UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

André Leonardo Zipf

Aquisição e classificação de sinais vibratórios e sonoros em sistemas de segurança automotiva

Blumenau
2023

André Leonardo Zipf

Aquisição e classificação de sinais vibratórios e sonoros em sistemas de segurança automotiva

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.
Orientador: Prof. Ciro André Pitz, Dr.

Blumenau
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Zipf, André Leonardo
Aquisição e classificação de sinais vibratórios e sonoros em sistemas de segurança automotiva / André Leonardo Zipf ; orientador, Ciro André Pitz, 2024.
57 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Blumenau, Graduação em Engenharia de Controle e Automação, Blumenau, 2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Algoritmo de classificação. 3. Aquisição de sinais. 4. Segurança automotiva. 5. Sensor de vibração. I. Pitz, Ciro André. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

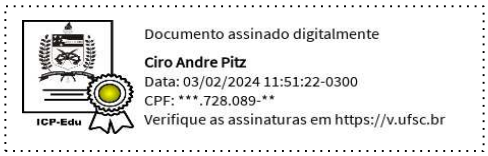
André Leonardo Zipf

Aquisição e classificação de sinais vibratórios e sonoros em sistemas de segurança automotiva

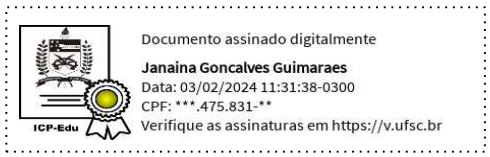
Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Blumenau, 31 de janeiro de 2024.

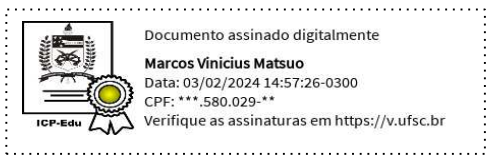
Banca examinadora



Prof. Ciro André Pitz, Dr.
Universidade Federal de Santa Catarina



Profa. Janaina Gonçalves Guimarães, Dra.
Universidade Federal de Santa Catarina



Prof. Marcos Vinicius Matsuo, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus amados pais, Adila e Adilson, ao meu estimado irmão Thomas e ao meu amor Letícia.

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais por todo apoio incondicional durante toda a minha trajetória acadêmica. Sem eles eu não teria chegado aonde cheguei e me tornar a pessoa que sou hoje. São os exemplos que sigo e sempre seguirei. Agradeço também ao meu irmão e minha namorada por todo apoio ao longo desses anos.

Agradeço, em especial, a Universidade Federal de Santa Catarina por proporcionar um ambiente acadêmico excepcional, repleto de profissionais competentes e solícitos. Sou grato por poder fazer parte desta instituição que me inspira e já me abriu muitos caminhos.

Por fim, sou imensamente grato ao meu orientador, Prof. Dr. Ciro André Pitz. Sua dedicação, paciência e conhecimentos foram essenciais ao longo de toda a elaboração deste trabalho. Agradeço por todas as suas valiosas orientações, que não contribuiu apenas para este trabalho, mas também para o meu crescimento como estudante e pessoa.

RESUMO

Com o constante aumento da produção de veículos, há também o aumento do número de furtos e roubos. Atrelado a isso, verifica-se uma deficiência na detecção de eventos associados à segurança do veículo e na comunicação com o usuário por parte dos sistemas de segurança automotivos convencionais. Tendo em vista o exposto, o presente trabalho de conclusão de curso tem como objetivo desenvolver um sistema de baixo custo capaz de captar sinais vibratórios e sonoros, identificando cenários danosos para o veículo e reportando para o usuário através de mensagens SMS (*short message service*) transmitidas na rede de uma operadora de comunicação celular. Para isso, a primeira etapa deste trabalho consiste no desenvolvimento de um banco de dados referentes à batida na lataria, tentativa de furto, riscado na lataria e sons externos de intensidade alta e regular próximos ao veículo. Em seguida, a partir do banco de dados obtido, foi desenvolvido um algoritmo classificador de cenários. Por fim, o algoritmo proposto e o sistema de comunicação foram implementados em Arduino, resultando em um protótipo de baixo custo que alerta o usuário sobre a ocorrência de eventos relacionados à segurança do veículo. O desempenho do algoritmo de classificação foi avaliado a partir das métricas de acurácia, precisão e revocação, sendo obtidos valores superiores a 97%, 80% e 77,8%, respectivamente. Em relação ao sistema de comunicação, foi constatado que os cenários identificados foram reportados adequadamente ao usuário com um intervalo de resposta de aproximadamente 12s.

Palavras-chave: Algoritmo de classificação; Aquisição de sinais; Segurança automotiva; Sensor de som; Sensor de vibração.

ABSTRACT

With the constant increase in vehicle production, there's also a rise in the number of thefts and robberies. Linked to this, it turns out a deficiency in detecting events associated with vehicle safety and in the communication with the user by conventional automotive security systems. In view of the above, the present end of course work aims to develop a low-cost system capable of capturing vibrational and sound signals, identifying harmful scenarios to the vehicle and reporting them to the user by SMS message (*short message service*) transmitted over a cellular communication network. For this, the first step of this work consists of developing a database related to hit on the vehicle's body, attempted theft, scratches on the vehicle's body and sounds with high and regular intensity next to the vehicle. Afterwards, based on the acquired database, a scenarios classifier algorithm was developed. Finally, the proposed algorithm and the communication system were implemented using Arduino, resulting in a low-cost prototype that warns the user about the events related to the vehicle security. The performance of the classification algorithm was evaluated by the accuracy, precision, and recall metrics, with values higher than 97%, 80% e 77.8%, respectively. Regarding the communication system, it was noted that the identified scenarios were appropriately reported to the user with a response interval of approximately 12 seconds.

Keywords: Classifier algorithm; Signals acquisition; Automotive security; Sound sensor; Vibration sensor.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A fabricação em massa de carros populares começou a partir do modelo de produção proposto por Henry Ford, fazendo com que a disseminação de carros ocorresse, levando a um aumento significativo no número de veículos mundiais. Assim como informa Souza (2017), “um processo que revolucionou a fabricação de automóveis, e, graças à montagem em série, em 1925, um novo Ford ficava pronto a cada 15 segundos”.

Na atualidade, qualquer veículo automotor já se tornou algo essencial de se utilizar, devido à necessidade de deslocamento em grandes distâncias com o menor tempo possível. Atrelado a isso, levando em consideração o aumento da população mundial, fica claro o aumento da demanda por veículos. Segundo Albuquerque (2023), no Brasil, comparando o primeiro semestre de 2022 com o mesmo semestre de 2023, houve um aumento de 8,8% do número de vendas de veículos, alcançando 998,6 mil unidades.

Na mesma direção em que se aumenta a venda de veículos, aumenta-se também o número de furtos e roubos. Segundo o Anuário Brasileiro de Segurança Pública (2022), em 2022, o número de veículos que foram furtados ou roubados chegou a margem de 373.225, com um crescimento de cerca de 8% com relação ao ano 2021.

Com o objetivo em minimizar danos ao bem material, algumas empresas como a Tesla vêm focando no uso de novas tecnologias para segurança em seus veículos, desenvolvendo sistemas para monitoramento e alerta em casos críticos. Por exemplo, o sistema sentinela, presente nos automóveis da marca Tesla, é um sistema de segurança moderno capaz de identificar e alertar casos de ameaças à integridade do veículo (Tesla, 2023). Esse sistema conta com um conjunto de câmeras que consegue gravar os acontecimentos e identificar situações críticas, fornecendo assim um alto grau de segurança para o usuário que dispõe de um bem com alto valor agregado.

A implementação de sistemas de segurança sofisticados, tal como o citado anteriormente, tem um custo muito elevado quando considerado os veículos populares fabricado em países emergentes. No Brasil existem alguns sistemas de segurança disponíveis no mercado, porém, menos tecnológicos e com um custo mais barato do que o modelo utilizado pela Tesla. Dentre os sistemas de segurança

disponíveis no mercado, destacam-se o alarme convencional, travas para volante, localizadores e chaves codificadas.

Segundo Alves (2023), os principais tipos de alarmes convencionais são o perimétrico, ideal para detectar tentativas de violação de alguma parte de possível entrada do veículo, e o volumétrico, que através de sensores de movimento interno detecta presença no interior do veículo. Ambos os métodos podem ser utilizados para verificar possíveis tentativas de furto. Entretanto, tais métodos não reconhecem vandalismo externo ao veículo e também não permitem uma identificação antecipada ao possível furto.

Nesse contexto, fica evidente a necessidade de sistemas mais completos e baratos para segurança veicular. Atualmente, existem sensores que podem auxiliar no desenvolvimento desses sistemas, tal como os sensores piezoelétricos, para detecção de vibração, e sensores sonoros. Além do mais, uma vantagem da utilização dos sensores mencionados, é que podem ser utilizados em sistemas embarcados de baixo custo, como é o caso do Arduino.

Outro ponto importante a se destacar é a falta de informação em tempo real dos sistemas de segurança veiculares convencionais. Nesse aspecto, há equipamentos, aplicáveis em sistemas embarcados de baixo custo, que são capazes de enviar informação para o usuário através de redes de comunicação móvel.

Com base nas informações apresentadas anteriormente, este trabalho de conclusão de curso tem como objetivo desenvolver um sistema de baixo custo para identificação de possíveis danos e furtos ao veículo. O novo sistema deve ser capaz de alertar os usuários em tempo real utilizando a tecnologia de comunicação móvel.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo deste trabalho é desenvolver um sistema automatizado e de baixo custo, capaz de identificar e alertar o usuário sobre eventos que possam colocar a integridade do veículo em risco.

1.1.2 Objetivos específicos

Os objetivos específicos do trabalho são listados a seguir:

1. Gerar um banco de dados com sinais provenientes de sensores de vibração (vibração mecânica) e microfone (vibração sonora) instalados em uma placa metálica que simula a lataria do veículo;
2. Propor um algoritmo para identificação e classificação de eventos baseado em amostras provenientes dos sensores;
3. Implementar um sistema de alerta ao usuário baseado na comunicação por mensagens de texto, SMS (*short message service*);
4. Desenvolver um protótipo para validação do sistema proposto.

1.2 ORGANIZAÇÃO DO DOCUMENTO

O presente documento está organizado conforme descrito a seguir. O segundo capítulo contempla os materiais utilizados e os métodos aplicados para se alcançar os objetivos do trabalho. No terceiro capítulo é descrito o desenvolvimento do sistema proposto. O quarto capítulo explicita os resultados obtidos com o trabalho. Por fim, o quinto capítulo apresenta as conclusões deste trabalho.

2 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais utilizados na confecção do protótipo, juntamente com a metodologia para se alcançar os objetivos propostos.

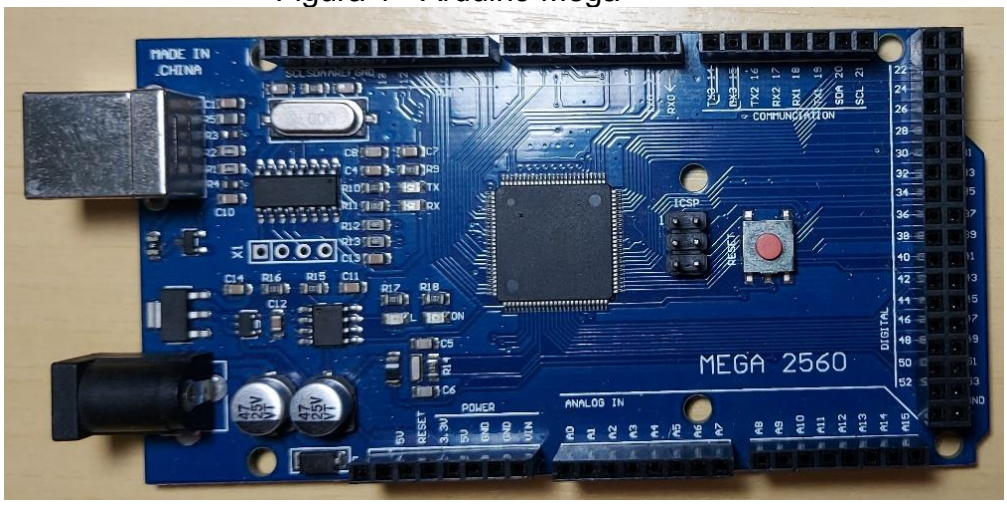
2.1 MATERIAIS

Como o objetivo do trabalho proposto é o desenvolvimento de um sistema funcional, que reconheça eventos críticos e alerte o usuário, foi necessário o desenvolvimento de uma plataforma de testes para validação do sistema desenvolvido. Para isso, utilizou-se o Arduino Mega, sensores piezoelétricos, sensor sonoro, módulo para comunicação via tecnologia GSM (*global system for mobile communications*) e tabuleiro de forno para simular a estrutura da funilaria.

2.1.1 Arduino Mega

Para a aquisição dos dados dos sensores e controle do sistema, utilizou-se um Arduino Mega, conforme pode ser visto na Figura 1, que possui um microcontrolador ATmega2560 de 8 bits. O sistema embarcado possui uma frequência de amostragem do conversor A/D de até 125 kHz, uma memória flash de 256 kB, 54 portas digitais e 16 portas analógicas (MICROCHIP, 2023).

Figura 1 - Arduino Mega



Fonte: O autor.

2.1.2 Sensores piezoelétricos

Utilizou-se quatro sensores piezoelétricos para a identificação de sinais de vibração na lataria do veículo. Devido ao princípio piezoelétrico, a tensão gerada (V) nos terminais de um cristal piezoelétrico é a razão entre o deslocamento de cargas (x) e uma constante de sensibilidade de carga (D), além do mais, o deslocamento de cargas é proporcional a razão entre a força aplicada (F) e uma constante de rigidez do cristal piezoelétrico (k) (BENTLEY, 2005), isto é,

$$V = \frac{x}{D} \tag{1}$$

e

$$x = \frac{F}{k}. \tag{2}$$

Abaixo, na Figura 2, segue uma ilustração de um sensor piezoelétrico com cristal de quartzo.

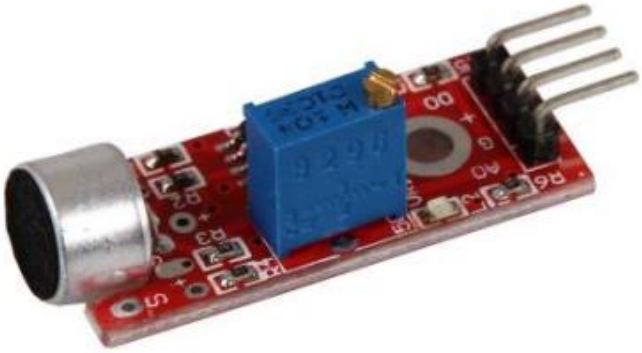
Figura 2 – Sensor piezoelétrico

Fonte: (ELETROGATE, 2024).

2.1.3 Sensor sonoro

Para a medição de sinais sonoros, utilizou-se um módulo sensor de microfone KY-037 (ELETROGATE, 2023). Esse sensor também utiliza do princípio piezoelétrico, convertendo vibrações sonoras em sinais de tensão. É constituído também de um circuito amplificador, amplificando o sinal gerado. O módulo utilizado pode ser visto na Figura 3.

Figura 3 – Módulo sensor KY-037



Fonte: (ELETROGATE, 2023)

2.1.4 Tabuleiro de forno

A estrutura da funilaria de um carro popular, nos dias atuais, é majoritariamente composta por aço ou alumínio sendo revestida com tinta (AUTOMOTIVA, 2023). Devido a isso, optou-se por utilizar o tabuleiro de forno tendo em vista que possui características similares às da estrutura de um carro. No decorrer deste trabalho, o tabuleiro de forno em questão será chamado de “funilaria de ensaios” ou “chapa”. A chapa utilizada pode ser vista na Figura 4.

Figura 4 – Tabuleiro de forno

Fonte: O autor

2.1.5 Módulo de comunicação GSM

Aplicou-se um módulo SIM800L para o envio de informações para o usuário através da tecnologia de comunicação de redes móveis. Sendo assim, o módulo permite enviar mensagens SMS para o usuário. Possui uma tensão de alimentação de 4 V e uma corrente de pico para conexão do módulo à rede de 2 A, sendo necessário uma alimentação por fonte externa (SIMCOM, 2015). O módulo utilizado pode ser visualizado na Figura 5.

Figura 5 – Módulo SIM800L

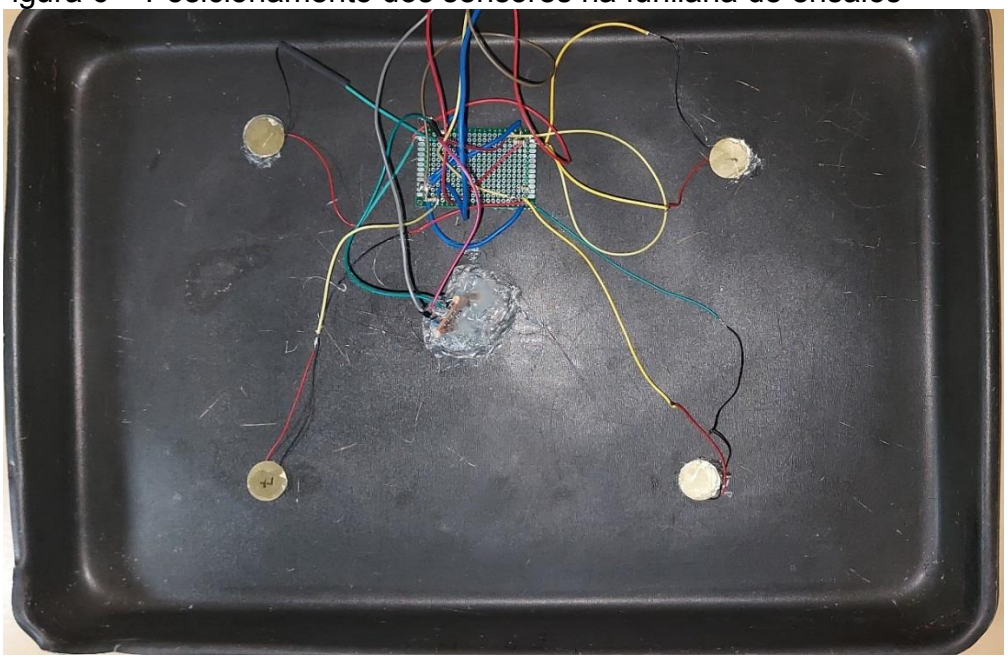
Fonte: O autor

2.2 METODOLOGIA

Com os equipamentos definidos, foram determinadas as melhores posições dos sensores para a aquisição de dados. Com isso, fixou-se os sensores piezoelétricos a uma distância de 12 cm em relação a borda horizontal e 7 cm em relação a borda vertical, devido ao tamanho do tabuleiro de forno e para uma melhor distinção de cada dado. Posicionou-se também o sensor sonoro no centro da placa

para uma melhor detecção de sons, conforme pode ser verificado na fotografia apresentada na Figura 6.

Figura 6 – Posicionamento dos sensores na funilaria de ensaios



Fonte: O autor.

Com os sensores posicionados, definiram-se os seguintes cenários de detecção:

- Batida na funilaria;
- Riscado na funilaria;
- Sons externos de intensidade alta;
- Sons externos de intensidade regular;
- Tentativa de furto.

Cada um desses cenários foi simulado a partir da funilaria de ensaios utilizada neste trabalho. Para facilitar a descrição dos eventos, foi utilizado um sistema de coordenadas, conforme ilustrado na Figura 7.

Figura 7 – Sistema de coordenadas na funilaria de ensaios

Fonte: O autor.

Com o intuito de desenvolver um banco de dados com as informações de cada cenário gerado, utilizou-se de interações entre as plataformas de programação do Arduino e do MATLAB, desenvolvendo um algoritmo para captar dados via porta serial e armazená-los em arquivos no formato MATLAB *Data* (MATHWORKS, 2023). A seguir serão apresentadas as metodologias utilizadas para a execução de cada cenário.

2.2.1 Batida

Na execução deste cenário, soltou-se uma caixa quadrada de massa definida igual a 0,09 kg, de uma altura fixa de 0,3 m. Com isso, obteve-se uma força de impacto de aproximadamente 15 N. A caixa utilizada nos testes pode ser visualizada na Figura 8. Foram realizadas batidas nos cinco pontos apresentados na Figura 7.

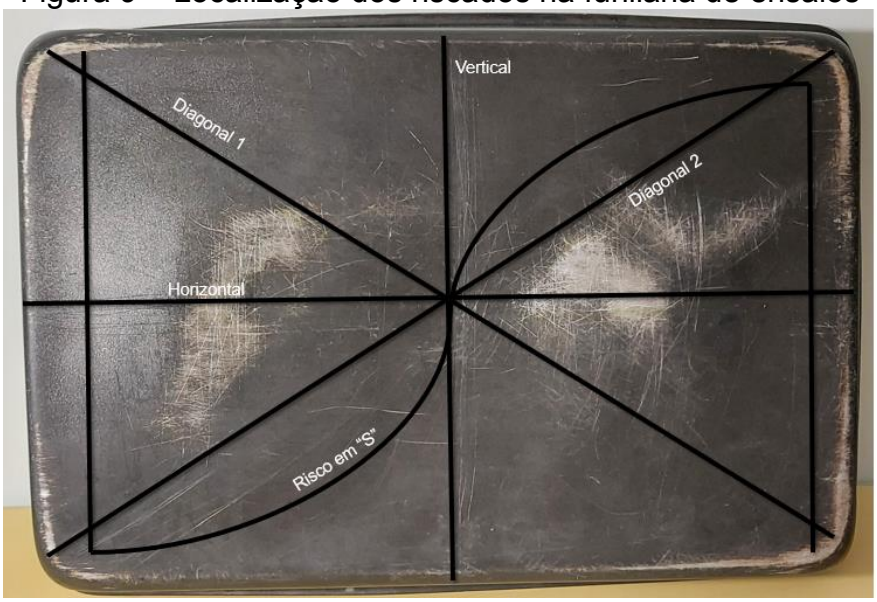
Figura 8 – Caixa utilizada na geração da batida

Fonte: O autor.

2.2.2 Riscado

Neste cenário, definiram-se algumas características especiais para a sua realização, devido a ser um caso em que o ponto de aplicação da força não permanece estático em uma posição. Para gerar este cenário, utilizou-se uma faca de cozinha serrilhada para a aplicação de força contra a funilaria de ensaios. Os casos gerados foram um riscado na horizontal, na vertical, em ambas diagonais e em um formato de “S”, assim como representado na Figura 9.

Figura 9 – Localização dos riscados na funilaria de ensaios



Fonte: O autor.

2.2.3 Sons externos de intensidade alta

Para executar este cenário, utilizou-se uma caixa de som da marca JBL modelo *Extreme 2* como fonte sonora externa (JBL, 2024), e, utilizando um *software* decibelímetro para celular, manteve-se a mesma intensidade de som, em 105 dB, para todos os casos. Os casos gerados para este cenário foram com a fonte sonora a uma distância da chapa de 0,3 m e 1 m.

2.2.4 Sons externos de intensidade regular

Neste cenário, aplicou-se a mesma metodologia do cenário anterior, apenas reduzindo a intensidade sonora da fonte externa. Os casos foram desenvolvidos com uma intensidade sonora de 85 dB e com as mesmas distâncias do cenário anterior.

2.2.5 Tentativa de furto

Para realizar este cenário, levou-se em consideração que em uma tentativa de furto, geralmente, necessita algum tipo de arrombamento, ou seja, é necessário aplicar grande pressão em alguma parte da estrutura do veículo. Sendo assim, aplicou-se pressão na funilaria de ensaios com a mão nos pontos ilustrados na Figura 7, com um intervalo de aproximadamente meio segundo.

2.2.6 Desenvolvimento do algoritmo de classificador

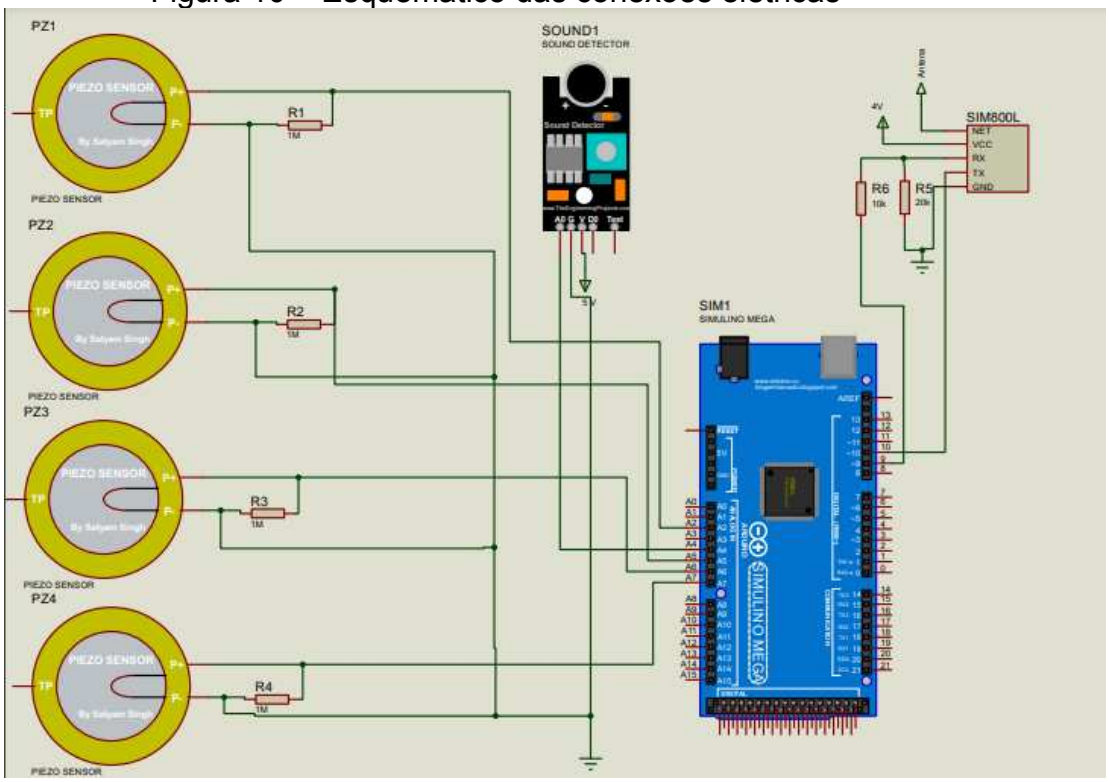
Com o banco de dados adquirido, desenvolveu-se, através do *software* MATLAB, um algoritmo para classificar cada cenário considerado. Após a validação do algoritmo no MATLAB, através do desenvolvimento de testes de classificação, implementou-se o algoritmo proposto no Arduino. Por fim, verificou-se a assertividade do algoritmo com relação ao banco de dados, adquirido previamente ao algoritmo de classificação, e com relação a novos dados, gerados após o desenvolvimento do algoritmo de classificação, através de uma matriz de confusão.

3 DESENVOLVIMENTO

3.1 ESTRUTURA FÍSICA

Para o protótipo desenvolvido, conectou-se os sensores piezoelétricos e o sensor sonoro em portas analógicas. Para a conexão dos sensores piezoelétricos, necessitou-se adicionar um resistor de 1 MΩ para o controle de carga e descarga do sensor. Como o módulo SIM800L possui um microcontrolador integrado, a comunicação entre o módulo e o Arduino se deu através de portas seriais. Neste caso, o Arduino estava como mestre e o SIM800L como escravo. Para a alimentação correta do módulo, utilizou-se um conversor *buck* regulável, mantendo-se fixo os 4 V de alimentação do módulo. Além disso, para a comunicação correta entre o microcontrolador e o módulo SIM800L, utilizou-se um divisor de tensão. Abaixo, na Figura 10, é apresentado o esquemático das conexões elétricas.

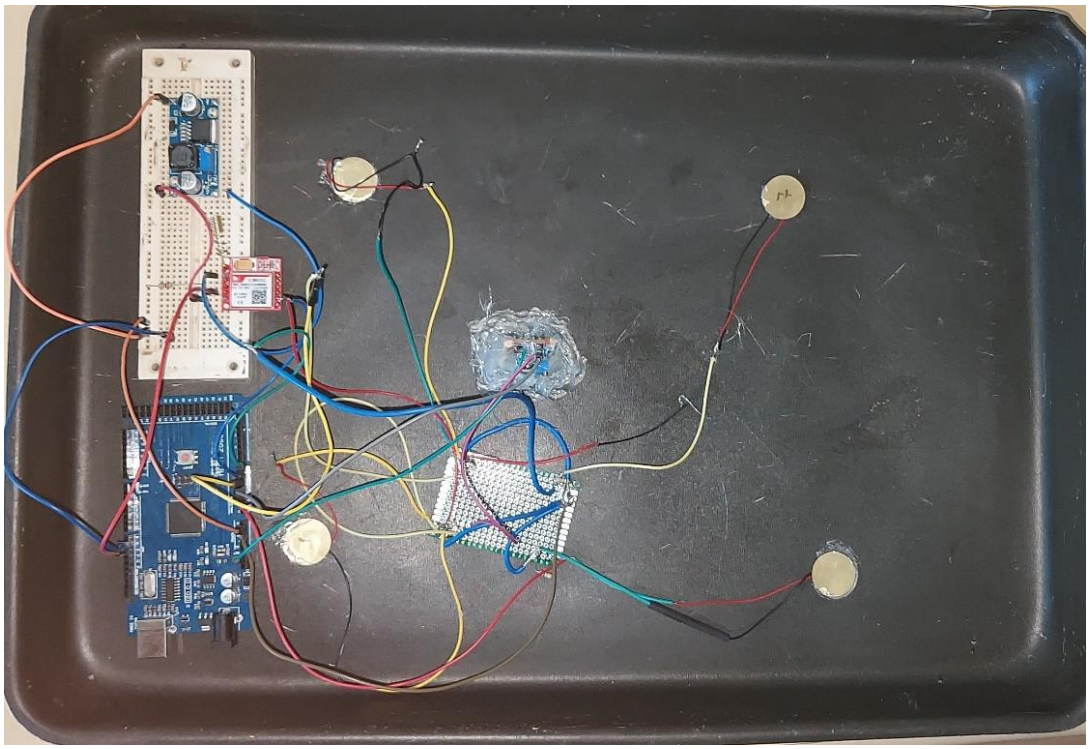
Figura 10 – Esquemático das conexões elétricas



Fonte: O autor.

Com as conexões definidas, implementou-se o sistema na prática fixando os sensores piezoelétricos e o sensor sonoro, com cola quente, na funilaria de ensaios, conforme apresentado na Figura 11.

Figura 11 – Interface física da funilaria de ensaios



Fonte: O autor.

3.2 LÓGICA DE PROGRAMAÇÃO

Na presente seção são apresentadas as lógicas de programação desenvolvidas, acompanhadas das devidas explicações dos algoritmos.

3.2.1 Aquisição do banco de dados

3.2.1.1 Algoritmo para aquisição de dados

Desenvolveu-se um algoritmo na plataforma Arduino para a aquisição e transferência de dados em tempo real pela porta serial. Em um primeiro momento, calibra-se o sensor sonoro para a referência em zero, devido que em alguns momentos pode possuir um baixo ruído. Além disso, é enviado também um separador

de informações para a porta serial, que nesse caso é a vírgula, auxiliando na distinção da informação de cada sensor. O código descrito pode ser verificado no Quadro 1.

Quadro 1: Leitura e envio de dados via porta serial pelo Arduino

```
if (ciclo == 0) {
    som = analogRead(sensorsom);
    calibra_som = som;
    ciclo ++;
  }

sensorvalorpiezo1 = analogRead(piezoPin1);
sensorvalorpiezo2 = analogRead(piezoPin2);
sensorvalorpiezo3 = analogRead(piezoPin3);
sensorvalorpiezo4 = analogRead(piezoPin4);
som = analogRead(sensorsom);
sensorsom_calibrado =som-calibra_som;
valorAbsoluto = abs(sensorsom_calibrado);

Serial.print(sensorvalorpiezo1);
Serial.print(",");
Serial.print(sensorvalorpiezo2);
Serial.print(",");
Serial.print(sensorvalorpiezo3);
Serial.print(",");
Serial.print(sensorvalorpiezo4);
Serial.print(",");
Serial.println(valorAbsoluto);
Serial.print(",");
```

Fonte: O autor.

Com as informações disponíveis na porta serial, desenvolveu-se um código em MATLAB para ler e armazenar estes dados. Com isso, utilizou-se uma biblioteca de suporte para a plataforma Arduino, disponível no MATLAB, para auxiliar nessa aquisição (MATHWORKS, 2024). Para inicializar o processo de aquisição dos dados no MATLAB, identificou-se a porta serial que seriam transferidos os dados, criou-se

um arquivo para armazenar as informações adquiridas e gravou-se, no arquivo criado, os dados lidos, de acordo com o Quadro 2.

Quadro 2: Identificação da porta serial, criação do arquivo e armazenamento de dados

```
serialPort = serial('COM5', 'BaudRate', 9600);

fopen(serialPort);

[arquivoNome, caminho] = uiputfile('*.mat', 'Salvar o arquivo');

nomeArquivo = fullfile(caminho, arquivoNome);

arquivo = fopen(nomeArquivo, 'w');
```

Fonte: O autor.

Define-se então um intervalo para a aquisição dos dados e cria-se uma variável vetor com o número de colunas de acordo com o número de sensores do sistema. Com isso, implementou-se um laço de repetição *for*, que realiza a leitura das informações da porta serial, e aplicou-se uma lógica de segmentação em partes da informação adquirida, onde cada parte corresponde a informação de um sensor. A lógica descrita pode ser observada no Quadro 3.

Quadro 3: Lógica de leitura e segmentação de informações

(continua)

```
numLeituras = 1500;
batida_lateral_esquerda = zeros(numLeituras, 5);
t=0;
for i = 1:numLeituras
    if t==0
    linha = fgetl(serialPort);
    disp(linha);
    end
```

Quadro 3: Lógica de leitura e segmentação de informações

(conclusão)

```
    if t>0
    linha = fgetl(serialPort);

    disp(linha);
    partes = strsplit(linha, ',');
    batida_lateral_esquerda(i, 1) = str2double(partes{2});
    batida_lateral_esquerda(i, 2) = str2double(partes{3});
    batida_lateral_esquerda(i, 3) = str2double(partes{4});
    batida_lateral_esquerda(i, 4) = str2double(partes{5});
    batida_lateral_esquerda(i, 5) = str2double(partes{6});

    end
    t=t+1;
end
```

Fonte: O autor.

Após finalizar a amostragem definida, salva-se o arquivo criado, juntamente com os dados adquiridos, e libera-se a porta serial utilizada, conforme apresentado no Quadro 4.

Quadro 4: Finalização da aquisição

```
save(nomeArquivo, 'batida_lateral_esquerda');

fclose(arquivo);
fclose(serialPort);
delete(serialPort);
```

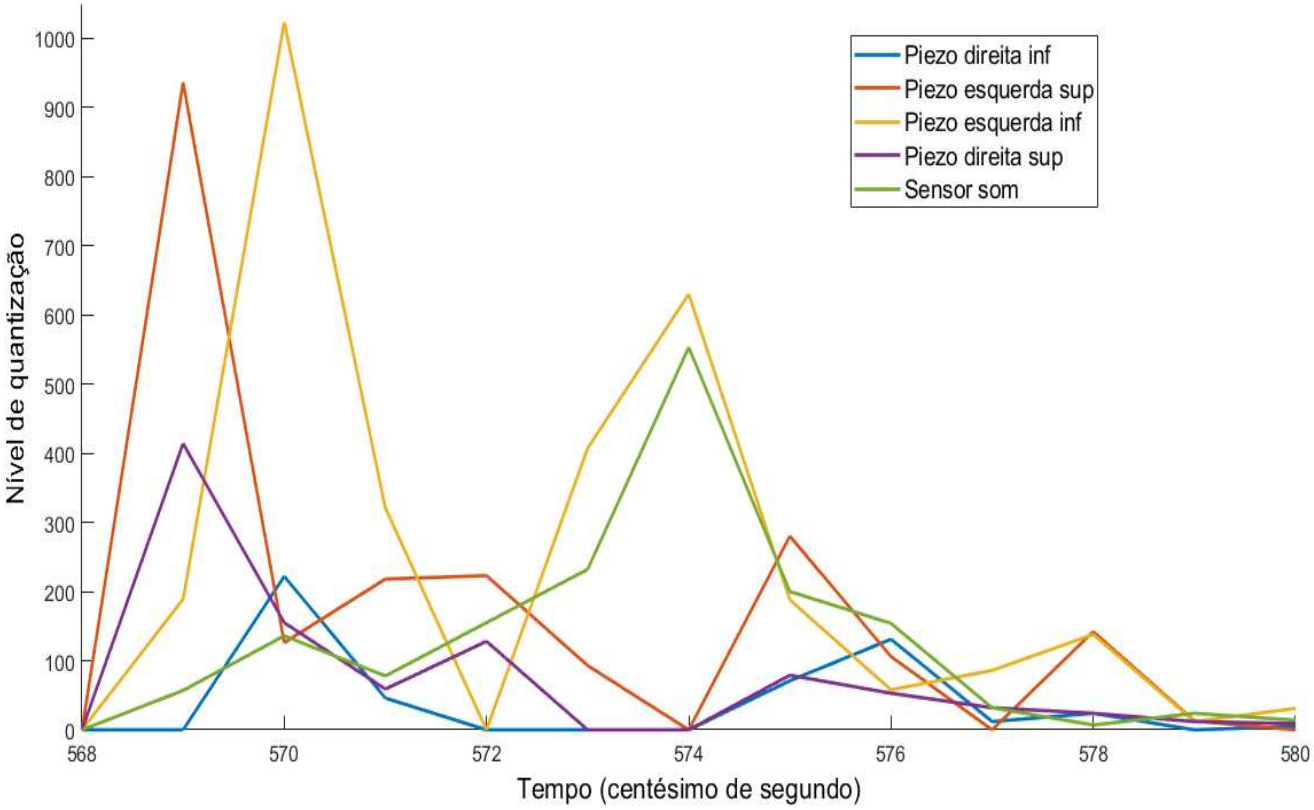
Fonte: O autor.

Realizou-se este processo para todos os cenários considerados, modificando apenas o nome do arquivo e o nome da variável vetor de acordo com o cenário gerado.

3.2.1.2 Banco de dados

Com o banco de dados totalmente adquirido de acordo com a metodologia adotada, pode-se analisar os resultados e verificar as características específicas presentes em cada cenário. Primeiramente, analisou-se os dados adquirido para o cenário de batida na parte central da lataria. O gráfico obtido pode ser visualizado na Figura 12.

Figura 12 – Curvas referentes à batida na parte central da lataria

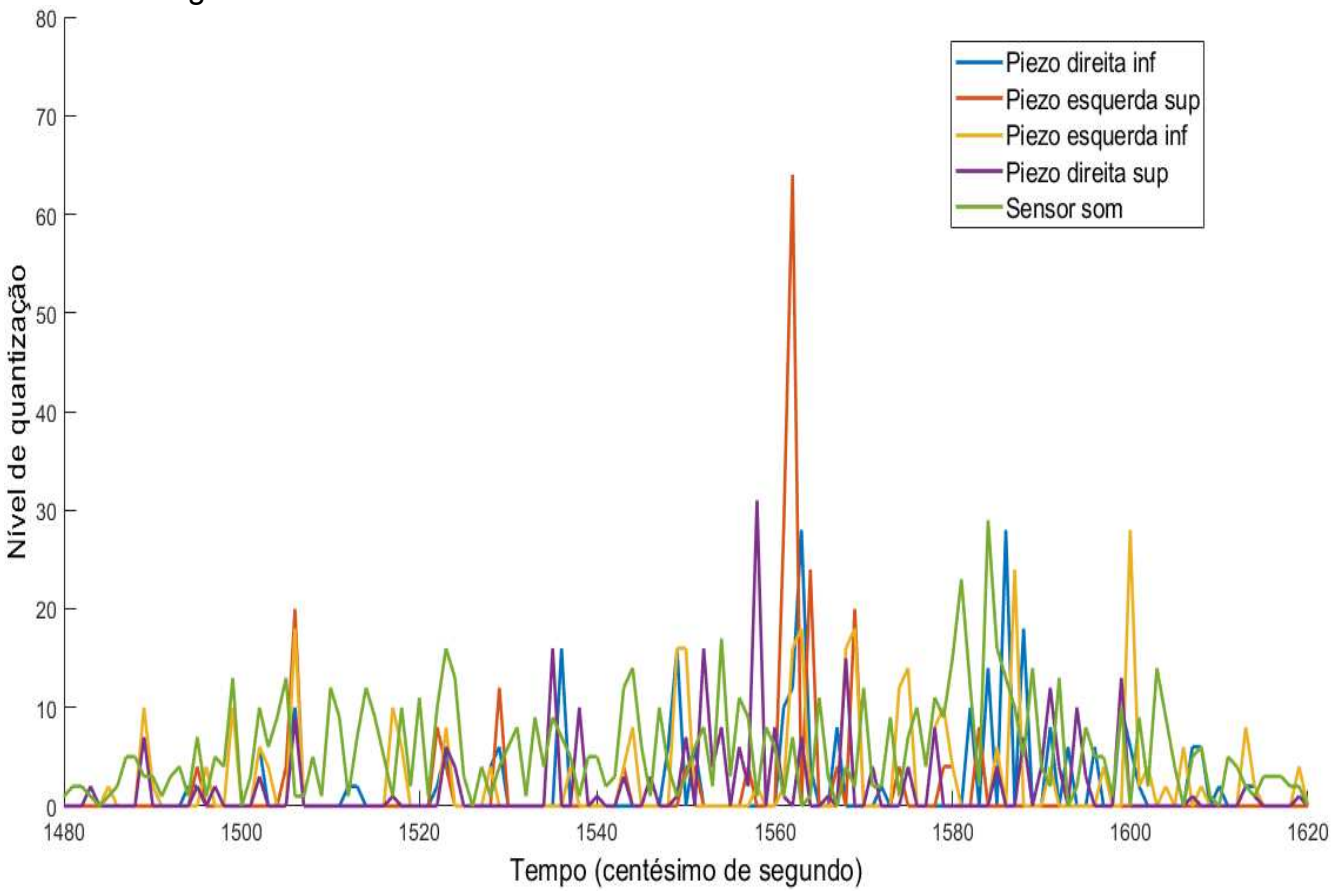


Fonte: O autor.

De acordo com a Figura 12, fica evidente que em uma batida na lataria a excitação de todo o sistema é muito alta. Foram realizadas batidas em todas as regiões ilustradas na Figura 7, resultando na mesma dinâmica observada na Figura 12.

Para o cenário de riscado na horizontal da lataria, conforme ilustrado na Figura 9, obteve-se o resultado apresentado na Figura 13.

Figura 13 – Curvas referentes à riscado na horizontal da lataria

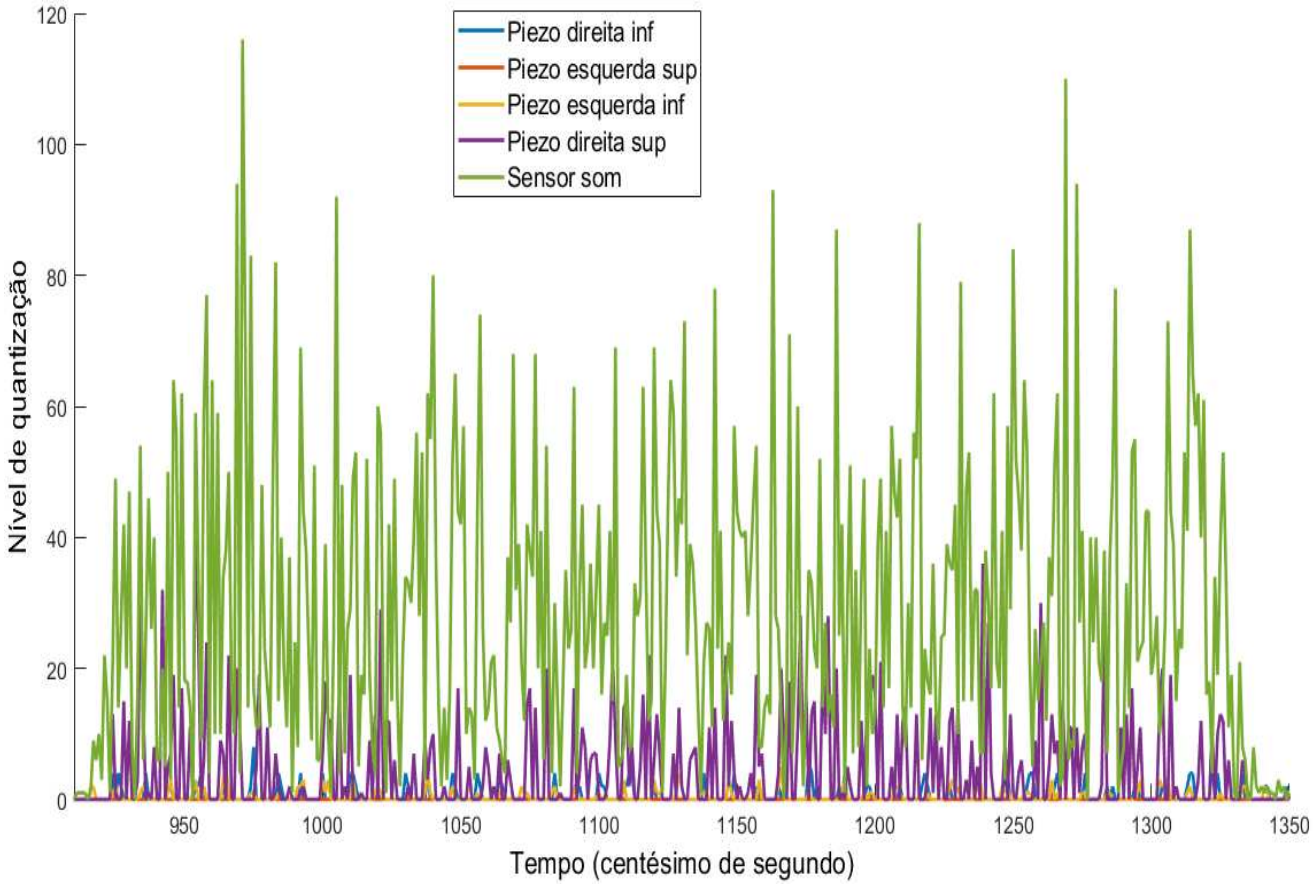


Fonte: O autor.

Analisando-se a resposta obtida, observa-se que houve uma excitação de intensidade média em todos os sensores. Verificando os resultados dos outros casos gerados ilustrados na Figura 9, notou-se o mesmo padrão do exposto na Figura 13.

Com relação ao cenário de sons externos de intensidade alta, obteve-se o resultado apresentado na Figura 14.

Figura 14 – Curvas referentes à sons externos de intensidade alta a 0,3 m da funilaria de ensaios

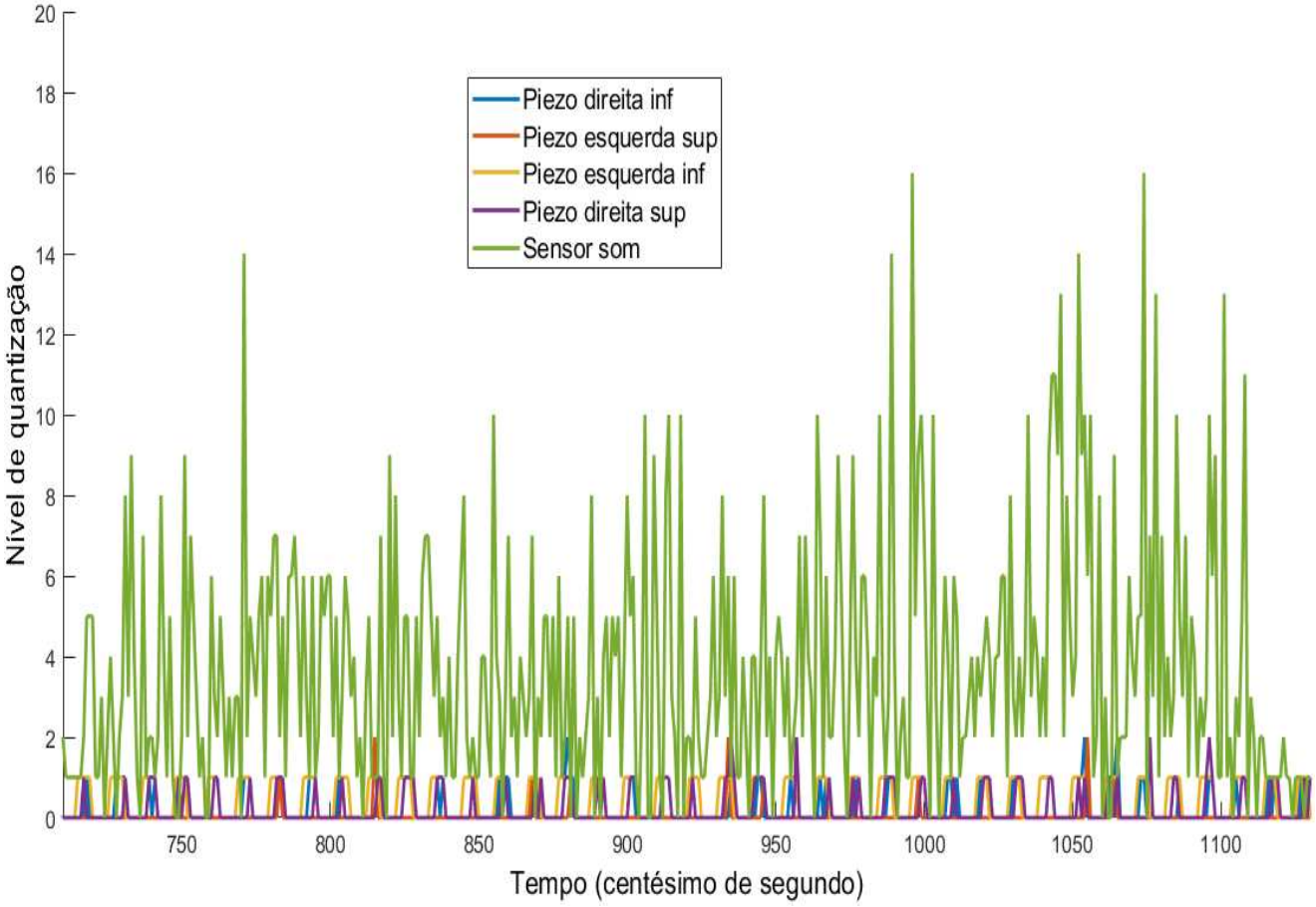


Fonte: O autor.

Para este cenário, devido a intensidade sonora ser muito elevada, pode-se observar pequenas vibrações na funilaria de ensaios, excitando levemente os sensores piezoelétricos. Além disso, verifica-se que a intensidade do sinal do sensor sonoro é muito alta. Para o caso em que a fonte sonora está a 1 m de distância da funilaria de ensaios, apenas verificou-se que houve uma redução na amplitude do sinal em todos os sensores, porém, ainda se manteve a característica comentada no resultado da Figura 14.

Analisando-se o cenário de sons externos de intensidade regular, observou-se quase todas as características apresentadas no caso anterior, apenas diferenciando que neste caso houve uma redução na amplitude do sinal sonoro e uma excitação extremamente baixa ou até mesmo nula nos sensores piezoelétricos. O resultado comentado pode ser verificado na Figura 15.

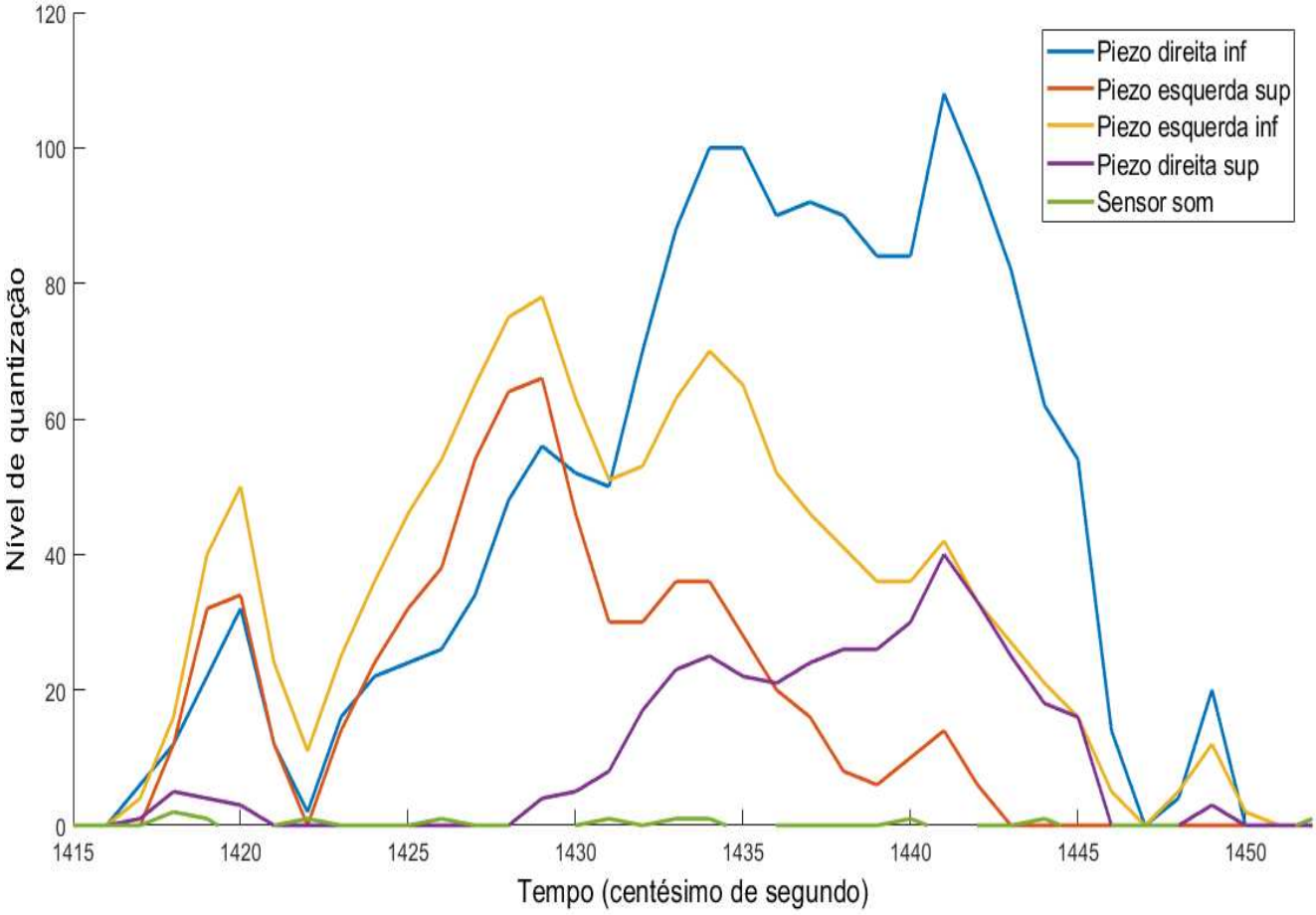
Figura 15 – Curvas referentes à sons externos de intensidade regular a 0,3 m da funilaria de ensaios



Fonte: O autor.

O último cenário gerado foi uma tentativa de furto do veículo. Com isso, os dados adquiridos para o cenário de furto forçando a parte central da funilaria do automóvel, de acordo com a Figura 7, pode ser visualizado na Figura 16.

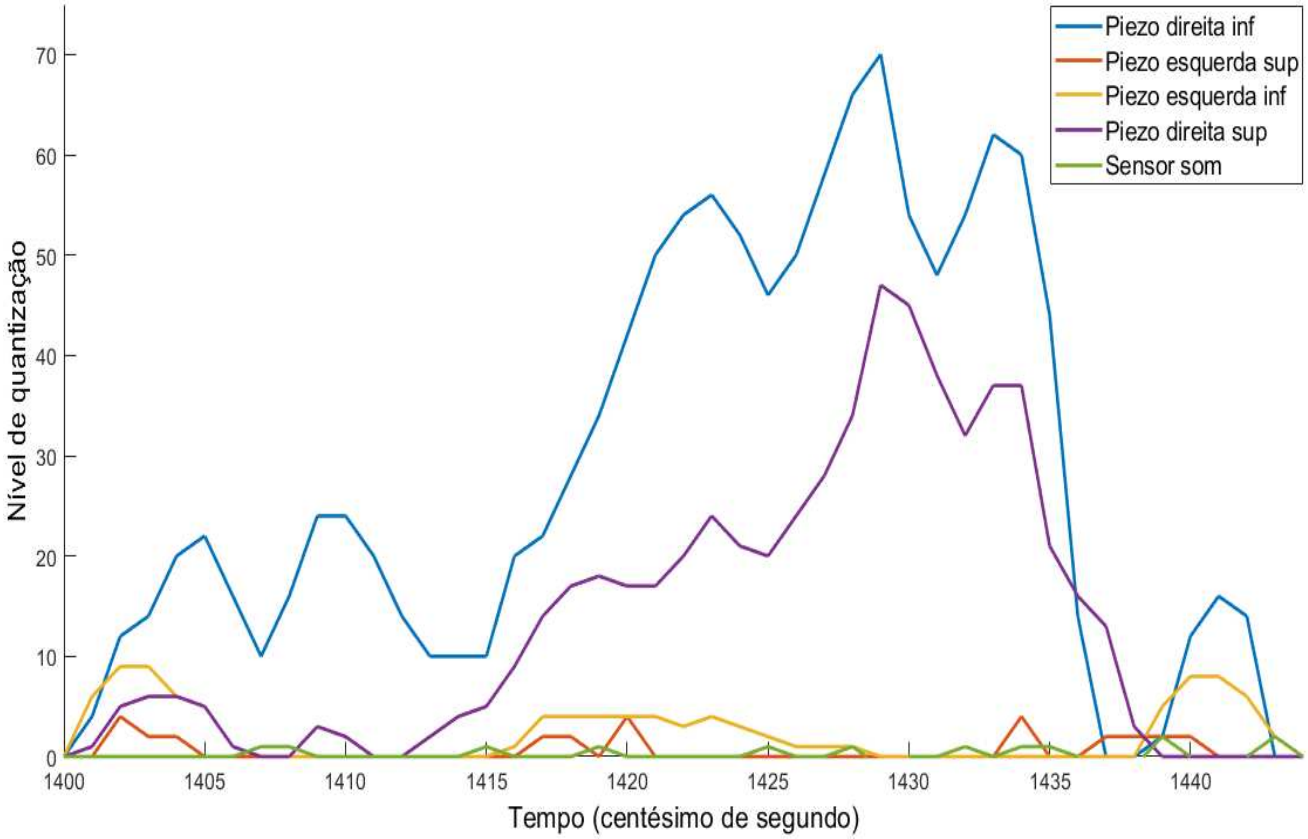
Figura 16 – Curvas referentes à uma tentativa de furto forçando a parte central do veículo



Fonte: O autor.

Com base no resultado apresentado na Figura 16, fica evidente que há uma excitação muito elevada nos sensores piezoelétricos e extremamente baixa ou nula no sensor sonoro. Os demais casos gerados para este cenário, com base na Figura 7, apresentaram as mesmas características do caso visto na Figura 16. Observou-se que nos demais casos, quando mais próxima a força aplicada da posição de um sensor piezoelétrico, maior é a sua excitação, justificável pelo princípio piezoelétrico já relatado. Um exemplo dessa dinâmica pode ser observado na Figura 17, onde a força foi aplicada na lateral direita da funilaria de ensaios e houve uma maior excitação dos sensores localizados naquela posição.

Figura 17 – Curvas referentes à uma tentativa de furto forçando a parte lateral direita do veículo



Fonte: O autor.

3.2.2 Desenvolvimento do algoritmo classificador em MATLAB

Com os dados de cada cenário adquiridos e armazenados, de acordo com a metodologia apresentada, desenvolveu-se um algoritmo em MATLAB para classificação dos cenários. Utilizou-se uma metodologia para análise dos dados através de médias aritméticas de janelas deslizantes, onde definiu-se um intervalo de início entre as janelas e o tamanho da janela. Em cada janela, retorna-se então o valor correspondente a média dos sinais adquiridos para cada sensor. Com isso, criou-se uma variável para armazenar as médias das janelas, de acordo com o apresentado no Quadro 5.

Quadro 5: Parâmetros de inicialização do código

(continua)

```
clc
load('100cm_105db.mat');
tamanho_janela = 100;
```

Quadro 5: Parâmetros de inicialização do código

(conclusão)

```
intervalo = 5;


num_amostras = size(som_100_cm, 1);
num_sensores = size(som_100_cm, 2);
medias_sensores = zeros(ceil(num_amostras / intervalo),
num_sensores);
```

Fonte: O autor.

Sendo assim, implementou-se um laço de repetição *for* para varrer todo o número de amostras do vetor com os dados, identificando também o início e o fim de cada janela. Previamente as classificações, realizou-se a média dos dados armazenados da janela correspondente. O código descrito pode ser visualizado no Quadro 6.

Quadro 6: Laço de varredura dos dados e cálculo de médias

```
for i = 1:intervalo:num_amostras
    janela_inicio = i;
    janela_fim = min(i + tamanho_janela - 1, num_amostras);

    if janela_fim - janela_inicio + 1 < tamanho_janela
        janela_fim = num_amostras;
    end

    media_janela = mean(som_100_cm(janela_inicio:janela_fim, :));
```

Fonte: O autor.

Com o vetor de médias da janela obtido, tendo os elementos de 1 a 4 do vetor correspondente aos sensores piezoelétricos e o elemento 5 do vetor correspondente ao sensor sonoro, desenvolveu-se o algoritmo classificador com base nas análises dos dados já adquiridos. Sendo assim, para a condição de batida na lataria, é verificado se a média aritmética dos dados dos sensores fornece um valor elevado, caracterizando assim uma alta excitação. Além disso, para a validação do cenário de batida, verifica-se também se houve uma grande excitação no sensor sonoro. O código implementado pode ser observado no Quadro 7.

Quadro 7: Lógica de verificação de batida

```
if (media_janela(5)>5 && (media_janela(1) + media_janela(2) +
media_janela(3) + media_janela(4)+ media_janela(5))/5>10)

fprintf('batida \n');

end
```

Fonte: O autor.

O algoritmo desenvolvido para a detecção do cenário de riscados é o mesmo que do cenário de batida. A única diferença presente neste cenário é com relação ao nível de excitação que é verificado, que neste caso verifica-se uma excitação média nos sensores. O código relatado pode ser visualizado no Quadro 8.

Quadro 8: Lógica de verificação de riscado

```
if ( (media_janela(1) + media_janela(2) + media_janela(3) +
media_janela(4)+ media_janela(5))/5 > 2 && media_janela(5)<5
&& media_janela(5)>2)

fprintf('riscado \n');

end
```

Fonte: O autor.

Como o cenário de sons externos não é algo crítico para a estrutura do veículo e as características dos cenários sonoros diferem muito pouco entre si, desenvolveu-se uma única lógica de verificação para os casos de sons externos de intensidade alta e regular. Assim, para englobar as características de sons externos altos, é verificado se há uma alta diferença entre a média do sensor sonoro e média aritmética dos demais sensores. Para englobar os casos de sons externos de intensidade regular, é analisado se há uma diferença mínima entre a excitação no sensor sonoro com relação a média de cada sensor piezoelétrico individualmente. Ainda assim, é necessário que a média aritmética dos sensores piezoelétricos esteja baixa, caracterizando sinais nulos ou quase nulos nos sensores. A lógica desenvolvida pode ser verificada no quadro 9.

Quadro 9: Lógica de verificação de sons externos

```
if (media_janela(5)-((media_janela(1) + media_janela(2) +
media_janela(3) + media_janela(4))/4)>10 || (media_janela(5)-
media_janela(1)>2 && media_janela(5)-media_janela(2)>2 &&
media_janela(5)-media_janela(3)>2 && media_janela(5)-
media_janela(4)>2 && ((media_janela(1) + media_janela(2) +
media_janela(3) + media_janela(4))/4)<1))

fprintf('som \n');

end
```

Fonte: O autor.

Para o cenário de furto, implementou-se uma lógica onde verificar-se a diferença entre a média de cada sensor piezoelétrico com relação à média do sensor sonoro. Para este caso, também é analisado se a média do sensor sonoro é muito baixa ou nula. A lógica desenvolvida pode ser observada no Quadro 10.

Quadro 10: Lógica de verificação de furto

```
if ((media_janela(1) - media_janela(5)>4 || media_janela(2)-
media_janela(5)>4|| media_janela(3)- media_janela(5)>4||
media_janela(4)- media_janela(5)>4) && media_janela(5)<1)

fprintf('furto \n');

end
```

Fonte: O autor.

Por fim, é salvo em uma variável todas as médias das janelas de acordo com o índice correspondente da janela, podendo ser visto no Quadro 11.

Quadro 11: Armazenamento das médias dos sensores

```
indice_media = ceil(janela_inicio / intervalo);
medias_sensores(indice_media, :) = media_janela;
```

Fonte: O autor.

3.2.3 Implementação do algoritmo na plataforma Arduino

Com o algoritmo classificador criado e testado no MATLAB, desenvolveu-se um algoritmo no Arduino para a verificação, classificação e reporte dos eventos para o usuário. Primeiramente, no cabeçalho do código defiram-se os seguintes argumentos:

- Portas seriais para a comunicação do Arduino com o módulo SIM800L;
- Uma função para o envio de mensagens SMS;
- Entradas analógicas dos sensores;
- Algumas variáveis gerais;
- Constantes para definição do intervalo de início de cada janela deslizante e o tamanho de cada janela;
- Vetores para armazenamento de dados em tempo real.

As definições implementadas podem ser verificadas no código apresentado no Quadro 12.

Quadro 12: Cabeçalho de inicialização do código

(continua)

```
#include <SoftwareSerial.h>
#define TX_PIN 10
#define RX_PIN 9
SoftwareSerial serialGSM(TX_PIN, RX_PIN);

void enviaSMS(String telefone, String mensagem);

const int piezoPin1 = A7;
const int piezoPin2 = A6;
const int piezoPin3 = A5;
const int piezoPin4 = A2;
const int sensorsom = A4;

int ciclo=0;
int ciclo2=0;
float calibra_som=0;
```

Quadro 12: Cabeçalho de inicialização do código

(conclusão)

```
float calibra_p1=0;
//contadores de cenários
int cont_batida=0;
int cont_furto=0;
int cont_som=0;
int cont_riscado=0;

int sensorsom_calibrado=0;
int valorAbsoluto=0;

int sensorvalorpiezo1=0;
int sensorvalorpiezo2=0;
int sensorvalorpiezo3=0;
int sensorvalorpiezo4=0;
int som=0;



int contador_ciclo=0;
const int intervalo=5;
const int n = 100;

int amostras_p1[n];
int amostras_p2[n];
int amostras_p3[n];
int amostras_p4[n];
int amostras_s1[n];

float media_p1=0;
float media_p2=0;
float media_p3=0;
float media_p4=0;
float media_s1=0;
```

Fonte: O autor.

Posteriormente, calibra-se o sensor sonoro para a referência em zero. Com todos os sinais na referência, adquire-se os dados lidos dos sensores, armazenando-os em um vetor que vai deslizando as amostras adquiridas, removendo os dados mais antigos e adicionando no início do vetor os novos. Além disso, é incrementada uma variável para a verificação do início da próxima janela. O código é explicitado no Quadro 13.

Quadro 13: Leitura e armazenamento de dados adquiridos

(continua)

```
int somap1=0;
int somap2=0;
int somap3=0;
int somap4=0;
int somas1=0;

// envia o sinal para origem
if(ciclo==0){
som = analogRead(sensorsom);
calibra_som=som;
ciclo ++;
}

sensorvalorpiezo1 = analogRead(piezoPin1);
sensorvalorpiezo2 = analogRead(piezoPin2);
sensorvalorpiezo3 = analogRead(piezoPin3);
sensorvalorpiezo4 = analogRead(piezoPin4);
som = analogRead(sensorsom);
sensorsom_calibrado=som-calibra_som;
valorAbsoluto = abs(sensorsom_calibrado);

for(int i=n-1;i>0;i--){
    amostras_p1[i]= amostras_p1[i-1];
    amostras_p2[i]= amostras_p2[i-1];
    amostras_p3[i]= amostras_p3[i-1];
```

Quadro 13: Leitura e armazenamento de dados adquiridos

(conclusão)

```
    amostras_p4[i]= amostras_p4[i-1];
    amostras_s1[i]= amostras_s1[i-1];

    amostras_p1[0]= sensorvalorpiezo1;
    amostras_p2[0]= sensorvalorpiezo2;
    amostras_p3[0]= sensorvalorpiezo3 ;
    amostras_p4[0]= sensorvalorpiezo4;
    amostras_s1[0]= valorAbsoluto;
}
contador_ciclo=contador_ciclo + 1;
```

Fonte: O autor.

Com isso, é verificado se o intervalo entre o início das janelas foi atendido, caso a resposta for afirmativa, realiza-se a média dos dados presentes na janela, e caso for negativa, continua-se o *loop*. A lógica realizada está explícita no Quadro 14.

Quadro 14: Algoritmo de execução de média das janelas

```
if(contador_ciclo==intervalo){
for(int i=0;i<n;i++){
      somap1+=amostras_p1[i];
      somap2+=amostras_p2[i];
      somap3+=amostras_p3[i];
      somap4+=amostras_p4[i];
      somas1+=amostras_s1[i];
}
    media_p1=somap1/n;
    media_p2=somap2/n;
    media_p3=somap3/n;
    media_p4=somap4/n;
    media_s1=somas1/n;
    contador_ciclo=0;
  }
```

Fonte: O autor.

Para a identificação dos eventos, transferiu-se a lógica de classificação desenvolvida no MATLAB para o Arduino. Além disso, implementou-se também uma lógica para verificar se o mesmo cenário ocorreu duas vezes seguidas. Validando a ocorrência do mesmo cenário mais de uma vez, envia-se uma mensagem para o número de telefone informado, relatando o cenário ocorrido, de acordo com o Quadro 15.

Quadro 15: Algoritmo classificador na plataforma Arduino

(continua)

```
if (media_s1 - ((media_p1 + media_p2 + media_p3 + media_p4) / 4) > 10 || (media_s1 - media_p1 > 2 && media_s1 - media_p2 > 2 && media_s1 - media_p3 > 2 && media_s1 - media_p4 > 2 && ((media_p1 + media_p2 + media_p3 + media_p4) / 4) < 1)) {
    if (cont_som == 2) {
      enviaSMS("Telefone", "SOM");
      cont_som = 0;
    }
    cont_riscado = 0;
    cont_batida = 0;
    cont_som ++;
    cont_furto = 0;
  }
if ((media_p1 - media_s1 > 4 || media_p2 - media_s1 > 4 || media_p3 - media_s1 > 4 || media_p4 - media_s1 > 4) && media_s1 < 1) {
    if (cont_furto == 2) {
      enviaSMS("Telefone", "FURTO");
      cont_furto = 0;
    }
    cont_riscado = 0;
    cont_batida = 0;
    cont_som = 0;
    cont_furto ++;
  }
```

Quadro 15: Algoritmo classificador na plataforma Arduino

(conclusão)

```
   if ( (media_p1 + media_p2 + media_p3 + media_p4 + media_s1)
/ 5 > 2 && media_s1 < 5 && media_s1 > 2) {
    if (cont_riscado == 2) {
      enviaSMS("Telefone", "RISCADO");
      cont_riscado = 0;
    }
    cont_riscado ++;
    cont_batida = 0;
    cont_som = 0;
    cont_furto = 0;
  }

  if (media_s1 > 5 && (media_p1 + media_p2 + media_p3 + media_p4
+ media_s1) / 5 > 10) {
    if (cont_batida == 2) {
      enviaSMS("Telefone", "BATIDA");
      cont_batida = 0;
    }
    cont_riscado = 0;
    cont_batida ++;
    cont_som = 0;
    cont_furto = 0;
  }
```

Fonte: O autor.

Para o reporte das informações ao usuário, implementou-se uma função que recebe como parâmetro o número de telefone do usuário e a mensagem que se deseja enviar. Portanto, o Arduino repassa a informação para o módulo SIM800L, que por sua vez encaminha para o número de telefone fornecido. A função mencionada pode ser visualizada no Quadro 16.

Quadro 16: Função de envio de mensagem SMS

```
void enviaSMS(String telefone, String mensagem) {
  serialGSM.print("AT+CMGS=\"" + telefone + "\"\n");
  serialGSM.print(mensagem + "\n");
  serialGSM.print((char)26);
}
```

Fonte: O autor.

3.2.4 Desenvolvimento do algoritmo da matriz de confusão

A matriz de confusão é empregada para avaliar o desempenho de um algoritmo através de métricas estatísticas (ALPAYDIN, 2014). Para o desenvolvimento de uma matriz de confusão, informa-se os casos verdadeiros e os casos previstos. Com isso, atualiza-se os índices da matriz de acordo com o resultado real analisado e a resposta fornecido pelo algoritmo. No quadro 17 é ilustrado um exemplo de uma matriz de confusão de quarta ordem.

Quadro 17: Exemplo de uma matriz de confusão de quarta ordem

	Classe prevista				
Classe verdadeira	Classe 1	Classe 2	Classe 3	Classe 4	Total
Classe 1	$tC1$	$fC2$	$fC3$	$fC4$	$p1$
Classe 2	$fC1$	$tC2$	$fC3$	$fC4$	$p2$
Classe 3	$fC1$	$fC2$	$tC3$	$fC4$	$p3$
Classe 4	$fC1$	$fC2$	$fC3$	$tC4$	$p4$
Total	$p1'$	$p2'$	$p3'$	$p4'$	N

Fonte: O autor.

Quando a classe prevista pelo algoritmo for igual a classe verdadeira, significando que o algoritmo acertou a classificação, retorna-se o resultado *true* (t), caso contrário, retorna-se o resultado *false* (f). Além disso, a composição da classificação do algoritmo leva em consideração qual a classe prevista, sendo assim, no exemplo a cima, há a Classe 1 ($C1$), Classe 2 ($C2$), Classe 3 ($C3$) e Classe 4 ($C4$). Com isso, para a Classe 1, o resultado $tC1$ significa que a classe prevista pelo algoritmo é realmente a classe real, já o resultado $fC1$ determina que o algoritmo

previu a Classe 1, porém de forma incorreta, não sendo essa a classe verdadeira. O processo comentado se estende para as demais classes.

Com os dados já adquiridos, armazenou-se em um vetor os arquivos salvos, definiu-se as váriaveis de cada arquivo, inicializou-se a matriz de confusão e definiu-se os cenários verdadeiros para cada caso. O código desenvolvido é apresentado no Quadro 18.

Quadro 18: Inicialização da matriz de confusão

```
arquivos = {'batida.mat', 'riscado.mat', 'som.mat',
'furto.mat'};

dado = {batida, riscado, som, furto};
confusion_matrix_global = zeros(4, 4);
rotulos_verdadeiros{1} = "batida"
rotulos_verdadeiros{2} = "riscado"
rotulos_verdadeiros{3} = "som"
rotulos_verdadeiros{4} = "furto"
```

Fonte: O autor.

Posteriormente, utilizou-se um laço de repetição *for* para realizar a matriz de confusão em todos os arquivos. Definiu-se a matriz de confusão sendo de quarta ordem quadrada devido aos quatro possíveis cenários. Ao final, somou-se todas as matrizes de confusão para se obter uma matriz de confusão geral, assim como apresentado no Quadro 19.

Quadro 19: Aquisição da matriz de confusão geral

(continua)

```
for arquivo_idx = 1:length(arquivos)
load(arquivos{arquivo_idx});

tamanho_janela = 100;

intervalo = 5;

num_amostras = size(dado{arquivo_idx}, 1);
```

Quadro 19: Aquisição da matriz de confusão geral

(conclusão)

```
num_sensores = size(dado{arquivo_idx}, 2);
medias_sensores = zeros(ceil(num_amostras / intervalo),
num_sensores);

confusion_matrix = zeros(4, 4);

rotulos_classe = ["som", "furto", "riscado", "batida"];
confusion_matrix_global = confusion_matrix_global +
confusion_matrix;
end
```

Fonte: O autor.

Através de outro laço *for*, é verificado e classificado todos os dados dos arquivos utilizando o algoritmo desenvolvido anteriormente, de acordo com o Quadro 20.

Quadro 20: Classificação de dados para incremento na matriz de confusão

(continua)

```
    if (media_janela(5)-((media_janela(1) + media_janela(2) +
media_janela(3) + media_janela(4))/4)>10 || (media_janela(5)-
media_janela(1)>2 && media_janela(5)-media_janela(2)>2 &&
media_janela(5)-media_janela(3)>2 && media_janela(5)-
media_janela(4)>2 && ((media_janela(1) + media_janela(2) +
media_janela(3) + media_janela(4))/4)<1))
        rotulo_previsto = "som";
    end

    if ((media_janela(1) - media_janela(5)>4 ||
media_janela(2)- media_janela(5)>4|| media_janela(3)-
media_janela(5)>4|| media_janela(4)- media_janela(5)>4) &&
media_janela(5)<1)
        rotulo_previsto = "furto";
    end
```

Quadro 20: Classificação de dados para incremento na matriz de confusão (conclusão)

```
    if ( (media_janela(1) + media_janela(2) + media_janela(3)
+ media_janela(4)+ media_janela(5))/5 > 2 && media_janela(5)<5
&& media_janela(5)>2)
        rotulo_previsto = "riscado";
    end
    if     (media_janela(5)>5     &&     (media_janela(1)     +
media_janela(2)    +    media_janela(3)    +    media_janela(4)+
media_janela(5))/5>10)
        rotulo_previsto = "batida";
    end
```

Fonte: O autor.

Além disso, dentro do mesmo laço de repetição do caso anterior, identifica-se o rótulo verdadeiro, ou seja, o cenário que foi simulado realmente, e incrementa-se na matriz o cenário identificado, atualizando seus índices. O código implementado pode ser visualizado no Quadro 21.

Quadro 21: Atualização dos índices da matriz de confusão

```
rotulo_verdadeiro = rotulos_verdadeiros{arquivo_idx};
confusion_matrix(strcmp(rotulo_verdadeiro,  ["som",  "furto",
"riscado", "batida"]),strcmp(rotulo_previsto, ["som", "furto",
"riscado","batida"]))                                        =
confusion_matrix(strcmp(rotulo_verdadeiro,  ["som",  "furto",
"riscado", "batida"]),strcmp(rotulo_previsto, ["som", "furto",
"riscado", "batida"])) + 1;
```

Fonte: O autor.

Por fim, para a avaliação do algoritmo classificador, levou-se em consideração as métricas de acurácia, precisão e revocação adquiridas através dos dados obtidos na matriz de confusão (ver o Quadro 17). A acurácia (A) avalia o desempenho geral do algoritmo em classificar todos os casos corretamente, a qual é obtida através da razão entre todos os resultados corretos obtidos ($\sum_{i=1}^{4} tCi$) e o número total de classificações realizadas pelo algoritmo (N), isto é

$$A = \frac{(\sum_{i=1}^{4} tCi)}{N}. \quad (3)$$

A precisão (P), por sua vez, é obtida individualmente para cada caso analisado e avalia a efetividade do algoritmo para prever corretamente um caso, sendo caracterizada pela razão entre a previsão correta obtida ($tC1$) e o número total de previsões obtidas pelo algoritmo ($p1'$), podendo ser descrita como,

$$P = \frac{tC1}{p1'}. \quad (4)$$

Já a revocação (R), também chamada de *recall*, assim como na precisão, é obtida individualmente para cada caso analisado e avalia a capacidade do algoritmo de identificar corretamente todos os casos reais existentes. Sendo assim, é obtida pela razão da classificação correta do cenário ($tC1$) pelo número total de elementos presentes no cenário real ($p1$), isto é,

$$R = \frac{tC1}{p1}. \quad (5)$$

O código com as métricas desenvolvidas pode ser observado no Quadro 22.

Quadro 22: Classificação do algoritmo

```
confusion_table = array2table(confusion_matrix_global,
'VariableNames',rotulos_classe, 'RowNames', rotulos_classe);
disp(confusion_table);
acuracia = sum(diag(confusion_matrix_global)) /
sum(confusion_matrix_global, 'all');
precisao = diag(confusion_matrix_global) ./
sum(confusion_matrix_global, 1)';
revocacao = diag(confusion_matrix_global) ./
sum(confusion_matrix_global, 2);
disp('Acurácia:');
disp(acuracia);
disp('Precisão:');
disp(precisao);
disp('Revocação:');
disp(revocacao);
```

Fonte: O autor.

4 RESULTADO

4.1 CLASSIFICAÇÃO DO BANCO DE DADOS

Para avaliar o algoritmo classificador projetado, desenvolveu-se uma matriz de confusão utilizando o banco de dados adquirido antes do desenvolvimento do algoritmo. Para este caso, a matriz geral obtida é apresentada na Tabela 1.

Tabela 1: Matriz de confusão com base no banco de dados

	Som	**Furto**	**Riscado**	**Batida**
Som	77	0	0	0
Furto	0	23	0	0
Riscado	2	0	7	0
Batida	0	0	1	20

Fonte: O autor.

Com a matriz de confusão obtida, pode-se observar um elevado número de classificações no cenário sonoro em relação aos demais, devido a este evento possuir um maior número de amostras adquiridas. Já para o baixo número de classificações no cenário de riscado, isso deve-se ao menor número de amostras desse cenário disponível no banco de dados. Vale mencionar que a geração desse cenário tem maior complexidade devido à necessidade de um controle preciso da força aplicada sobre a chapa.

Utilizando os dados adquiridos, obteve-se uma acurácia do algoritmo de 97,7%. Na Tabela 2, podem ser visualizados os percentuais de precisão e revocação obtidos para cada cenário.

Tabela 2: Resultados de precisão e revocação do algoritmo com base no banco de dados

	Precisão	**Revocação**
Som	97,5%	100%
Furto	100%	100%
Riscado	87,5%	77,8%
Batida	100%	95,2%

Fonte: O autor.

A partir dos resultados obtidos, determinou-se que o algoritmo possui uma boa assertividade geral dos cenários. Observou-se também, uma alta taxa de acerto das previsões de cada cenário, sendo o menor valor de precisão obtido na faixa de 87,5% para os casos de riscados. Ademais, afirma-se também que, para cada cenário levantado, o algoritmo consegue identificar corretamente um alto percentual de cenários reais existentes, sendo o menor valor de revocação obtido no cenário de riscados com um total de 77,8%.

4.2 CLASSIFICAÇÃO DE NOVOS DADOS

Para validar o desempenho do algoritmo classificador após seu desenvolvimento, desenvolveu-se uma matriz de confusão com base em novos dados adquiridos. A matriz obtida é ilustrada na Tabela 3. Para gerar o novo cenário batida, soltou-se um multímetro de 0,15 kg, em um ponto aleatório da chapa, de uma altura de 0,15 m, gerando uma força de impacto de aproximadamente 17 N. No novo cenário de riscado, utilizou-se uma tesoura para realizar força na funilaria de ensaios. Na geração do novo cenário sonoro, utilizou-se a própria voz como fonte sonora, sendo mantida uma intensidade do som de aproximadamente 80 dB a uma distância de 0,4 m. Por fim, para o novo cenário de furto gerado, aplicou-se pressão com uma caneta em um ponto aleatório da funilaria de ensaios.

Tabela 3: Matriz de confusão com novos dados

	Som	**Furto**	**Riscado**	**Batida**
Som	55	0	0	0
Furto	0	21	0	0
Riscado	1	0	4	0
Batida	0	0	1	19

Fonte: O autor.

Assim como no caso anterior, obteve-se um elevado número de classificações no cenário sonoro, devido a um maior número de amostras adquiridas neste cenário, e um baixo número de classificações no cenário de riscado, por causa das características de geração desse evento.

Com isso, para a acurácia, obteve-se um valor de 98%. Na tabela 4 são apresentados os percentuais de precisão e revocação obtidos para cada cenário.

Tabela 4: Resultados de precisão e revocação do algoritmo com base em novos dados

	Precisão	**Revocação**
Som	98,2%	100%
Furto	100%	100%
Riscado	80%	80%
Batida	100%	95%

Fonte: O autor.

Com as métricas obtidas, determinou-se que o algoritmo possui uma boa assertividade geral dos cenários. Analisando-se a precisão obtida, observa-se uma boa capacidade do algoritmo de prever corretamente os cenários e verifica-se que a menor precisão adquirida foi nos casos de riscados com um valor de 80%. Além disso, para cada cenário analisado, obteve-se um alto valor de identificações corretas em relação ao número total de cenários reais existentes, visto que, o menor valor de revocação obtido foi de 80% para o cenário de riscados.

Comparando-se os resultados obtidos na seção anterior com a seção atual, verifica-se que houve uma diferença muito baixa entre ambos. Além disso, verificou-se que as métricas analisadas para o algoritmo forneceram valor elevados, sendo o menor valor de precisão obtido em 80% e o menor valor de revocação obtido em 77,8%. Dado o exposto, determina-se então que o algoritmo desenvolvido é capaz de atender seu propósito, obtendo valores satisfatórios de acurácia, precisão e revocação.

4.3 REPORTE DE EVENTOS PARA O USUÁRIO EM TEMPO REAL

Com o algoritmo final implementado na plataforma do Arduino, necessitou-se de um chip de rede móvel, instalado no módulo SIM800L, e a informação no código do número de telefone que recebe a informação via SMS. Primeiramente, para exemplificar o reporte para o usuário, criou-se um cenário de furto, onde a resposta enviada para o usuário pode ser vista na Figura 18.

Figura 18 – Reporte para o usuário através de mensagem SMS de um cenário de furto

Fonte: O autor.

Analisando-se a resposta enviada pelo sistema, ilustrado na Figura 18, é possível determinar que o algoritmo desenvolvido é capaz de identificar e reportar o cenário identificado pelo algoritmo de classificação. Além disso, obteve-se um tempo de resposta da mensagem SMS de aproximadamente 12s.

Posteriormente, criou-se um cenário de batida, onde a resposta enviada pelo sistema pode ser verificada na Figura 19.

Figura 19 – Reporte para o usuário através de mensagem SMS de um cenário de batida

Fonte: O autor.

Assim como para o cenário de furto desenvolvido, neste cenário obteve-se uma correta identificação e reporte para o usuário do cenário em questão. Ademais, verificou-se um tempo de resposta igual ao caso anterior, de aproximadamente 12s.

Por fim, criou-se os demais cenários faltantes, de sons externos e batida na lataria, para se avaliar a resposta do algoritmo para o usuário em todos os cenários possíveis. A resposta fornecida pelo sistema pode ser visualizada na Figura 20.

Figura 20 – Reporte dos cenários para o usuário através de mensagens SMS

Fonte: O autor.

5 CONSIDERAÇÕES FINAIS

Quando se iniciou este trabalho de conclusão de curso, verificou-se a necessidade de um sistema de segurança automotivo mais completo e robusto que os sistemas convencionais presentes no mercado. Com isso, foi proposto um sistema baseado na aquisição e classificação de sinais vibratórios e sonoros.

Dado o exposto, definiu-se os materiais que foram utilizados no projeto como, quatro sensores piezoelétricos, um sensor de microfone, um módulo SIM800L para a comunicação através de mensagens SMS, um tabuleiro de forno e um Arduino MEGA. Definiu-se, então, os métodos para o desenvolvimento de um banco de dados de possíveis cenários de vandalismo levantados, sendo eles, tentativa de furto, riscado na lataria, batida no veículo e sons externos de intensidade alta e normal. Além disso, foi definida a metodologia para a análise de desempenho do algoritmo classificador projetado.

Após a definição dos materiais e métodos, definiram-se as conexões físicas dos equipamentos e desenvolveu-se, através do tabuleiro de forno, uma estrutura de funilaria de ensaios. Determinou-se então, como objetivo, desenvolver um banco de dados dos cenários levantados, sendo desenvolvido um algoritmo em MATLAB para aquisição e armazenamento dos dados dos sensores através da interação com a plataforma do Arduino. Além disso, com base no banco de dados adquirido, desenvolveu-se um algoritmo classificador que possibilitou a identificação dos cenários considerados.

Para a implementação efetiva do projeto que foi desenvolvido, implementou-se o algoritmo classificador na plataforma do Arduino. Além disso, implementou-se um algoritmo que possibilitou o reporte dos cenários identificados para o usuário através de mensagens SMS. Para verificar a qualidade do algoritmo classificador proposto, foram avaliadas as métricas de acurácia, precisão e revocação através da matriz de confusão gerada em MATLAB.

Para as métricas avaliadas do algoritmo, adquiriu-se resultados a cima de 97% para a acurácia, a cima de 80% para a precisão e a cima de 77% para a revocação, tanto nos cenários gerados antes do desenvolvimento do algoritmo quanto nos cenários gerados após o desenvolvimento do algoritmo, para sua validação. Além disso, verificou-se que o algoritmo conseguiu reportar para o usuário os cenários identificados com um tempo de resposta de aproximadamente 12s.

Em virtude dos fatos mencionados, pode-se concluir que o objetivo do presente trabalho de conclusão de curso foi alcançado com êxito. O sistema proposto é capaz classificar adequadamente os cenários considerados, além de alertar o usuário em tempo real dos cenários identificados.

Para projetos futuros, sugere-se a utilização do banco de dados gerado neste trabalho para o treinamento de uma rede neural para a classificação dos cenários considerados. O banco de dados está disponibilizado na plataforma GUITHUB (nota de rodapé).[1]

[1] ZIPF, André Leonardo. Repositório Pessoal. Github, 2024. Disponível em: https://github.com/AndrelZipf/Banco-de-dados-cenarios. Acesso em: 20 jan. 2024.

REFERÊNCIAS

ALBUQUERQUE, Flávia (org.). **Produção de veículos aumenta 3,7% no primeiro semestre**: total de unidades produzidas chegou a 1,13 milhão. Total de unidades produzidas chegou a 1,13 milhão. 2023. Disponível em: https://agenciabrasil.ebc.com.br/economia/noticia/2023-07/producao-de-veiculos-aumenta-37-no-primeiro-semestre#:~:text=ouvir%3A,mesmo%20per%C3%ADodo%20do%20ano%20anterior.. Acesso em: 15 nov. 2023.

TESLA. **Funcionalidades de proteção e segurança do veículo**. Disponível em: https://www.tesla.com/pt_pt/support/vehicle-safety-security-features. Acesso em: 15 nov. 2023. 2023.

ANUÁRIO BRASILEIRO DE SEGURANÇA PÚBLICA. São Paulo: Fórum Brasileiro de Segurança Pública, v. 16, 2022. Issn 1983-7364. Disponível em: https://forumseguranca.org.br/wp-content/uploads/2023/07/anuario-2023.pdf. Acesso em: 21 jan. 2024.

SOUZA, Jean Carlos Litz. A INFLUÊNCIA DE HENRY FORD PARA A ATUALIDADE. **Uniuv em Revista**, União da Vitória, v. 15, n. 1, p. 91-95, nov. 2017. Disponível em: https://periodicos.uniuv.edu.br/uniuvemrevista/article/view/373. Acesso em: 15 nov.

ALVES, Ana Julia. **Antifurto veicular: conheça os 6 tipos mais comuns**. 2023. Disponível em: https://chiptronic.com.br/blog/antifurto-veicular-conheca-os-6-tipos-mais-comuns. Acesso em: 21 nov. 2023.

MICROCHIP. **ATmega640/V-1280/V-1281/V-2560/V-2561/V [DATASHEET]**. Disponível em: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf. Acesso em: 18 dez. 2023.

SIMCOM. **SIM800H&SIM800L_Hardware Design_V2.02**. Shanghai: [s.n.], 2015.

ELETROGATE. **Módulo Sensor de Som KY-037**. Disponível em: https://www.eletrogate.com/modulo-sensor-de-som-ky037?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&utm_source=google&utm_medium=cpc&utm_campaign=[MC4]_[G]_[PMax]_ArduinoRoboticaSensoresModuloss&utm_content=&utm_term=&gad_source=1&gclid=CjwKCAiA-P-rBhBEEiwAQEXhH_GsiP8OMK7LkKBwno5U9sdfUE7KO7sDELVSad99Thl9rlj1X9SwshoCmyIQAvD_BwE. Acesso em: 18 dez. 2023.

AUTOMOTIVA, Kmc Tecnologia. **Estrutura Básica de um Veículo**. Disponível em: https://kmctecnologia.com/mecanica-automotiva/estrutura-de-um-veiculo/. Acesso em: 18 dez. 2023.

BENTLEY, John P.. **Principles of Measurement Systems**. 4. ed. Middlesbrough: Pearson, 2005.

MATHWORKS. **MATLAB Data**. Disponível em: https://www.mathworks.com/help/matlab/matlab_external/matlab-data.html?s_tid=srchtitle_site_search_3_MATLAB%20data. Acesso em: 19 dez. 2023.

MATHWORKS. **MATLAB Support Package for Arduino Hardware**. Disponível em: https://www.mathworks.com/matlabcentral/fileexchange/47522-matlab-support-package-for-arduino-hardware. Acesso em: 08 jan. 2024.

ALPAYDIN, Ethem. **Introduction to machine learning**. 3. ed. London: Mit Press, 2014. Disponível em: https://dl.matlabyar.com/siavash/ML/Book/Ethem%20Alpaydin-Introduction%20to%20Machine%20Learning-The%20MIT%20Press%20(2014).pdf. Acesso em: 09 jan. 2024.

ELETROGATE. **Transdutor/Pastilha Piezoelétrica 35mm**. Disponível em: https://www.eletrogate.com/transdutor-pastilha-piezoeletrica-35mm. Acesso em: 10 jan. 2024.

JBL. **JBL Xtreme 2**. Disponível em: https://www.jbl.com.br/refurbished/JBL+Xtreme+2.html. Acesso em: 10 jan. 2024.