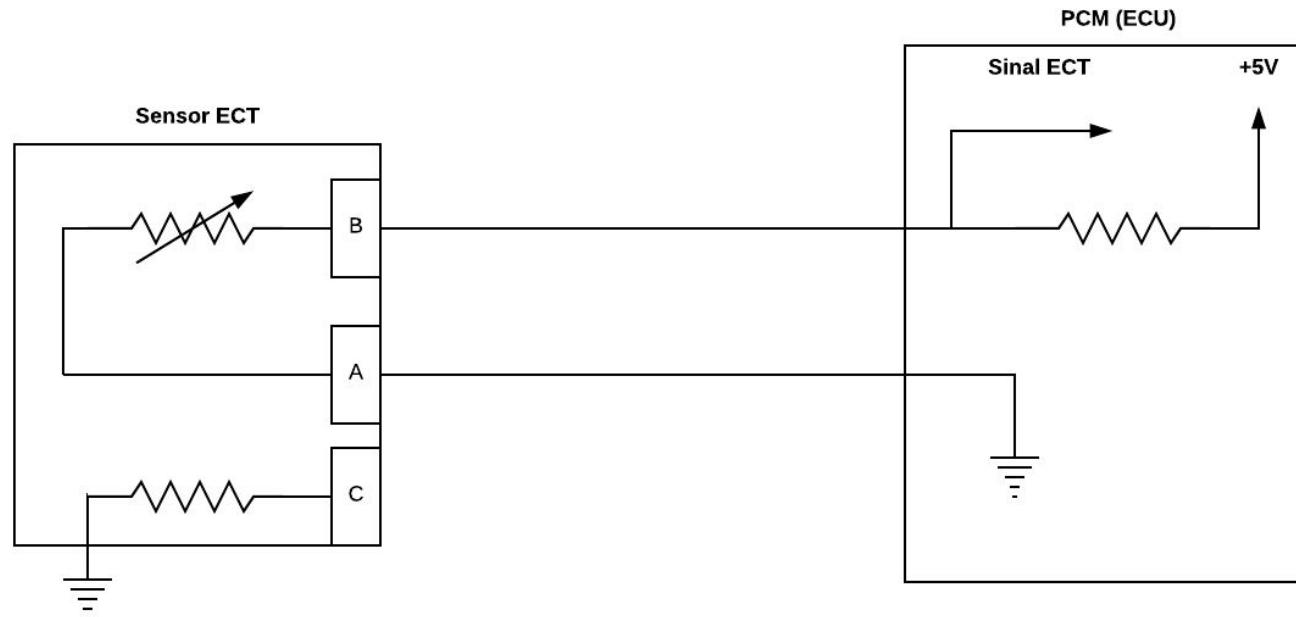


Fonte: Própria





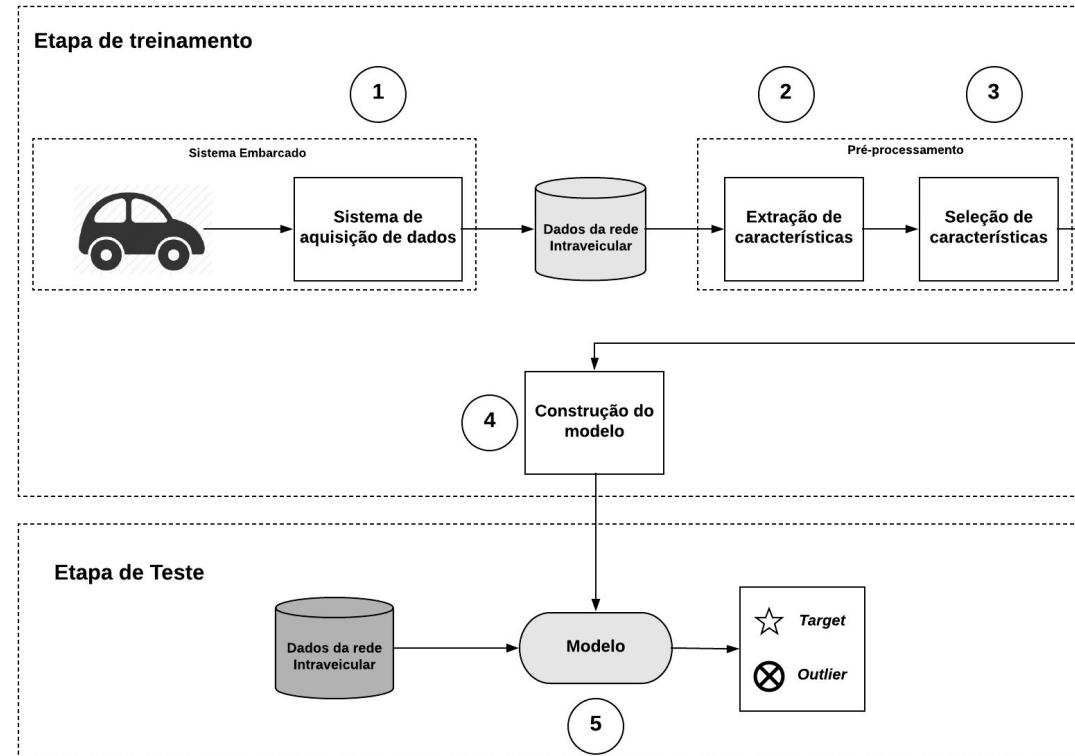
## Sensor de Temperatura do Líquido de Arrefecimento



Engine Coolant Temperature (ECT) Sensor

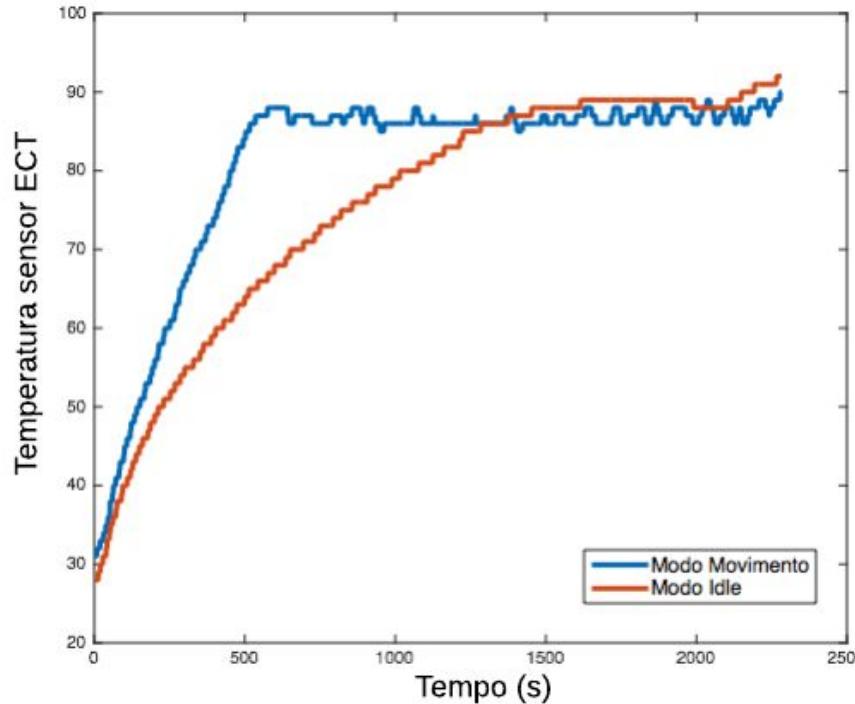


## Visão geral do sistema de detecção de anomalias veiculares



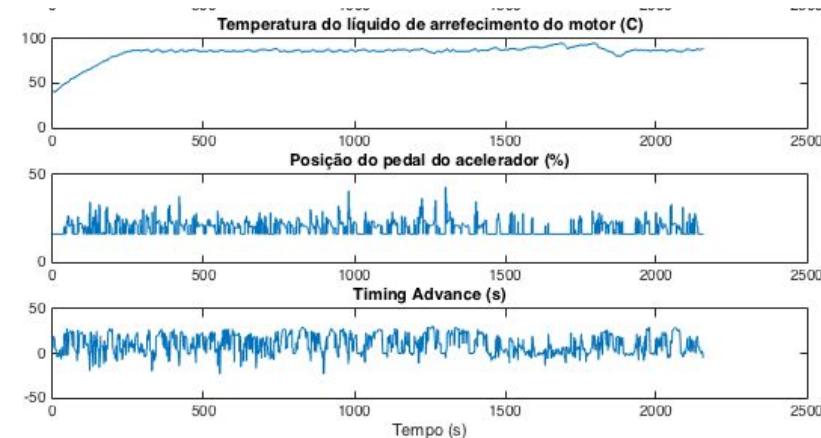
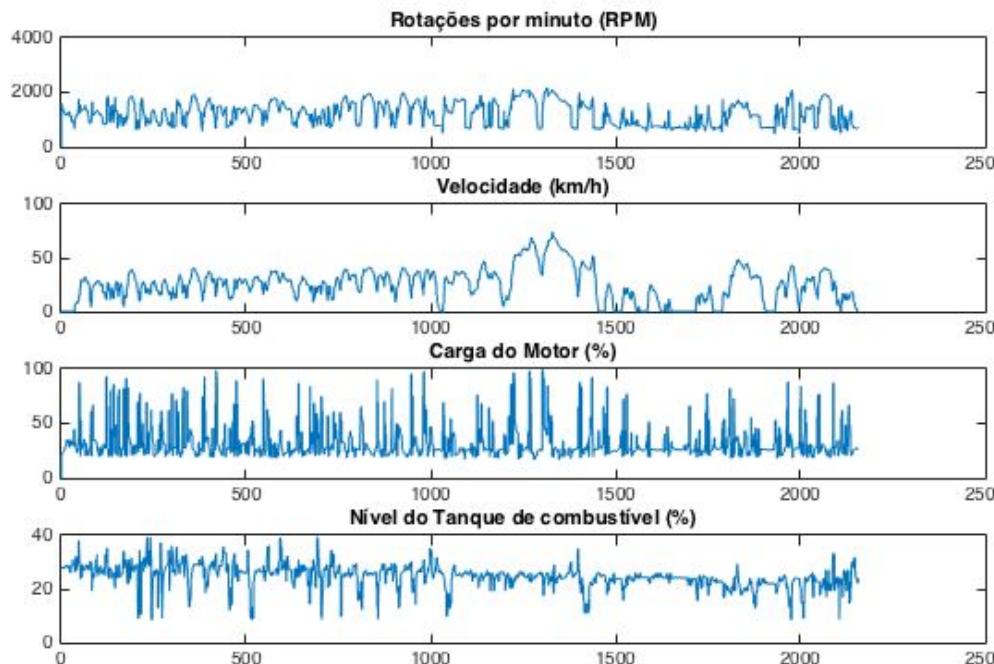


## Sinal do sensor ECT em detalhe





## Dados Capturados e sensor ECT





## Etapa II e III: extração de características

- Como parte do pré-processamento dos dados, todos os atributos são normalizados.

$$y_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

- O sinal de uma *trip* (ciclo veicular) de  $k$  segundos pode ser representado então por:

$$X_{trip_1} = \begin{pmatrix} x_{1,t_1} & x_{2,t_1} \\ x_{1,t_2} & x_{2,t_2} \\ \vdots & \vdots \\ x_{1,t_k} & x_{2,t_k} \end{pmatrix}$$

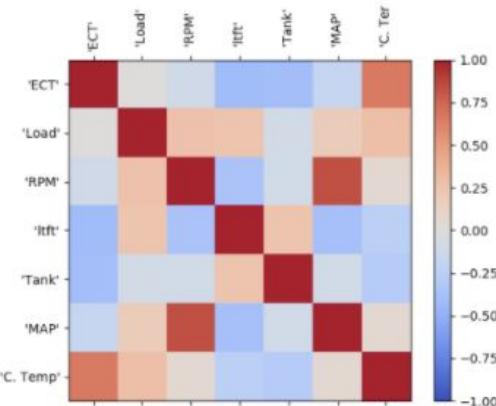
- Torna-se interessante analisar intervalos de operação do sensor. Assim, é usada uma **janela de tamanho  $N$**  para extração da média, variância e desvio padrão de cada atributo:

$$Y_{t_j \dots t_{j+N-1}} = \left( \bar{x}_1 \quad s_{x_1}^2 \quad s_{x_1} \quad \bar{x}_2 \quad s_{x_2}^2 \quad s_{x_2} \right)$$



## Etapa III: seleção de características

- Análise da correlação média destes parâmetros com outros quatro (*Long Term Fuel Trim*, Nível do tanque, *Manifold Absolute Pressure* e Temperatura do Catalisador).

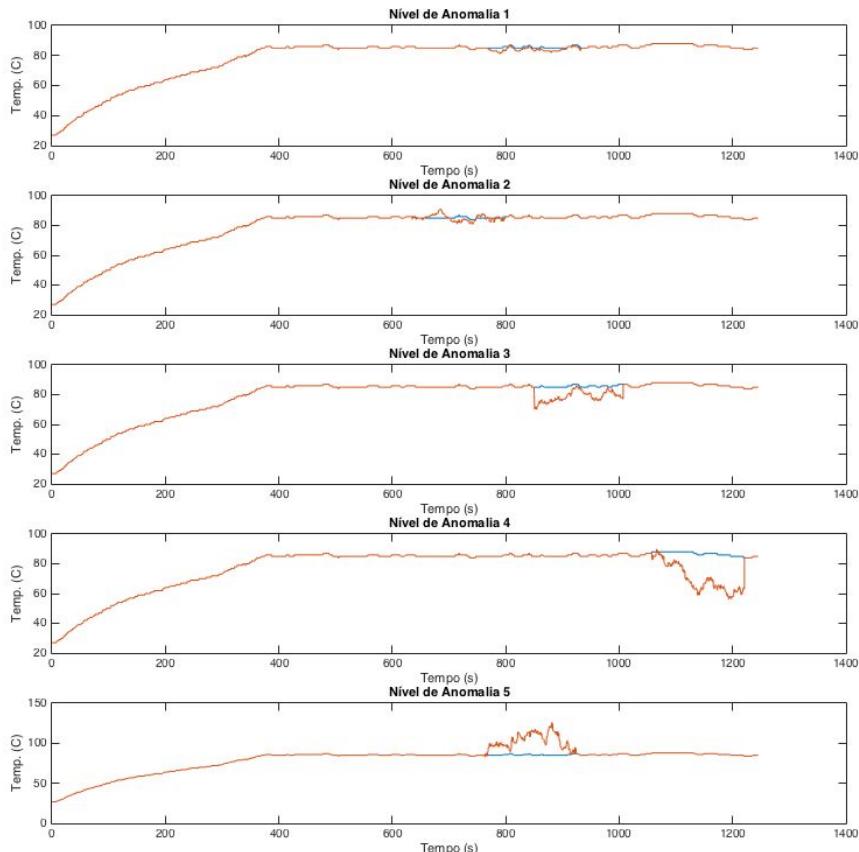


- Após análise dos atributos, é definida a instância de entrada para o sistema:

$$F_i = \begin{pmatrix} \bar{x}_1 & s_{x_1}^2 & s_{x_1} & \bar{x}_2 \end{pmatrix}$$



## Inserção de Anomalias



- Com a coleta de dados do funcionamento normal do veículo, anomalias são inseridas de maneira artificial.
- Inserção a partir de análise do mau funcionamento do sensor, baseado em níveis de ruído presente no sinal.



# Construção do Modelo



## Avaliação dos classificadores One-Class

- Uso da toolbox DDTools (desenvolvida por David Tax) versão 2.1.3 no *software* MATLAB.
- Processo de treinamento apenas com dados do funcionamento normal do veículo.
- Separação de dois grupos ,  $||A||$  e  $||B||$  para treinamento e teste.
- Para fins estatísticos, são treinados e testados 30 diferentes modelos.



## Avaliação dos classificadores One-Class

- Classificador é a técnica utilizada para detectar a anomalia.

- *Gauss data description*
- *k-NN data description*
- *Mahalanobis classifier*
- *Minimum Spanning Tree (MST)*
- *Naive Parzen data description*
- *Self-organized map (SOM)*
- *Parzen data description*
- *Support Vector Data Description (SVDD)*
- *Extreme Value*
- *One Class SVM (OC-SVM)*



## Métricas de performance

- Avaliação de um classificador pode ser realizada a partir de sua matriz de confusão.
- Objetivo de avaliar a detecção de maneira conjunta, tanto *outliers* quanto *target*.
- Uso da métrica F2-score, que considera tanto a precisão quanto a sensibilidade em sua fórmula.

$$sens = \frac{TP}{TP + FN}$$

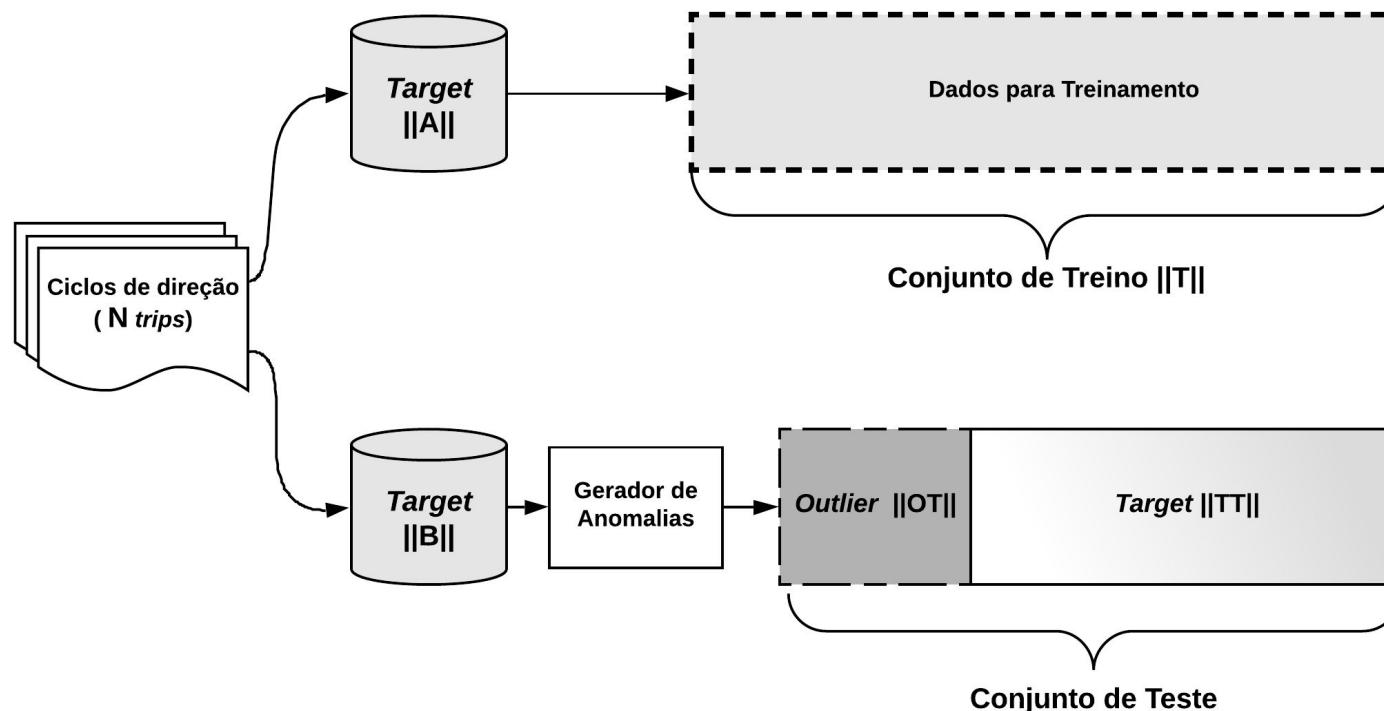
$$prec_{out} = \frac{TN}{TN + FN}$$

$$F_2 = \frac{5 \times prec \times sens}{4 \times prec + sens}$$

		Resultado da classificação	
Rótulo da classe	Normal	Anomalia	
Normal	TP	FN	
Anomalia	FP	TN	



## Divisão dos grupos de treinamento e teste





## Resultados

- O processo de treinamento e teste foram executados trinta vezes e então os resultados analisados:

ANOMALY DETECTION SYSTEM RESULTS FOR THE VEHICLE ON IDLE MODE

	Selected window size	Level I			Level II			Level III		
		TPR	Prec	F2	TPR	Prec	F2	TPR	Prec	F2
Gauss	10	0.920	0.225	0.568	0.935	0.375	0.720	0.935	0.375	0.720
k-NN	5	0.957	<b>0.972</b>	<b>0.960</b>	0.941	<b>0.983</b>	<b>0.949</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
Mahalanobis	10	0.487	0.017	0.008	0.532	0.142	0.343	0.641	0.474	0.598
MST	6	0.687	0.109	0.333	0.738	0.318	0.583	0.869	0.672	0.820
Naive Parzen	6	0.717	0.325	0.577	0.821	0.617	0.770	0.927	0.859	0.912
SOM	7	0.698	0.217	0.483	0.682	0.172	0.428	0.692	0.196	0.459
Parzen	6	0.692	0.278	0.533	0.747	0.462	0.665	0.898	0.813	0.879
SVDD	4	0.718	0.383	0.611	0.814	0.649	0.774	0.832	0.724	0.807
Extreme Value	10	0.669	0.075	0.259	0.749	0.663	0.729	0.803	0.942	0.827
OC-SVM	3	<b>0.972</b>	0.488	0.811	<b>0.972</b>	0.485	0.809	0.977	0.587	0.862

ANOMALY DETECTION SYSTEM RESULTS FOR THE VEHICLE ON MOTION MODE

	Selected window size	Level I			Level II			Level III		
		TPR	Prec	F2	TPR	Prec	F2	TPR	Prec	F2
Gauss	10	0.654	0.189	0.438	0.670	0.246	0.498	0.799	0.616	0.754
k-NN	6	0.939	0.873	0.925	0.927	0.847	0.909	0.934	0.868	0.920
Mahalanobis	6	0.685	0.005	0.024	0.717	0.129	0.375	0.823	0.550	0.748
MST	7	0.680	0.109	0.332	0.763	0.318	0.596	0.873	0.718	0.836
Naive Parzen	10	0.722	0.336	0.587	0.848	0.617	0.788	0.941	0.873	0.926
SOM	6	0.639	0.185	0.428	0.672	0.187	0.442	0.653	0.227	0.474
Parzen	10	0.669	0.301	0.537	0.771	0.448	0.673	0.884	0.842	0.875
SVDD	3	0.704	0.379	0.600	0.771	0.659	0.745	0.900	0.724	0.858
Extreme Value	10	0.831	0.071	0.264	0.872	0.319	0.648	0.921	0.816	0.897
OC-SVM	9	<b>0.998</b>	<b>0.957</b>	<b>0.989</b>	<b>0.998</b>	<b>0.957</b>	<b>0.989</b>	<b>0.998</b>	<b>0.943</b>	<b>0.986</b>



# Quais são os próximos passos?



## Detecção em tempo real no embarcado?

O'REILLY®

# TinyML

Machine Learning with TensorFlow Lite on  
Arduino and Ultra-Low-Power Microcontrollers



Pete Warden &  
Daniel Situnayake



## Quer saber mais?

- Versão completa do trabalho: <https://bit.ly/3c1adVM>
- Artigo publicado no 2019 IEEE 90th Vehicular Technology Conference <https://bit.ly/2WWprHy>
- Base dados carOBD (utilizada no trabalho) - [github.com/eron93br/carOBD](https://github.com/eron93br/carOBD)



## Sobre mim

- LinkedIn: Eronides da Silva Neto
- GitHub: [github.com/eron93br](https://github.com/eron93br)
- Base dados carOBD (utilizada no trabalho) - [github.com/eron93br/carOBD](https://github.com/eron93br/carOBD)

# Seminário de Sistemas

## Embarcados e IoT **2020**

**OBRIGADO!**



Apoio

