TECNIACÚSTICA
CUENCA 2023

# OTIMIZAÇÃO DE HÍPER-PARÂMETROS EM CLASSIFICADORES DE SOM

*André Silva Mendes*[1]
*Paulo M. Trigo*[1]
*Joel Preto Paulo*[1]

[1]ISEL – Instituto Superior de Engenharia de Lisboa, Rua Conselheiro Emídio Navarro, 1; 1959-007 Lisboa; Portugal

## RESUMO

O contributo deste trabalho é o de explorar o espaço de HyP (híper-parâmetros) na procura dos valores que optimizem o desempenho de todo o processo de classificação automática de sons impulsivos. Esse processo inicia na geração do conjunto de dados ("dataset") a processar, prossegue com a aprendizagem de modelos de classificação e termina com a avaliação dos modelos aprendidos.

O conceito de HyP refere-se, em geral, aos valores escolhidos para configuração de determinado algoritmo a executar num processo de Aprendizagem Automática. Esse algoritmo tem como "input" um "dataset" (os excertos de áudio) que o algoritmo processa visando aprender (descobrir) padrões contidos nesses dados. Neste trabalho propomos estender o conceito de HyP de modo a englobar também a construção do próprio "dataset". Temos 2 conjuntos de HyP: a) os usados para geração do "dataset" (HyP_data), e b) os usados pelo algoritmo que processa o "dataset" (HyP_learn). De modo mais formal, temos HyP = {HyP_data, HyP_learn}, e HyP_data = {HyP_data_global (e.g.: Event-Length, Number-of-Segments, Bandwidth), HyP_data_feature (e.g.: Root-Mean-Square, Band-Energy-Ratio, Spectral-Centroid, Zero-Crossing-Rate)}.

A validação do processo irá recorrer a áudio extraído de vídeos de competições de eventos desportivos, nomeadamente, ténis e padel onde se procuram identificar os sons de pancadas da raquete na bola.

## ABSTRACT

The contribution of this work is to explore the HyP (hyperparameter) space in search of values that optimize the performance of the entire process of automatic classification of impulsive sounds. This process begins with the generation of the dataset to be processed, continues with the training of classification models, and ends with the evaluation of the learned models.

The concept of HyP generally refers to the values chosen for configuring a specific algorithm to be executed in a Machine Learning or Deep Learning process. This algorithm takes as its input a dataset (audio excerpts), which the algorithm processes in order to learn (discover) patterns contained within that data. In this work, we propose to extend the concept of HyP to also encompass the construction of the dataset itself. As such, we have two sets of HyP: a) those used for generating the dataset (HyP_data), and b) those used by the algorithm that processes the dataset (HyP_learn). In a more formal manner, we have HyP = {HyP_data, HyP_learn}, and HyP_data = {HyP_data_global (e.g., Event-Length, Number-of-Segments, Bandwidth), HyP_data_feature (e.g., Root-Mean-Square, Band-Energy-Ratio, Spectral-Centroid, Zero-Crossing-Rate)}.

The validation of the process will involve audio extracted from videos of sports events, specifically tennis and paddle tennis, where the goal is to identify the sounds of racket hits on the ball.

***Keywords*** **—** Machine Learning, Deep Learning, Audio Analysis, Sound Event Dataset, Hyperparameter Optimization.

## 1. INTRODUÇÃO

Em determinada modalidade desportiva, a performance de um atleta pode melhorar quando ele tem uma perspetiva externa da sua atividade. Essa perspetiva pode ser dada pelo seu treinador. Complementarmente, ou mesmo quando a supervisão humana não é possível, essa análise só pode ser feita vendo ou ouvindo novamente a gravação da atividade. Como tal, é de grande utilidade uma ferramenta para a deteção e identificação de sons impulsivos aplicada, em particular, a eventos desportivos (por exemplo, na modalidade de ténis e padel, onde se procura identificar os sons de pancadas da raquete na bola). Neste trabalho, propõe-se o desenvolvimento de um sistema para identificação automática de sons impulsivos aplicado a eventos desportivos, ou seja, um assistente de treinador.

Um som impulsivo sonoro pode ser caracterizado pela sua curta duração, onde a sua energia tem um crescimento rápido inicialmente (“onset or attack”) seguido de um decaimento (“decay/release”) que depende do ambiente acústico onde o som é gerado. Efetivamente, a reverberação do espaço tem uma grande influência na duração do som impulsivo. Todavia, em detalhe, existem outras características que é necessário extrair do sinal acústico de

forma a distinguir e identificar um som impulsivo, nomeadamente, as tonais. Todas essas características, assim como as variáveis associadas à escolha do modelo de classificação a utilizar, representam um vasto conjunto de híper-parâmetros (HyP), que geram espaço de pesquisa de grande dimensão. É sabido que, por maior que seja o conhecimento e experiência na área do áudio e da Inteligência Artificial, é praticamente impossível prever, à partida, a configuração dos HyP mais adequada ao problema. Além disso, os algoritmos de aprendizagem profunda ("deep-learning") têm demonstrado bons resultados, contudo, esses algoritmos têm um espaço de pesquisa ("search space") ainda maior do que os algoritmos tradicionais de Aprendizagem Automática ("machine learning"). Nesse sentido, este trabalho está focado na exploração do espaço de HyP na procura dos valores que optimizem o desempenho de todo o processo de classificação automática de sons impulsivos.

## 2. ESTADO DA ARTE

A utilização de uma abordagem de Inteligência Artificial (AI) no campo da deteção e identificação de sinais acústicos, mais concretamente, na classificação de sons impulsivos, tem tido bastante investigação e avanços significativos [1, 2, 3, 4, 5, 6, 7, 8]. Têm sido propostas varias técnicas para a detecção e identificação de sons impulsivos. No trabalho [1] é feita uma análise sobre algumas das técnicas mais utilizadas, e é proposto um método caracterizado por duas fases: primeiro detecta os sons impulsivos e, em seguida, a janela encontrada é passada para a parte de identificação. Técnicas de machine learning e deep-learning, têm provado conseguir classificar sons impulsivos com boa precisão. Uma das técnicas mais utilizadas é a aprendizagem supervisionada, onde os modelos são treinados com conjuntos de dados etiquetados ("labeled datasets"). Com o avançar da tecnologia e a disponibilidade de melhores recursos computacionais, modelos mais complexos, nomeadamente, redes neuronais profundas, podem ser treinados para se alcançar o estado da arte sob o ponto de vista do desempenho, na classificação de sons impulsivos. Atualmente, existem plataformas que permitem criar, treinar e lançar modelos de Machine Learning e Deep Learning na nuvem ("cloud-based solutions"), como é o caso da Amazon SageMaker, da Microsoft Azure Machine Learning, da Google Cloud AI Platform, da H2O.ai, entre outras. Aquelas plataformas disponibilizam os recursos computacionais necessários para lidar de forma eficiente com conjuntos de dados de grande dimensão. Existem também estruturas/bibliotecas de software, tais como, o TensorFlow (desenvolvido pela Google), o PyTorch (desenvolvido pela Facebook), Apache MXNet, e o Scikit-Learn, que permitem a criação de algoritmos de machine learning e deep-learning do zero.

As Redes Neuronais Convolucionais (CNNs) têm sido uma das técnicas mais utilizadas na classificação de sons. Um exemplo disso é o trabalho [9]. Este tipo de rede neuronal é o mais adequado atualmente para trabalhar com classificação de imagens. A ideia é representar o som através de imagem, e para isso, considera-se a imagem do espectrograma ou Mel Frequency Cepstral Coefficients (MFCCs). No artigo [10] é demonstrado que este tipo de redes é capaz de obter excelentes resultados na classificação de áudio quando comparado com uma rede neuronal simples ("fully connected"), ou com arquiteturas anteriores de classificação de imagem.

Para explorar o espaço de híper-parâmetros, existem atualmente várias técnicas e algoritmos. As abordagens mais conhecidas e tradicionais são a grid-search e a random-search, as quais estão explicadas na literatura sobre optimização de híper-parâmetros ("Hiperparameter Optimization") [11, 12]. Com o aumento do volume de dados e do espaço de HyP, a técnica grid-search tem um custo computacional muito elevado [12, 13]. Testar uma única configuração de híper-parâmetros em datasets de grandes dimensões, pode, nos dias de hoje, facilmente ultrapassar várias horas e levar até vários dias [13]. Existem outras abordagens mais avançadas para uma exploração mais eficiente do espaço de híper-parâmetros, como a Otimização Bayesiana (Bayesian-optimization) e o HyperBand, que aceleram as avaliações das diferentes configurações de híper-parâmetros em comparação com os métodos exaustivos, como o grid-search [11]. O algoritmo Bayesian-optimization é considerado um dos algoritmos que representam o estado da arte na optimização de híper-parâmetros [12, 13].

Nos últimos anos, foram lançadas várias bibliotecas e estruturas que implementam aquelas técnicas. Algumas das bibliotecas mais populares são: Scikit-optimize; Hyperopt; Keras tuner; e Optuna.

A classificação de sons impulsivos é uma área de investigação em pleno desenvolvimento, com vários desafios a serem enfrentados. É necessário lidar com conjuntos de dados desequilibrados, enquanto a quantidade e qualidade dos dados deve permitir manter a robustez do modelo em diferentes condições (e.g. ambiente indoor; outdoor), e ainda, lidar com o ruído do mundo real.

## 3. METODOLOGIA

O sistema para identificação automática de sons impulsivos é implementado com foco na exploração do espaço de híper-parâmetros, na procura dos valores que otimizem o desempenho de todo o processo de classificação automática de sons impulsivos. A implementação contempla duas fases distintas: uma fase em que são gerados os datasets considerando os HyP_data e guardando-se os parâmetros que deram origem a cada dataset; e uma outra fase que é o processamento de cada dataset usando os HyP_learn.

Com esta metodologia pretende-se estudar o impacte de diferentes características do dataset no desempenho do modelo. Como tal, é essencial que os conjuntos de dados sejam gerados automaticamente e com código genérico.

O desenvolvimento do sistema é feito usando a linguagem de programação Python. A escolha foi feita considerando o vasto ecossistema de bibliotecas e frameworks desenhadas especialmente para o desenvolvimento de modelos de Inteligência Artificial (e.g. TensorFlow; Keras; Scikit-Learn); as bibliotecas disponíveis para o processamento de sinais de áudio (e.g., Librosa); e as bibliotecas para manipulação e processamento de dados (e.g. NumPy; Pandas).

### 3.1. Recolha de dados

Os dados utilizados na construção dos datasets provêm de vídeos de competições (profissionais ou amadoras) de eventos desportivos, nomeadamente, ténis e padel. Desses vídeos, é extraído o áudio e é gravado em formato wav (waveform), uma vez que este é um formato de arquivo sem perdas e, como tal, a representação mais próxima do áudio original.

Existem diversos conjuntos de dados (datasets) de áudio disponíveis na internet para fins de pesquisa e desenvolvimento na área da inteligência artificial e processamento de sinais, em aplicações como, processamento de fala, reconhecimento de voz, classificação de sons ambientais, entre outros. Alguns exemplos: URBAN-SED; ESC-50; DCASE 2016; Freesound; e MIVIA Audio Events Dataset. Considerando os sons impulsivos em particular, os dados mais comuns disponíveis são de sons de disparos de arma (“gun shots”). Assim, não sendo fácil encontrar conjuntos de dados específicos de áudio de jogos de padel disponíveis publicamente, é necessário criar manualmente o próprio conjunto de dados, através da captação directa de vídeos, ou a recolha de vídeos já gravados e disponíveis publicamente – no YouTube por exemplo.

No caso de ser feita a captação manual de vídeos, é utilizada câmara com microfone externo, e os sinais de áudio são capturados a uma alta frequência de amostragem (“sampling rate”) de forma a preservar-se a fidelidade da natureza impulsiva do som. No estudo [4] por exemplo, o áudio foi capturado a uma frequência de amostragem de 204,8 kHz.

O conjunto total dos vídeos deve representar variabilidade das condições com efeito nas características acústicas. Ou seja, ambientes interiores e exteriores (indoor e outdoor), e diferentes condições, tais como, humidade no ar, nível de ruído ambiental, e diferentes distâncias do microfone que correspondem a diferentes níveis da relação sinal-ruído.

### 3.2. HyP (híper-parâmetros)

O conceito de HyP refere-se, em geral, aos valores escolhidos para configuração de determinado algoritmo a executar num processo de Aprendizagem Automática [11]. Esse algoritmo tem como "input" um "dataset" (os excertos de áudio) que o algoritmo processa visando aprender (descobrir) padrões contidos nesses dados. O "dataset" é, portanto, a base (a "matéria-prima") de todo o processo. Neste trabalho propomos estender o conceito de HyP de modo a englobar também a construção do próprio "dataset".

Temos 2 conjuntos de HyP: a) os usados para geração do "dataset" (HyP_data), e b) os usados pelo algoritmo que processa o "dataset" visando aprender a partir desses dados (HyP_learn); de modo mais formal, temos HyP={HyP_data, HyP_learn}. Neste trabalho, os dados representam excertos de áudio, pelo que HyP_data separa-se ainda em dois sub-conjuntos: a) os que estão ligados à amostragem do sinal de áudio, e como tal, são "globais" a todo o restante processamento (HyP_data_global), e b) os que são usados na extração de características ("features") desse sinal de áudio (HyP_data_feature); temos então HyP_data={HyP_data_global, HyP_data_feature}.

Os esquemas de HyP (HyP_data e HyP_learn) a considerar, são definidos e representados formalmente numa estrutura de dados persistente e separada do código, contendo o intervalo de valores dos HyP. Assim, o utilizador pode alterar aquele intervalo de valores, e reiniciar o sistema, sem a necessidade de alteração do código.

### 3.3. Extração de características do áudio

Com base na noção geral do que distingue um som impulsivo, são identificadas as características potencialmente relevantes. Essas características são extraídas do áudio a partir da sua representação em vários domínios: no domínio do tempo; da frequência; e na representação tempo-frequência.

É feito um varrimento sobre o áudio com uma janela temporal de dimensão fixa – ‘Event_length’ – de acordo com a duração do evento sonoro que se pretenda considerar. Esse varrimento é feito em várias iterações, onde em cada iteração a janela desliza um número específico de amostras – ‘Number_of_shifted_samples’. A janela é dividida em segmentos – ‘Number_of_segments’ – e a extração das características é feita sob as amostras abrangidas por cada segmento (e não sob o total de amostras da janela), de forma a preservar o contexto temporal. O número de amostras abrangidas depende também da frequência de amostragem do áudio – ‘Bandwidth’. Esta técnica é inspirada no trabalho desenvolvido em [14]. A Figura 1 ilustra o processo de varrimento sobre o áudio para extração de características.

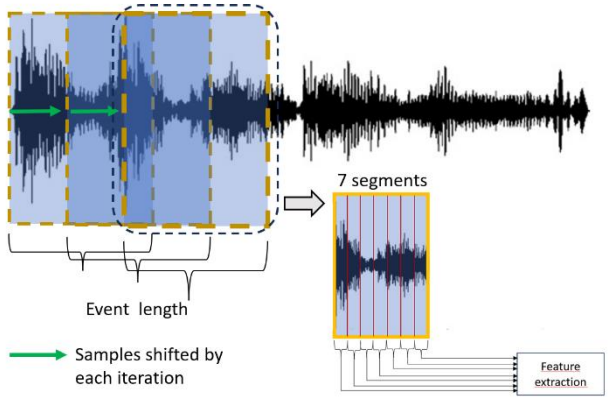


**Figura 1.** Varrimento sobre o áudio para extração de características, com um exemplo de Number_of_segments = 7**.**

### 3.4. Conjunto de dados (Dataset)

Propõe-se, neste trabalho, estender o conceito de HyP de modo a englobar também a construção do próprio "dataset". Assim, e tal como explicado no capítulo 3.2 deste documento, podem ser explorados diferentes esquemas de HyP_data (e.g. diferentes features; diferentes intervalos de valores dos parâmetros do varrimento do áudio para a extração de features), sem a necessidade de alteração do algoritmo do sistema. Os datasets são gerados de raiz, automaticamente, com código genérico e apenas dependente da declaração dos HyP_data. Cada configuração de valores dos HyP_data dá origem a um dataset diferente.

A matriz de características é construída com os valores obtidos através do processo descrito no capítulo 3.3 deste documento.

Sobre os dados recolhidos para a geração dos datasets, é feita a anotação manual de eventos sonoros através da perceção do ouvido humano sobre o som impulsivo em questão. Essa anotação é usada na criação automática do vector de classes. Assim, temos aprendizagem supervisionada, guiada pela relação entre o ouvido humano e a construção do dataset.

Para equilibrar o dataset, é considerada a técnica de sub-amostragem, uma vez que, o número pancadas da raquete na bola é muito inferior ao ruído de fundo. O objetivo é reduzir o número da classe em maioria [15], neste caso, a classe do ruído de fundo.

Para gerar os dataset, o processo proposto pode ser descrito pela Figura 2.

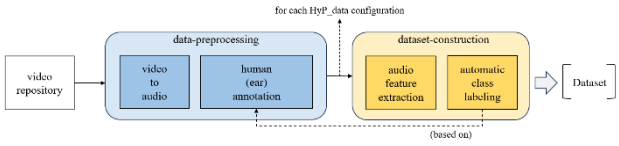


**Figura 2.** Processo de construção do dataset**.**

### 3.5. Modelo

Conforme referido anteriormente, é praticamente impossível prever, à partida, a configuração dos HyP mais adequada ao problema. O sistema explora automaticamente o espaço de HyP_learn definido, alimentado pelos datasets gerados.

É utilizada a técnica da validação cruzada (cross-validation) para avaliar a capacidade de generalização do modelo, e para evitar uma medição enviesada do desempenho (resultante de um possível conjunto de teste com exemplos mais fáceis de classificar). É também utilizada a técnica de Stratified K-Fold, de forma a preservar uma distribuição idêntica de exemplos de cada classe em cada fold, em relação ao dataset original.

Para explorar o espaço de HyP_learn, são utilizadas as técnicas grid-search e Bayesian-optimization, sendo o espaço de pesquisa definido no esquema HyP_learn. A técnica Bayesian-optimization é especialmente útil quando a busca é realizada em híper-parâmetros que têm espaços de valores contínuos, como é o exemplo do 'Learning-rate' das Feed-forward Neural Network.

Quando o mesmo procedimento de validação cruzada e o mesmo dataset são usados tanto para ajustar (os HyP_learn) como para selecionar um modelo, é provável que isso leve a uma avaliação otimisticamente tendenciosa do desempenho do modelo [16]. Para evitar esse problema, e obter uma medida de desempenho e capacidade de generalização mais realista do modelo, neste trabalho é utilizada a técnica Nested Cross-Validation (também conhecida por ‘double cross-validation’). Esta técnica combina duas camadas de validação cruzada: uma camada externa para avaliar o desempenho geral do modelo, e uma camada interna para optimizar os HyP_learn do modelo. A camada interna é ‘aninhada’ (“nested”) na camada externa. A implementação desta técnica consiste em dividir o dataset em K folds para a camada externa (“outer cross-validation”), e com o conjunto de treino de cada fold (da camada externa) é executada a camada interna, na qual é feita a optimização dos HyP_learn. Esse conjunto (set) da camada interna é dividido noutros k folds (“inner cross-validation”) com conjunto de treino e conjunto de validação. Assim, a avaliação de cada configuração de HyP_learn não tem a oportunidade de se superajustar ao dataset original, pois está exposta apenas a um subconjunto do conjunto de dados fornecido pelo procedimento de validação cruzada externa. Nesta implementação é utilizada a biblioteca Scikit-learn, em particular, as funções ‘GridSearchCV’ e ‘BayesSearchCV’, para a camada interna. Na camada externa é avaliado, com o conjunto de teste, o desempenho geral do modelo otimizado (pela camada interna). No final, o desempenho apresentado será a média dos resultados dos K folds da camada externa. Na Figura 3 está representada uma ilustração demonstrativa

da técnica Nested-cross-validation, com 5 folds na camada externa e 3 folds na camada interna. É também possível visualizar, na Figura 3, o conjunto de dados de validação na camada interna (associado à avaliação de híper-parâmetros na fase de treino), e o conjunto de dados de teste na camada externa (associado à avaliação do modelo).

Antes do dataset ser processado pelo modelo, é usada uma técnica de pré-processamento para standardizar a escala das features. É usada a função 'StandardScaler' da biblioteca 'sklearn.preprocessing', para remover a média e dimensionar o valor das features de acordo com a variância.

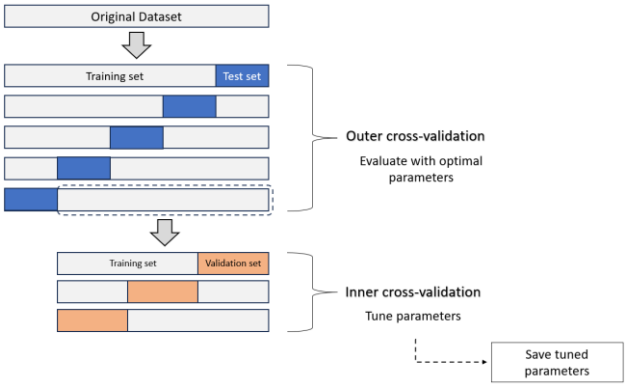


**Figura 3.** Técnica Nested-cross-validation, com 5 folds na camada externa e 3 folds na camada interna**.**

Para automatizar o passo de pré-processamento ('StandardScaler') seguido do classificador, foi criado um Pipeline (da biblioteca sklearn.pipeline).

No final do processo, são apresentados os resultados do desempenho de todos os modelos, assim como, os melhores híper-parâmetros de cada modelo.

## 4. DESENVOLVIMENTO DO TRABALHO

Neste trabalho o objetivo de aprendizagem é o da identificação automática de sons impulsivos relacionados com atividades desportivas.

De forma a realizar-se um teste e obter-se os primeiros resultados intermédios, e seguindo a metodologia descrita no capítulo 3 deste documento, foram considerados os HyP_data_global = {Event_length, Number_of_segments, Bandwidth, Number_of_shifted_samples}, e os HyP_data_feature = {Onset-Detect; Root-Mean-Square, Band-Energy-Ratio, Spectral-flux}. A escolha de alguns daqueles HyP_data foi baseada em trabalho relacionado [14], onde a sua utilização teve bons resultados. Foi testado apenas um híper-parâmetro, o 'Number_of_shifted_samples', com o espaço de pesquisa de valores [4096, 2048, 1024], o que resultou em 3 configurações de HyP_data, e assim, foram gerados 3 datasets (descritos na Tabela 1 como 'DS-1', 'DS-2' e 'DS-3' respetivamente). Como fonte de dados para a geração dos datasets, foram recolhidos vídeos de jogos de padel que totalizam um tempo de gravação de aproximadamente 29 minutos. A amostragem (Bandwidth) foi feita a 44100Hz; considerou-se um 'Event_length' de 0.5 segundos; e o 'Number_of_segments' foi ajustado com a variação de 'Number_of_shifted_samples'. Com as configurações acima indicadas, foram gerados datasets com cerca de 8000, 15800 e 31700 instâncias respetivamente. Relativamente aos HyP_learn, foram considerados dois algoritmos para a classificação – Feed-forward Neural Network e Support Vector Machine – descritos na Tabela 1 como 'MLP' e 'SVM' respetivamente, e consideraram-se alguns dos HyP_learn pertencentes àqueles algoritmos, tendo-se definido um intervalo de valores para cada um deles. A escolha dos algoritmos MLP e SVM para os primeiros testes deveu-se ao facto de eles terem apresentado bons resultados em trabalhos relacionados [14, 9, 1, 2, 3, 5, 8]. Para a técnica Nested-cross-validation foi definida uma configuração com 10 folds na camada externa e 3 folds na camada interna, sendo que estes valores podem também ser considerados HyP_learn. Na Tabela 1 são mostrados os resultados com as técnicas grid-search e Bayesian-optimization, considerando o espaço de busca definido na Tabela 2.

**Tabela 1.** Resultados de alguns dos primeiros testes preliminares. Legenda: lr - 'learning_rate'; mi - 'max_iter'; BS - Bayesian Search; GS - Grid Search.

| Dataset | Modelo | Score (Accuracy) | Mellhores HyP_learn |
|---|---|---|---|
| DS-3 | MLP | 0.930653 | 'lr'0.000600:, 'mi': 76. (GS) |
| DS-3 | MLP | 0.929235 | 'lr': 0.001036, 'mi': 67. (BS) |
| DS-3 | SVM | 0.928574 | 'C': 31. (BS) |
| DS-3 | SVM | 0.928574 | 'C': 31. (GS) |
| DS-2 | MLP | 0.921098 | 'lr': 0.001360, 'mi': 44. (BS) |
| DS-2 | MLP | 0.920090 | 'lr': 0.0016, 'mi': 36. (GS) |
| DS-2 | SVM | 0.919018 | 'C': 19. (GS) |
| DS-2 | SVM | 0.918640 | 'C': 23. (BS) |
| DS-1 | MLP | 0.900521 | 'lr': 0.0041, 'mi': 86. (GS) |
| DS-1 | MLP | 0.899885 | 'lr': 0.002799, 'mi': 78. (BS) |
| DS-1 | SVM | 0.894536 | 'C': 36. (BS) |
| DS-1 | SVM | 0.894408 | 'C': 28. (GS) |

Este trabalho encontra-se em desenvolvimento. Contamos ter mais resultados na data da conferência.

**Tabela 2.** Espaço de busca de HyP_learn para os testes preliminares.

| Model | HyP_learn | Range of values (start, stop, step (if applicable)) |
|---|---|---|
| MLP | learning rate | 0.0001, 0.01, step = 0.0005 for the grid-search |
| MLP | max iterations | 1, 100, step = 5 for the grid-search |
| SVM | C | 1, 40, step = 3 for the grid-search |

## 5. CONCLUSÕES

Os testes iniciais permitem algumas conclusões preliminares. Desde logo, sobre os datasets gerados a partir das várias configurações de HyP_data – em particular, a variação do 'Number_of_shifted_samples' – verifica-se que o varrimento sobre o áudio com um deslizamento menor da janela resulta num dataset com mais instâncias, e o modelo tem melhor desempenho. Esse melhor desempenho do modelo pode justificar-se não apenas pela dimensão do dataset, mas também pelo significado de um deslizamento menor da janela, ou seja, uma maior probabilidade de captar a janela temporal exata da ocorrência do evento. Sobre os algoritmos de classificação testados – MLP e SVM – o resultado mostra um desempenho equivalente, embora com ligeira vantagem do MLP, pelo que terão de ser realizados testes mais completos e com todos os HyP_learn daqueles algoritmos. Quanto às técnicas de pesquisa de híper-parâmetros – grid-search e Bayesian-optimization – verificaram-se as suas vantagens e desvantagens já documentadas na literatura [11, 13], e nesse sentido, a Bayesian-optimization tem mostrado ser a técnica mais adequada para este problema, uma vez que lidamos com uma grande dimensão do espaço de configurações.

## 6. REFERÊNCIAS

[1] Arslan, Yüksel (2017) "Impulsive Sound Detection by a Novel Energy Formula and its Usage for Gunshot Recognition" *arxiv.org* https://doi.org/10.48550/arXiv.1706.08759.

[2] Rabaoui, A., H. Kadri, and N. Ellouze (2008) "New approaches based on One-Class SVMs for impulsive sounds recognition tasks" *Published in: 2008 IEEE Workshop on Machine Learning for Signal Processing*, DOI https://doi.org/10.1109/MLSP.2008.4685494.

[3] Suliman, Azizah, Batyrkhan Omarov, and Zhandos Dosbayev (2020) "Detection of impulsive sounds in stream of audio signals" *Published in: 2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, DOI https://doi.org/10.1109/ICIMU49871.2020.9243540.

[4] Tardif, Bruno, David Lo, and Rafik Goubran (2021) "Gunshot Sound Measurement and Analysis" Published in: 2021 IEEE Sensors Applications Symposium (SAS) DOI https://doi.org/10.1109/SAS51076.2021.9530145.

[5] Hrabina, Martin, and Milan Sigmund (2018) "Gunshot recognition using low level features in the time domain" *Published in: 2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA)*, DOI https://doi.org/10.1109/RADIOELEK.2018.8376372.

[6] Nalla, Ravali, Macarena Varela, and Marc Oispuu (2021) "Evaluation of Image Classification Networks on Impulse Sound Classification Task" *Published in: 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, DOI https://doi.org/10.1109/MFI52462.2021.9591202.

[7] Smailov, Nurzhigit; Dosbayev, Zhandos; Omarov, Nurzhan; Sadykova, Bibigul; Zhekambayeva, Maigul; et al (2023) "A Novel Deep CNN-RNN Approach for Real-time Impulsive Sound Detection to Detect Dangerous Events" *International Journal of Advanced Computer Science and Applications, West Yorkshire*, DOI:10.14569/IJACSA.2023.0140431.

[8] Ahmed, Talal, Momin Uppal, and Abubakr Muhammad (2013) "Improving efficiency and reliability of gunshot detection systems" *Published in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, DOI https://doi.org/10.1109/ICASSP.2013.6637700.

[9] Boixeda, Martí Bolet, and Elisa Sayrol (2019) "Urban Sounds Classification using Deep Learning" *Universitat Politècnica de Catalunya*, Thesis, Degree in Telecommunications Engineering.

[10] S. Hershey, S. Chaudhuri, Ellis, DP, Gemmeke, JF, et al. (2017) "CNN Architectures for Large-Scale Audio Classification" *arxiv.org* https://doi.org/10.48550/arXiv.1609.09430.

[11] Agrawal, Tanay (2021) "Hyperparameter Optimization in Machine Learning" *Apress*, ISBN-13 (electronic): 978-1-4842-6579-6.

[12] Elshawi, Radwa, Mohamed Maher, and Sherif Sakr (2019) "Automated Machine Learning: State-of-The-Art and Open Challenges" *arxiv.org* https://doi.org/10.48550/arXiv.1906.02287.

[13] Feurer, Matthias, and Frank Hutter (2019) "AutoML Methods - Hyperparameter Optimization", in Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren (eds) *Automated Machine Learning. Methods, Systems, Challenges*, The Springer Series on Challenges in Machine Learning, ISBN 978-3-030-05317-8.

[14] Fernandes, Carina, Paulo Trigo, Joel Paulo, and Paulo Vieira (2022) "Anotação de Eventos Sonoros em Vídeo" *Instituto Superior de Engenharia de Lisboa (ISEL)*, final course project, Degree in Informatics and Multimedia Engineering.

[15] Badr, Will (2019) "Having an imbalanced dataset? Here is how you can fix it." *Towards Data Science*.

[16] Brownlee, Jason (2021) "Nested Cross-Validation for Machine Learning with Python" *Machine Learning Mastery*.