Arbitragem de dispositivos acionados por voz em redes com nós heterogêneos

Juliana C. Inácio, Vinícius J. D. Vieira, Guilherme C. Pereira e Renato Candido

***Resumo*— A arbitragem de dispositivos (AD) ocorre quando, em um mesmo ambiente, existem dispositivos equipados com assistentes virtuais inteligentes acionados por voz que possuem a mesma palavra-chave de ativação. Recentemente, foram propostas soluções baseadas em redes neurais profundas, com processamento centralizado e considerando dispositivos com características idênticas. Neste trabalho, é proposta uma solução para AD que utiliza modelos de classificação baseados em árvores de decisão, com processamento descentralizado e considerando dispositivos com características distintas. A solução proposta obteve uma melhora na acurácia de até 14% comparada à solução utilizando a medida de energia.**

***Palavras-Chave*— arbitragem de dispositivos, localização de fontes, processamento de fala, aprendizado de máquina**

***Abstract*— Device arbitration occurs when there are devices equipped with smart virtual assistents activated by voice and by the same keyword in a same environment. Recently, deep-learning-based solutions were proposed with centralized processing and considering devices with identical characteristics. In this work, we propose an arbitration solution that uses classification models based on ensembles of decision trees, with descentralized processing and considering devices with distintic features. The proposed solution obtained an accuracy improvement up to 14% compared to the solution using energy-based arbitration.**

***Keywords*— device arbitration, source-localization, speech processing, machine learning**

I. Introdução

Nos últimos anos, o número de dispositivos equipados com assistentes virtuais inteligentes acionados por voz, como a Alexa da Amazon ou a Bixby da Samsung aumentou. É comum em casas e escritórios, por exemplo, ter diversos dispositivos como *smartphones*, *tablets*, relógios, televisores e refrigeradores a espera da palavra-chave de ativação (PCdA) do assistente virtual. Nesses cenários, onde vários dispositivos aguardam a mesma PCdA para interagir com o usuário, surge o problema de arbitragem de dispositivos (AD) [1], [2].

Para realizar a arbitragem, o primeiro passo é definir o critério de escolha do dispositivo adequado. Os critérios podem variar de acordo com a aplicação e podem envolver diversas regras de negócio, mas um cenário comum é a necessidade da ativação do dispositivo mais próximo do usuário, utilizando um critério de distância como em [3].

O problema de AD se assemelha ao problema de localização de fontes [4], amplamente abordado na literatura. No entanto, a maioria dos dispositivos equipados com assistentes virtuais inteligentes possuem poucos ou um único microfone, o que torna difícil o uso dessas técnicas, já que a maioria delas depende da diversidade de sinais obtida com múltiplos microfones, como é o caso dos algoritmos que medem a diferença do tempo de chegada (*time difference of arrival* - TDOA) [5]–[7].

Partindo da premissa de que dispositivos mais próximos do usuário recebem sinais de áudio com maior energia, alguns trabalhos propuseram métodos de localização de fontes baseados nessa medida [8]–[11]. No entanto, esses métodos apresentam desempenho ruim quando há fontes de ruído no ambiente, tais como pessoas conversando, televisores ou outros eletrodomésticos. Além disso, também são afetados pelos efeitos de reverberação, dependendo do formato da sala e dos objetos dentro dela. Outro problema é que os nós da rede de dispositivos podem ser heterogêneos, ou seja, compostos por dispositivos diferentes. Dessa forma, é comum haver diferenças de *hardware* e *software* entre os dispositivos, como microfones com características diversas ou sistemas de pré-processamento como o controle automático de ganho. Nesse contexto, os algoritmos baseados na medida da energia do sinal de áudio têm o desempenho bastante comprometido, pois não é raro que um dispositivo mais distante da fonte tenha um sinal com energia maior do que um dispositivo mais próximo, devido às diferenças em suas características.

No contexto de aprendizado de máquina (AM), há diversos trabalhos explorando a predição da distância entre fonte e destino [12]–[15]. No entanto, apesar de ser semelhante ao problema de localização de fontes, o problema de AD possui características e limitações exclusivas, como a utilização de um único microfone por dispositivo, e restrições quanto ao compartilhamento de áudios entre dispositivos por questões de privacidade. Essas restrições fazem com que esse seja um tema pouco explorado e interessante para ser investigado.

Apesar de ter sido mencionado pela primeira vez nas patentes [1], [2] em 2016, ainda existem poucos trabalhos que abordam o problema específico de AD na literatura, destacando-se os trabalhos [3], [16], e [17], sendo [17] uma revisão bibliográfica sobre esse tema. Em [3], os autores propuseram uma solução baseada em AM para o problema de AD, em que cada dispositivo utiliza uma rede neural profunda para obter um vetor de atributos a partir de uma entrada baseada na variação do envelope do sinal de áudio [18]. Esse vetor de atributos é então transmitido a um nó de processamento central, que pode ser um dispositivo com maior capacidade de processamento ou até mesmo um sistema em nuvem e, nesse nó central, uma segunda rede neural é utilizada para fazer arbitragem de qual dispositivo deve ser ativado. Apesar de existirem redes separadas nos dispositivos e no nó central de processamento, o treinamento das redes é feito de maneira conjunta, fim-a-fim. Em [16], foi proposta uma melhoria na abordagem de [3] adicionando uma etapa de pré-treinamento auto-supervisionado capaz de separar características acústicas do ambiente do sinal de fala sem a necessidade de dados rotu-

Juliana C. Inácio, Vinícius J. D. Vieira, Guilherme C. Pereira e Renato Candido, Depto. de Processamento de Voz, SiDi, Alphaville, Campinas, SP - Brasil, e-mails: {j.camilo, vinicius.dv, g.pereira, r.candido}@sidi.org.br.

lados. Essa nova abordagem apresentou melhor desempenho, especialmente em casos com poucos dados para treinamento. Apesar dos resultados interessantes, essas abordagens consideram os nós da rede como homogêneos e as redes utilizadas têm complexidade computacional elevada.

No contexto de redes com nós heterogêneos, devido às diferenças em termos de *hardware* entre os nós da rede, é interessante considerar soluções de baixa complexidade e que consumam pouca memória. Assim, neste trabalho, a proposta é avaliar soluções para ADs baseadas em AM nesse cenário. Em particular, são utilizados algoritmos baseados em árvores de decisão e, diferentemente de [3], a solução aqui proposta opera de maneira descentralizada, utilizando processamento local em cada nó para a arbitragem.

Este trabalho está organizado da seguinte forma. Na Seção II, é apresentada a formulação do problema, na Seção III são apresentados detalhes do modelo e das características utilizadas como entradas, na Seção IV são apresentados o banco de dados utilizado para o treinamento e a forma de avaliação dos modelos e, por fim, nas Seções V e VI são apresentados os resultados obtidos e as conclusões, respectivamente.

II. Formulação do problema

Consideremos um ambiente com N_D dispositivos D_i, $i = 1, \cdots, N_D$ e uma fonte de áudio S que representa um usuário falando a PCdA dos dispositivos, todos posicionados aleatoriamente. Eventualmente, pode existir também uma fonte de ruído S_N no ambiente, também em uma posição aleatória. Cada dispositivo D_i capta um sinal de áudio x_i e extrai um vetor de características $\boldsymbol{\varphi}_i$.

Em seguida, os dispositivos compartilham os vetores de características $\boldsymbol{\varphi}_i$ entre si por meio de algum tipo de conexão sem fio (Wi-Fi, *Bluetooth*, etc), de forma que todos os dispositivos da rede tenham acesso a todos os vetores $\boldsymbol{\varphi}_i$, conforme ilustrado no esquema da Fig. 1-(a) para o dispositivo D_1.

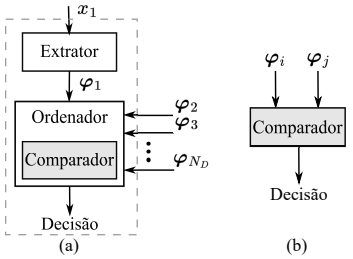

Fig. 1: Diagrama de blocos da AD. a) Blocos presentes em cada dispositivo e compartilhamento dos vetores de características. b) Detalhamento do bloco de comparação.

Após receber os vetores de características, cada dispositivo usa um algoritmo de ordenação para decidir qual dispositivo é o mais adequado a responder, com base na proximidade da fonte. Caso um dispositivo verifique que ele próprio é o mais adequado a responder, ele é ativado. Dessa forma, a decisão é feita localmente em cada dispositivo, sem a necessidade de um nó central para processamento.

Para ilustrar o processo de decisão, podemos considerar um caso simplificado, em que o vetor de características $\boldsymbol{\varphi}_i$ é reduzido ao escalar $\varphi_i^{(1)}$, composto apenas pelo valor da energia do sinal. Desconsiderando os efeitos de reverberação da sala e os possíveis efeitos introduzidos no processo de captação dos áudios, a energia do áudio captado por um dispositivo mais próximo da fonte tende a ser maior do que a energia referente aos áudios de dispositivos mais distantes. Dessa forma, ao receber os valores de energia $\varphi_i^{(1)}$, cada dispositivo ordena os valores, buscando pelo maior dentre eles, permitindo identificar o dispositivo cujo áudio tem o maior valor de energia.

No caso de um vetor de características multidimensional, é necessário que o algoritmo de ordenação seja capaz de ordenar os vetores, de forma a decidir o vetor referente ao áudio mais próximo da fonte. Em geral, esse processo de ordenação não é trivial, não sendo possível utilizar uma simples norma dos vetores para fazer a ordenação, por exemplo. Dado que diversos algoritmos de ordenação são baseados em comparações de elementos dois a dois como, por exemplo, o *bubble sort* [19], o problema da ordenação pode ser reduzido a obtenção de um bloco que receba dois vetores como entrada e funcione como comparador, indicando o vetor referente ao dispositivo mais próximo da fonte, como ilustrado no esquema da Fig. 1-(b).

Neste trabalho, são avaliadas algumas alternativas de soluções para funcionar como comparador nesse contexto. A proposta é utilizar algoritmos de AM que recebem vetores de pequena dimensão, com características extraídas dos áudios para fazer a classificação binária do vetor correspondente ao áudio mais próximo da fonte. Para fins de comparação de desempenho, levando em conta que trata-se de uma solução descentralizada, é utilizado o resultado obtido com um classificador baseado na energia (CBE), usando os valores $\varphi_i^{(1)}$, conforme descrito anteriormente. A seguir, são descritos os modelos utilizados e as características utilizadas para composição dos vetores $\boldsymbol{\varphi}_i$.

III. Modelos e características utilizadas

Diferentemente da abordagem proposta em [3], baseado em AM com redes profundas, neste trabalho são avaliadas soluções de baixo custo computacional, a fim de permitir o uso em dispositivos com menor capacidade computacional, utilizando processamento local. Para tanto, como entrada do sistema de classificação ilustrado na Fig. 1-(b), são considerados vetores compostos de 1 a 9 características extraídas conforme detalhado na Seção III-A.

Para os modelos, são utilizados soluções de comitê (*ensemble*) de algoritmos baseados em árvores de decisão, amplamente utilizados na literatura para aplicações tabulares com dados de baixa dimensionalidade na entrada [20]. Mais especificamente, são avaliados modelos utilizando os algoritmos *extra trees* (ET) [21] e *gradient boosting* (GB) [22] para diversas variações de vetores de entrada, considerando diferentes características, detalhadas a seguir.

A. Características utilizadas

Para obter uma referência do desempenho do sistema, foi utilizada a arbitragem baseada no simples cálculo da energia do sinal dado por $E_i = \sum_{n=0}^{N_{x_i}} x_i^2(n)$.

Para compor os vetores de características utilizados como entrada dos modelos, além da energia, foi utilizada uma

proposta baseada em [23]. Nesse trabalho, foram propostas 8 características para a tarefa de estimação de distância do locutor a partir de sinais de áudio de um único canal. São características estatísticas, obtidas a partir da divisão do áudio x_i em trechos com duração entre 16 ms e 30 ms[1], utilizando 50% de sobreposição entre os trechos. A partir destes trechos, utilizando um detector de atividade vocal depois de um janelamento de Hanning, é selecionado o conjunto de quadros f_{ij}, $j = 1, \cdots, N_{f_i}$ para os quais é detectada atividade vocal, onde N_{f_i} é o número de quadros selecionados. As características são extraídas desse conjunto de quadros e são divididas em dois grupos, o primeiro composto por 4 características obtidas a partir dos áudios originais, brevemente descritas a seguir, e o segundo composto pelas mesmas características, mas usando como entrada os áudios processados por um filtro passa-faixas, de acordo com a metodologia em [23].

A *Razão LP* é obtida pela razão entre duas medidas calculadas a partir dos coeficientes de predição linear (*Linear Prediction Coefficients* - LPCs). Para cada quadro f_{ij} de um sinal x_i, são calculados os LPCs de ordem 32 e, a partir deles, são obtidos os resíduos de predição, que são utilizados para obter duas medidas: (i) o valor quadrático médio, denotado por r_{ij} e (ii) o percentil de 90% da distribuição dos resíduos denotado por p_{ij}. A partir desses valores para cada quadro, obtém-se o valor da *Razão LP* para o áudio x_i, calculando

$$RLP_i = \textstyle\sum_{j=1}^{N_{f_i}} p_{ij} / \sum_{j=1}^{N_{f_i}} r_{ij}. \quad (1)$$

Ainda utilizando os resíduos dos LPCs, para cada f_{ij} de um sinal x_i, é calculada a curtose estatística, definida como $E(y(n) - \mu_y)^4/\sigma_y^4$ para um sinal $y(n)$, onde μ_y é a média e σ_y é o desvio padrão das amostras de y e a partir desses valores para cada quadro, obtém-se o valor da *Curtose* para o áudio x_i, calculando

$$C_i = (1/N_{f_i}) \textstyle\sum_{j=1}^{N_{f_i}} k_{ij}. \quad (2)$$

Para levar em conta o efeito da reverberação, é utilizada uma medida baseada na representação espectral do sinal de áudio [24]. Para cada f_{ij} de um sinal x_i, primeiramente é calculado o espectro de magnitude em dB e, com base nessa representação, é calculada a assimetria estatística (*skewness*), definida como $E(y(n) - \mu_y)^3/\sigma_y^3$ para um sinal $y(n)$. A partir dos valores de s_{ij} para cada f_{ij}, obtém-se o valor da *Assimetria* para o áudio x_i, calculando

$$A_i = (1/N_{f_i}) \textstyle\sum_{j=1}^{N_{f_i}} s_{ij}. \quad (3)$$

A última das quatro características principais também é obtida a partir de uma medida de assimetria estatística. Cada f_{ij} de um sinal x_i é filtrado pelo filtro com função de transferência $H(z) = (1 - z^{-1})/(1 + 0{,}9z^{-1})$ e, em seguida, é passado por um bloco retificador de meia onda. Após a retificação, dois sinais são gerados, filtrando-se o sinal por dois filtros distintos com funções de transferência dadas por $H_1(z) = [1 - 0{,}99z^{-1}]^{-1}$ e $H_2(z) = [1 - 0{,}998z^{-1}]^{-3}$. Para cada um desses sinais, calcula-se a energia ao longo do tempo, em dB. Por fim, calcula-se o sinal de diferença entre os sinais de energia, o qual é utilizado para obter a assimetria estatística e_{ij}, referente à f_{ij}. A partir dos valores de e_{ij} para cada quadro, obtém-se o valor da *Assimetria da Diferença de Energia* para o áudio x_i, calculando

$$ADE_i = (1/N_{f_i}) \textstyle\sum_{j=1}^{N_{f_i}} e_{ij}. \quad (4)$$

As outras quatro características são obtidas seguindo os mesmos passos utilizados para as características RLP_i, C_i, A_i e ADE_i mas utilizando como entrada os sinais dos f_{ij} filtrados por um filtro passa-faixa com frequências de corte de 4 kHz e 7,8 kHz, resultando nas características denotadas, respectivamente, por RLP_i^{BP}, C_i^{BP}, A_i^{BP} e ADE_i^{BP}.

Com base nessas características, foram consideradas as combinações apresentadas na Tabela I para composição dos vetores $\boldsymbol{\varphi}_i^{(j)}$. Vale notar que $\varphi_i^{(1)}$ utiliza apenas a medida de energia, sendo utilizada para fins de comparação de desempenho e $\boldsymbol{\varphi}_i^{(6)}$ é composto pelas oito características propostas em [23] para o problema de localização de fontes.

TABELA I: Conjuntos de características usados nos modelos de classificação baseados em árvore de decisão.

Rótulo	Características								
$\varphi_i^{(1)}$	E_i	–	–	–	–	–	–	–	–
$\boldsymbol{\varphi}_i^{(2)}$	E_i	C_i	–	–	–	–	–	–	–
$\boldsymbol{\varphi}_i^{(3)}$	E_i	C_i	RLP_i	–	–	–	–	–	–
$\boldsymbol{\varphi}_i^{(4)}$	E_i	C_i	RLP_i	ADE_i	–	–	–	–	–
$\boldsymbol{\varphi}_i^{(5)}$	E_i	C_i	RLP_i	ADE_i	A_i	C_i^{BP}	RLP_i^{BP}	ADE_i^{BP}	A_i^{BP}
$\boldsymbol{\varphi}_i^{(6)}$	–	C_i	RLP_i	ADE_i	A_i	C_i^{BP}	RLP_i^{BP}	ADE_i^{BP}	A_i^{BP}

IV. Banco de dados e métrica de desempenho

Conforme mencionado em [17], um dos obstáculos para o desenvolvimento de pesquisas sobre o problema de ADs é a falta de bancos de dados que representem o problema. Além disso, em geral, os poucos bancos de dados reais existentes são restritos por questões de confidencialidade. Por essa razão, neste trabalho os bancos de dados para treinamento e testes das soluções propostas são gerados sinteticamente utilizando a biblioteca *Pyroomacoustics* [25], a qual simula os efeitos de reverberação das salas. Para as PCdAs, foi utilizado o banco de dados *Google Speech Commands v2* (GSCv2) [26], que possui 35 PCdAs diferentes, com milhares de gravações para cada uma delas, totalizando 105.829 gravações de 2.618 locutores diferentes, dos quais 1.833 locutores foram usados para treinamento e 524 para teste.

Nas simulações, são considerados $N_D = 5$ dispositivos e, no caso da rede com nós heterogêneos, para simulação da diversidade entre os dispositivos e a influência dos diferentes blocos de *hardware* e *software*, é considerado que cada dispositivo D_i atenua o áudio recebido por um fator G_i sorteado aleatoriamente no intervalo $[0{,}4; 1]$. No caso da rede com nós homogêneos, os fatores G_i foram fixados em $G_i = 1$.

Para criar sinteticamente um cenário de arbitragem, simula-se um ambiente de uma sala retangular, com suas propriedades físicas e acústicas, considerando uma fonte de áudio S e as posições P_j, $j = 1, \ldots, N_P$ para os N_D dispositivos, conforme esquematizado na Fig. 2. A largura e o comprimento de cada sala são escolhidos aleatoriamente, considerando valores distribuídos uniformemente no intervalo $[3\text{ m}, 6\text{ m}]$ e a altura é

[1]Os valores foram ajustados por *grid search* para cada característica: 16 ms para A_i e ADE_i; 20 ms para C_i^{BP} e RLP_i^{BP}; 25 ms para A_i^{BP}; e 30 ms para as demais.

escolhida aleatoriamente no intervalo $[2{,}5\text{ m}, 3\text{ m}]$. A posição da fonte S dentro da sala é escolhida aleatoriamente segundo uma distribuição uniforme e as posições P_j são alocadas nas órbitas circulares com centro em S com raios de j metros e fase θ_j, escolhida aleatoriamente. Dependendo das dimensões da sala e da localização da fonte S, as órbitas com raios maiores podem situar-se totalmente na região fora da sala e, nesses casos, são consideradas $N_P < N_D$ posições de forma que todas posições P_j, estejam situadas dentro da sala. Nos cenários com ruído, também é considerada uma fonte de ruído S_N localizada em uma posição aleatória dentro da sala. Todas as fontes e dispositivos são simulados à altura de 1 m e, por simplicidade, a diretividade das fontes e dos microfones dos dispositivos é considerada omnidirecional.

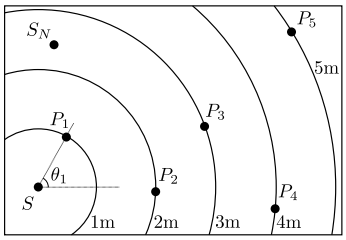

Fig. 2: Exemplo de sala gerada para o banco dados sintéticos com uma fonte de áudio e de ruído e cinco dispositivos, todos posicionados aleatoriamente.

Dado o cenário de arbitragem com N_P posições, são calculadas as respostas ao impulso da sala a partir da posição de S para cada uma das N_P posições dos dispositivos usando [25], assim como as respostas ao impulso a partir da posição de S_N nos cenários com ruído. Com essas respostas ao impulso e considerando sinais de áudio para as fontes S e S_N, são obtidos os áudios resultantes para cada dispositivo D_i, situados em cada posição P_j, considerando a atenuação pelo fator G_i em cada dispositivo. Por exemplo, para o caso de uma sala com $N_P = 3$ posições e $N_D = 5$ dispositivos, são obtidos 15 sinais de áudio, simulando cada um dos 5 dispositivos em cada uma das posições.

São gerados quatro bancos de dados, variando os casos com ruído ou sem e com a rede com nós heterogêneos ou não. Para cada banco, são simuladas 1000 salas para treinamento e 286 salas para teste, cada uma utilizando 4 PCdAs escolhidas aleatoriamente a partir dos respectivos conjuntos de PCdAs para treinamento e teste, sem contaminação entre os conjuntos. Todos os áudios gerados têm frequência de amostragem de 16 kHz e são armazenados no formato PCM de 16-bits. Como fonte de ruído, adotou-se um ruído de conversação disponibilizado em [27].

Como mencionado, a proposta deste trabalho é avaliar classificadores binários para serem utilizados como comparadores, conforme ilustrado na Fig. 1-(b). Dessa forma, a fim de avaliar o desempenho dos sistemas, para cada sala e PCdA, são consideradas todas as comparações possíveis de dispositivos D_i e D_j com D_i mais próximo à fonte S que D_j e saída igual a 1 e D_i mais distante da fonte S que D_j e saída igual a 0. Dado o cenário das possíveis comparações entre os pares de dispositivos e a decisão esperada, avalia-se o modelo em termos da *acurácia* obtida.

Para cada PCdA e cada par de posições P_i e P_j com $i \neq j$, considerando 5 dispositivos, temos um total de 25 comparações entre pares de dispositivos sendo (i) 20 comparações envolvendo dispositivos distintos entre si (arranjo de 5 dispositivos tomados dois a dois) e (ii) 5 comparações envolvendo dispositivos idênticos. Já em relação ao número de pares de posições P_i e P_j com $i \neq j$, em uma sala com $N_P = 5$ posições, temos 10 possibilidades, dadas pela combinação de 5 posições tomadas duas a duas. Dessa forma, para uma sala, com $N_D = N_P = 5$, temos 250 comparações entre pares de dispositivos a distâncias distintas para cada PCdA. Vale notar que, na prática, esse número será um pouco menor pois nem toda sala é capaz de alocar as 5 posições, devido as suas dimensões e à posição da fonte S. Os números de comparações práticos para os diferentes bancos de dados gerados são apresentados na Tabela II.

TABELA II: Número prático de comparações para os diferentes casos e cenários.

Casos	Número de comparações			
	sem ruído		com ruído	
	treino	teste	treino	teste
Homogêneo	485.340	129.056	486.896	122.380
Heterogêneo	468.244	130.510	480.488	125.479

V. Resultados

Para realização dos experimentos, foi utilizada a linguagem Python e os modelos GB e ET foram implementados utilizando a biblioteca *Scikit-learn*. Para ambos os modelos, o ajuste dos hiperparâmetros foi feito considerando 100 estimadores, o número mínimo de 2 exemplos por nó folha e, para o GB, foi utilizada a taxa de aprendizagem de 0,1. A função utilizada como função custo nos modelos GB e como critério para medir a qualidade de um corte nos modelos ET foi a *log-loss*.

A comparação de desempenho em termos de acurácia para os cenários sem e com ruído entre o CBE com $\varphi_i^{(1)}$ e as soluções utilizando modelos de classificação é apresentada na Tabela III. Como pode-se observar, o CBE com $\varphi_i^{(1)}$ apresenta uma piora de $13{,}85\%$ na acurácia quando comparamos os bancos de dados dos casos homogêneo e heterogêneo para o cenário sem ruído. Esse comportamento já era esperado dada a característica da medida da energia, que é sensível à aplicação de um ganho ou atenuação no sinal. Como isso é comum em cenários práticos, o CBE se torna uma solução pouco robusta para o problema de AD.

Para o caso homogêneo no cenário sem ruído, ambos os modelos de classificação apresentaram desempenho muito próximo ao do CBE com $\varphi_i^{(1)}$, sendo que o melhor resultado encontrado, de $85{,}82\%$, foi obtido com o modelo ET com $\boldsymbol{\varphi}_i^{(5)}$, apresentando um ganho de $2{,}15\%$ no desempenho em relação ao CBE. No entanto, no caso heterogêneo no cenário sem ruído, todos os experimentos com modelos de classificação apresentaram um ganho significativo na acurácia, variando de $3{,}34\%$ à $14{,}36\%$ de melhoria no desempenho em comparação ao CBE com $\varphi_i^{(1)}$. Além disso, no caso heterogêneo, a característica $\varphi_i^{(1)}$ baseada na energia tem pouca influência quando são utilizadas as outras 8 características, o que pode ser notado comparando os modelos com $\boldsymbol{\varphi}_i^{(5)}$ e $\boldsymbol{\varphi}_i^{(6)}$, cujas diferenças de desempenho são sempre menores que $0{,}5\%$.

Para os cenários com ruído, pode-se observar que o CBE com $\varphi_i^{(1)}$ também apresenta uma piora de desempenho de $5{,}12\%$ no caso heterogêneo quando comparado ao caso homogêneo. Nesse cenário, todos os resultados com os modelos GB ou ET, tanto no caso homogêneo quanto no caso heterogêneo, apresentaram ganhos de desempenho quando comparados ao CBE com $\varphi_i^{(1)}$. Para o caso homogêneo, a melhoria na acurácia variou de $1{,}91\%$ à $8{,}9\%$ comparado ao CBE com $\varphi_i^{(1)}$, enquanto que, no caso heterogêneo, essa melhoria variou de $4{,}89\%$ à $14{,}23\%$. Assim como no cenário sem ruído, para o cenário com ruído a característica de energia é pouco relevante para o desempenho no caso heterogêneo quando comparamos os modelos com $\boldsymbol{\varphi}_i^{(5)}$ e $\boldsymbol{\varphi}_i^{(6)}$, sendo a diferença de desempenho menor que $0{,}17\%$ para esse cenário.

Cabe ressaltar, que a solução proposta apresentou ganho de desempenho para os diferentes conjuntos de características $\boldsymbol{\varphi}_i$ testadas, para os diferentes casos e condições de ruído. Dessa forma, havendo a necessidade de uma solução com complexidade reduzida, é possível obter um bom desempenho em termos de acurácia utilizando uma quantidade reduzida de características. Além disso, os melhores resultados para todos os cenários considerados foram obtidos com o modelo ET.

TABELA III: Comparação de acurácia para os casos homogêneos e heterogêneos, sem e com ruído, entre CBE com $\varphi_i^{(1)}$ e modelos de classificação baseados em árvore de decisão. Melhores resultados destacados em negrito.

	Sem ruído				Com ruído			
	Homogêneo		Heterogêneo		Homogêneo		Heterogêneo	
$\varphi_i^{(1)}$	83,67%		69,82%		65,54%		60,42%	
$\boldsymbol{\varphi}_i^{(J)}$	E.T	G.B	E.T	G.B	E.T	G.B	E.T	G.B
$\boldsymbol{\varphi}_i^{(2)}$	84,21%	83,60%	73,27%	73,16%	67,77%	67,45%	65,31%	65,79%
$\boldsymbol{\varphi}_i^{(3)}$	83,72%	84,15%	77,73%	77,64%	72,79%	72,11%	71,98%	72,16%
$\boldsymbol{\varphi}_i^{(4)}$	84,28%	84,09%	81,13%	80,03%	73,93%	72,54%	72,93%	72,15%
$\boldsymbol{\varphi}_i^{(5)}$	**85,82%**	85,02%	**84,18%**	82,83%	**74,44%**	73,61%	**74,65%**	73,83%
$\boldsymbol{\varphi}_i^{(6)}$	84,44%	82,94%	84,06%	83,34%	73,55%	72,66%	74,48%	73,72%

VI. Conclusões

Neste trabalho, foi proposta uma solução para o problema de AD que utiliza comitês de modelos de classificação baseados em árvores de decisão com processamento descentralizado, considerando redes com nós homogêneos ou heterogêneos para cenários com e sem ruído. Os resultados obtidos indicam que a solução proposta é capaz de melhorar o desempenho em termos de acurácia em todos os cenários testados quando comparados à solução com um CBE, proporcionando um ganho de acurácia de cerca de 14% no caso da rede com nós heterogêneos nos cenários com e sem ruído. Além disso, dependendo do conjunto de características escolhido, pode-se ajustar o compromisso entre desempenho e complexidade da solução, o que traz um grande potencial para aplicações práticas no contexto de AD. Em trabalhos futuros, pretende-se fazer uma comparação detalhada em termos de custo computacional em relação à outras soluções baseadas em AM.

Agradecimentos

Os resultados apresentados neste artigo foram desenvolvidos como parte de um projeto do SiDi, financiado pela Samsung Eletrônica da Amazônia Ltda., com o apoio da Lei Federal de Informática no. 8248/91.

Referências

[1] Microsoft Technology Licensing LLC Redmond, WA (US), "Device arbitration for listening devices," US Patent 2016/0155443 A1, Jun. 2016.

[2] Microsoft Technology Licensing LLC Redmond, WA (US), "Device arbitration for listening devices," US Patent 9,812,126 B2, Nov. 2017.

[3] J. Barber, Y. Fan, and T. Zhang, "End-to-end alexa device arbitration," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, May 2022, pp. 926–930.

[4] J.C. Chen, K. Yao, and R.E. Hudson, "Source localization and beamforming," *IEEE Signal Proc. Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.

[5] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transac. on Acoustics, Speech, and Signal Proc.*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[6] M. Brandstein, J. Adcock, and H. Silverman, "Practical time-delay estimator for localizing speech sources with a microphone array," *Comput., Speech, and Language*, vol. 9, no. 2, pp. 153–169, Apr. 1995.

[7] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transac. on Signal Proc.*, vol. 56, no. 5, pp. 1770–1778, 2008.

[8] X. Sheng and Y.-H. Hu, "Energy based acoustic source localization," in *Inform. Proc. in Sensor Networks*, Apr. 2003, pp. 285–300.

[9] M. Shashanka, B.S.-Cunningham, and M. Cooke, "Effects on reverberant energy on statistics of speech," in *Workshop on Speech Separation and Compreh. in Complex Acoustic Environ.*, 2004.

[10] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Transac. on Signal Proc.*, vol. 56, no. 1, pp. 365–377, 2008.

[11] W. Meng and W. Xiao, "Energy-based acoustic source localization methods: a survey," *IEEE Transac. on Signal Proc.*, vol. 17, no. 2, pp. 376, 2017.

[12] F. Vesperini and et al, "A neural network based algorithm for speaker localization in a multi-room environment," in *IEEE 26th Int. Workshop on Mach. Learn. for Signal Proc. (MLSP)*, Sept. 2016, pp. 1–6.

[13] E. L. Ferguson, S.B. Williams, and C.T. Jin, "Sound source localization in a multipath environment using convolutional neural networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, Apr. 2018, pp. 2386–2390.

[14] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Proc.*, vol. 13, no. 1, pp. 34–48, 2019.

[15] M.R. Celsi, S. Scardapane, and D. Comminiello, "Quaternion neural networks for 3d sound source localization in reverberant environments," in *IEEE 30th Int. Workshop on Mach. Learn. for Signal Proc. (MLSP)*, Sept. 2020, pp. 1–6.

[16] J. Harvill, J. Barber, A. Nair, and R. Pishehvar, "SPADE: self-supervised pretraining for acoustic disentanglement," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, Jun. 2023, pp. 1–5.

[17] G. Ciccarelli and et al, "Challenges and opportunities in multi-device speech processing," in *arXiv preprint: 2206.15432*, 2022.

[18] M. Wolf and C. Nadeu, "Channel selection measures for multi-microphone speech recognition," *Speech Commun.*, vol. 57, pp. 170–180, Feb. 2014.

[19] E.H. Friend, "Sorting on Electronic Computer Systems," *Journal of the ACM*, vol. 3, no. 3, pp. 134–168, Jul 1956.

[20] R.S.-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, May 2022.

[21] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[22] J.H. Friedman, "Stochastic gradient boosting," *Comput. Statistics and Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[23] E. Georganti and et al, "Single channel sound source distance estimation based on statistical and source-specific features," in *Convention of the Audio Eng. Soc.*, May 2009, pp. 1–11.

[24] E. Georganti, J. Mourjopoulos, and F. Jacobsen, "Analysis of room transfer function and reverberant signal statistics," *The Journal of the Acoustical Soc. of America*, vol. 123, no. 5, pp. 3761, May 2008.

[25] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, Apr. 2018, pp. 351–355.

[26] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," in *arXiv preprint:1804.03209*, 2018.

[27] C. K. Reddy and et al, "A scalable noisy speech dataset and online subjective test framework," in *arXiv preprint: 1909.08050*, 2019.