

A era digital e suas implicações sociais: Desafios e contribuições

MODELO DE IDENTIFICAÇÃO DE MOSQUITOS *Aedes Aegypti* ATRAVÉS DE SOM EM *HARDWARE* EMBARCADO UTILIZANDO ARDUINO E *TENSORFLOW*

Guilherme Marques de Oliveira, Lucas Agostinho de Sousa, João Paulo dos Santos Gomez, Amita Muralikrishna, Carlos Eduardo Gomes, Carlos Eduardo Oliveira da Silva.

Instituto Federal de São Paulo/Engenharia de Controle e Automação, Rodovia Presidente Dutra, 145, Jardim Diamante - 12223-201 - São José dos Campos-SP, Brasil,
guilherme.marques@aluno.ifsp.edu.br, lucas.agostinho@aluno.ifsp.edu.br,
j.gomez@aluno.ifsp.edu.br, amita@ifsp.edu.br, c.e.gomes@ifsp.edu.br, carlossilva@ifsp.edu.br

Resumo

Este artigo visa encontrar uma maneira mais automatizada e precisa de realizar a identificação de mosquitos, usando áudios das batidas de asa em dispositivo embarcado e realizando o treinamento de modelos em nuvem, a fim de prevenir focos de mosquitos vetores de doenças como a Dengue e Chikungunya. Para isso, utiliza redes neurais convolucionais, uma técnica de *Machine Learning*, aliada a uma tecnologia embarcada, a *Tiny ML*. Com o uso de conjuntos de dados com sons de mosquitos e sons de ruído, foi possível captar, tratar o áudio e classificar a informação em espécies de mosquito ou se é som ambiente. O projeto atingiu alta precisão em simulação, mostrando ser possível a identificação utilizando áudio, já os testes em situação prática foram preliminares e mostraram predominância de falsos e verdadeiros negativos para mosquitos, além de precisão abaixo do esperado para verdadeiros positivos de mosquitos, sugerindo análises e incrementos no projeto para ser viável em situações reais.

Palavras-chave: *Machine Learning*, *Hardware* Embarcado, Arduino, *TensorFlow*, *Aedes Aegypti*.

Área do Conhecimento: Ciência da Informação - Tecnologia da Informação.

Introdução

O mosquito *Aedes Aegypti* é um dos principais vetores de doenças como Dengue, Zika, Chikungunya e febre amarela no Brasil e em outras regiões tropicais do mundo. A identificação precoce e precisa desse mosquito é fundamental para o controle e a prevenção dessas doenças, que afetam milhões de pessoas todos os anos. (Bhatt *et al.*, 2013) No entanto, os métodos tradicionais de identificação, baseados na identificação em campo, como montagem de armadilhas, são caros, demorados e requerem pessoal especializado (Fernandes, Cardoso e Recamonde-Mendoza, 2021).

Nesse contexto, surge a possibilidade de identificar o mosquito *Aedes Aegypti* através de métodos não invasivos e automatizados, como na classificação de áudios do mosquito em voo ou através de imagens, pois mosquitos de espécies distintas possuem características diferentes o suficiente para uma classificação assertiva. (Dieng *et al.*, 2016) Realizar essas classificações manualmente é inviável, por motivos de custo, imprecisão e saúde dos trabalhadores, pois existem dificuldades de identificar padrões e manter a assertividade por longos períodos, além de gerar riscos à saúde pela exposição a insetos transmissores de doenças.

Com o avanço das tecnologias nos últimos anos, o *Machine Learning* se tornou presente no dia a dia de todos, visto em redes sociais como Instagram, Tik Tok ou Youtube, em assistentes virtuais como Alexa, Siri ou Google Assistente e em diversas outras situações. O uso dessa tecnologia é amplamente utilizado nestes sistemas por envolver uma arquitetura altamente escalável, consistente, de baixo custo e capaz de fornecer lógicas e reconhecimento de padrões que humanamente são muito difíceis de identificar e programar de maneira determinística, como sons, vídeos e imagens (Warden e Situnayake, 2019).

No artigo, apresentamos uma proposta de identificação de mosquitos *Aedes Aegypti* através de classificação de áudio utilizando *Machine Learning* utilizando técnica de Redes Neurais Convolucionais com *TensorFlow Lite* em *hardware* embarcado com o Arduino *Tiny Machine Learning Kit*. O objetivo é verificar a assertividade e o funcionamento simulado do modelo, e com isso desenvolver um sistema portátil, eficiente e de fácil uso que possa auxiliar na detecção e no monitoramento do mosquito em diferentes ambientes. Para isso, utilizamos o microcontrolador com um sensor de som acoplado, que

A era digital e suas implicações sociais: Desafios e contribuições

capta o áudio do mosquito e utiliza um algoritmo de *Machine Learning* treinado com *TensorFlow*, que realiza a classificação do áudio de acordo com as características e padrões identificados pelo modelo.

Metodologia

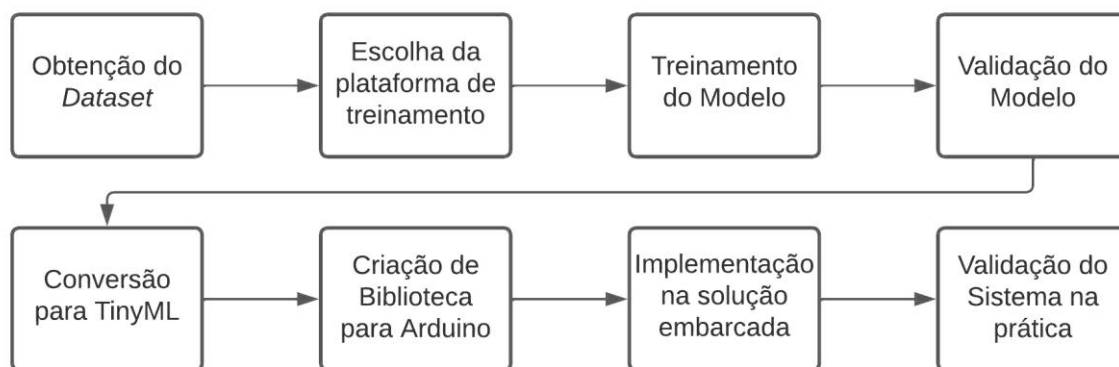
O primeiro trabalho realizado foi a decisão da arquitetura do projeto como um todo, desde a captação dos dados até a execução do modelo e tomada de decisão. Para a elaboração de um projeto envolvendo *Machine Learning*, é necessário que se identifique a plataforma onde se deseja treinar o modelo e qual tipo de dado será utilizado para a execução do processo de decisão. A plataforma final de execução escolhida foi o *Kit Tiny Machine Learning* do Arduino, com um modelo Nano 33 BLE Sense, ele possui suporte para *TensorFlow Lite* e microfone e sistema de visão embutidos.

A escolha do tipo de dado escolhido para treinamento foi áudio, por ser um formato de arquivo menos volumoso que os vídeos, e possui filtros bem efetivos e complexos para sistemas de detecção com *hardware* menos potentes, considerando a execução em Arduino. Após a escolha do tipo de dados, é obtido o *Dataset*, que pode ser gerado manualmente, artificialmente ou utilizando dados coletados por outro autor, que foi o caso desse projeto.

Obtido o *dataset*, foi escolhida a plataforma onde os dados serão treinados para a geração do modelo, utilizando linguagem de programação como *Python* ou plataformas de treinamento como o *Edge Impulse*. Depois de escolhida a plataforma, o modelo é finalmente treinado e validado, para garantir assertividade de funcionamento e tomada de decisão, usando para isso dados de teste nunca antes verificados pelo modelo.

Por fim, após ter o modelo funcional foi possível convertê-lo para a plataforma desejada, que é uma biblioteca Arduino usando *TensorFlow Lite*, onde fazemos o porte do modelo para sistemas embarcados e garantimos que terá um tamanho compatível com as limitações de *hardware*, que logo em seguida recebe o programa gerado e pode ser testado fora de ambiente simulado, onde é colocado à prova em situação real. O desenho completo da solução é representado pelo fluxograma apresentado na Figura 1.

Figura 1 - Arquitetura do Projeto.



Fonte: Autor.

Estruturado o escopo e fases do projeto, o *Dataset* é encontrado utilizando fontes grátis de dados como institutos sem fins lucrativos, empresas com fins científicos e mecanismos de busca como o *Google Scholar*. Foram identificados os *Datasets do Humbug* (Kiskin et al, 2020) e *Wingbeats* (Potamitis e Rigakis, 2016), ambos com dados de 1s de duração, que foram avaliados e selecionado o que era de mais simples utilização no treinamento. Após a decisão do *Dataset* a ser utilizado e a separação dos áudios de *Aedes Aegypti*, de *Anopheles Gambiae* (mosquito vetor da malária) e de sons de ruído de fundo, foram carregados 8 minutos de cada categoria de classificação para dentro da plataforma, que seguiu para a etiquetagem e logo em seguida à escolha da arquitetura de treinamento, baseada em classificação, espectrograma e transformada rápida de fourier (TRF) para uso de dados no domínio da frequência.

A era digital e suas implicações sociais: Desafios e contribuições

Para abordar temas envolvendo identificação de sons, o uso de Redes Neurais Convolucionais (RNC) se torna importante para o treinamento e geração da lógica computacional, por fornecer um algoritmo capaz de identificar padrões em sistemas de N dimensões. Para o caso de sons, é trabalhado um vetor de 2 dimensões - tempo e frequência - em forma de espectrograma para a identificação dos padrões. (Warden e Situnayake, 2019).

Decidida a arquitetura de treinamento, foi realizada a primeira execução de treinamento do modelo, escolhendo os parâmetros de funcionamento do tratamento de áudio com *Mel-filterbank Energy* (MFE) como número de pontos de TRF, número de filtros e normalização do áudio para remoção de ruídos. Para a parte de classificação usando rede neural, foram medidos parâmetros como quantidade de épocas, taxa de aprendizado, quantidade de camadas e neurônios da rede neural. A Tabela 1 mostra os parâmetros trabalhados e os valores utilizados para o treinamento do modelo.

Tabela 1 - Parâmetros do treinamento do modelo

Arquitetura da Rede Neural		Configurações da Rede Neural		Parâmetros do filtro Mel-filterbank energy	
Camada	Valor	Configuração	Valor	Configuração	Valor
<i>Input Layer</i>	3.960 features	<i>Number of training cycles</i>	100	<i>Frame lenght</i>	0.02
<i>Reshape Layer</i>	40 columns	<i>Learning rate</i>	5	<i>Frame stride</i>	0.01
<i>1D conv, pool layer</i>	8 neurons, 3 kernel size, 1 layer	<i>Validation set size</i>	20%	<i>Filter number</i>	40
<i>Dropout rate</i>	0.25	<i>Profile int8 model</i>	true	<i>FFT lenght</i>	256
<i>Flatten layer</i>	-			<i>Low frequency</i>	0
				<i>High frequency</i>	-
				<i>Noise floor (dB)</i>	-52

Fonte: Autor.

Feito o treinamento, começaram as fases de testagem, onde foram selecionados os dados separados anteriormente para teste para colocar o modelo à prova, onde é gerada a precisão de teste do modelo. Feita essa etapa e com o modelo validado, tem-se a conversão para *TensorFlow Lite* na forma de uma biblioteca de Arduino, que é modificada para gerar uma saída em visual via led, ou seja, caso o som analisado seja um *Aedes Aegypti*, ele irá indicar com cor azul, caso não, vermelho e no console serial com a probabilidade de cada uma das opções, considerando o áudio captado no momento pelo microfone do Arduino e os parâmetros de peso obtidos no treinamento do modelo. Por fim, o código foi embarcado e o funcionamento foi validado em situação real.

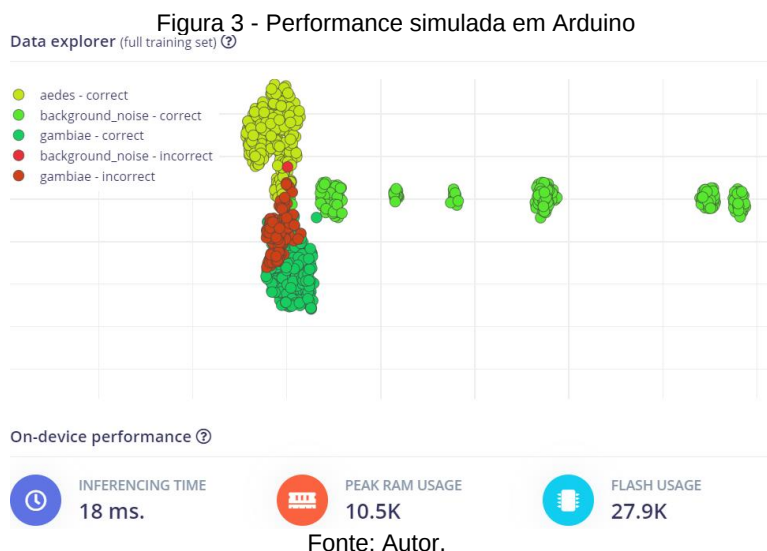
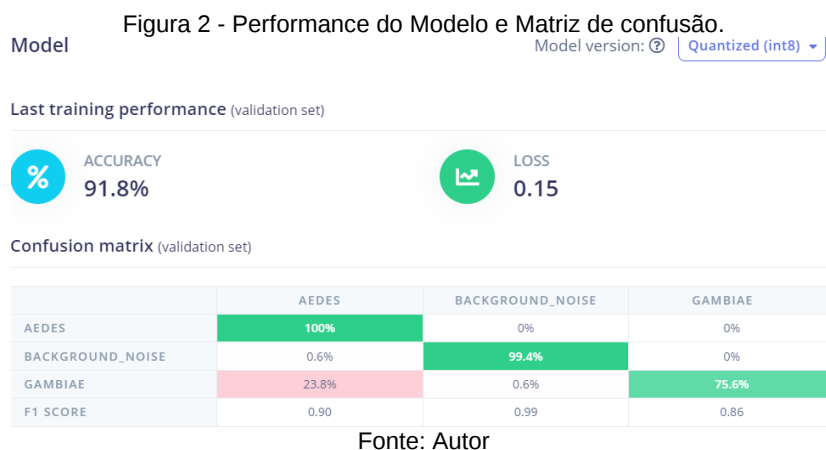
Resultados

Com a pesquisa realizada nos *Datasets* públicos ou para usos de fins científicos, foi identificado que o melhor *Dataset* para a utilização de áudios de *Aedes* e *Anopheles* foi o *Wingbeats* (Potamitis e Rigakis, 2016), por serem áudios já separados em arquivos de 1 segundo e estarem etiquetados corretamente por dia e tipo de mosquito. Porém, para o funcionamento do modelo, não basta ter áudios apenas de mosquito, mas também áudios de sons ambientes e ruídos de fundo, para que o modelo indique essa opção para evitar confusões do modelo.

Considerando essa necessidade, foi selecionado o *Dataset* do Google (Warden, 2018) com áudios públicos de sons de ruído, que foram adicionados ao modelo junto a áudios captados por um celular

A era digital e suas implicações sociais: Desafios e contribuições

Samsung Galaxy S23 inseridos na plataforma Edge Impulse pelos autores. Feita a separação de dados em 80% para treinamento e 20% para teste, o treinamento foi realizado com os parâmetros da Figura 2, e os resultados de performance junto com a matriz de confusão são mostrados nas Figuras 3 e 4.

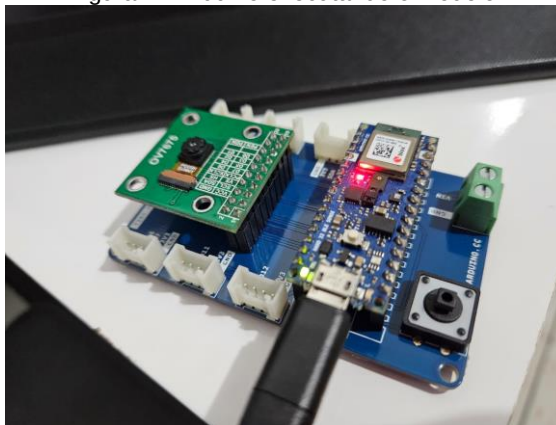


Com precisão acima de 90%, mostrada pela Figura 2, foi identificado que na matriz de confusão a maior taxa de erro foi na identificação errada de mosquitos gambiae como se fossem *Aedes*, e em outros casos ele performou de maneira satisfatória, com precisão acima de 99% de assertividade, considerando que a performance estimada em um Arduino Nano 33 BLE Sense foi de 18ms para inferência, uso de ram de 10.5k e 27.9k de memória flash, como é possível verificar na Figura 3. Usando os dados de treinamento, o teste de classificação teve assertividade de mais de 95%, mostrando grande capacidade de lidar com sons novos com as mesmas características.

A biblioteca gerada usando *TensorFlow* se baseia na probabilidade estimada de cada um dos parâmetros selecionados de corresponder a alguma das possibilidades definidas, seja *Aedes*, *Anopheles* ou ruído de fundo. Utilizando a plataforma, foi gerada a biblioteca em Arduino usando *TensorFlow Lite* e C++ com o código do modelo, que foi modificado para adicionar um indicativo visual nos casos de identificação de porcentagem acima de 80% de cada uma das etiquetas definidas, e logo em seguida o código foi carregado no Arduino. A Figura 4 mostra o Arduino em funcionamento com o código gerado, onde ele está programado para indicar com luz vermelha caso não tenha mosquito, luz verde em caso de identificação de *Aedes* e luz azul em caso de identificação de *Anopheles*.

A era digital e suas implicações sociais: Desafios e contribuições

Figura 4 - Arduino executando o modelo.



Fonte: Autor.

Quanto ao desempenho na prática, a média de tempo para execução de predição foi de cerca de 20ms, muito próximo do estimado na plataforma *Edge Impulse*. A Figura 5 mostra um exemplo de funcionamento do console enquanto o Arduino está em funcionamento com o código gerado.

Figura 5 - Saída Serial do Arduino.

```
Output Serial Monitor x
Message (Enter to send message to 'Arduino Nano 33 BLE' on 'COM9')

gambiae: 0.00000
Predictions (DSP: 50 ms., Classification: 19 ms., Anomaly: 0 ms.):
aedes: 0.00000
background_noise: 0.99609
gambiae: 0.00000
Predictions (DSP: 50 ms., Classification: 19 ms., Anomaly: 0 ms.):
aedes: 0.00000
background_noise: 0.99609
gambiae: 0.00000
Predictions (DSP: 50 ms., Classification: 19 ms., Anomaly: 0 ms.):
aedes: 0.00000
background_noise: 0.99609
gambiae: 0.00000
```

Fonte: Autor.

No funcionamento para identificação dos mosquitos em situação prática, os testes foram qualitativos e preliminares, funcionando com o uso de celulares com amostras de áudios não utilizados no treinamento, e os resultados mostram uma tendência grande do algoritmo de representar falsos negativos para mosquitos, muitas vezes não identificando áudios legítimos de *Aedes* e *Anopheles*, porém mostrando uma boa precisão na identificação de verdadeiros negativos de áudio de mosquito. Não foi possível testar o sistema com mosquitos e ruídos reais, o teste pelo celular, como já esperado, não trouxe a mesma acurácia na classificação, uma vez que o modelo não foi treinado com a qualidade de som oferecida pelo celular.

Conclusão

O projeto foi bem-sucedido em situações simuladas e mostrou alguns resultados preliminares práticos importantes, mostrando possibilidades de implementação do modelo em situações controladas e com potencial de implementação em situações reais. Podemos visualizar uma precisão acima de 90% de acerto e trazendo bons indicadores nos testes simulados. Mesmo repetindo testes, adicionando ruídos, sons externos, ainda foi possível diferenciar quais representavam *Aedes* ou não considerando situações simuladas e gravadas.

A era digital e suas implicações sociais: Desafios e contribuições

A utilização dos filtros e do treinamento do modelo foram funcionais e com adição de uma maior massa de dados e um melhor porte do código para o Arduino, é possível realizar testes utilizando mosquitos reais com armadilhas e situações controladas. Avaliando mais situações práticas, usando *Datasets* mais abrangentes e podendo ampliar o número de insetos detectados, é possível levar o projeto adiante e levar para locais em que a identificação é necessária, como em locais públicos, residências e até mesmo outros estudos de auxílio.

Considerando a situação prática em que foi submetida a solução, o modelo treinado se mostrou impreciso para identificar verdadeiros positivos de mosquitos, exigindo novas análises e implementação de melhorias no modelo e código da solução, para que seja confiável o suficiente para mais testes práticos com mosquitos reais. Nesse artigo o foco foi direcionado para a avaliação da hipótese de identificação dos mosquitos utilizando áudio, e a validação prática ficou limitada pela inviabilidade, no momento, de testes com mosquitos reais.

Em próximos estudos, pretende-se analisar mais aspectos práticos da solução e maiores validações em situações reais, além da adição de novas espécies de mosquitos e melhoria do modelo utilizando melhores técnicas de filtragem e volume de dados, deixando assim o projeto mais robusto. Com essas tratativas e soluções, espera-se que seja possível dar enfoque na implementação prática do projeto em locais em que é necessária essa identificação para prevenção de doenças, ou em estudos, como de instituições de pesquisa, ou ainda em locais que possuem alta taxa de casos de doenças relacionadas a esses mosquitos em análise.

Referências:

BHATT, Samir; GETHING, P. W.; BRADY, O. J.; MESSINA, J. P.; FARLOW, A. W.; MOYES, C. L.; DRAKE, J. M.; BROWNSTEIN, J. S.; HOEN, A. G.; SANKOH, O.; MYERS, M. F.; GEORGE, D. B.; JAENISCH, T.; WINT, G. R.; SIMMONS, C. P.; SCOTT, T. W.; FARRAR, J. J.; HAY, S. I. **The global distribution and burden of dengue**. *Nature* 496, 504–507, 2013. DOI: 10.1038. Disponível em: <https://www.nature.com/articles/nature12060>. Acesso em 10 jun. 2023

DIENG, H.; ABANG, F.; AHMAD, A.; ABD GHANI, I.; SATHO, T.; MIAKE, F.; AHMAD, H.; ZUHARAH, W.; MAJID, A.; MORALES, R.; MORALES, N.; HIPOLITO, C.; NOWEG, G. **Physical characteristics and reproductive performance in Aedes** (Diptera: Culicidae). *Journal of Entomological and Acarological Research*, [S. l.], v. 48, n. 3, p. 323–331, 2016. DOI: 10.4081/jear.2016.5883. Disponível em: <https://www.pagepressjournals.org/index.php/jear/article/view/5883>. Acesso em: 15 aug. 2023.

FERNANDES, M.; CORDEIRO, W.; RECAMONDE-MENDOZA, M.. **Detecting Aedes Aegypti mosquitoes through audio classification with convolutional neural networks**. *Computers in Biology and Medicine*, v. 129, p. 104152, 2021.

KISKIN, I.; COBB, A. D.; WANG, L.; ROBERTS, S. **Humbug zooniverse: A crowd-sourced acoustic mosquito Dataset**. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). p. 916-920. IEEE, 2020.

POTAMITIS, I.; RIGAKIS, I. **Large aperture optoelectronic devices to record and time-stamp insects' wingbeats**. *IEEE Sensors Journal*, v. 16, n. 15, p. 6053-6061, 2016.

WARDEN, P. **Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition**. ArXiv e-prints, 2018. DOI: <https://doi.org/10.48550/arXiv.1804.03209>.

WARDEN, P.; SITUNAYAKE, D. **Tinyml: Machine learning with tensorflow lite on Arduino and ultra-low-power microcontrollers**. O'Reilly Media, 2019.