TinyML aplicado ao reconhecimento de teclas a partir do sinal sonoro produzido pelo pressionamento em teclado físico

Aguinaldo Cardozo da Costa Filho, João Sinohara Silva Sousa
Renam Silva, Rodrigo Lantyer Marques Dantas

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)
Campus São José dos Campos
Curso de Engenharia de Controle e Automação

1. Nome do projeto

TinyML aplicado ao reconhecimento de teclas a partir do sinal sonoro emitido no pressionamento de teclado físico.

Membros da equipe

a. Aguinaldo Cardozo da Costa Filho
b. João Sinohara Silva Sousa
c. Renam Silva
d. Rodrigo Lantyer Marques Dantas

2. Considerações Iniciais

Os teclados mecânicos ainda são uma das principais *interfaces* homem-máquina devido à sua facilidade de operação, eficiência e baixo custo. São amplamente utilizados em fechaduras para controle de acesso e em terminais de acesso à serviços bancários. A possibilidade de identificação de uma sequência de teclas digitadas é uma séria ameaça à segurança de sistemas.

A proposta é estudar a possibilidade de identificação de teclas pressionadas em um teclado mecânico a partir da análise de sinais sonoros captados por meio de um microfone fixado próximo ao teclado. Alguns estudos já foram desenvolvidos e ilustram que os sons gerados ao se pressionar teclas de teclados de computador, teclas de terminais de caixas eletrônicos e ataques acústicos em outros dispositivos podem sugerir distinções que podem ser estudadas sob a abordagem proposta no trabalho.

A abordagem desenvolvida é a de análise sonora e/ou de vibração mecânica em teclado e originada pelo ato de teclar.

Uma mesma pessoa pode apertar uma tecla de várias formas, com maior ou menor intensidade, maior ou menor permanência da pressão na tecla, variação nas componentes de força dadas pelo ângulo do dedo relativo ao teclado e etc. Por outro lado, mesmo que se esforce para pressionar uma tecla de modo semelhante, o som e a vibração gerados no equipamento podem ainda ser bem variados.

Um outro elemento complicador de uma possível modelagem é o fato do sistema ser perturbado em posições físicas distintas, correspondentes às diferentes teclas. Vibrações aparentemente espúrias ocorrem no teclar de algumas teclas específicas, devido à oscilação de partes internas do teclado, que ressonam na presença de energia naquela vizinhança. Este fenômeno pode não ocorrer em todos os pressionamentos, entretanto em situações em que a pressão exercida ultrapassa algum limiar.

Esse trabalho aborda o problema como um caso de classificação de padrões via aprendizagem de máquina.

Obtenção das Amostras

O processo de amostragem envolveu uma única pessoa, pressionando aproximadamente 200 vezes cada uma das teclas correspondentes aos dígitos de 0 a 9 do teclado utilizado no experimento. A coleta das amostras pode ser enriquecida com a adição de mais pessoas uma vez que cada pessoa possui uma maneira diferente de teclar, que envolve variações na intensidade e na posição de toque na tecla. Após a amostragem (Dos 2912 pressionamentos), 80% das amostras foram reservadas para o treinamento do modelo e 20% para o teste.

3. Objetivos

O objetivo geral do projeto é expor a ameaça de captação de informações confidenciais tais como senhas e outras opções de códigos em dispositivos equipados com teclados mecânicos. Para esse fim, será explorada a possibilidade de discriminação da tecla pressionada pelo sinal sonoro produzido e/ou sinais de unidades de medidas inerciais [1] utilizando TinyML.

Os objetivos específicos incluem:

- a construção de um conjunto de treinamento;
- desenvolvimento, otimização, implementação e avaliação de uma rede neural artificial para realizar a predição de teclas pressionadas usando como entrada sinais capturados pelo microfone e/ou acelerômetro/giroscópio. A plataforma de desenvolvimento com um embarcado de baixo consumo de energia, o Arduino Nano 33 Sense, será empregada para aquisição de dados e implementação do modelo.

4. Descrição do Projeto

Para atingir o objetivo definido, será desenvolvido uma prova de conceito empregando a plataforma de desenvolvimento Arduino Nano 33 Sense [2] para aquisição de sinais de som e/ou vibração produzidos ao pressionar um teclado numérico. O projeto será desenvolvido nas etapas mostradas na Figura 1 e discutidas a seguir:

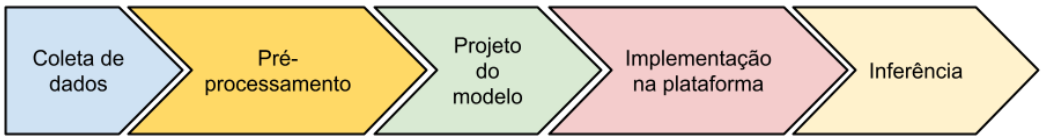


Figura 1 - Esquemático das etapas do projeto

- **Coleta de dados:** essa etapa consiste na coleta de sinais de som e/ou vibração produzidos ao pressionar um teclado numérico de 0 a 9. O projeto busca confirmar ou refutar a ideia de que o pressionamento de cada tecla produz um padrão de som e/ou vibração característico que pode ser explorado para discriminar a tecla pressionada. Naturalmente, essa possibilidade torna real a ameaça de revelação de dados sensíveis, como, por exemplo, senhas e votos depositados em urnas eletrônicas. Um ataque como o descrito, seria possível fixando um dispositivo eletrônico equipado com unidade de medidas inerciais no equipamento alvo do ataque ou explorando a disponibilidade da unidade no próprio dispositivo. Ideias similares já foram exploradas anteriormente na literatura técnica. Em [3] é exposta a ameaça de revelação de senha em fechaduras eletrônicas usando a vibração produzida no pressionamento de teclas. Após adquiridos, os sinais são enviados para uma unidade de processamento para inferência da senha. O trabalho [4] explorou o sinal sonoro, captado pelos microfones, produzido pelo acionamento de teclas virtuais em *smartphones* e *tablets* para inferir senhas e palavras digitadas.
- **Pré-processamento:** os sinais serão pré-processados para eliminação de erros da aquisição, checagem da consistência, anotação de rótulos e formatação para consumo no treinamento do modelo. Além disso, serão exploradas transformações/operações nos sinais com o objetivo de produzir uma representação mais adequada para a tarefa de classificação. Os sinais sonoros capturados são transformados para a representação tempo-frequência conhecida Mel Filterank Energy (MFE).
- **Projeto do modelo:** a etapa de projeto seguirá de forma iterativa, alternando entre projeto da arquitetura, treinamento do modelo e avaliação do desempenho. O objetivo é produzir um modelo equilibrado em termos de desempenho e recursos empregados, tendo em vista que o modelo final a ser embarcado na plataforma de Arduino Nano 33 Sense. O projeto da arquitetura inclui a definição do número de camadas, não linearidades/funções de ativações empregadas, quantidade/tamanho dos filtros, entre outros aspectos da arquitetura do modelo. Em seguida, o modelo será otimizado para minimizar a função objetivo usando pares de entrada/saída do conjunto de treinamento. O modelo treinado será avaliado no conjunto de validação e as métricas de desempenho serão usadas para realizar ajustes no modelo.
- **Implementação na plataforma:** essa etapa consiste em converter e embarcar o modelo resultante da conversão na plataforma Arduino Nano 33 Sense.
- **Inferência:** a plataforma Arduino Nano 33 Sense embarcada com o modelo final será usada para realizar inferência em sinais novos.

5. Descrição

a. Diagrama de blocos

A Figura 2 exibe um diagrama com blocos funcionais para implementar a solução. Ressalta-se que o diagrama está sujeito a alterações ao longo da implementação, o diagrama atual reflete o estágio de planejamento. A plataforma do Arduino Nano 33 Sense poderá ser colocada nas proximidades do teclado para captação de sons teclados.

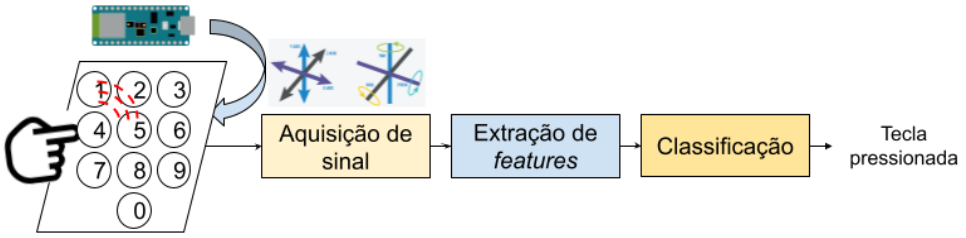


Figura 2 - Diagrama de blocos genérico da solução

b. Hardware

A Figura 3 mostra o esquemático da montagem da prova de conceito. A montagem consiste de uma plataforma sobre a qual estão distribuídas espacialmente teclas de 0 a 9 e da plataforma Arduino Nano 33 Sense que irá capturar o som e realizar a inferência das teclas pressionadas.

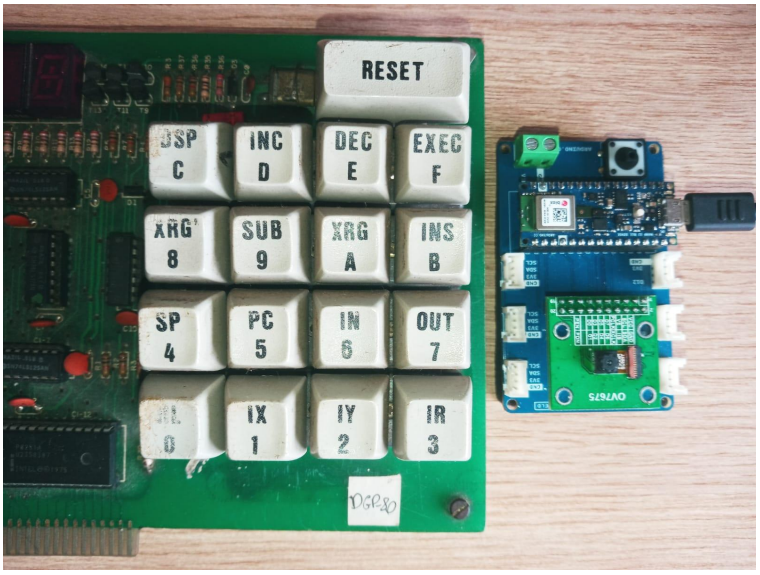


Figura 3 - Montagem da prova de conceito.

c. Conjunto de dados

Para o desenvolvimento do projeto foi realizada a aquisição do som produzido pelo pressionamento de cada tecla empregando o microfone disponível na plataforma Arduino Nano 33 Sense e fixada na plataforma de teclado. Para criar um conjunto de amostras diversificado e representativo da distribuição do sinal sonoro, a aquisição será realizada em várias etapas, com algumas variações na configuração. As variações previstas incluem a utilização de diferentes usuários, posicionamento da plataforma Arduino Nano 33 BLE Sense, entre outras que contribuam para a diversificação do conjunto de dados.

d. Pré-processamento

Como discutido anteriormente, os sinais foram pré-processados para garantir a consistência do volume de dados tanto para treinamento quanto para validação. Além disso, serão exploradas transformações/operações nos sinais com o objetivo de produzir uma representação mais adequada para a tarefa de classificação, como o uso dos sinais no domínio da frequência e utilizar técnicas que explorem melhor a separação de classes desse conjunto de recursos. Para o treinamento do modelo, dividimos o conjunto de dados em 2 partes: treinamento e dados de teste, em uma proporção de 80/20.

e. Projeto do modelo

No atual estágio do projeto, foi empregada uma rede neural artificial *feedforward.* Nas camadas iniciais que realizam a extração de características empregadas unidades totalmente conectadas ou unidades convolucionais em que os pesos são compartilhados entre neurônios. Sem perda de generalidade, as camadas iniciais da arquitetura exibida na Figura 4 empregam unidades totalmente conectadas. Após a extração de *features,* a representação produzida é geralmente empilhada para alimentar camadas totalmente conectadas que realizam a classificação. No caso em particular do projeto, a etapa de classificação produziu a probabilidade de o sinal sonoro ter sido originado no pressionamento de cada uma das teclas disponíveis no teclado, ou seja, o classificador irá produzir um vetor de probabilidade. A classe predita será aquela com maior probabilidade.

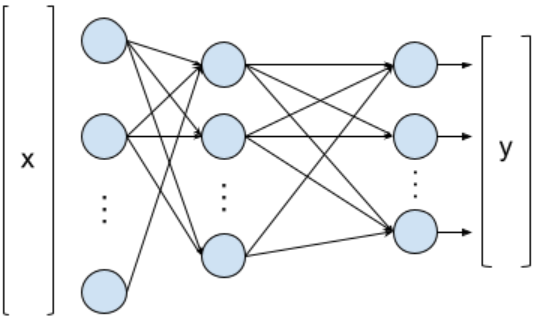


Figura 4 - Arquitetura genérica da rede a ser explorada

A arquitetura da rede neural artificial empregada para a classificação dos sinais sonoros capturados é exibida na Tabela a seguir. O modelo final foi obtido após várias iterações de adição/remoção de módulos, variação de parâmetros e otimizações.

Tabela: Arquitetura da rede

Layer (type)	Output Shape	Param #
input_32 (InputLayer)	[(None, 3960)]	0
reshape_39 (Reshape)	(None, 99, 40)	0
conv1d_32 (Conv1D)	(None, 99, 8)	1608

```
average_pooling1d_32         (None, 50, 8)             0
_________________________________________________________________
dropout_27 (Dropout)         (None, 50, 8)             0
_________________________________________________________________
conv1d_33 (Conv1D)           (None, 50, 8)             200
_________________________________________________________________
average_pooling1d_33         (None, 25, 8)             0
_________________________________________________________________
flatten_14 (Flatten)         (None, 200)               0
_________________________________________________________________
y_pred (Dense)               (None, 12)                2412
=================================================================
Total params: 4,220
Trainable params: 4,220
Non-trainable params: 0
```

f. Otimizações

Para o argumento de otimização foi utilizado RMSprop (parâmetros tabela 2), uma vez que esta tem uma taxa de aprendizado adaptativa, o que faz com que o modelo aprenda de forma mais eficiente.

Tabela 2 - Parâmetros do otimizador

Optimizer	RMSprop
Batch size	32
Épocas	150
Conjunto de treinamento	2.426
Conjunto de teste	486

g. Treinamento

Para medir a acurácia do modelo, foi analisado por épocas, após 100 épocas, tantos os dados de treinamento e de teste do modelo apresentaram uma estabilidade da acurácia, assim demonstrado na figura 5.

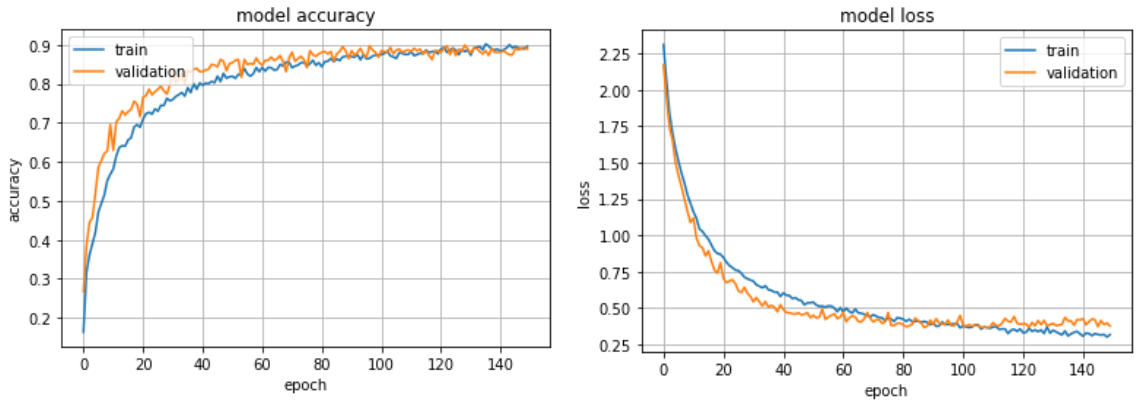
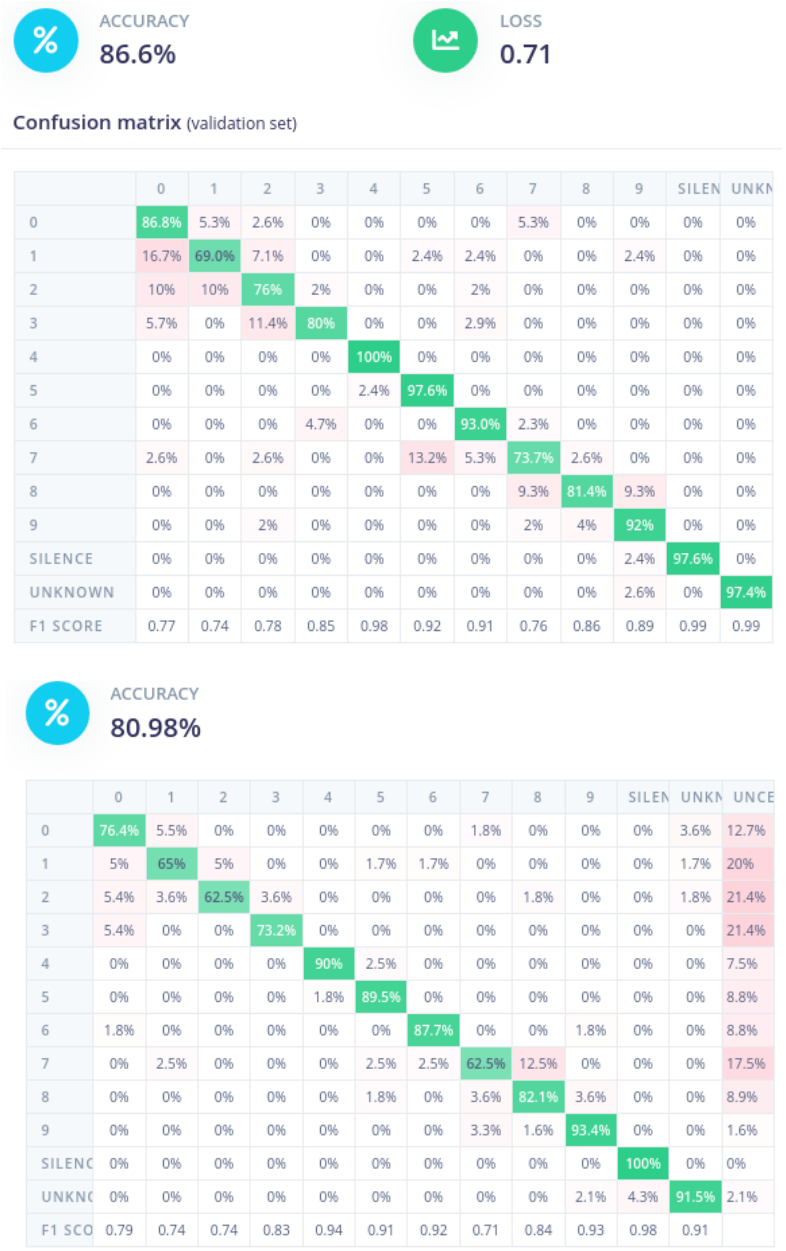


Figura 5 - Acurácia do modelo dos dados de treinamento (azul) e de validação (laranja) ao longo das épocas. A estabilidade da acurácia e uma baixa *loss* nos dados de treinamento e de validação representa que o sistema foi bem treinado indicando um bom desempenho do modelo desenhado e que não ocorreu *overfitting*.

h. Matriz de Confusão

A matriz de confusão, figura 6, foi possível avaliar o desempenho da classificação do modelo. Cada coluna da matriz de confusão corresponde a um rótulo previsto, em ordem (“0”, ao, “9”). Cada linha, de cima para baixo, corresponde ao rótulo real. Na nossa matriz de confusão, observa-se que a grande maioria das previsões concordam com os rótulos reais. Também podemos ver os lugares específicos na matriz dos dados de teste onde confusão está ocorrendo: houve uma significância sobre a classificação “incerto”, indicando os possíveis pontos fracos do modelo. Indicando que o ruído do ambiente no momento da inferência ou alterar o arduino de lugar, pode estar afetando o modelo que ele não consegue inferir corretamente a classe.



	0	1	2	3	4	5	6	7	8	9	SILEN	UNKN
0	86.8%	5.3%	2.6%	0%	0%	0%	0%	5.3%	0%	0%	0%	0%
1	16.7%	69.0%	7.1%	0%	0%	2.4%	2.4%	0%	0%	2.4%	0%	0%
2	10%	10%	76%	2%	0%	0%	2%	0%	0%	0%	0%	0%
3	5.7%	0%	11.4%	80%	0%	0%	2.9%	0%	0%	0%	0%	0%
4	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
5	0%	0%	0%	0%	2.4%	97.6%	0%	0%	0%	0%	0%	0%
6	0%	0%	0%	4.7%	0%	0%	93.0%	2.3%	0%	0%	0%	0%
7	2.6%	0%	2.6%	0%	0%	13.2%	5.3%	73.7%	2.6%	0%	0%	0%
8	0%	0%	0%	0%	0%	0%	0%	9.3%	81.4%	9.3%	0%	0%
9	0%	0%	2%	0%	0%	0%	0%	2%	4%	92%	0%	0%
SILENCE	0%	0%	0%	0%	0%	0%	0%	0%	0%	2.4%	97.6%	0%
UNKNOWN	0%	0%	0%	0%	0%	0%	0%	0%	0%	2.6%	0%	97.4%
F1 SCORE	0.77	0.74	0.78	0.85	0.98	0.92	0.91	0.76	0.86	0.89	0.99	0.99

	0	1	2	3	4	5	6	7	8	9	SILEN	UNKN	UNCE
0	76.4%	5.5%	0%	0%	0%	0%	0%	1.8%	0%	0%	0%	3.6%	12.7%
1	5%	65%	5%	0%	0%	1.7%	1.7%	0%	0%	0%	0%	1.7%	20%
2	5.4%	3.6%	62.5%	3.6%	0%	0%	0%	0%	1.8%	0%	0%	1.8%	21.4%
3	5.4%	0%	0%	73.2%	0%	0%	0%	0%	0%	0%	0%	0%	21.4%
4	0%	0%	0%	0%	90%	2.5%	0%	0%	0%	0%	0%	0%	7.5%
5	0%	0%	0%	0%	1.8%	89.5%	0%	0%	0%	0%	0%	0%	8.8%
6	1.8%	0%	0%	0%	0%	0%	87.7%	0%	0%	1.8%	0%	0%	8.8%
7	0%	2.5%	0%	0%	0%	2.5%	2.5%	62.5%	12.5%	0%	0%	0%	17.5%
8	0%	0%	0%	0%	0%	1.8%	0%	3.6%	82.1%	3.6%	0%	0%	8.9%
9	0%	0%	0%	0%	0%	0%	0%	3.3%	1.6%	93.4%	0%	0%	1.6%
SILENC	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
UNKNC	0%	0%	0%	0%	0%	0%	0%	0%	0%	2.1%	4.3%	91.5%	2.1%
F1 SCO	0.79	0.74	0.74	0.83	0.94	0.91	0.92	0.71	0.84	0.93	0.98	0.91	

Figura 6 - Matriz de Confusão dos dados de validação e dos dados de teste, respectivamente. Observa-se que a acurácia dos dados de teste preservou-se alta.

i. Modelo

A solução final composta da plataforma Arduino Nano 33 BLE Sense e o modelo treinado embarcado na plataforma foi usado para realizar inferências sobre novos sinais capturados. Reafirmando o objetivo, a ideia é expor a possibilidade de ataques que revelem dados sensíveis de usuários quando estes pressionam teclas. O sinal sonoro emitido sobre a tecla inevitavelmente produz características específicas, bastando apenas a utilização de unidades para captação de sinais sonoros e um microcontrolador executando uma rede neural artificial.

6. Perspectivas, desafios e obstáculos previstos e soluções potenciais

Os desafios e obstáculos identificados até o momento são descritos a seguir, juntamente com possíveis formas de mitigação ou evidência anteriores.

-Conjunto pouco representativo:

a. aumentar a coleta de dados

- Inexistência de padrões sonoros e vibratórios que possibilitem a discrimanação da tecla pressionada:

b. pesquisas anteriores sugerem que existe um padrão

-Treinamento para diferentes configurações (particularidades de teclados/aplicações)

c. análise detalhada da modelagem variando o posicionamento e fixação dos sensores para melhor caracterização e discriminação de diferentes estruturas (“funções de transferência”) .
d. particularização para grupos de utilização ou produtos: como tipos e configurações de teclados para aplicação em produtos de mercado.

Perspectivas:

Fusão de sensores: a integração com sinais inerciais como acelerômetros e giroscópios está no escopo da continuidade da pesquisa e propiciará um aumento significativo de sinais indicadores propícios para esse tipo de pesquisa. O desenvolvimento está em fase inicial e se mostra bastante promissor. A fusão dos sinais dos sensores poderá garantir ou melhorar a consistência, confiabilidade e acurácia na identificação de teclas e/ou de padrões de pressionamento. Vários trabalhos de pesquisa caminham nessa direção [],[].

Diversidade e possibilidades de novas aplicações: utilização semelhante poderão ser desenvolvidas para detectar interações, sequenciamento inadequado de carregamento, choques e/ou impactos em sistemas e/ou plataformas de transporte ou serviços como guindastes e assemelhados. Nesse sentido, sistemas de TinyML poderão evitar acidentes e/ou alertarem para manutenções preventivas.

A pesquisa faz parte de um grupo de pesquisa do CNPQ, sua continuidade será fundamental para o desenvolvimento de um trabalho de conclusão de curso envolvendo bioengenharia e também contará com a contribuição dos trabalhos de uma iniciação científica voluntária e, provavelmente, com mais um estudante através de uma bolsa de IC do programa PIBIFSP.

7. Maior questão não resolvida

Não foram identificadas questões que impossibilitem a implementação do projeto.

8. Referências

[1] Howtomechatronics, https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/

[2] https://www.arduino.cc/en/Guide/NANO33BLESense

[3] Youngmok Ha, Soo-Hee Jang, Kwang-Won Kim, Ji Won Yoon, “Side channel attack on digital door lock with vibration signal analysis: Longer password does not guarantee higher security level”, *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Daegu, Korea, 2017.

[4] Ilia Shumailov, Laurent Simon, Jeff Yan, Ross Anderson, "Hearing your touch: A new acoustic side channel on smartphones", preprint https://arxiv.org/abs/1903.11137 (https://www.cl.cam.ac.uk/~is410/Posters/hearing_touch_poster.pdf)

[5]https://www.hackster.io/news/building-an-rf-side-channel-attack-using-machine-learning-6544e691542e

[6] https://elie.net/blog/security/hacker-guide-to-deep-learning-side-channel-attacks-the-theory/

[7] https://www.tau.ac.il/~tromer/handsoff/

[8] Vijay Janapa Reddi *et al.*, "Widening Access to Applied Machine Learning with TinyML", preprint https://arxiv.org/abs/2106.04008]

[9] https://www.tau.ac.il/~tromer/handsoff/