

# Collection de jeu de données via la plateforme Google Tiny Motion Trainer

Réalisé par : Mehdi NAIT HAMMOU & Abdelkarim BEN BRAHIM (5A ASTRE 2022)

---

## Introduction :

Le but de ce document est de présenter un exemple d'utilisation de la plateforme Google Tiny Motion Trainer pour la collecte de données. Dans ce tutoriel, on propose de collecter des données provenant d'un capteur LSM9DS1 intégré dans une carte Arduino BLE 33 Sense, et qui mesure l'accélération linéaire et angulaire.

Google Tiny Motion Trainer est une application faisant partie de la collection Google Experiments, qui est une collection d'applications liées à l'IA, et développées par les équipes de Google ou par des développeurs tiers en utilisant des outils fournis par Google.

Cette plateforme permet de simplifier le processus de développement d'applications IA pour les microcontrôleurs via une application web. Cette application propose la simplification complète du processus de développement de bout en bout ; l'acquisition des données via une communication bluetooth avec la carte ciblée, l'entraînement du modèle et le packaging de l'application sous forme d'un firmware à flasher directement sur la carte ciblée.

Le présent document représente une prise en main de l'usage de cette plateforme pour le développement d'applications Tiny ML.

Notes :

- 1- Le présent développement peut être appliqué à d'autres cartes électroniques dotés d'un module de communication Bluetooth.
- 2- D'autres plateformes peuvent être utilisées pour le même but. Tels que Edge Impulse, qui propose une documentation plus riche, destinée aux utilisateurs avancés.

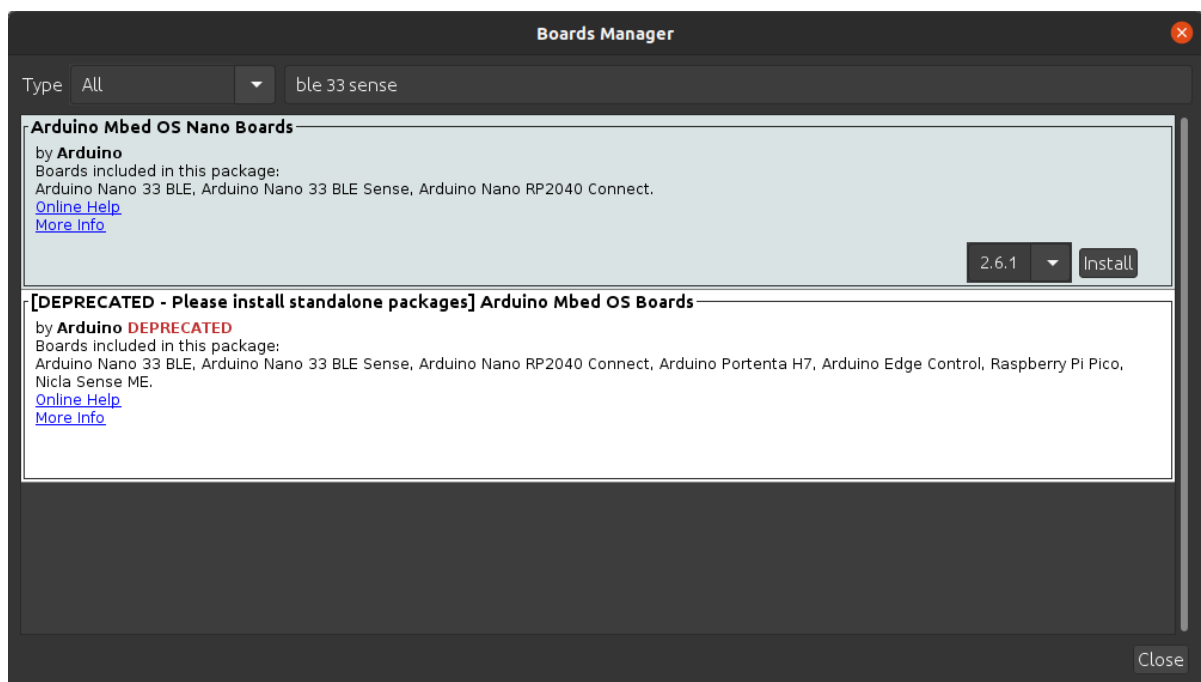
## Pré-requis :

- Une carte arduino BLE 33 Sense.
- Un ordinateur doté d'une interface Bluetooth.

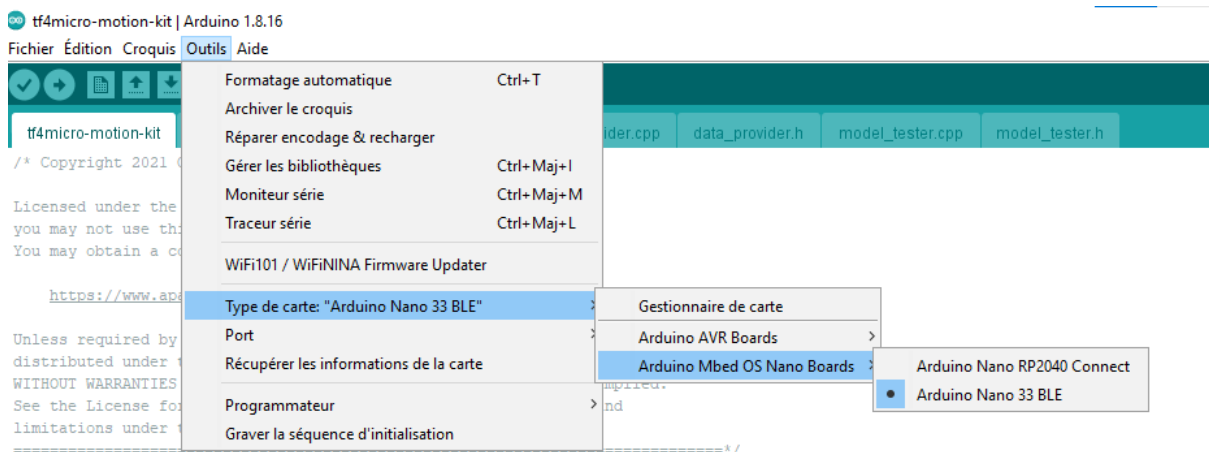
# Prise en main :

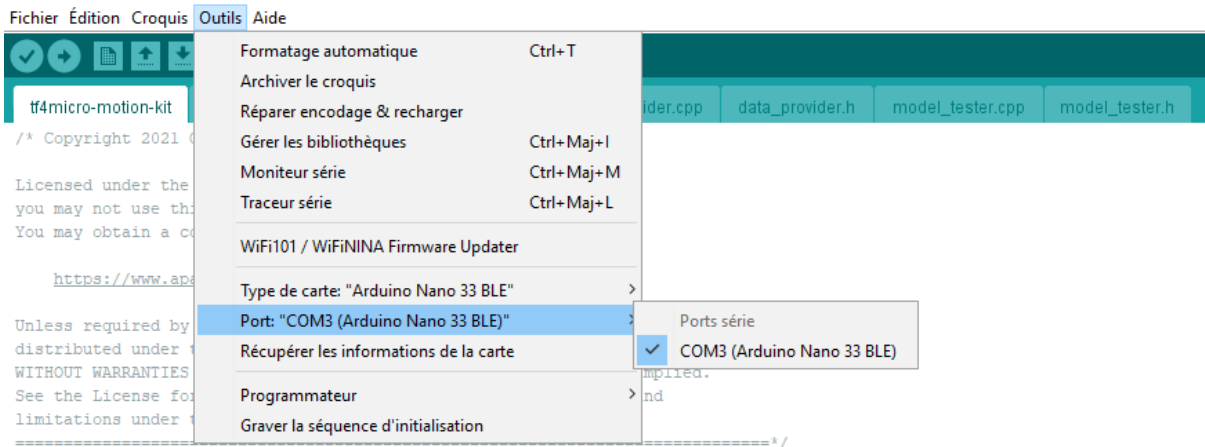
## 1 - Configuration de la carte Arduino :

La première étape est de connecter la carte arduino à l'ordinateur, et installer le package contenant les bibliothèques de la carte ciblée. Pour utiliser la BLE 33 Sense avec votre IDE Arduino, il faut télécharger le package Arduino MBED OS Nano Boards à l'aide du board manager comme illustré sur la figure suivante :



Une fois le package installé, l'étape suivante est de sélectionner la carte ainsi que le port sur lequel la carte est connectée. Les deux figures suivantes illustrent cette étape.



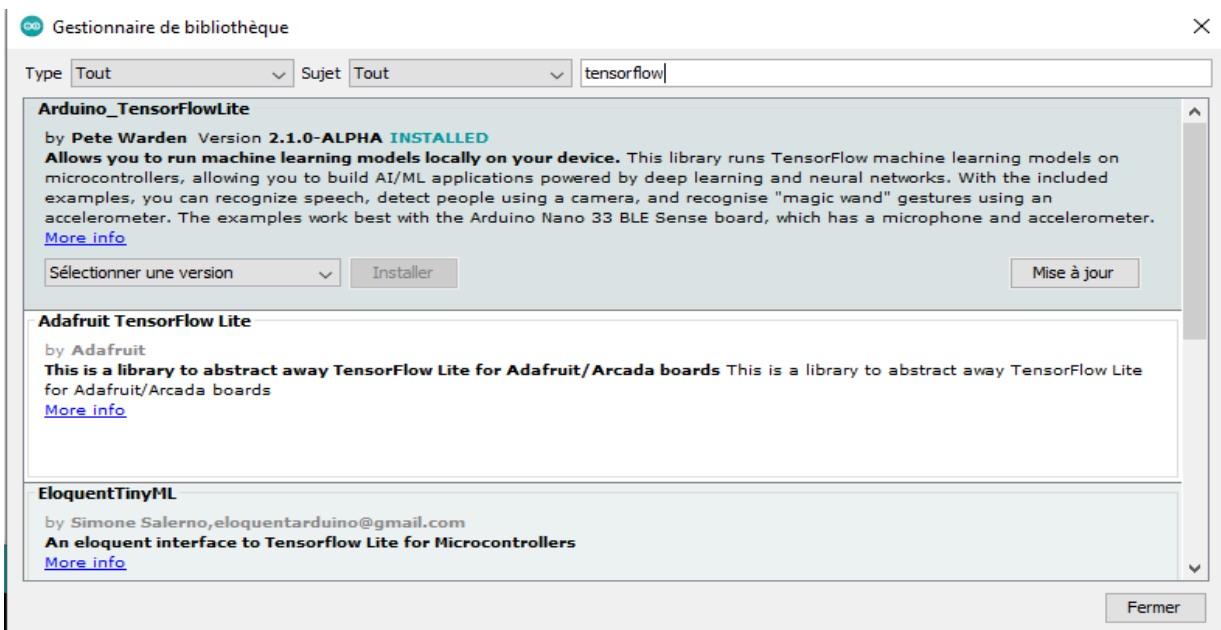


## 2 - Programmation de la carte pour la transmission bluetooth:

Cette étape requiert l'installation des bibliothèques suivantes :

- **ArduinoBLE** : Bibliothèque qui permet de gérer le module de communication Bluetooth à basse consommation (BLE).
- **Arduino\_LSM9DS1** : Bibliothèque de gestion du module LSM9DS1, qui représente notre source de données.
- **Arduino\_TensorFlowLite** : Il s'agit d'une bibliothèque précompilée du module Tensorflow Lite, qui permet de gérer l'inférence des modèles d'intelligence artificielle entraînés.

L'ensemble de ces bibliothèques est accessible sur le gestionnaire de bibliothèques de l'IDE Arduino présenté sur la figure suivante.



L'étape suivante consiste à utiliser (ou à coder) un script permettant la lecture des données capteurs et de les envoyer par bluetooth. Dans le présent document, nous allons utiliser un script fourni par le Google Creative Lab, qui permet d'assurer l'ensemble de ces fonctions. Le script est présent sur le repo Github du creative lab, accessible via ce [lien](#).

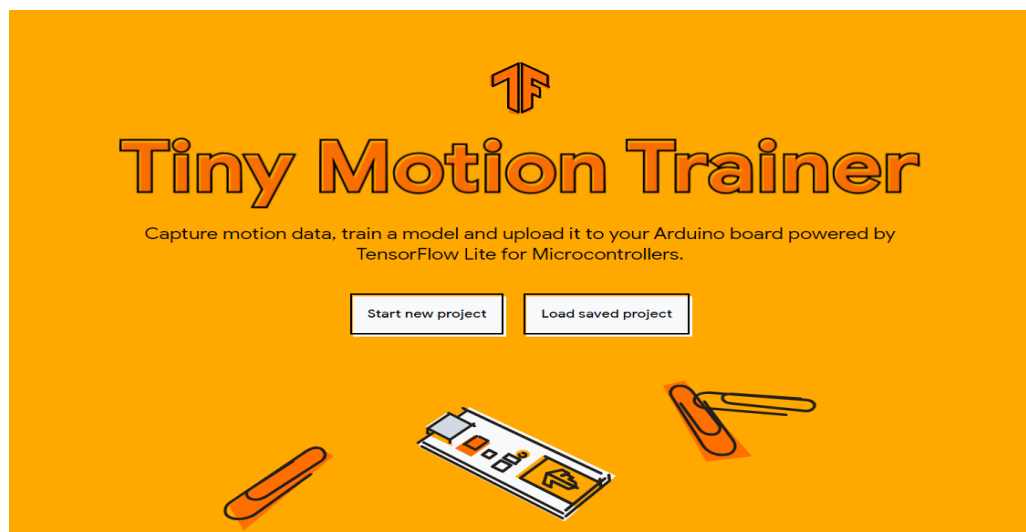
L'étape suivante consiste à compiler et téléverser le code sur la carte arduino. La compilation peut prendre quelques minutes, ce qui est normal le nombre de bibliothèques à charger.

Une fois la compilation terminée, on constate le clignotement de la LED alternativement en rouge, vert et bleu pour indiquer qu'elle est en attente de connexion comme le montre la figure suivante.

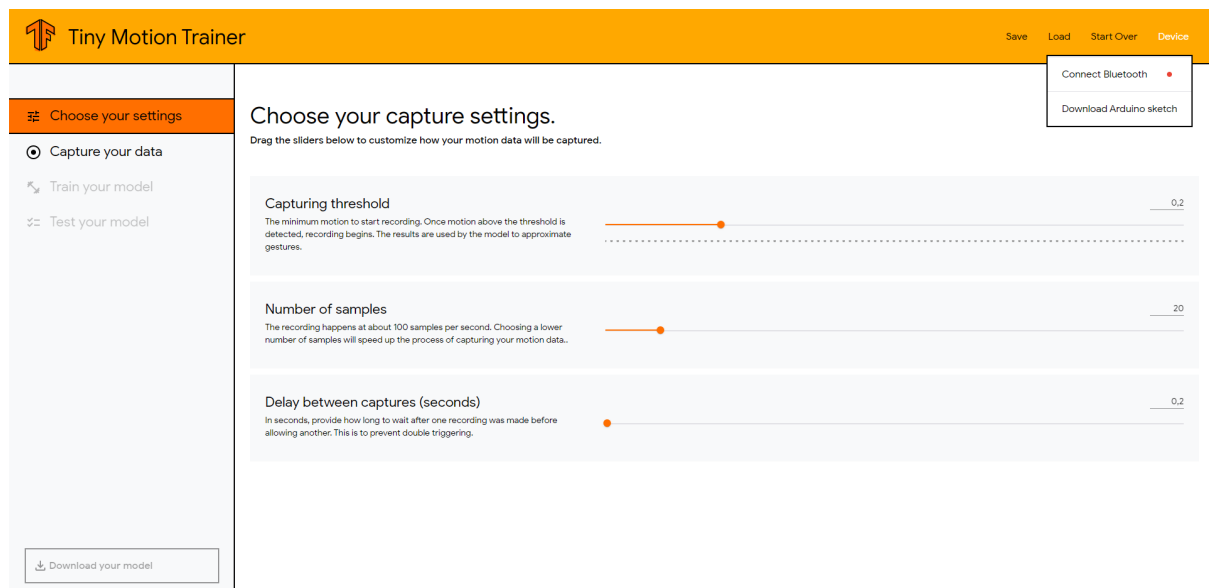


### 3 - Connexion sur la plateforme Tiny Motion Trainer:

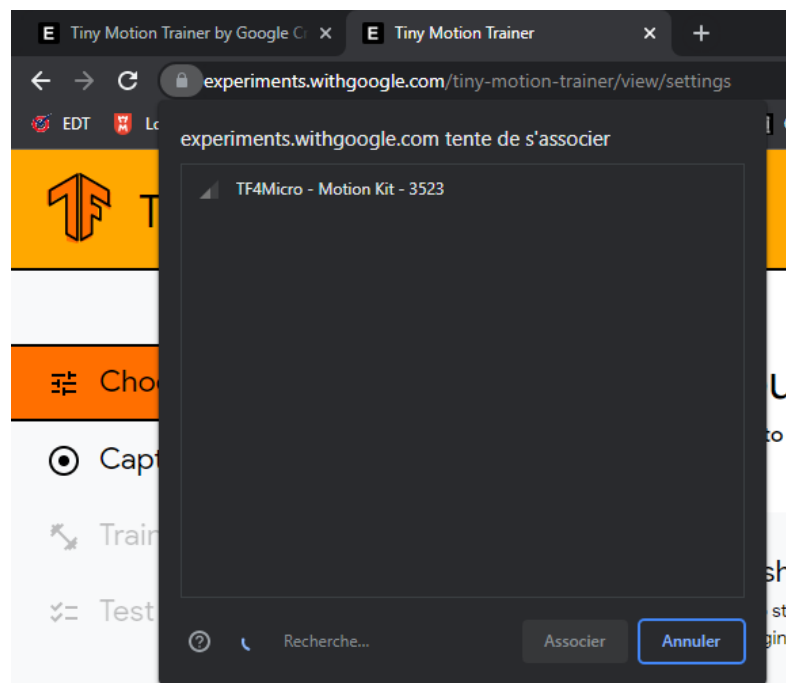
Une fois la carte arduino est prête, on passe à l'étape suivante, qui est la connecter à l'ordinateur à travers la plateforme web de [Tiny Motion Trainer](#). La figure suivante montre la page d'accueil de la plateforme.



On commence un nouveau projet, et à travers l'interface de configuration on peut connecter la carte via bluetooth à travers le menu Device.



Une fois cliqué, le bouton connect Bluetooth affiche l'interface des périphériques Bluetooth disponibles, on choisit le périphérique nommé "TF4Micro - Motion Kit".



Une fois la carte arduino connectée par bluetooth, la LED passe en bleu. Et la transmission commence.

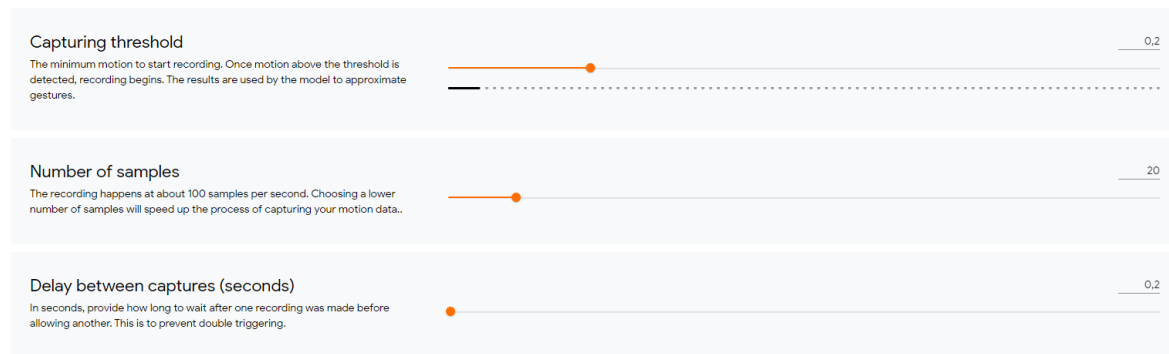
## 4 - Choix des paramètres de capture:

La plateforme nous propose 3 configurations à choisir pour nos signaux à capturer. Notamment :

- Capturing threshold : Représente le seuil au-delà duquel l'enregistrement est effectué. Il représente le module des trois composantes du signal d'accélération et de gyroscope.
- Number of samples : Représente le nombre d'échantillons par signal.
- Delay between captures (seconds) : Le temps d'attente entre deux captures.

Choose your capture settings.

Drag the sliders below to customize how your motion data will be captured.



The interface shows three sliders for customizing motion data capture. Each slider has a title, a description, and a numerical value on the right.

- Capturing threshold**: The minimum motion to start recording. Once motion above the threshold is detected, recording begins. The results are used by the model to approximate gestures. Value: 0.2
- Number of samples**: The recording happens at about 100 samples per second. Choosing a lower number of samples will speed up the process of capturing your motion data.. Value: 20
- Delay between captures (seconds)**: In seconds, provide how long to wait after one recording was made before allowing another. This is to prevent double triggering. Value: 0.2

Pour notre cas, il faut garder en tête la fréquence d'échantillonnage du signal capteur, qui est fixée pour le LSM9DS1 à 104 Hz, ce qui signifie que 20 échantillons correspond à une durée de signal de 0.2 secondes.

## 5 - Capture des données:

L'étape suivante consiste à créer les différents labels correspondant aux classes de mouvements à reconnaître.

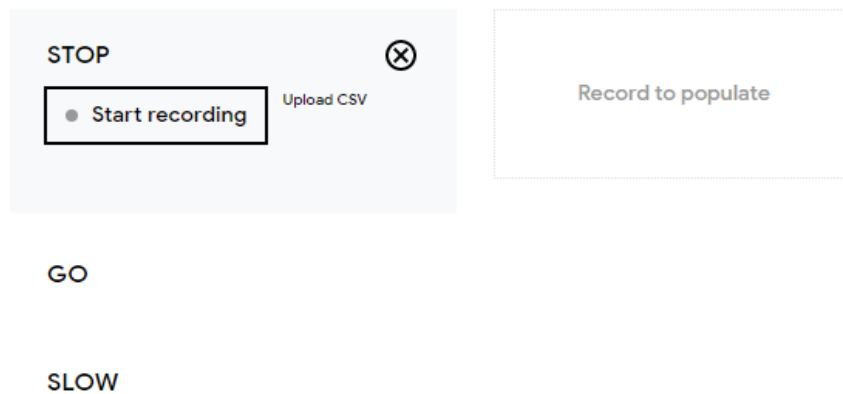
Capture your data

Create a new label to record data for each gesture. A minimum of two gestures and three samples of each gesture is required to train your model, though we recommend recording at least 20 samples for better results. Choosing a lower number of samples in settings will speed up this process

Add unique label

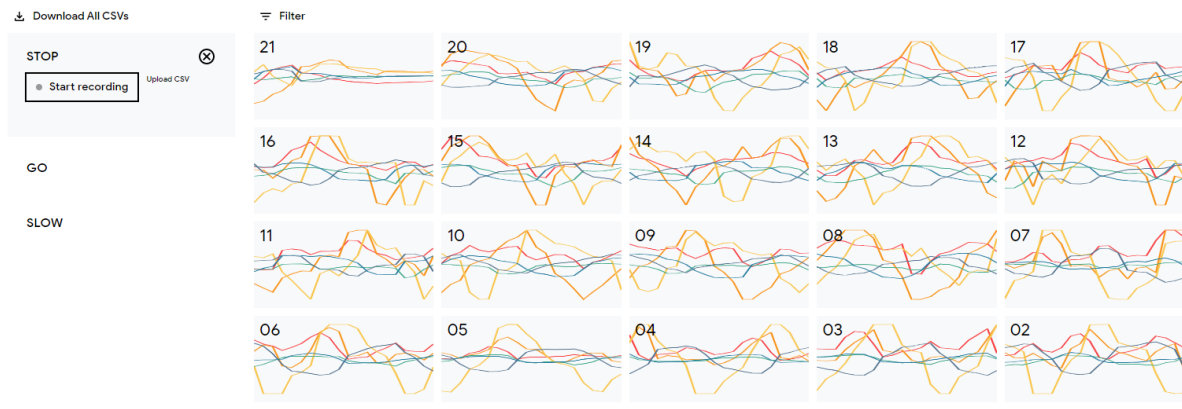
Create new

Une fois la création des labels terminés, on peut commencer l'enregistrement des mouvements.



Il est conseillé d'avoir plusieurs dizaines de mouvements par classe, enregistrés par plusieurs personnes tout en gardant en tête que les classes doivent contenir quasiment le même nombre de signaux. Ce qui va permettre d'éviter les problèmes de sur-apprentissage ou d'overfitting.

Une fois les données capturées, on peut exporter notre dataset en format .csv ou l'utiliser pour entraîner un réseau de neurones sur la plateforme, ce qui n'est pas vraiment optimal puisqu'on n'a pas vraiment d'idées sur l'architecture utilisée, ni les transformations qui sont appliquées sur la dataset. Il s'agit donc d'un modèle en boîte noire, donc difficile à déboguer et à maintenir.



## Conclusion :

Tiny Motion Trainer est une plateforme qui facilite énormément le processus de capture de données, mais elle reste limitée à un usage de découverte. Pour des utilisateurs expérimentés, il est conseillé d'utiliser Edge impulse qui offre des pipelines beaucoup plus riches et une expérience plus complète pour une multitude de cartes électroniques.