

UTILIZAÇÃO DE FERRAMENTAS DE APRENDIZAGEM AUTOMÁTICA PARA PREVENÇÃO DE ANOMALIAS EM MOTORES DE AUTOMÓVEIS

Use of Machine Learning tools in predictive maintenance of motor vehicles

Cabral, J.¹, Paulo, J.^{1,2}, Trigo, P.¹

¹ ISEL-Instituto Superior de Engenharia de Lisboa, Portugal

² LAA-Lab. de Áudio e Acústica do ISEL, Portugal

{a46357@alunos.isel.pt, joel.paulo@isel.pt, paulo.trigo@isel.pt}

PACS: 07.05.Mh, 43.60.-c

Keywords: Artificial intelligence; Automatic Learning; Audio analysis; Sound Events Dataset; Predictive maintenance; Neural Networks

RESUMO.

Visando a segurança dos passageiros e a redução dos custos de manutenção dos veículos, os fabricantes começam a fornecer aos carros sistemas sofisticados para que os veículos se desloquem às oficinas somente quando necessário, sem que estas tenham problemas acrescidos por falta de manutenção adequada (manutenção preditiva). Nesse sentido, é necessário padronizar a manutenção com base na verificação constante da condição física dos equipamentos, utilizando as mais variadas informações sobre o estado dos diversos componentes do veículo. Este estudo pretende realizar uma análise dos sons produzidos pelo motor do veículo para identificar anomalias, através do uso de técnicas de aprendizagem automática, como as redes neuronais. Os resultados dessa análise são apresentados e armazenados em tempo real em uma plataforma centralizada para posterior análise e visualização. Isso mantém um histórico do veículo e possibilita ouvir as anomalias produzidas pelo motor. Como os diferentes sons produzidos pelos carros seguem uma sequência, regimes de funcionamento, foi utilizada uma máquina de estados para separação de eventos para isolar os regimes motores. Resultados experimentais mostraram que a separação de eventos auxilia o classificador na sua análise em comparação ao uso de segmentos igual duração. Os resultados são muito promissores, atingindo cerca de 90% de precisão. Este é um trabalho em curso que está em desenvolvimento no Laboratório de Áudio e Acústica do ISEL.

ABSTRACT.

Aiming at passenger safety and reducing vehicle maintenance costs, manufacturers are already beginning to provide cars with sophisticated systems so that vehicles travel to workshops only when necessary, without these having added problems due to lack of proper maintenance (predictive maintenance). In this sense, it is necessary to standardize maintenance based on constant verification of the physical condition of the equipment, using the most varied information on the status of the various components of the vehicle. This study intends to carry out an analysis of the sounds produced by the vehicle engine to identify anomalies, through using automatic machine learning techniques, such as neural networks. The results of this analysis are presented and stored in real time on a centralized platform for further analysis and visualization. This keeps a history of the vehicle and makes it possible to hear the anomalies produced by its engine. Since the different sounds produced by cars follow a sequence order, a state machine for events separation was used to isolate motor regimes. Experimental results showed that the separation of events helps the classifier in its analysis compared of using segments of the same duration. The results are very promising, achieving 85% of accuracy. This is an on-going work which are developed at the Laboratório de Áudio e Acústica do ISEL.

1. INTRODUÇÃO

Imagine-se o dia, em que veículos autónomos, podem marcar as suas próprias idas à oficina ou o dia no qual os condutores não precisam de estar constantemente preocupados com o estado do funcionamento do seu automóvel. Uma maneira de alcançar este horizonte é através da manutenção preditiva. Esta abordagem pode ser definida como a manutenção baseada nas condições físicas dos equipamentos, tendo como objetivo agendar a reparação do veículo antes que ocorra qualquer avaria, e pode ser realizada através de várias técnicas [1]. Neste estudo faz-se o uso do som produzido pelo motor da viatura, para os varios regimes de funcionamento, como forma de informação para monitorização de potenciais anomalias. A Figura 1 mostra exemplos das características de 3 tipos funcionamento anormal dos sistemas de tração de um veículo.

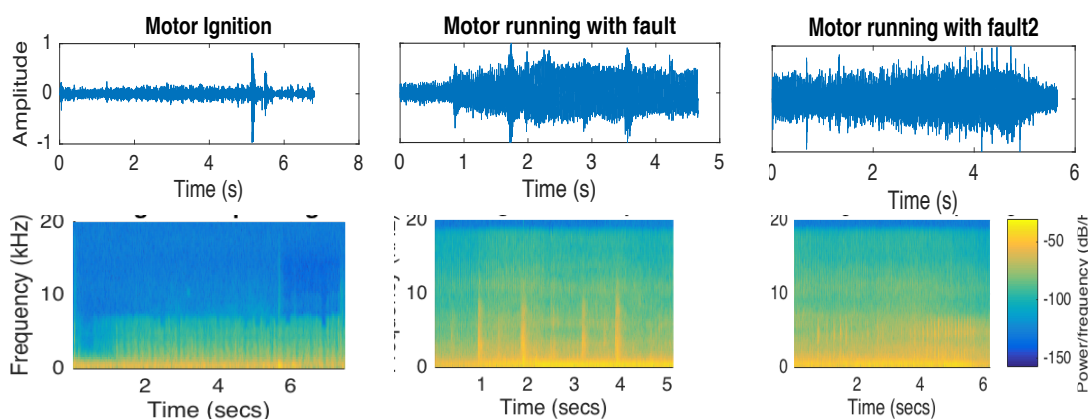


Figura 1 – Características de 3 tipos funcionamento anormal de motor.

Através do uso de técnicas de aprendizagem automática e processamento digital de sinais de áudio, é possível realizar uma análise constante do som gerado pelo motor e conseqüentemente, conhecer o estado que este se encontra [2]. Esta inteligência será composta por redes neuronais treinadas com inúmeras amostras de padrões de som de motores, que se encontram organizadas num conjunto de dados, o dataset.

Com a criação de uma plataforma será possível manter um histórico da viatura, com as avaliações realizadas, bem como permitir aos utilizadores/técnicos ouvir as anomalias produzidas pelos motores das suas viaturas. Numa situação limite, esta plataforma poderá gerir uma frota de veículos.

2. TRABALHO RELACIONADO

O som é uma grandeza que transporta grande quantidade de informação sobre o mundo que nos rodeia. Por exemplo, uma pessoa com limitações visuais consegue orientar-se essencialmente com base nos estímulos sonoros existentes à sua volta. Desta forma, cada vez mais o som é um elemento importante a ter em conta quando se pretende construir sistemas automáticos de análise de situações onde existe som. Assim, existem inúmeros projetos que utilizam o som como informação básica para tomada de decisão.

Um dos trabalhos que serve de base para a realização deste projeto é Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring [3]. Este estudo tem o objetivo de detetar anomalias em máquinas, através do som. E tinha como principal desafio detetar sons anómalos, completamente, desconhecidos. Para isso os autores criaram um dataset composto por sons de miniaturas de máquinas e máquinas reais. Estes sons são constituídos por sons dos equipamentos a funcionarem de forma correta e de sons destes com anomalias. Para as anomalias foi necessário danificar as máquinas deliberadamente.

O trabalho relacionado permite a compreensão de como se realiza um dataset e de que forma este permite treinar um modelo composto por redes neuronais para detetar anomalias com sons

desconhecidos. Esta base é útil, pois tal como no trabalho mencionado este projeto não tem acesso às anomalias que se propõe detetar. Em vez disso, são utilizados sons de motores a funcionar de forma correta e incorreta provenientes de base de dados na internet como o AudioSet disponibilizado pela Google.

Outro projeto estudado que, em parte, possui o mesmo intuito, é o Acoustic Anomaly Detection for Machine Sounds based on Image Transfer Learning [4-6], onde se realiza a análise de falhas de máquinas com base nos sons produzidos. Para tal, é utilizado um classificador do tipo CNN que é treinado através de imagens obtidas da característica Mel-Spectrogram.

3. MODELO PROPOSTO

O projeto que apresentamos propõe também realizar outros objetivos que não estão presentes nos trabalhos mencionados no estado-da-arte, tal como a separação de eventos, pois um motor de um veículo não produz um som constante como as máquinas de fábricas e por também se pretender detetar quando o motor se encontra a funcionar.

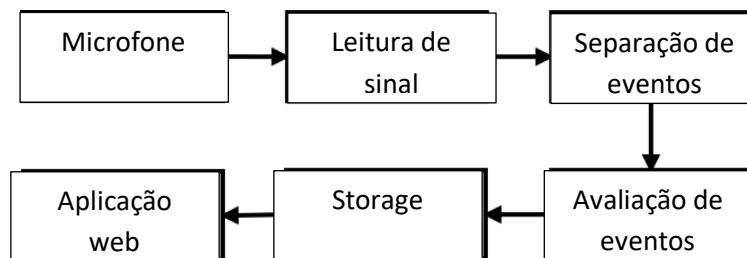


Figura 2 – Diagrama simplificado do modelo implementado.

Como está demonstrado na Figura 2, pretende-se obter o som do motor da viatura através de um microfone, ler o sinal recebido, separar este em eventos, classificar os eventos separados e por fim, guardar os eventos e mostrá-los numa Aplicação Web.

Como se pretende a atualização constante dos dados, e tendo em conta que o som recolhido poderá ter uma duração de várias horas, é necessário realizar a análise com a separação do sinal obtido em segmentos de um determinado número de amostras. Estes segmentos são à posteriori passados ao Separador de eventos, que serve não só para facilitar a classificação dos eventos, por parte do Avaliador, pois sem este seriam enviados sempre segmentos de cinco segundos (independentemente do seu conteúdo), como também permite a diminuição de consumos energéticos, porque tenciona-se que este considere que apenas existem eventos quando o motor está ligado. Esta funcionalidade é necessária porque neste projeto não se pretende conectar o dispositivo ao painel de controlo do carro. Quando o Separador de eventos considera que existe um evento este passa-o ao Avaliador. No Avaliador pretende-se perceber se o som obtido é normal ou uma anomalia, e com base nesta avaliação enviar o som recebido para a plataforma.

Na plataforma o utilizador poderá observar o comportamento do motor do seu veículo ao longo do tempo, bem como também poderá ouvir as anomalias produzidas pelo mesmo.

O trabalho foca-se principalmente na separação de eventos e na avaliação do som do motor, e para tal é necessária a existência de um conhecimento teórico de algumas características sonoras. Neste projeto analisam-se características como: Zero Crossing Rate, Spectral Centroid, Spectral Rolloff, Spectral Flux e Root Mean Square, para separar o som recolhido e, Spectrogram, Mel-Spectrogram, Chromagram e Mel Frequency Cepstral Coefficients, para a avaliação de eventos. Faz-se de seguida uma descrição dos várias features utilizadas no projeto.

4. IMPLEMENTAÇÃO DO MODELO

Na execução deste projeto escolheu-se a linguagem de programação python para a realização do código, que permite a criação e treino do modelo de redes neuronais. Esta também foi utilizada na implementação do módulo de processamento de sinal. A escolha deve-se ao facto da

linguagem python ser simples e consistente, mas principalmente pelo seu número elevado de bibliotecas, que permitem realizar tarefas complexas de uma forma simples e rápida [19]. Dá-se uso a bibliotecas como o librosa para realizar a análise de áudio, o keras e o tensorflow que possibilitam a utilização de técnicas de machine learning (sem a necessidade de ter um conhecimento aprofundado sobre este tema).

4.1. Processamento

No código python implementado para além do main, existem mais duas threads a funcionar simultaneamente (Apêndice A). Uma resulta do callback da biblioteca PyAudio, que permite receber o áudio obtido no microfone do computador, e outra thread responsável pela separação, classificação de eventos e pelo envio dos sons das anomalias para a plataforma. Estas tarefas têm de estar separadas, porque se estivessem na mesma, enquanto o código realizaria operações sobre o sinal recolhido não estaria a obter o sinal proveniente do microfone.

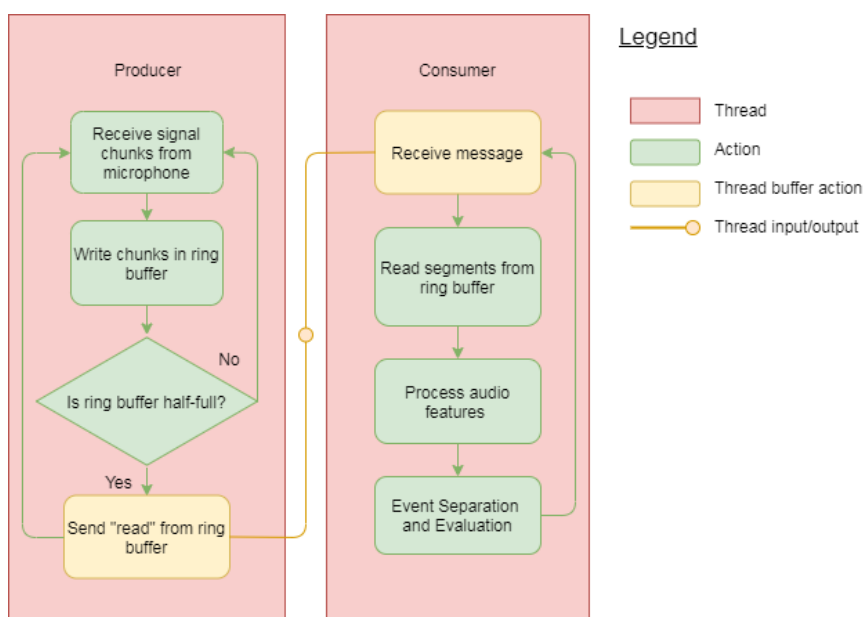


Figura 3 - Funcionamento das threads da plataforma implementada.

Como se observa na Figura 3 enquanto a thread Consumer realiza as operações mais demoradas (processamento das features de áudio, a separação de eventos e avaliação dos mesmos), a thread Producer apenas lê pedaços do sinal proveniente do microfone e passa estes para o buffer. Isto permite à aplicação obter o sinal ao mesmo tempo que realiza operações sobre este. Para poder ter um código a funcionar em tempo real criou-se um buffer. Este não só permite passar os dados recolhidos de uma thread para a outra, como também possibilita escolher o tamanho dos segmentos que se pretende analisar. Para tal, utiliza-se um buffer circular, com dois ponteiros que permitem saber o quão cheio o buffer se encontra, e conseqüentemente alertar a thread consumidora que deve ler o seu conteúdo. Também é muito útil, pois consegue facilmente obter os valores de uma variável array e quando os seus valores são lidos para outro array esses mesmos valores são eliminados do buffer. Além disso, o buffer lança uma exceção em caso de overflow, se os novos valores sobrescrevem os valores anteriores [20].

Na parte de separação de eventos utilizam-se as features RMS, Spectral Rolloff e Spectral Flux. A feature Zero Crossing Rate não é utilizada, porque depois de alguns testes esta demonstra ter a mesma utilidade que a RMS, em termos de verificar se o motor se encontra a funcionar. Devido à maior complexidade da variação e da escolha de um threshold para esta característica acabou-se por optar pela escolha da RMS, que também tem a vantagem de apresentar uma representação do sinal mais intuitiva para o “olho humano”. Também não se utiliza a característica Spectral Centroid por esta avaliar os mesmos aspetos que a Spectral Rolloff, apesar de possuir algumas diferenças. A escolha da Spectral Rolloff deveu-se apenas à observação do comportamento de ambas quando aplicadas a vários sons. Não foi observada uma diferença muito grande entre elas nos testes, como seria expectável. Normalmente a

Spectral Rolloff aparenta ser mais previsível, depois de serem ouvidos os áudios nos quais as duas características são testadas.

É importante referir, que na maior parte das vezes, as features apresentadas encontram-se normalizadas, isto é, as imagens mostradas não apresentam os seus valores reais, e por isso características como a RMS, parecem descrever melhor o sinal do que na realidade o fazem. Apesar da normalização ser muito útil para observar o comportamento das features ao longo do sinal, esta não pode ser utilizada no Separador de eventos, por este realizar o cálculo das features em segmentos com apenas algumas frames. Apresenta-se na Figura 4 as features estudadas são aplicadas a um som produzido por um motor.

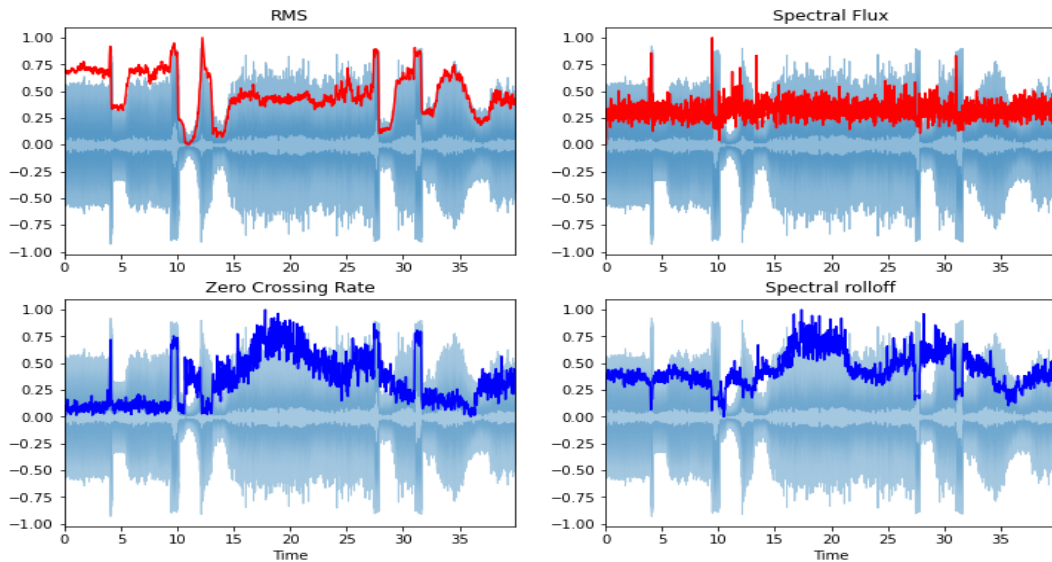


Figura 4 - Normalização de features no fim do processamento de sinal.

4.2. Separação de Eventos

Embora existam muitas possibilidades [5], para utilizar as features escolhidas de uma forma eficiente e lógica construiu-se uma máquina de estados como apresenta a Figura 5.

Como já foi referido, a separação de eventos é realizada através de uma máquina de estados. A máquina em questão possui três estados: Estado 1 - Disposable Sound para o som que não é considerado evento, Estado 2 - Non-Stationary Sound que considera que o som que provém do microfone é candidato a ser de um evento não estacionário e Estado 3 - Stationary Sound que separa eventos considerados estacionários.

A máquina inicia-se pelo estado Estado 1 - Disposable Sound, permanecendo neste estado enquanto considera que o som não tem interesse (ex: motor desligado, ruído de fundo). Neste estado apenas é comparado o threshold da RMS, $rmsTh$, com a RMS do sinal. Se o valor RMS for superior à $rmsTh$ a máquina de estados passa para o estado Estado 2 - Non-Stationary Sound. A máquina passará para Estado 2 - Non-Stationary Sound sempre que considerar que há uma possibilidade do som ser não estacionário, como por exemplo quando o carro inicia ou o condutor toca a buzina. Neste estado continua-se a verificar o valor da RMS, para se saber se o motor continua a funcionar. Este também possui um timer, que é decrementado ao longo do tempo. Inicialmente coloca-se uma margem de tempo no timer, que permite verificar se o evento é não estacionário, esta verificação é realizada ao longo deste estado pela Spectral Flux. Sempre que esta feature ultrapassa o seu threshold, $sFTh$, o timer é atualizado para o seu valor inicial, indicando assim que está a decorrer um evento não estacionário. Quando o timer descrito for decrementado e atingir zero considera-se que se está no evento estacionário e muda-se para o estado Estado 3 - Stationary Sound. A característica que é utilizada para se separar eventos neste estado é a Spectral Rolloff. Esta possui uma margem de frequência que indica se existe ou não uma grande variação de frequência, tanto para valores superiores como inferiores.

Sempre que esta feature atinge um valor fora da sua margem, o seu ponto de referência é atualizado. Caso a RMS seja menor que o seu threshold volta-se para o estado inicial.

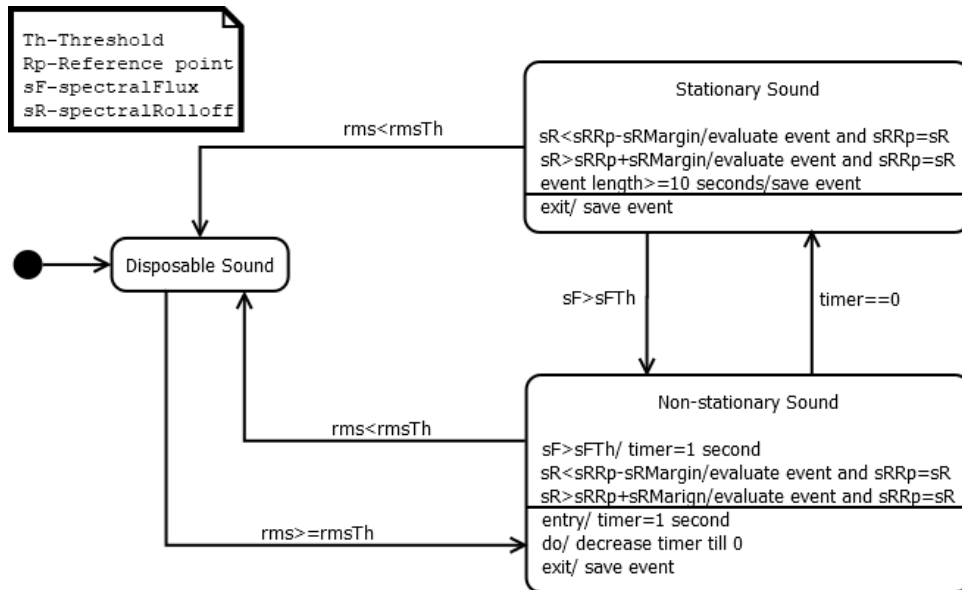


Figura 5 - Diagrama UML da Máquina de Estados.

A máquina inicia-se pelo estado Estado 1 - Disposable Sound, permanecendo neste estado enquanto considera que o som não tem interesse (ex: motor desligado, ruído de fundo). Neste estado apenas é comparado o threshold da RMS, rmsTh, com a RMS do sinal. Se o valor RMS for superior à rmsTh a máquina de estados passa para o estado Estado 2 - Non-Stationary Sound. A máquina passará para Estado 2 - Non-Stationary Sound sempre que considerar que há uma possibilidade do som ser não estacionário, como por exemplo quando o carro inicia ou o condutor toca a buzina. Neste estado continua-se a verificar o valor da RMS, para se saber se o motor continua a funcionar. Este também possui um timer, que é decrementado ao longo do tempo. Inicialmente coloca-se uma margem de tempo no timer, que permite verificar se o evento é não estacionário, esta verificação é realizada ao longo deste estado pela Spectral Flux. Sempre que esta feature ultrapassa o seu threshold, sFTh, o timer é atualizado para o seu valor inicial, indicando assim que está a decorrer um evento não estacionário. Quando o timer descrito for decrementado e atingir zero considera-se que se está no evento estacionário e muda-se para o estado Estado 3 - Stationary Sound. A característica que é utilizada para se separar eventos neste estado é a Spectral Rolloff. Esta possui uma margem de frequência que indica se existe ou não uma grande variação de frequência, tanto para valores superiores como inferiores. Sempre que esta feature atinge um valor fora da sua margem, o seu ponto de referência é atualizado. Caso a RMS seja menor que o seu threshold volta-se para o estado inicial.

Quando a máquina de estados se encontra no estado Estado 3 - Stationary Sound pode-se presumir que nos encontramos num evento estacionário. Assim, continua-se a verificar a RMS e a Spectral Rolloff como no estado anterior, a diferença deste estado para o anterior é que quando a Spectral Flux ultrapassa o seu threshold volta-se para o estado Estado 2 - Non-Stationary. Neste regime, é provável haverem eventos muitos longos, isto é, haver pouca variação das features durante muito tempo, por isso também se segmenta um evento sempre que a sua duração ultrapasse 10 segundos.

Sempre que existe uma deteção de evento ou se muda de estado (exceto na passagem do estado Estado 1 - Disposable Sound para o Estado 2 - Non-Stationary), usa-se a função evaluate_event. Esta função verifica se o evento passado possui uma duração superior a meio segundo, se sim, envia o evento para o classificador e inicia um novo evento, se não, não realiza nenhuma operação. Esta verificação existe para evitar enviar eventos demasiado curtos para o Avaliador, pois muitas vezes as features variam com diferenças de tempo muito pequenas. Também se deve ao facto de frequentemente esta verificação impedir o envio de eventos que

não são do motor do carro. Por exemplo, nos testes constatou-se que o bater de uma porta do carro pode fazer com que o RMS ultrapasse o seu threshold. Como o “bater de porta” é um evento muito curto com esta averiguação este já não seria considerado evento.

4.3. Classificação

Na estimação do tipo de eventos utilizou-se a feature Mel-Spectrogram, devido à suas vantagens logarítmicas em relação ao espectrograma e devido à sua maior informação quando comparado com os MFCCs. Inicialmente começou-se pelo uso dos MFCCs, mas devido à complexidade da computação do mesmo, não se estava a aproveitar todas as suas capacidades. Usou-se a média e o desvio de padrão da imagem apresentada pelos MFCCs. Quando se realizaram testes, com a média e o desvio de padrão, o Mel-Spectrogram foi a feature que mais vezes conseguiu distinguir, de forma correta, o som de motor normal de um com anomalias.

A aplicação da média e o desvio de padrão às features faz perder informação. Realizou-se esta abordagem, pois pretende-se avaliar sons, normalmente, estacionários, devido à elevada complexidade dos modelos alternativos e porque o vetor de features tem que ter comprimento constante, independentemente do evento sonoro detetado.

O modelo de redes neuronais usado é o Dense Only Network, este é do tipo Sequential, isto é, as camadas deste comunicam de forma linear [22]. A primeira camada do modelo é do tipo flatten, esta tem como objetivo tornar os arrays multidimensionais recebidos em array de apenas uma dimensão [23]. Os arrays passados ao modelo têm duas dimensões, sendo que a primeira representa as médias da Mel-Spectrogram em cada frequência e a segunda contém os desvios de padrão da Mel-Spectrogram em cada frequência. A arquitetura implementada possui três camadas do tipo Dense. As duas primeiras camadas têm activation do tipo relu, isto significa que todos os resultados que sejam inferiores a zero são alterados para zero [24]. Também possuem dropout de 0.5, ou seja, metade dos valores obtidos por estas camadas são alterados para zero [25]. Isto permite prevenir o overfitting (sobreajuste) do modelo. A última camada que compõe a arquitetura tem activation softmax, isto é, converte um vetor de valores numa distribuição de probabilidades [24], sendo estas apresentadas como resultado final da avaliação.

O dataset é separado em dois grupos, um de treino e outro para teste (80% treino e 20% teste). Nesta separação é utilizado o modo stratify que, obriga a que a separação do dataset seja uniforme (são retirados 80% para treino de cada classe e 20% para teste de cada classe).

O modelo é treinado em 100 epochs com batches de tamanho 32. Sendo que, batch é um conjunto de amostras que é usado para treinar a rede neuronal. Este pode possuir diversos tamanhos, sendo que os mais utilizados são 32, 64 ou 128. O epoch é um ciclo por todo o dataset de treino. O número de epochs pode ser fixo, mas deve ser orientado com base no erro do conjunto de validação. Um epoch é composto por um ou mais batches [27]. Depois de ser treinado e testado o modelo encontra-se pronto para ser utilizado.

Foram utilizados sons de datasets online como o AudioSet da Google e o ToyADMOS e o MIMII Dataset disponibilizados no DCASE2020 [4]. Porém, devido à escassez e à falta de qualidade dos sons, os autores completaram o dataset realizando um conjunto de gravações reais em motores dos automóveis. O dataset atualmente possui mais de mil sons. Os datasets online foram essenciais para a obtenção de sons de anomalias.

4.4. Plataforma

Os dados recolhidos ao longo do tempo são armazenados numa plataforma online com recurso à Firebase suportada pela Google, que permite guardar ficheiros sem utilização de lógica SQL. Esta ferramenta tem um plano gratuito podendo armazenar até 1 GB de dados e realizar diariamente 50000 leituras e 20000 gravações de documento [28].

Sempre que o Avaliador considera que ocorreu uma anomalia, envia o áudio do evento, com o nome da data e hora atual para um folder da storage do Firebase. Assim os dados podem ser acedidos por qualquer equipamento com acesso à internet. O folder no qual são guardados os ficheiros de áudio dependem do id de cada utilizador, assim é possível guardar na mesma storage os dados de múltiplos utilizadores sem que estes tenham acesso aos dados de outro utilizador.

Com o intuito de apresentar os dados recolhidos, criou-se uma aplicação Web simples através de javascript e HTML na qual é possível criar uma conta para um novo utilizador, fazer login na

mesma, observar o comportamento do motor ao longo do tempo e ouvir o som das anomalias que possam ser produzidas pelo mesmo.

Para gerir a autenticação foram usadas as bibliotecas do Firebase. No caso do python foi necessário o uso da biblioteca pyrebase, que permite trabalhar com o Firebase em python, de uma forma similar a outras linguagens de programação.

5. VALIDAÇÃO E TESTES

O Separador de eventos tem como principais objetivos detetar, quando o motor se encontra a funcionar, isolar sons não estacionários e separar eventos com valores de frequência muito díspares. Para tal, foram testadas as features estudadas com o objetivo de observar o seu comportamento em casos práticos.

Para o primeiro objetivo, todas as características estudadas têm, teoricamente, o potencial de serem úteis, tal como a Figura 5.1 demonstra.

Observa-se que, todas as features apresentadas detetam de forma correta o carro a iniciar. A RMS exhibe valores quase nulos quando o carro não se encontra a funcionar e o Spectral Flux apresenta dois picos no início dos eventos. As características Zero Crossing Rate e a Spectral Rolloff manifestam valores muito elevados quando o carro está desligado (devido à presença de frequências elevadas do ruído). Apesar disto, a Spectral Flux demonstra não conseguir representar quando o motor é desligado e a Spectral Rolloff apresenta uma grande variação quando o carro se encontra ligado. A Figura 6 tem o objetivo de mostrar o comportamento das mesmas features focalizando num carro a desligar.

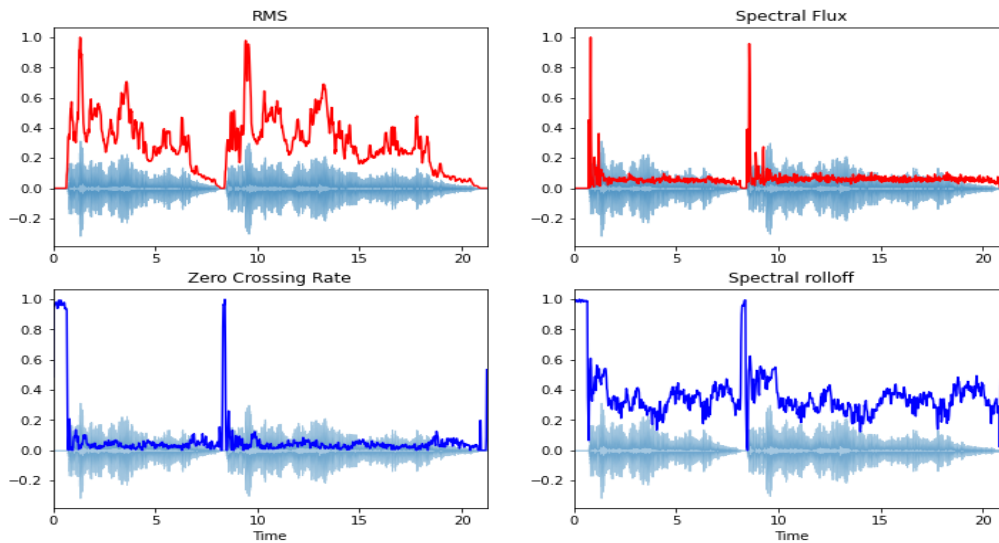


Figura 6 - Features aplicadas a um sinal de um carro a iniciar duas vezes.

O Avaliador tem como tarefa classificar o evento que recebe. Para obter os melhores resultados possíveis é necessário escolher a característica que distingue maior número de vezes, corretamente, as anomalias dos sons normais. As features analisadas neste estudo são os MFCCs e a Mel-Spectrogram. A razão desta escolha prende-se com o facto de apresentarem geralmente bons resultados e também porque como resultam em imagens podem ser utilizadas em CNNs, no nosso estudo subsequente. Estas features são avaliadas com base na sua precision, recall e f1-score. Como apresentado na Figura 7 a precision representa a proporção de identificações positivas que são corretas, a recall reflete a proporção de número de casos positivos que são identificados corretamente e a f1-score é obtida através da média ponderada da precision e da recall.

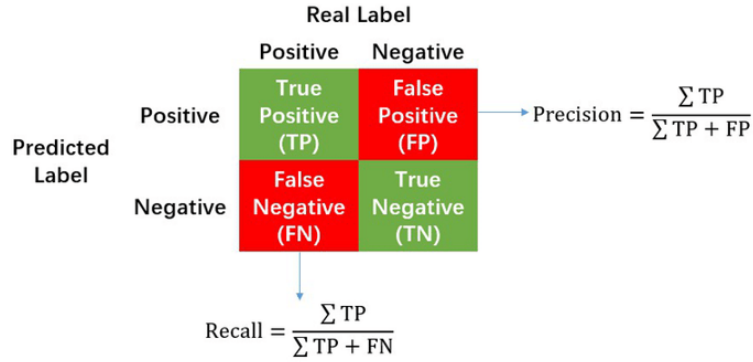


Figura 7 - Cálculo do indicador *precision* e da *recall*.

Segue-se a Tabela 1 com os resultados das features testadas.

Tabela 1 Percentagem de acertos dos Modelos testados

Model MFCCs	precision	recall	f1-score
anomaly	0.99	1.00	0.99
engine	0.99	0.98	0.99
Model Mel-Spectrogram			
anomaly	1.00	0.99	0.99
engine	0.99	1.00	0.99

Observa-se que as features utilizadas, Mel-Spectrogram e os MFCCs, apresentam muito bons resultados. Mas é importante salientar que, todos os resultados finais encontram-se inflacionados, pois estes testes foram realizados a sons que são muito similares aos do dataset utilizado para o treino dos modelos, devido à escassez de sons de anomalias.

Tabela 2 Classificação de anomalias pelos modelos

Sons	MFCCs		Mel-Spectrogram	
	Engine	Anomaly	Engine	Anomaly
Motor a funcionar	0.01	0.99	0.99	0.01
Motor inundado	0.01	1.00	0.00	1.00
Motor com a biela partida	0.00	1.00	0.00	1.00
Motor gripado	0.00	1.00	0.00	1.00
Motor com junta de velocidade constante partida	0.00	1.00	0.00	1.00
Motor com transmissão partida	0.99	0.01	0.00	1.00

Os resultados mostram que a feature Mel-Spectrogram é mais indicada para classificação dos eventos destes tipos, embora marginalmente.

Após a aplicação ter sido implementada foram realizados inúmeros testes compostos por um motor a funcionar de forma correta e com sons de anomalias sobrepostos a este. No melhor dos casos a aplicação conseguiu identificar duas das cinco anomalias sobrepostas ao motor. Também se verificou que o som do motor a funcionar corretamente nunca foi considerado anomalia.

Realizaram-se testes ao Separador de eventos utilizando duas abordagens: uma abordagem simples de segmentação de som em intervalos de cinco segundos e outra com a máquina de estados. Concluiu-se que quanto menor é o evento da anomalia maior é importância de utilizar o Separador de eventos e que a separação de eventos facilita a classificação de anomalias, pois sem a sua utilização o Avaliador tem de classificar segmentos, muitas vezes, compostos por mais de um evento.

3. CONCLUSÕES

O projeto utiliza os sons produzidos por motores de viaturas com o objetivo de identificar se estes possuíam alguma anomalia, seguindo uma abordagem de 'manutenção preditiva'.

Foi necessária a criação de um dataset composto por sons provenientes do AudioSet da Google, ToyADMOS disponibilizados no DCASE2020 e por gravações realizadas pelos autores a um conjunto de viaturas disponíveis.

Desenvolveu-se um algoritmo para separação de eventos baseado em um máquina de estados que utiliza uma abordagem de pré-classificação sobre um conjunto de features. Observou-se, que a feature RMS é a mais indicada para verificar se o motor do carro se encontra a funcionar, que a Spectral Flux permite a identificação de sons não estacionários e que a Spectral Rolloff descreve melhor a variação de frequência produzida pelo motor.

Na classificação constatou-se, que a Mel-Spectrogram é a mais eficaz na identificação de anomalias, como se observa na Tabela 2, e o que o seu tempo de execução não impossibilita a aplicação de funcionar em tempo real.

Verificou-se que a separação de eventos contribui para um melhor desempenho do classificador de eventos, pois sem este o Avaliador teria de classificar segmentos de som com mais do que um evento ou com eventos partidos a meio.

Os resultados obtidos são excelentes revelando taxas de identificação de anomalias elevadas, mais de 90%, para os indicadores precision, recall e f1-score, para determinados tipos de eventos sonoros, mas não podem ser vistos de uma forma quantitativa, uma vez que, por escassez de dados com anomalias, os sons utilizados são muito similares. Contudo, o sistema de classificação está muito dependente da qualidade dos eventos sonoros (segmento de áudio), quer em termos de separação quer em termos de ruído de fundo inerente a este ambiente. Outro problema relevante é a sobreposição de tipos de sons (sons de vários sistemas com anomalias).

Este estudo encontra-se em desenvolvimento, assim, os resultados são ainda preliminares e necessitam ser melhorados. Neste sentido, iremos aplicar técnicas de redução/cancelamento de ruído, melhorar o modelo da máquina de estados para separação de eventos sonoros e estudar outras abordagens de classificadores de redes neuronais do tipo CNNs.

AGRADECIMENTOS

Este trabalho está a ser suportado pelo Laboratório de Áudio e Acústica do Instituto Superior de Engenharia de Lisboa, ISEL, LAA (<https://acusticaudiolab.isel.pt>)

REFERENCIAS

- [1] Manutenção Preditiva vs. Preventiva: Qual É a Diferença?
<https://blog.infraspeak.com/pt-pt/manutencao-preditiva-vs-preventiva/>
- [2] Abeßer, J. (2020). A review of deep learning based methods for acoustic scene classification. Applied Sciences, 10(6).
- [3] Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring
<http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds-results>
- [4] Müller, R., Ritz, F., Illium, S. and Linnhoff-Popien, C. (2021). Acoustic Anomaly Detection for Machine Sounds based on Image Transfer Learning. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2, pages 49-56 ISBN: 978-989-758-484-8
- [5] Rushe, E. and Namee, B. M. (2019). Anomaly detection in raw audio using deep autoregressive networks. ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3597–3601, 2019.
- [6] Marchi, E., Vesperini, F., Eyben, F., Squartini, S., and Schuller, B. (2015). A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 1996–2000. IEEE.
- [7] Duman, T. B., Bayram, B., and Ince, G. (2019). Acoustic anomaly detection using convolutional autoencoders in industrial processes. In International Workshop on Soft Computing Models in Industrial and Environmental Applications, pages 432–442. Springer.