UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Taciano Douglas Pereira Prudêncio

Sistema detector de comandos de voz através de LPC e utilização de redes neurais artificiais para aplicação em aeronaves

João Pessoa - PB
2023

Taciano Douglas Pereira Prudêncio

Sistema detector de comandos de voz através de LPC e utilização de redes neurais artificiais para aplicação em aeronaves

Trabalho de Conclusão de Curso apresentado à Coordenação de Engenharia Elétrica do Centro de Energias Alternativas e Renováveis da Universidade Federal da Paraíba como parte dos requisitos necessários para a obtenção do título de Engenheiro Eletricista.

Universidade Federal da Paraíba

Centro de Energias Alternativas e Renováveis

Curso de Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Juan Moises Mauricio Villanueva

João Pessoa - PB

2023

Catalogação na publicação
Seção de Catalogação e Classificação

P971s Prudencio, Taciano Douglas Pereira.
Sistema detector de comandos de voz através de LPC e utilização de redes neurais artificiais para aplicação em aeronaves / Taciano Douglas Pereira Prudencio. - João Pessoa, 2023.
41 f. : il.

Orientação: Juan Moises Mauricio Villanueva.
TCC (Graduação) - UFPB/CEAR.

1. Transportes 5.0. 2. Reconhecimento de voz. 3. Redes Neurais Artificiais. 4. Linear Predictive Coding. 5. Aviação. I. Villanueva, Juan Moises Mauricio. II. Título.

UFPB/CT/BSCT CDU 652.3(043.2)

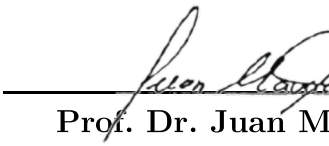
Elaborado por ONEIDA DIAS DE PONTES - CRB-15/198

Taciano Douglas Pereira Prudêncio

Sistema detector de comandos de voz através de LPC e utilização de redes neurais artificiais para aplicação em aeronaves

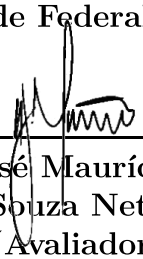
Trabalho de Conclusão de Curso apresentado à Coordenação de Engenharia Elétrica do Centro de Energias Alternativas e Renováveis da Universidade Federal da Paraíba como parte dos requisitos necessários para a obtenção do título de Engenheiro Eletricista.

Data de Aprovação: _____/_____/_____



Prof. Dr. Juan Moises Mauricio Villanueva
(Orientador)
Universidade Federal da Paraíba

Prof. Dr. Fabrício Braga Soares de Carvalho
(Avaliador)
Universidade Federal da Paraíba



Prof. Dr. José Maurício Ramos de Souza Neto
(Avaliador)
Universidade Federal da Paraíba

João Pessoa - PB
2023

Dedico este trabalho à minha avó Avani Correia.

Agradecimentos

Agradeço à minha família espiritual que está comigo nesta e nas próximas existências. Em especial ao trabalho desenvolvido no Lavário do Amanhecer da cidade de Itaporanga-PB, a todos os seus profissionais da fé, encarnados e desencarnados, o meu mais sincero agradecimento.

Agradeço também à minha mãe Tatiana Geysa e meu pai Domingos Sávio por acreditarem em mim desde antes do meu nascimento. A meus irmãos Beatriz e Domingos agradeço por serem minhas âncoras. A meus padrinhos e avós Auricélia e João agradeço por todo apoio que sem dúvida é o motivo de eu nunca ter desistido.

Agradeço também a meus amigos Júlio, Sam e Whélley por me impulsionarem a escrever este trabalho e por serem minha casa onde quer que estejamos. Agradeço finalmente a Djalma e Theodoro por tudo que fazem por mim e por todo amor e carinho que me fazem sentir todos os dias.

Agradeço também à toda minha família e meus amigos que sempre me lembram de ser quem eu sou.

Agradeço finalmente à minha primeira professora, Avani Correia, este trabalho de conclusão de curso é uma pequena etapa do que eu ainda vou alcançar e será tudo graças aos seus ensinamentos. Obrigado Vózinha.

A todos, muito obrigado!

Resumo

A indústria de transporte está em rápida evolução com a integração de tecnologias como Inteligência Artificial, robótica e assistentes virtuais, conhecida como Transportes 5.0. Este Trabalho de Conclusão de Curso (TCC) se concentra na aviação, explorando a utilização de Redes Neurais Artificiais (RNA) em sistemas de reconhecimento de voz. O objetivo principal é descrever um sistema que utiliza RNA para reconhecer comandos de voz durante o voo, permitindo a interação entre passageiros e os sistemas de conectividade e entretenimento das aeronaves. A voz é processada com detecção de palavras e Linear Predictive Coding (LPC) para garantir maior precisão de comandos, melhorando a eficácia da comunicação. O estudo analisa diversas topologias de camadas em RNAs, comparando seus resultados por meio de matrizes de confusão. O método de processamento de dados escolhido oferece robustez e precisão, contribuindo para o avanço das tecnologias que moldarão o futuro da comunicação na aviação.

Palavras-chave: Transportes 5.0. Reconhecimento de voz. Redes Neurais Artificiais. Linear Predictive Coding. Aviação.

Abstract

The transportation industry is rapidly evolving with the integration of technologies such as Artificial Intelligence, robotics, and virtual assistants, known as Transport 5.0. This undergraduate final thesis (TCC) focuses on aviation, exploring the utilization of Artificial Neural Networks (ANNs) in voice recognition systems. The main objective is to describe a system that utilizes ANNs to recognize voice commands during flight, enabling interaction between passengers and aircraft connectivity and entertainment systems. Voice is processed using word detection and Linear Predictive Coding (LPC) to ensure higher command accuracy, improving communication effectiveness. The study examines various layer topologies in ANNs, comparing their results through confusion matrices. The chosen data processing method provides robustness and accuracy, contributing to the advancement of technologies that will shape the future of communication in aviation.

Keywords: Transportation 5.0. Voice Reccognition. Artificial Neural Networks. Linear Predictive Coding. Aviation.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IA	Inteligência artificial
IFC	*In-Flight Connectivity*
IFE	*In-Flight Entertainment*
IFEC	*In-Flight Connectivity and Entertainment*
PED	*Personal Electronic Device*
RNA	Rede neural artificial
LPC	*Linear Predictive Coding*
FF	*Feed Foward*
MLP	*Multilayer Perceptron*
FIR	*Finite Impulse Response*

Sumário

1 Introdução

A evolução da indústria 4.0 e da 5ª Geração (5G) possibilita avanços tecnológicos em todos os setores produtivos, incluindo meios de transporte, com o surgimento do conceito de *Transportation 5.0* (WANG et al., 2023), que define a nova geração deste setor da indústria e tem a Inteligência Artificial(IA) como principal solução para seus paradigmas técnico-sociais.

A indústria de meios de transporte busca inovar para atender necessidades humanas básicas de atividades sociais como viajar a trabalho e visitar familiares, por exemplo. Outra necessidade humana é a de estar conectado virtualmente através da internet, e na aviação essa atividade social gera a demanda por sistemas de conectividade que consigam manter o usuário conectado, independente da sua localização ao longo da viagem (ONEWEB, 2022).

Dentro da cabine das aeronaves comerciais existe sistemas interligados chamados de *In-Flight Connectivity*(IFC) e *In-Flight Entertainment*(IFE), que juntos podem ser chamados de IFEC, suprem a demanda por conectividade ao mesmo tempo que, segundo (JIN; KIM, 2022), geram mais valor percebido pelo passageiro melhorando sua experiência como cliente da marca responsável pelo voo. No contexto de *Trasportation 5.0* os sistemas de IFEC podem ir além de apenas entreter passageiros entediados graças a maior capacidade de processamento de dados, novos tipos de sensores, novas formas de comunicação de dados e interfaces.

Ainda nesse contexto é possível pensar novos sistemas IFEC utilizando como base inovações ocorridas no mercado de assistentes virtuais. Presentes nos mais diversos dispositivos pessoais eletrônicos (do inglês PED) os assistentes virtuais, através de IA, processam comandos de voz e tomam decisões para simplificar atividades corriqueiras de seu usuário.

Em viagens, independente da distância, as pessoas buscam manter essa conexão por comodidade e familiaridade, portanto o sistema elaborado descreve o reconhecimento de comandos de voz dentro do vocabulário de comandos comumente utilizados em veículos inteligentes para funções básicas como ligar e desligar som ou abrir e fechar uma persiana da janela do avião.

Contudo, a cabine de uma aeronave comercial é um ambiente ruidoso, com a presença de diversas fontes sonoras contínuas que podem atrapalhar o reconhecimento de comandos de voz. É necessário uma ferramenta que possa atenuar os efeitos do ruído sonoro para que o sistema consiga interpretar corretamente cada palavra capturada. As fontes de ruído descritas pelo órgão de referência regulador internacional da aviação, a

Federal Aviation Administration (DEFAZIO, 2017) são:

- Fluxo de ar;
- Motores da aeronave;
- Sistema de ar-condicionado;
- Conversas entre passageiros;
- Sistema de avisos(alto-falantes);
- Serviços ao passageiro.

Existe atualmente diversas tecnologias que processam sinais de áudio e possibilitam o reconhecimento automático de voz extraindo as características principais do sinal. Essas características são utilizadas como padrões que identificam as referências registradas previamente como uma assinatura do sinal de voz, podendo destacar letras, fonemas, palavras inteiras e até locutores dependendo da aplicação de reconhecimento.

Entre as principais tecnologias utilizadas hoje para processamento de sinais de voz, destacam-se Modelos Ocultos de Markov (LIANG et al., 2011), Variação Temporal Dinâmica (DING; YEN; DA-CHENG, 2014), Redes Neurais Artificiais, entre outros métodos. Estas técnicas podem ser utilizadas separadamente ou podem integrar um único sistema de processamento da voz. Este trabalho foca na utilização da técnica de Redes Neurais Artificiais (RNA).

Redes Neurais Artificiais são modelos matemáticos que emulam o funcionamento do cérebro humano, realizando a função de conexão entre memória, escolha por associação e capacidade de aprendizagem (KOHONEN; OJA, 1987). É um método programável de aprendizagem a partir de dados selecionados previamente. No caso desse estudo os dados inseridos são comandos de voz de autoria própria e a RNA será alimentada com esses dados para construção de seu vocabulário matemático.

Porém, antes que o sinal de áudio esteja pronto para processamento dentro da RNA é filtrado para atenuação de ruído e passa pela etapa de preparação matemática chamada de *Linear Predictive Coding*(LPC), que consiste em reescrever cada amostra de uma lista como combinação linear das amostras passadas (ADAMI, 1997). Esse processamento matemático é necessário para extração das características principais do sinal de voz.

1.1 Objetivo Geral e Específicos

O objetivo geral do trabalho é descrever o sistema constituído pelo tratamento dos dados para construção do vocabulário através de LPC, criação, treinamento e teste da

rede neural artificial para reconhecimento das palavras de comando "On", "Off", "Open Window"e "Close Window"a partir de sinal contendo ruído ambiente da cabine de um avião comercial.

Entre os objetivos específicos estão:

a) Criação de uma rotina em Matlab que faça a leitura de sinais de áudio, com algoritmo de detecção de limiar de palavras para separar a parte útil da gravação, ou seja, onde está localizada a palavra de comando;

b) Geração de coeficientes LPC para descrever as amostras de áudio como combinação linear de suas amostras anteriores. Preparando as palavras para serem os itens da camada de entrada da Rede Neural Artificial;

c) Criação da RNA para reconhecimento dos comandos de voz, realização dos treinamentos e análise de testes para escolha da melhor topologia de camadas de rede.

1.2 Organização do trabalho

O projeto será descrito através de 4 capítulos além da introdução. O capítulo dois apresentará os fundamentos teóricos sobre cada um dos temas abordados no trabalho, apresentando uma revisão técnica sobre processamento de sinais de voz, Linear Predictive Coding e Redes Neurais Artificiais.

O capítulo três detalha todas as metodologias utilizadas no estudo de caso, apresentando o passo a passo para criação dos dados de comando de voz, a preparação matemática dos sinais de áudio e finalmente a programação do treinamento da RNA. Ainda neste capítulo são apresentados os algoritmos utilizados no projeto e as métricas utilizadas para obtenção dos resultados.

Os resultados estão descritos no capítulo quatro comparando diferentes respostas do sistema à mudança de parâmetros e resultados para cada uma das palavras de comando. É descrito também o método de testagem da Rede Neural e como foi avaliado sua acurácia. Por fim, o capítulo cinco sintetiza as conclusões que o projeto possibilitou bem como um ponto de partida para diferentes formas de continuidade deste trabalho no futuro.

2 Fundamentação Teórica

Este capítulo apresenta a fundamentação dos conceitos técnicos envolvidos no trabalho, sendo a primeira seção sobre aquisição de um sinal da voz humana, a segunda seção sobre detecção de palavras em sinais de voz, a terceira descreve a tecnologia de Linear Predictive Coding, a quarta seção é sobre Redes Neurais Artificiais e a quinta e última seção descreve o ambiente interno de aviões comuns.

2.1 Aquisição de sinais de voz

O mecanismo vibratório que ocorre nas cordas vocais humanas é capaz de produzir voz em um espectro de frequências que podem ir de 20 Hz até 10 kHz. As frequências fundamentais da voz variam de pessoa para pessoa, entre gêneros diferentes e também ao longo da vida de um mesmo indivíduo. Mas a maior parte da energia desse sinal vai estar localizada entre 300 Hz e 3400 Hz de acordo com (CARVALHO; DIAS,).

Na etapa de amostragem podem ser consideradas as componentes localizadas na faixa mais útil da voz que é entre 300 Hz e 3400 Hz, sem perdas consideráveis na qualidade do que foi falado. Essa operação é feita selecionando através de filtros a parte do sinal abaixo de 4 kHz, portanto, de acordo com o Teorema de Nyquist a taxa de amostragem deve ser pelo menos o dobro da frequência de corte, ou seja 8000 amostras por segundo.

Qualquer valor de amostragem menor que esse corre risco de sofrer distorções por sobreposição do sinal. Para selecionar a faixa de frequência de um sinal de voz é utilizado filtro passa-baixa com frequência de corte por volta de 4 kHz. Isso vai fazer com que a porção do áudio com frequência maior que esse valor seja atenuada até ser desconsiderada.

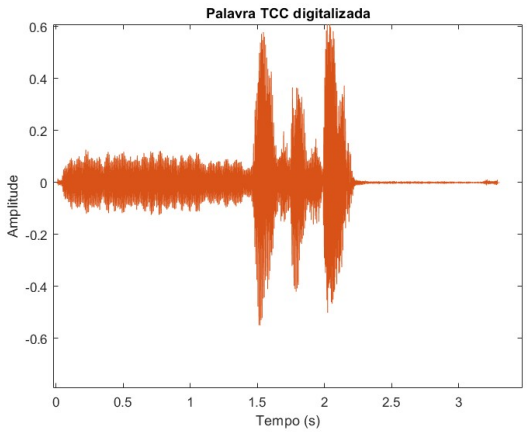
A Figura 1a é um exemplo de sinal de voz digitalizado no formato amplitude *versus* tempo, a palavra dita neste exemplo foi "TCC". Na imagem é possível perceber o formato de onda da palavra, ou seja, os valores de amplitude em cada amostra definem o que pode ser observado como cada uma das 3 letras: a sequência de amostras com maior amplitude, a separação entre elas, bem como o início e o final da palavra completa.

2.2 Seleção de palavras

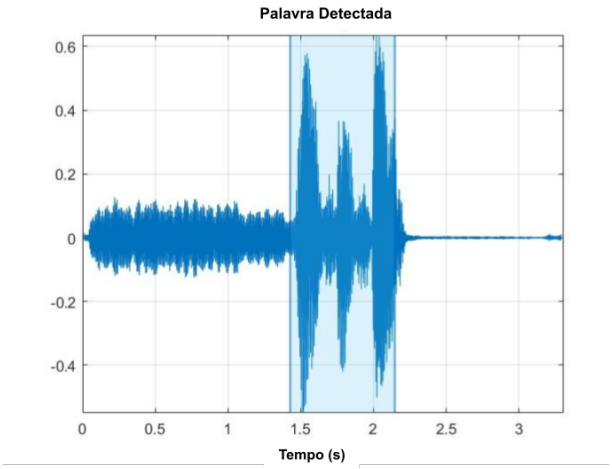
O processamento digital de voz busca obter as características fundamentais de frequência e amplitude desse sinal para que seja manipulado, a análise qualitativa de (BECKER, 2009) demonstra que é possível distinguir palavras, fonemas e intervalos de silêncio encontrando limiares de amplitude no sinal de voz. Os benefícios de selecionar

Figura 1 – Exemplo de sinal de áudio contendo a palavra "TCC"

(a) Palavra "TCC"no domínio do tempo



(b) Palavra TCC detectada pelo algoritmo



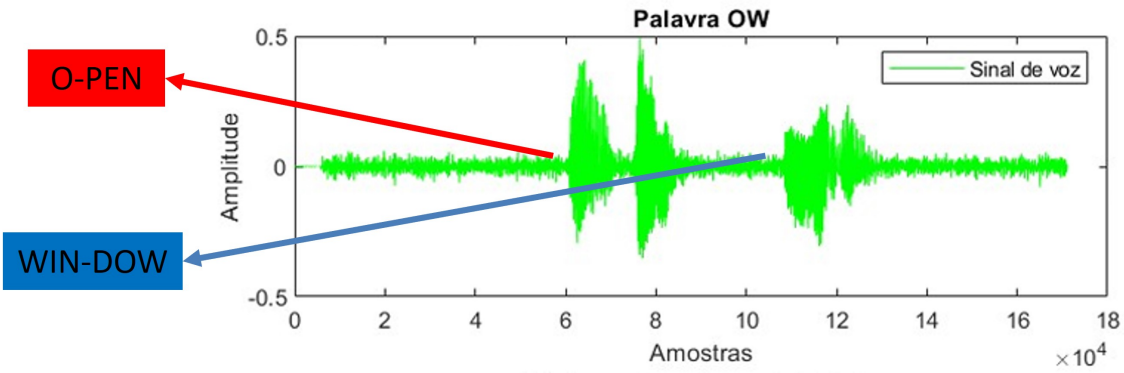
Fonte: Autoria própria.

palavras e fonemas são:

- Melhor transmissão no sinal;
- Melhor codificação e decodificação de palavras;
- Economia de banda maximizada, uma vez que apenas a parte útil do sinal será utilizada.

Na Figura 2 é ilustrada a localização das duas palavras contidas no comando *Open window*, é possível observar que existe um tempo de silêncio entre as formas de onda de cada palavra, bem como a diferença na amplitude que marca a pronúncia dos fonemas nestas duas palavras.

Figura 2 – Localização das palavras *open* e *window* no sinal de voz



Fonte: Autoria Própria.

Para maior eficácia na seleção de palavras o algoritmo de (GIANNAKOPOULOS, 2009) faz a diferenciação matemática de fonemas e espaços de silêncio estimando um limiar matemático e aplicando ao sinal este limiar para comparação da energia e a densidade espectral em cada intervalo analisado do sinal. Esse algoritmo se baseia no fato de que intervalos de silêncio possuem menor energia ao longo de si do que os intervalos que contém as palavras.

No exemplo da Figura 1a a palavra "TCC"o algoritmo de detecção de palavras identifica em quais pontos do sinal o limiar estabelecido foi ultrapassado, marcando início e fim da palavra possibilitando manipulação do sinal ou simples plotagem da detecção destacada por cor, como ilustrado na Figura 1b.

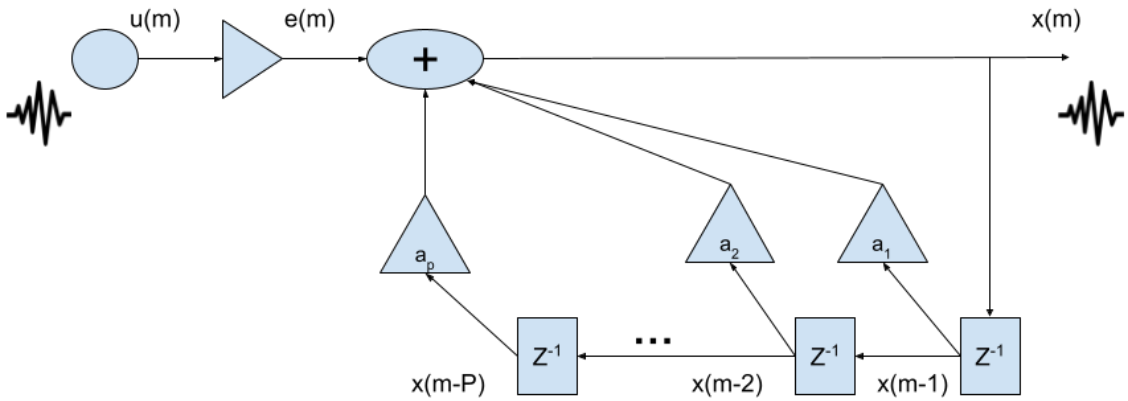
2.3 Linear Predictive Coding (LPC)

LPC é uma ferramenta computacional de predição onde o sinal pode ser reconstruído como correlação entre as variações aleatórias de cada amostra do sinal de voz. Em outras palavras, é a análise preditiva onde uma amostra pode ser reescrita como combinação linear das anteriores e é utilizada em diversas aplicações como diferenciação de locutores, de palavras e de letras. O diagrama ilustrado na Figura 3 representa o funcionamento de um sistema de LPC em blocos.

Os codificadores LPC modificam o sinal de voz, amostra por amostra, usando combinações lineares de P amostras passadas, como descreve a equação 2.2. Em que $\overline{x}$ é a previsão do sinal $x(m)$ e o vetor de coeficientes $a = [a_1, ..., a_p]$ contém os coeficientes conhecidos como *Linear Predictive Coefficients*. (VASEGHI, 2008)

$$\overline{x} = \sum_{k=1}^{P} a_k x(m-k) \tag{2.1}$$

Figura 3 – Diagrama de blocos de um LPC genérico



Fonte: Autoria Própria.

Os algoritmos geradores de LPC usam autocorrelação ou covariância para encontrar coeficientes e gerar o chamado filtro preditor, que na prática funciona nas seguintes etapas:

1. Estimar a amostra atual utilizando soma das P amostras passadas, reescrevendo a amostra assim:
$$s_m = \sum_{k=1}^{P} a_k s_(m-k) + e_m \tag{2.2}$$
onde, s_m: amostra atual, $s_(m-k)$: amostra anterior e e_m: erro de predição.

2. Coeficientes são calulados buscando minimizar a energia média de e_m:
$$E = \sum_{n=1}^{N} e_n^2 = \sum_{n=1}^{N} (\sum_{i=0}^{P} \alpha_i s_(n-i))^2 \tag{2.3}$$
onde N é nº total de amostras daquela janela de tempo onde o erro está sendo computado, valor típico é de 10 ms de janela para $N = 80$ amostras.

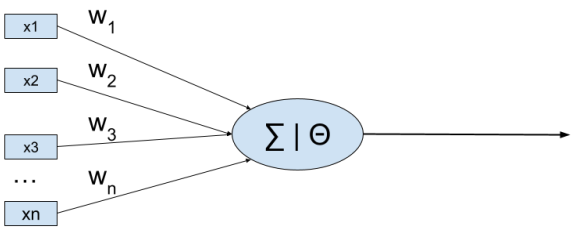
3. Para minimizar E é necessário derivá-la em relação a cada α_m :
$$\frac{\partial E}{\partial \alpha_m} = \sum_{i=0}^{P} \sum_{n=1}^{N} s_(n-m) s_(n-i) \alpha_i = 0 \tag{2.4}$$

4. A soma interna pode ser reescrita como coeficientes de correlação $C_i m$:
$$\sum_{i=1}^{P} C_i m \alpha_i = C_0 m \tag{2.5}$$

5. Por fim a equação 2.5 é usada para determinar os demais coeficientes preditores.

Os coeficientes preditores gerados podem ser utilizados diretamente como uma representação da voz que funcionará como entrada para etapa de aprendizado de máquina em sistemas de reconhecimento de locutor ou de palavras (ADAMI, 1997).

Figura 4 – Modelo matemático de neurônio de McCulloch e Pitts (1943)



Fonte: Autoria Própria.

Ao final da codificação por LPC cada exemplar de áudio de cada palavra de comando terá sido reescrito como combinação linear das amostras anteriores em uma matriz de ordem P, que determina a quantidade de coeficientes LPC gerados. É essa matriz que será manipulada pela Rede Neural Artificial para reconhecimento de comando.

$$Palavra = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_P \end{bmatrix}$$

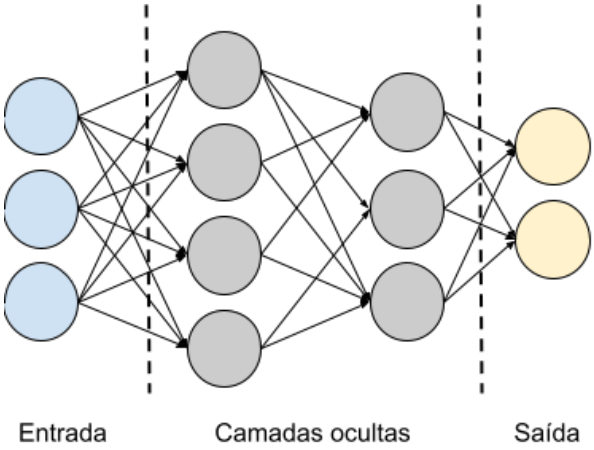
2.4 Redes Neurais Artificiais

O cérebro humano utiliza a memória e as associações para reter novos aprendizados o tempo inteiro e a partir de 1943 o trabalho de (MCCULLOCH; PITTS, 1943) equacionou matematicamente o funcionamento de um neurônio do cérebro humano, tornando possível a criação de sistemas complexos e cada vez mais realistas de réplicas artificiais do sistema de processamento dos neurônios, dando origem a neurociência computacional.

A grosso modo os neurônios reagem eletro-quimicamente quando a combinação linear das suas entradas atinge o limiar natural, como ilustrado na Figura 4. As redes neurais artificiais são um conjunto dessas unidades neuronais criadas artificialmente por computação, que matematicamente podem ser descritas como um conjunto de classificadores lineares (RUSSELL; NORVIG; MACEDO, 2013). RNAs tem sua relevância por realizarem processamento matemático complexo utilizando dados de entrada e gerando uma saída controlada sem necessidade de o usuário ter conhecimento da relações entre elas.

As RNAs são formadas por unidades, ou nós, conectados entre si e separados por camadas. Uma ligação entre dois nós i e j possui um peso numérico w_ij que serve para determinar propagação da ativação a_i ao longo da rede até a última camada. Então cada nó j realiza o cálculo da soma de suas entradas:

Figura 5 – RNA genérica com duas camadas ocultas



Fonte: Autoria Própria.

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \tag{2.6}$$

Logo após isso, aplica uma função de ativação g para obter o valor no nó j como saída:

$$out_j = g(in_j) = g \sum_{i=0}^{n} w_{i,j} a_i \tag{2.7}$$

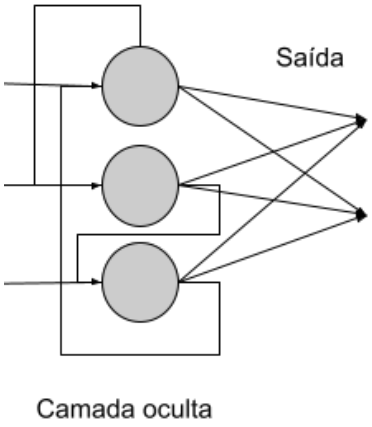
Sintetizando, as RNAs são capazes de gerar saídas controladas utilizando processamento matemático de pesos em suas unidades interligadas entre camadas. Além de mapeamento das relações entre entradas e saídas as RNAs são capazes de aprender utilizando grupos limitados de exemplos mas podendo tomar decisões a respeito de dados desconhecidos para rede, desde que tenham a mesma relação que o conjunto usado em treinamento (VALIATI, 2000).

Ao criar uma RNA o programador faz a separação dos exemplos de entrada como sendo exemplos de treinamento e exemplos de teste. O aprendizado de uma RNA se dá através do ajuste dos pesos $w_{i,j}$ durante a etapa de treinamento buscando reduzir o erro entre os valores de saída e os exemplos de teste. A Figura 5 descreve um exemplo da organização dos neurônios em camadas e as conexões entre eles que carregam pesos para aplicar a função de transferência.

De acordo com a topologia das ligações entre camadas as RNAs podem ser divididas em dois tipos principais:

- *Feedfoward*(FF) ou para frente, se a RNA não permitir que um nó na *i-ésima* envie sua informação de saída como parâmetro de entrada de nós em camadas com índice

Figura 6 – RNA recorrente genérica de uma camada



Fonte: Autoria Própria.

menor ou igual a i. Ou seja, a saída de cada neurônio sempre seguirá adiante para as camadas posteriores. A Figura 5 ilustra um exemplo;

- Recorrente, onde a saída de um neurônio *i-ésimo* pode ser utilizada como entrada de neurônios em camadas anteriores ou até na mesma camada. Ou seja, não é um fluxo de sentido único, ilustrado na Figura 6.

2.4.1 *Multi layer Perceptron* (MLP)

MLPs são uma arquitetura de RNA que possui uma camada de entrada, mais de uma camada oculta e uma camada de saída e o que possibilita a existência de várias camadas foi a criação do algoritmo de treinamento de *backpropagation*, que antes era apenas um algoritmo de iteração mas depois se tornou a denominação dada a redes MLP. *backpropagation* significa retro-propagar o erro da camada de saída para as camadas ocultas (RUSSELL; NORVIG; MACEDO, 2013).

Para entender melhor a retro-propagação é necessário entender dois parâmetros importantes (VALIATI, 2000) nesse tipo de rede:

- Taxa de aprendizado: quanto menor a taxa de aprendizado, menores são as variações de pesos durante a atualização. Maior taxa de aprendizado torna o treinamento mais rápido, mas pode conduzir a rede para saturação ou oscilação.

- Termo de momento: tem função de acelerar o aprendizado sem produzir oscilações, fazendo com que a atualização de parte dos pesos durante o treinamento continue sendo realizada na mesma direção das atualizações anteriores, evitando oscilações bruscas.

Figura 7 – Matriz de confusão genérica 2×2

	Previstos	
Realidade	Verdadeiro Positivo(Vp)	Falso Negativo (Fn)
	Falso Positivo (Fp)	Verdadeiro Negativo(Vn)

Fonte: Autoria Própria.

Com a definição desses parâmetros, os algoritmos de *backpropagation* então começam o treinamento incremental da RNA seguindo as etapas:

1. Inicialização: pesos, taxa de aprendizado, momento e critérios de parada;
2. Apresentação de um padrão de entradas vindo do conjunto de amostras separadas para treinamento;
3. Cálculo do erro na camada de saída;
4. Cálculo da atualização dos pesos;
5. Retro-propagação do erro para as camadas ocultas;
6. Cálculo do erro acumulado da rede. Caso o erro médio total atinja um valor abaixo do critério de parada é deduzido que a rede aprendeu, caso contrário retorna ao passo 2).

2.4.2 Matriz de confusão

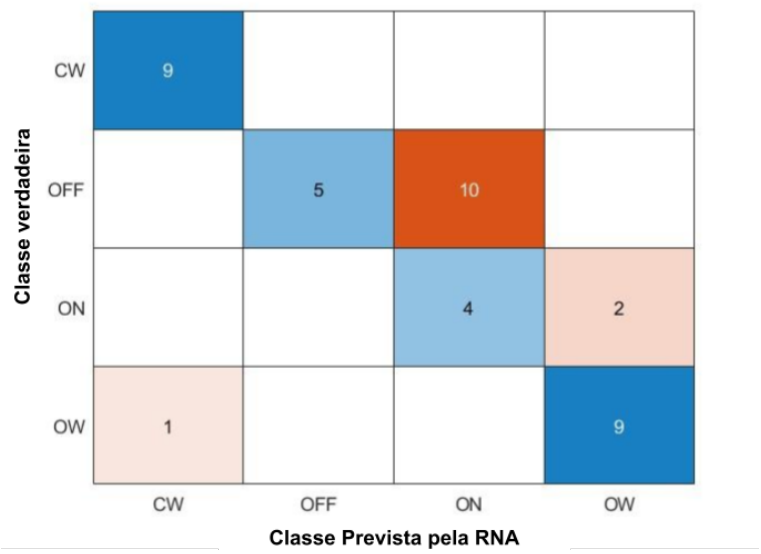
Matrizes de confusão são ferramentas de análise de previsão que comparam os valores previstos pelo sistema em questão com os valores reais da variável em questão (JUNIOR et al., 2022). O valores previstos são relacionados com os valores reais em uma matriz $n \times n$ onde n indica a quantidade de possibilidades de previsão.

Verdadeiros Positivos(V_p) acontecem quando o sistema previu determinado resultado como positivo e acertou, ou seja, o resultado era realmente positivo. Verdadeiros Negativos(V_n) acontecem quando o sistema previu que o resultado era negativo e acertou a previsão. Estes dois valores compõem a diagonal principal da matriz de confusão. Enquanto suas contra-partes Falso Negativo(F_n) e Falso Positivo(F_p) representam os momentos em que o sistema previu um resultado errado. Como mostra a Figura 7

A acurácia de uma RNA, portanto, pode ser medida avaliando o resultado dos testes, nesse caso com 30% das amostras, ou seja 30 testes, e é dada pela equação 2.8:

$$Acurácia = \frac{V_p + V_n}{Testes} \tag{2.8}$$

Figura 8 – Matriz de confusão 4×4



Fonte: Autoria Própria.

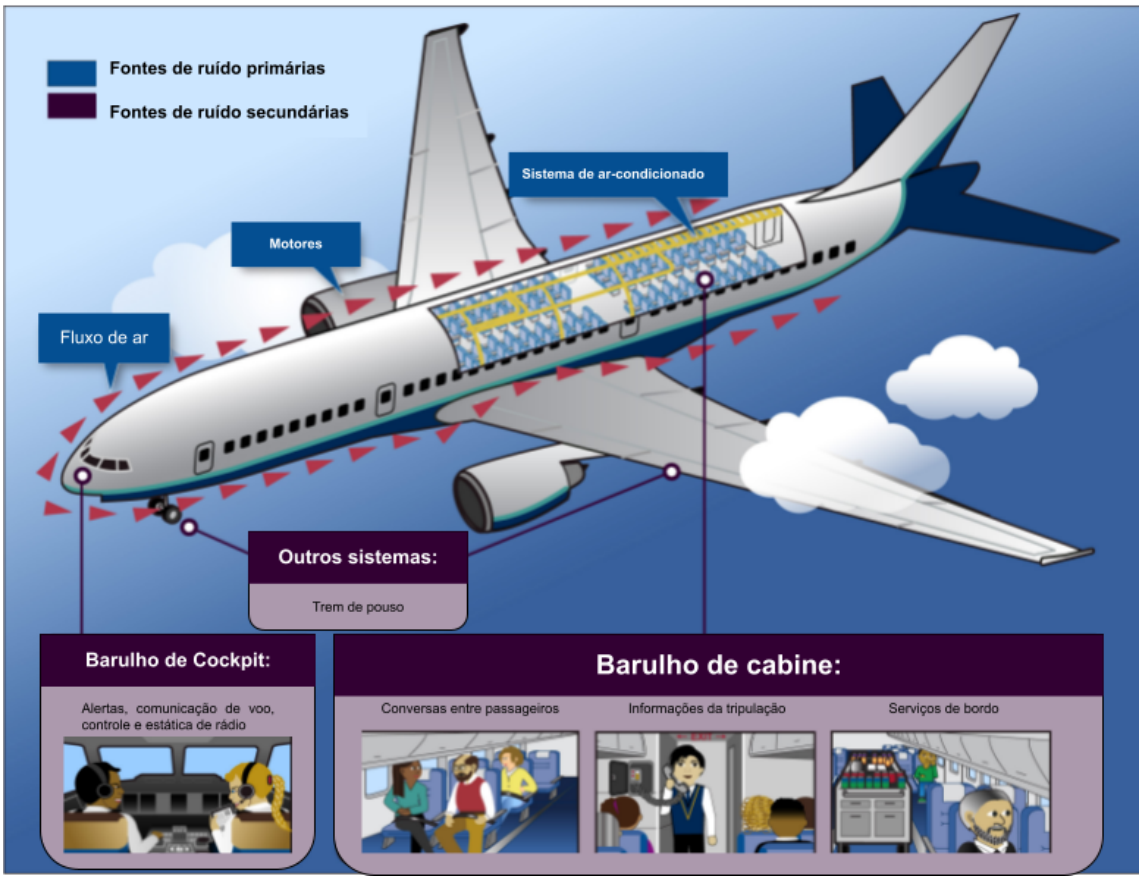
A Figura 8 representa a matriz de Confusão para o exemplo de validação com $67,5\%$ de acurácia. A matriz mostra que a RNA acertou 27 palpites de um total de 40 testes realizados, também é possível perceber os acertos e erros por comando, acertos na diagonal principal e erros nas demais posições da matriz.

2.5 Cabine de avião

O interior de uma aeronave comercial pode ser dividido em duas partes principais: o *cockpit* onde ficam piloto e co-piloto e a cabine onde ficam tripulação e passageiros do voo. Existem diversas fontes possíveis de ruídos sonoros durante um voo comercial, alguns são característicos apenas da cabine, como mostra a Figura 9. De acordo com os testes sumarizados em (DEFAZIO, 2017) o ruído interno pode chegar até 100 dB em alguns casos.

As fontes de ruído variam de intensidade de acordo com a idade da aeronave, tipo e localização do motor, fase do voo, velocidade da aeronave e localização do ouvinte. Existe atualmente muitas tecnologias sendo desenvolvidas para minimizar os efeitos de ruído e vibração causados por essas variáveis da cabine. No caso de sistemas de captura de voz, como foi visto anteriormente, são necessárias etapas de filtragem para que o ruído proveniente de barulho seja atenuado.

Figura 9 – Fontes de ruído no interior de aeronave comercial



Fonte: Traduzido de (DEFAZIO, 2017)

2.5.1 Sistemas de IFEC

O uso dos primeiros sistemas de IFEC na aviação está intimamente ligado aos primeiros voos de longa distância: quando os passageiros começavam a ficar entediados ou nervosos durante o voo era disponibilizado filmes para amenizar os efeitos da duração da viagem. Desde então os sistemas de IFEC vem se modificando e se atualizando para atender as novas necessidades dos passageiros.

Hoje os sistemas de IFEC dão margem à uma série de novos tipos de conectividade e possibilitam que os passageiros tenham acesso a entretenimento durante o voo, mas além disso possibilitam a implementação de sistemas inteligentes nas cabines (WANG et al., 2023). Para este projeto foi considerado um sistema de reconhecimento de voz cuja saída possa se conectar à acionamentos diversos no ambiente de cabine.

Tabela 1 – Evolução de sistemas IFEC

Estado da arte	Época	Características
Pré-internet	Antes de 1990	Sem internet, IFEC compartilhado
Consoles *seat-back*	1991	Telas *seat-back*, ou seja, nas costas da poltrona
Acesso à internet	2001	Introdução da internet a bordo (apenas *e-mails*)
Wi-Fi nos aviões	2006	Serviços de bordo via Wi-Fi
Wi-Fi grátis e rápido	2022	Wi-Fi gratuito e estável oferecido pela maioria das companhias aéreas

Fonte: Autoria própria.

3 Materiais e métodos

Neste capítulo serão descritos os materiais utilizados no trabalho na primeira seção e as metodologias aplicadas na segunda seção do capítulo. Materiais são todos os arquivos, *softwares*, comandos e demais recursos materiais produzidos e utilizados. Enquanto a metodologia diz respeito as técnicas de processamento, etapas de tratamento de dados, algoritmos e fluxograma utilizados durante o projeto.

3.1 Materiais

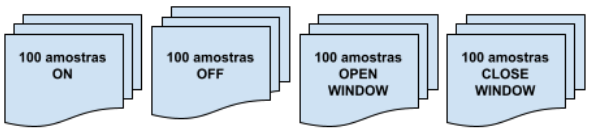
3.1.1 Dados de voz

A aquisição dos sinais de voz foi feita utilizando um grupo de microfones *Realtek(R) Audio* e driver nativo de notebook modelo Acer Nitro 5 AN515-55, o software Gravador de voz que é gratuito e nativo do sistema operacional *Windows 10* e captura áudio com taxa de bits de $173Kbps$. A voz pertence a apenas um locutor, cuja identidade é irrelevante para o trabalho.

Foram gravadas 100 amostras de no máximo 3 segundos para cada palavra de comando, totalizando 400 arquivos de áudio, como ilustra a Figura 10. Todas as amostras já continham ruído sonoro além da voz, mas foi adicionado som ambiente para simular o interior de uma aeronave comum, onde as fontes primárias de ruído foram descritas anteriormente. A junção das amostras de áudio deram origem ao banco de dados utilizado no projeto que posteriormente foi incrementado para obtenção de melhores resultados.

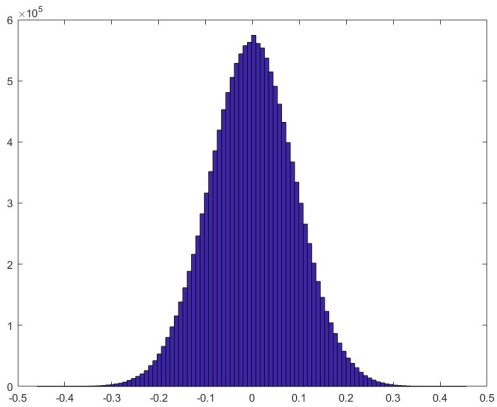
Para tornar as amostras mais próximas dos sinais de voz que serão capturados no ambiente de uma cabine de avião foi adicionado mais ruído, proveniente de um arquivo de áudio capturado durante um voo comercial. O ruído adicionado tem característica de uma distribuição normal, com valor médio situado em zero e desvio padrão $\sigma = 0,0913$, essas características foram extraídas do histograma do ruído ilustrado na Figura 11.

Figura 10 – Separação dos dados para RNA



Fonte: Autoria Própria.

Figura 11 – Histograma do sinal de ruído



Fonte: Autoria Própria.

3.1.2 Software Matlab

A ferramenta utilizada foi o MATLAB, versão 2023a devido a capacidade de processamento matemático e a possibilidade de realização de todas as etapas do sistema de reconhecimento de voz em um único *software*:

1. Leitura e escrita de dados em formato de matrizes e vetores essencial para preparação do banco de dados;
2. Geração dos coeficientes LPC;
3. Criação da Rede Neural Artificial;
4. Realização de testes para análise da precisão;

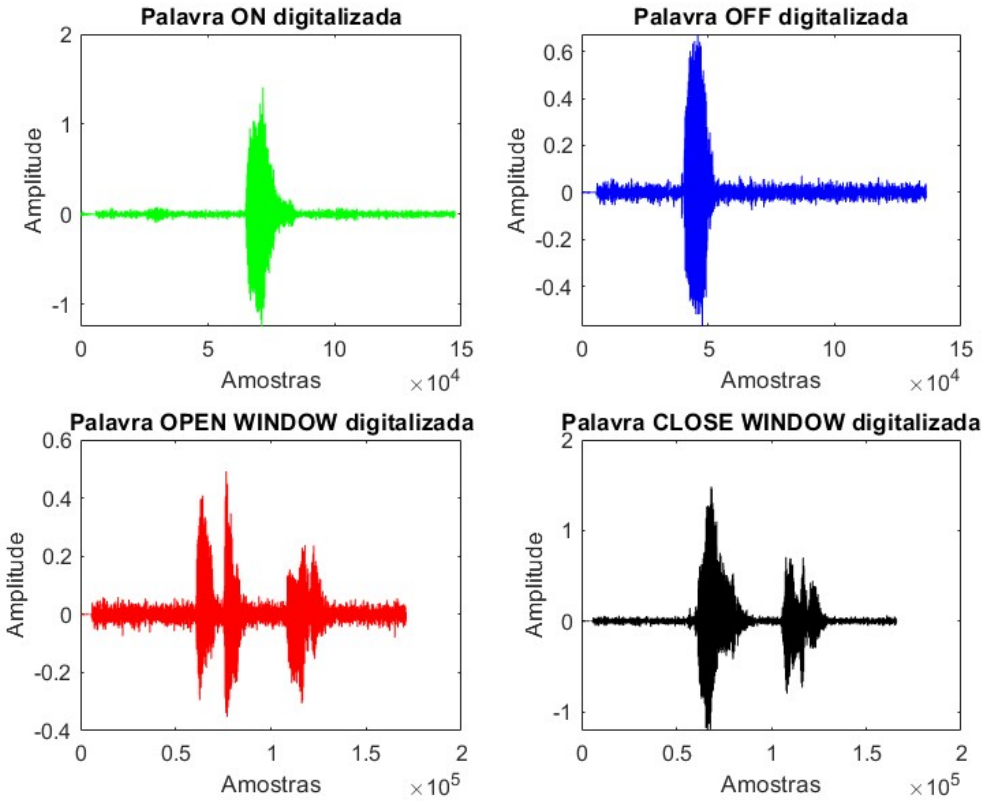
O Matlab dispõe de diversas funções e bibliotecas para leitura de arquivos de áudio, a função utilizada neste trabalho foi a função *built-in Audioread()* que recebe como entrada o nome do arquivo onde está o sinal de áudio, a quantidade de amostras e o tipo de amostras que se deseja capturar. As saídas dessa função são uma matriz contendo o valor discretizado de cada amostra e a frequência de captura. A Figura 12 mostra o exemplo da primeira amostra de cada palavra.

3.1.3 RNA

A função utilizada para criar uma RNA neste trabalho é uma função *built-in* do Matlab que gera RNAs do tipo *feedfoward*, sua forma básica de código está descrito a seguir:

```
rna = newff(PR,PT,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)
```

Figura 12 – Exemplo de amostras para cada palavra



Fonte: Autoria Própria.

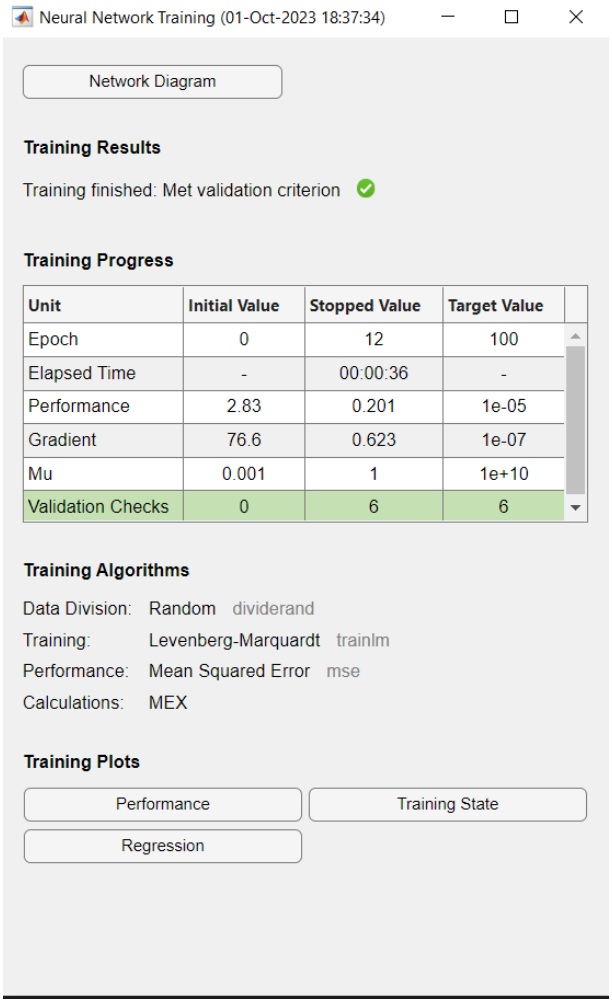
Os parâmetros de entrada dessa função são:

- PR: matriz com valores de entrada (treinamento e teste juntos);
- PT: matriz com valores da saída ou *targets*;
- S_n: Tamanho (quantidade de neurônios) da camada *n-ésima*;
- TF_n: Função de transferência da camada *n-ésima*, padrão: Sigmoide tangente hiperbólica;
- BTF: Função utilizada no treinamento *backpropagation*, padrão: *Levenberg-Marquadat*;
- PF: Função de performance, padrão: média dos erros quadráticos.

A Figura 13 ilustra a tela de carregamento nativa da função utilizada onde é possível acompanhar os parâmetros de treinamento e as características da rede neural. A RNA criada para validação da metodologia também contou com mais alguns parâmetros que detalham o processo de treinamento da rede, são eles:

- *Epochs*: Quantidade total de treinamentos previstos = 100;

Figura 13 – Tela de treinamento RNA



Fonte: Autoria Própria.

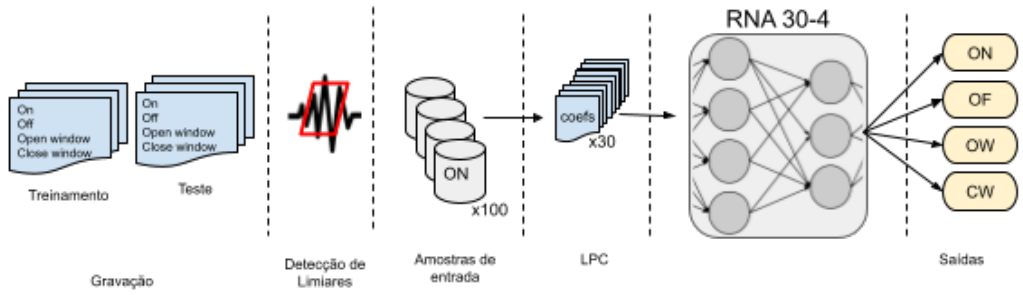
- Meta ou *target*: Valor de limiar que a RNA vai usar como um dos critérios de parada do treinamento = 10^{-5}.

3.2 Métodos

O projeto desenvolvido tem objetivo de validar a aplicação de RNA para reconhecimento de voz em cabines de avião utilizando coeficientes LPC para processamento das amostras de entrada da rede. Considerando que os atuais sistemas de IFEC em aeronaves possibilitam a conectividade de diversos sistemas de entretenimento, bem como a realização de atividades repetitivas dentro da cabine os comandos escolhidos para o projeto foram as expressões:

- *On* e *Off* para ligar e desligar respectivamente sistemas de ar-condicionado;

Figura 14 – Fluxo dos dados no sistema proposto



Fonte: Autoria Própria.

- *Open Window* e *Close Window* para abrir e fechar a persiana da janela, respectivamente.

O diagrama de blocos do sistema encontra-se descrito na Figura 14, contendo todas as etapas do sistema de reconhecimento de voz. Começando na gravação das amostras de voz e terminando nas saídas classificadas por comando.

3.2.1 Detecção de Limiares

Para detecção do limiar de amplitude do sinal de áudio foi constatado que tanto para comandos de uma só palavra (*On* e *Off*) quanto de duas palavras (*Open Window* e *Close Window*) o valor de amplitude de $0,2$ era suficientemente grande para diferenciar o que representa o som das letras e o que é ruído do som ambiente.

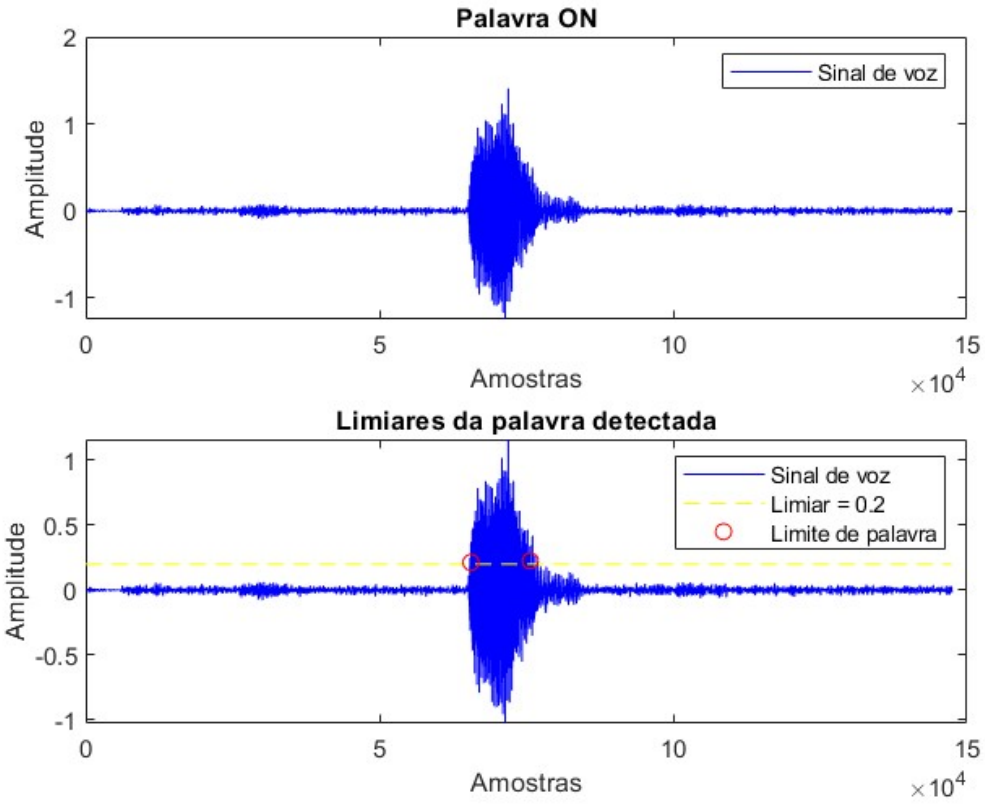
Para valores abaixo de $0,2$ houve captura de informações desnecessárias de som não útil, ou seja, algumas amostras de som ambiente tem amplitudes próximas do limiar sendo confundidas como começo ou fim de palavra, gerando erro na identificação. A Figura 15 mostra a detecção da palavra ON, marcando o limiar de $0,2$.

3.2.2 Geração dos coeficientes LPC

Encontrar coeficientes LPC implica em gerar os coeficientes de um filtro de Resposta ao Impulso Finita(FIR) que vai prever o valor de amostras atuais como combinação linear das amostras passadas. Os parâmetros de entrada da função *built-in* são a matriz do sinal a ser analisado e a ordem desejada para o filtro. A ordem define quantos coeficientes serão gerados e para o projeto foi definido um LPC de trinta e um coeficientes.

Como foi mencionado neste trabalho a geração de coeficientes acontece buscando minimizar a energia média do valor do erro acumulado. As variáveis de saída da função *lpc()* são uma matriz resposta contendo todos os coeficientes e o valor de erro de predição, que pode ser uma escalar ou um vetor. Ao final do processamento de lpc cada amostra

Figura 15 – Exemplo de detecção do limiar para palavra ON



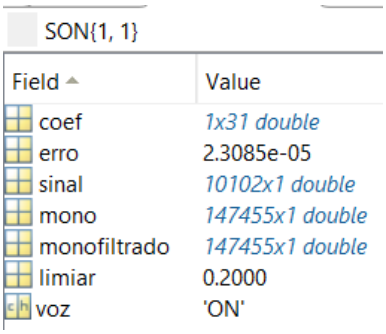
Fonte: Autoria Própria.

do sinal de voz fica na forma da *structure array* mostrada na Figura 16 com os seguintes campos de dados:

- coef: Contém o vetor de coeficientes LPC como ilustrado na Figura 17;
- erro: Contém o valor do erro de predição ao final do processamento;
- sinal: Contém apenas a palavra delimitada pelo algoritmo de detecção;
- mono e monofiltrado: Contém o sinal original e o sinal filtrado;
- limiar: Contém o valor do limiar que foi utilizado para detecção;
- voz: Contém o comando que aquele sinal representa.

O processo de gerar coeficientes foi realizado 400 vezes no total, sendo 100 vezes para cada palavra de comando. A partir disso, a ordem das amostras foi randomizada via software e salvas em matrizes que serão utilizadas como entradas e saídas da RNA.

Figura 16 – Exemplo da matriz gerada utilizando a função lpc, com adição de parâmetros sobre o sinal original



Fonte: Autoria Própria.

Figura 17 – Vetor de Coeficientes do 1º ao 10º termo

SON{1, 1}.coef

	1	2	3	4	5	6	7	8	9	10
1	1	-1.9454	0.6073	0.3753	0.1540	-0.0061	-0.0849	-0.0949	-0.0667	-0.0294

Fonte: Autoria Própria.

3.2.3 Banco de dados para treinamento da RNA

Antes de criar a rede neural é necessário separar quantas amostras serão utilizadas para treinamento da rede e quantas serão utilizadas para teste. Inicialmente para validação da metodologia e testes da função geradora de RNA foram utilizadas 30 amostras como entrada, divididas entre 20 para treinamento e 10 para teste, descrita anteriormente neste capítulo.

Depois de separadas a quantidade de amostras para treino e para teste é necessário escolher os parâmetros que definem a camada oculta da RNA, o número de camadas definido foi duas camadas e o modelo de RNA é o de *feedfoward* utilizando treinamento em formato de *backpropagation*.

As saídas da RNA são os quatro comandos de voz deste estudo, mas foi pensada uma codificação de palavra dois bits para que a saída possa ser utilizada em aplicações futuras. A codificação precisa ser um parâmetro enviado na função que cria a rede neural.

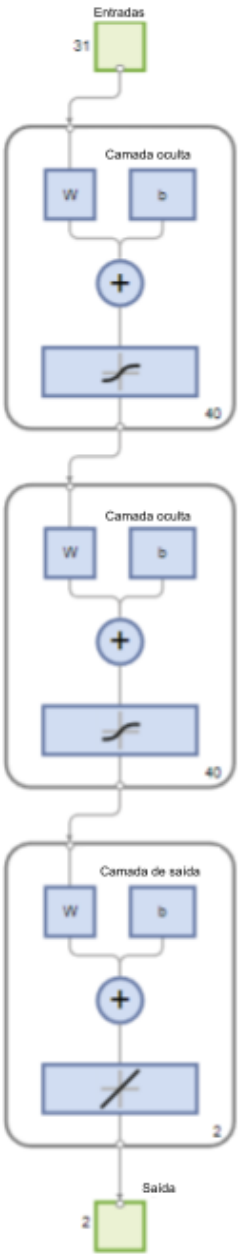
$$ON = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$OFF = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$OPEN - WINDOW = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$CLOSE - WINDOW = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Figura 18 – Diagrama de blocos da RNA 40-40-4



Fonte: Autoria Própria.

Ao final do processamento, o software cria uma RNA cujo diagrama de blocos está descrito na Figura 18. É possível observar a dimensão das entradas e das saídas, bem como o tamanho, em quantidade de neurônios das duas camadas ocultas.

4 Resultados

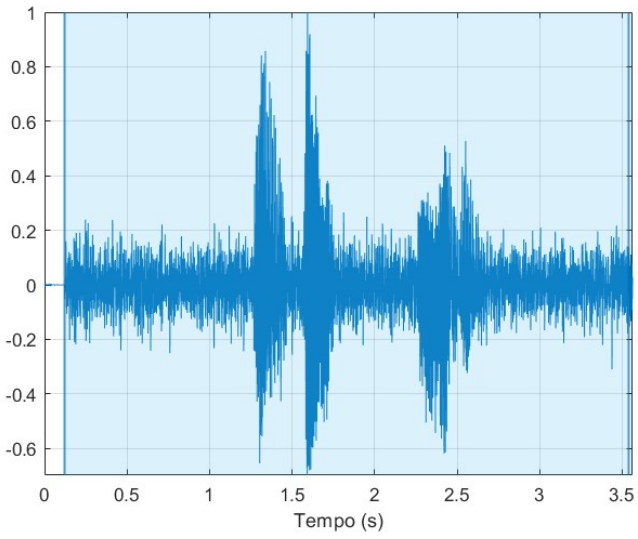
Este capítulo descreve os resultados obtidos, divididos por etapas de projeto: Detecção de comandos dentro do sinal de voz; Análise da performance da RNA para diferentes hiper-parâmetros; e Análise da acurácia através de Matriz de Confusão.

4.1 Detecção de comandos em sinais de voz

O algoritmo de detecção dos limiares de palavras que foi produzido neste trabalho por sua vez se mostrou mais eficaz que a função *built-in* do Matlab *detectspeech*, principalmente para comandos com duas palavras como mostra o exemplo na Figura 19 que mostra a dificuldade da função nativa em detectar os limites de começo e fim do comando enquanto o algoritmo produzido neste trabalho consegue detectar como mostra a Figura 20.

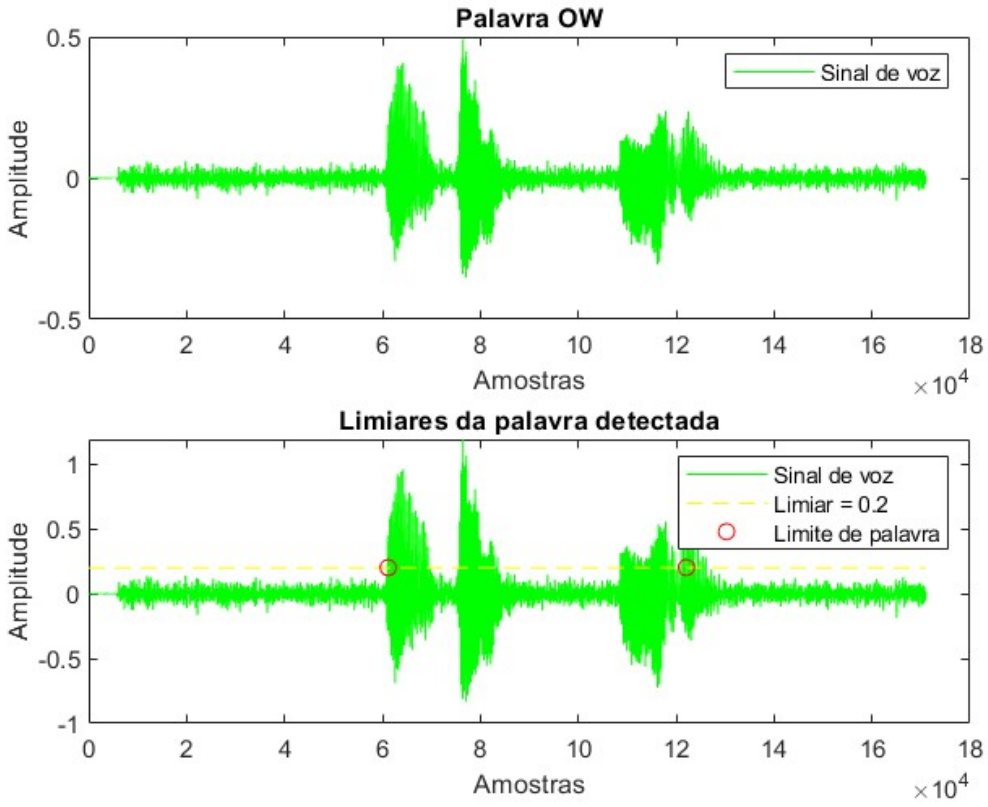
O algoritmo criado baseando-se em (GIANNAKOPOULOS, 2009) marca o ponto inicial e o ponto final da palavra de comando completa, mesmo se houver um espaço em silêncio entre palavras como é o caso dos comandos *Open Window* e *Close Window*. A Figura 20 mostra com o exemplo de *Open Window*(OW) que a detecção de comando mesmo com mais de uma palavra na expressão é eficaz e consegue englobar quase 100% da palavra final.

Figura 19 – Detectspeech() sendo utilizada em uma amostra do comando *Open Window*



Fonte: Autoria Própria.

Figura 20 – Detecção correta de comando Open Window



Fonte: Autoria Própria.

4.1.1 Reconstrução utilizando coeficientes LPC

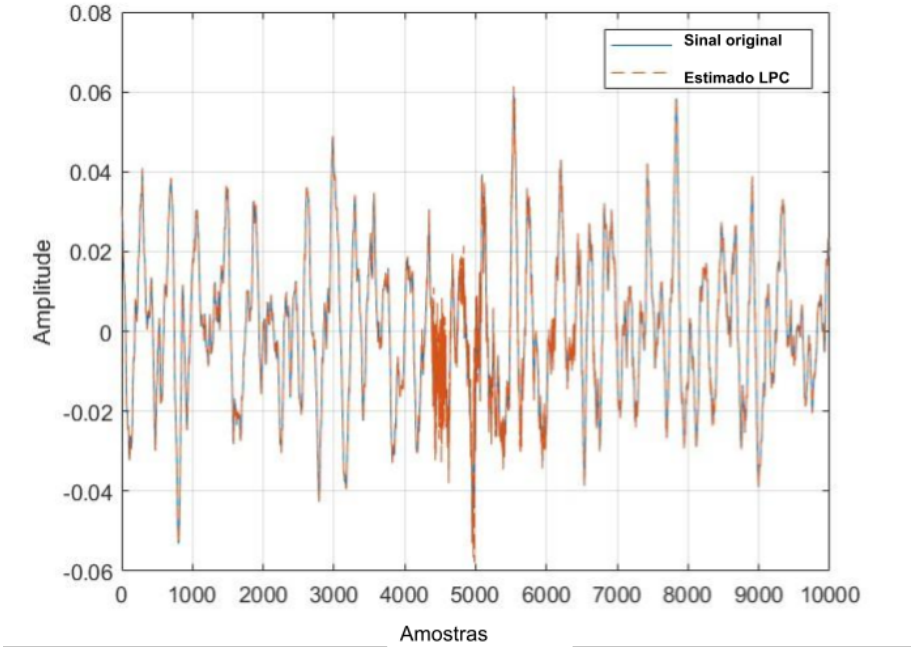
Os coeficientes LPC podem ser usados entre outras coisas para reconstrução do sinal de áudio uma vez que representam os coeficientes de um filtro aplicado ao sinal. A Figura 21 mostra como é possível reconstruir uma utilizando coeficientes LPC, provando a eficácia do método utilizado.

4.2 Análise da Performance da RNA

Inicialmente foi realizado testes criando redes de mesma topologia apenas repetindo o processamento das amostras e geração da RNA com 30 amostras de entrada, 2 camadas de 40 neurônios cada e 4 saídas. A RNA criada neste procedimento inicial teve acurácia média de $67,5\%$. Partindo desse resultado e analisando toda a metodologia apresentada foi definido que o banco de dados de entrada precisa crescer, ou seja, aumentar os padrões de treinamento e de testes para alcançar valores de acurácia mais próxima de 100%.

Foram feitas simulações com 30, 60 e 100 amostras por comando de sinal de voz e os resultados médios estão descritos na Tabela 2.

Figura 21 – Reconstrução do sinal de voz para uma amostra do comando *On*



Fonte: Autoria Própria.

Tabela 2 – Acurácia da RNA por quantidade de amostras

Amostras por comando	30	60	100
Qtde. Treinamento	20	42	70
Qtde. Testes	10	18	30
Camada 1 (Neurônios)	40	40	40
Camada 2 (Neurônios)	40	40	40
Acurácia (%)	67,5	84,72	97,5

Fonte: Autoria própria.

Foi definido então que a quantidade de amostras ideal para esse estudo é de 100 amostras de sinal de voz por comando, totalizando 400 amostras. Todos os resultados apresentados a seguir tiveram 400 mostras, divididas entre 280 para treinamento e 120 para testes.

Baseando-se nas análises realizadas em (ADAMI, 1997) e com a quantidade de entradas da RNA definida, o processo de obtenção dos resultados consistiu em variar então a quantidade de neurônios das camadas ocultas entre cinco, vinte e quarenta neurônios, gerando nove RNAs distintas para comparação de resultados via parâmetros de treinamento.

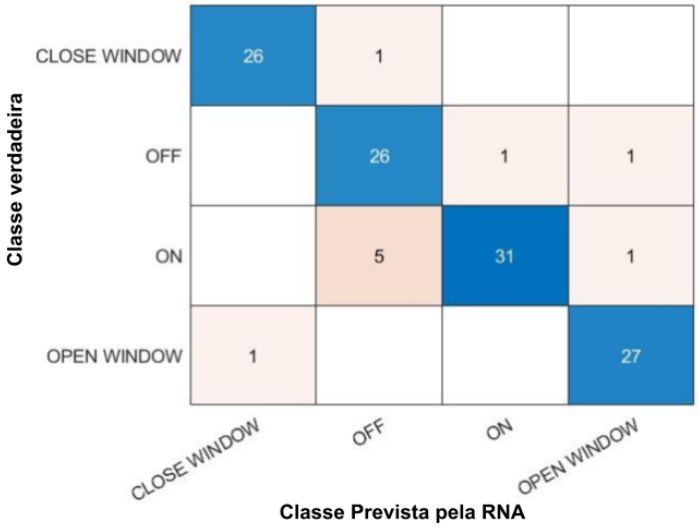
Para cada rede criada foi obtida uma acurácia, mostrando a eficácia de cada uma das topologias variando apenas o tamanho, em neurônios, das duas camadas. Os resultados estão descritos na Tabela 3 classificados de menor para maior acurácia.

Tabela 3 – Acurácia de RNAs com diferentes topologias

RNA	Camada 1	Camada 2	Acurácia medida
R1	5	20	71,67%
R2	20	5	70,60%
R3	5	5	72,50%
R4	5	40	80,00%
R5	20	40	90,00%
R6	40	5	90,83%
R7	20	20	91,67%
R8	40	20	92,50%
R9	40	40	97,50%

Fonte: Autoria própria.

Figura 22 – Matriz Confusão - RNA 20-20-4



Fonte: Autoria Própria.

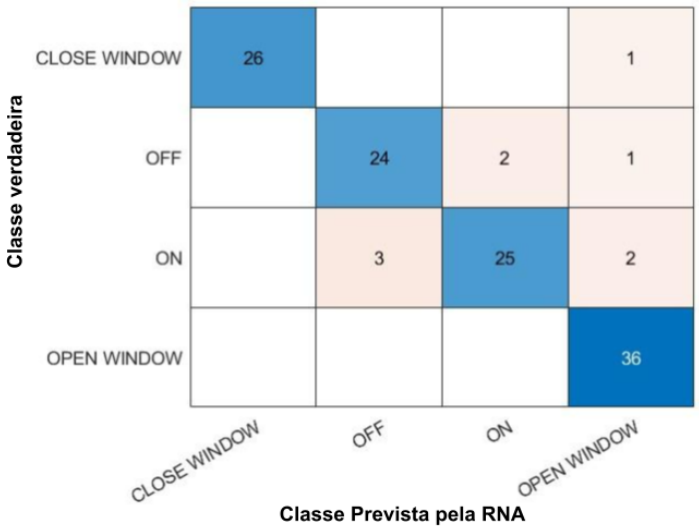
4.3 Análise de Acurácia através de Matriz de Confusão

As Figuras 22, 23 e 24 mostram a matriz de confusão das três melhores configurações de camadas enquanto a Figura 25 representa R1, a RNA de menor valor de acurácia. A Figura 18 é a representação em blocos da topologia de RNA de maior precisão e portanto a topologia escolhida para compor o sistema nesse trabalho.

Os gráficos de matriz de confusão utilizados estão no formato de mapa de calor, onde quanto maior o valor do termo for, mais quente será a cor escolhida para o termo da matriz. A diagonal principal representa os valores de verdadeiros positivos em azul pois a eficácia das RNAs descritas é alta, ao passo que os demais valores estão em cores mais frias.

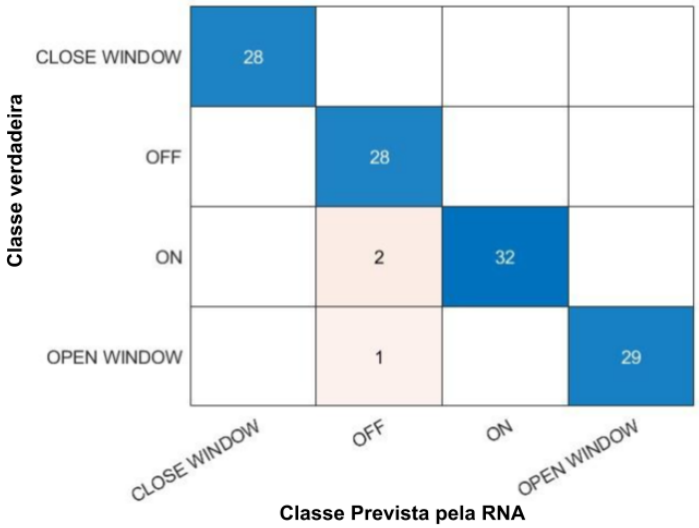
A rede R9 apresentou a eficácia mais próxima de 100% entre as redes criadas e

Figura 23 – Matriz Confusão - RNA 40-20-4



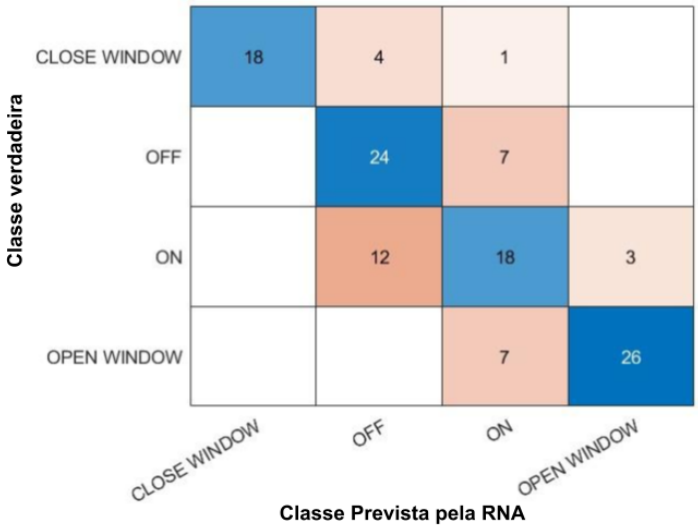
Fonte: Autoria Própria.

Figura 24 – Matriz Confusão - RNA 40-40-4



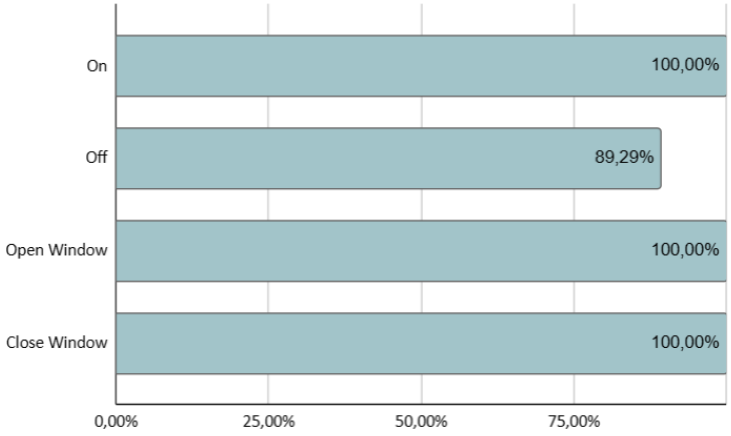
Fonte: Autoria Própria.

Figura 25 – Matriz Confusão - RNA 5-20-4



Fonte: Autoria Própria.

Figura 26 – Resultados da rede R9 por comando



Fonte: Autoria Própria.

seus resultados classificados por comando de voz estão descritos no gráfico da Figura 26 cujos dados foram retirados do teste realizado.

5 Conclusões

É importante ressaltar que os sinais de áudio analisados foram gravados fora de um estúdio de gravação audiovisual e portanto, a qualidade das variáveis de entrada pode ser ainda melhorada caso o sistema possa utilizar um processo de gravação com equipamentos profissionais em um ambiente acusticamente melhor. O algoritmo de detecção dos limiares de palavras que foi produzido neste trabalho por sua vez se mostrou mais eficaz que a função *built-in* do Matlab *detectspeech*.

Um vocabulário de comandos em assistentes virtuais é constituido por diferentes tipos de expressões, e o método desenvolvido mostra eficácia tanto para comandos de uma palavra curta como *On*, expressões de duas palavras como *Open Window* e *Close Window* e até para palavra *Off* que possui uma letra de som mudo, mais difícil de ser identificado.

O trabalho desenvolvido abre margem para futuras aplicações, principalmente considerando dois próximos passos:

- Adição de novos comandos: aumentando vocabulário também aumenta a quantidade de novos sistemas integrados ao reconhecimento de voz como por exemplo serviço de bordo na aeronave e interfaces com aplicativos de *streaming* (JIN; KIM, 2022);
- Adição de reconhecimento de Locutor: A mesma saída de comandos pode ser aproveitada adicionando apenas um ou mais bits à palavra final do processamento da RNA.

Contudo, a metodologia aplicada no sistema descrito neste trabalho torna possível analisar diferentes topologias de RNA e mostra que o uso de coeficientes LPC como parâmetros de entrada para redes neurais artificiais pode ser aplicado ao reconhecimento de voz com taxa de precisão de 97,5% na topologia R9, provando ser um sistema eficaz mesmo na presença do ruído sonoro de dentro da cabine de aviões. A acurácia pode ser considerada o resultado da escolha de topologia ideal e da robustez apresentada no método desenvolvido.

O trabalho cumpre seus objetivos geral e específicos descrevendo um sistema capaz de reconhecer comandos de voz dentro de um vocabulário pré-estabelecido utilizando algoritmo de limiar de detecção de palavra, coeficientes LPC como preparação de amostras para camada de entrada de Redes Neurais Artificiais em diferentes topologias de RNA. Todas as metodologias aplicadas puderam ser colocadas à prova ao longo do desenvolvimento e o software Matlab foi útil para centralizar todo o processamento.

Referências

ADAMI, A. G. *Sistema de Reconhecimento de Locutor utilizando Redes Neurais Artificiais*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 1997. Citado 3 vezes nas páginas 13, 18 e 36.

BECKER, R. *Análise qualitativa/quantitativa de algoritmos para a compressão de voz aplicados a redes de pacotes*. Tese (Doutorado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2009. Citado na página 15.

CARVALHO, J. L. A.; DIAS, D. Técnicas de codificação de voz aplicadas em sistemas móveis celulares. Citado na página 15.

DEFAZIO, P. *Commercial Aviation: Pilots' and Flight Attendants' Exposure to Noise aboard Aircraft*. [S.l.], 2017. Citado 3 vezes nas páginas 13, 23 e 24.

DING, I.-J.; YEN, C.-T.; DA-CHENG, O. A method to integrate gmm, svm and dtw for speaker recognition. *International journal of engineering and technology innovation*, Taiwan Association of Engineering and Technology Innovation, Taiwan, v. 4, n. 1, p. 38–47, 2014. ISSN 2223-5329. Citado na página 13.

GIANNAKOPOULOS, T. A method for silence removal and segmentation of speech signals, implemented in matlab. *University of Athens, Athens*, v. 2, 2009. Citado 2 vezes nas páginas 17 e 34.

JIN, M.-J.; KIM, J. K. Customer adoption factors for in-flight entertainment and connectivity. *Research in transportation business management*, Elsevier Ltd, v. 43, p. 100759, 2022. ISSN 2210-5395. Citado 2 vezes nas páginas 12 e 40.

JUNIOR, G. d. B. V. et al. DeterminaÇÃo das mÉtricas usuais a partir da matriz de confusÃo de classificadores multiclasses em algoritmos inteligentes nas ciÊncias do movimento humano. *Centro de Pesquisas Avançadas em Qualidade de Vida*, v. 14, n. v14n2, p. 1, 2022. ISSN 2178-7514. Citado na página 22.

KOHONEN, T.; OJA, E. Computing with neural networks. *Science*, v. 235, n. 4793, p. 1227–1227, 1987. Disponível em: <https://www.science.org/doi/abs/10.1126/science.235.4793.1227-a>. Citado na página 13.

LIANG, Y. et al. An improved noise-robust voice activity detector based on hidden semi-markov models. *Pattern recognition letters*, Elsevier B.V, AMSTERDAM, v. 32, n. 7, p. 1044–1053, 2011. ISSN 0167-8655. Citado na página 13.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 1943. Citado na página 19.

ONEWEB, T. *OneWeb Connected Passenger Report*. [S.l.], 2022. Disponível em: <oneweb.net/resources/oneweb-aviation-i-connected-passenger-report>. Citado na página 12.

RUSSELL, S.; NORVIG, P.; MACEDO, R. *Inteligência Artificial*. 3. ed. [S.l.]: Rio de Janeiro: Elsevier, 2013. ISBN 9788535237016. Citado 2 vezes nas páginas 19 e 21.

VALIATI, J. F. *Reconhecimento de voz para comandos de direcionamento por meio de redes neurais*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação, 2000. Citado 2 vezes nas páginas 20 e 21.

VASEGHI, S. V. *Advanced Digital Signal Processing and Noise Reduction*. 4. aufl.. ed. Newark: Wiley, 2008. ISBN 0470754060. Citado na página 17.

WANG, F.-Y. et al. Transportation 5.0: The dao to safe, secure, and sustainable intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, v. 24, n. 10, p. 10262–10278, 2023. Citado 2 vezes nas páginas 12 e 24.