

Table des matières

1	Motivation.....	3
1.1	Un marché en pleine expansion	3
2	But du projet.....	4
2.1	Objectifs du projet	4
2.1.1	Prise en main de la littérature et des techniques de <i>machine learning</i>	4
2.1.2	Réalisation d'un prototype pour l'acquisition du débit d'un tuyau	4
2.1.3	Analyse et préparation des données mesurées.....	4
2.1.4	Etude du résultat, précision du système, améliorations possibles.	4
2.1.5	Comparer une méthode de résolution classique à une résolution par intelligence artificielle.	4
3	Planification initiale.....	5
4	Outils employés.....	6
4.1	Logicielles	6
4.2	Librairies / API pour Python.....	6
5	Tâche 1 – Prise en main des outils de Machine Learning.....	7
5.1	Motivation.....	7
5.1.1	La classification automatique	7
5.2	Enoncé du problème	7
5.3	Obtention des données	8
5.3.1	Application de mesure.....	9
5.3.2	Format des données	9
5.4	Exploration des données	10
5.4.1	Caractéristique des variables	10
5.4.2	Visualisation des données.....	12
5.5	Préparation des données	14
5.5.1	Est-il nécessaire de supprimer les doublons ?.....	14
5.5.2	Suppression des SSID « Outliers »	14
5.5.3	Gestion des données qualitatives.....	14
5.5.4	Filtre sur le RSSI minimum.....	15
5.6	Entraînement du modèle	17
5.6.1	Sélection du jeu d'entraînement et du jeu de test	17
5.6.2	Sélection des modèles prometteurs	17
5.7	Validation du modèle.....	18
5.7.1	Matrice de confusion	18
5.8	Conclusion de la tâche	19
6	Tâche 2 – Mesure du débit par les vibrations d'un tuyau	20

6.1	Motivation.....	20
6.2	La physique derrière ce projet.....	21
6.3	Bancs de test hydraulique.....	23
6.3.1	Petit banc de test.....	23
6.3.2	Grand banc de test.....	23
6.4	Prototypes de mesure.....	24
6.4.1	Premier prototype : « Piézoélectrique ».....	24
6.4.2	Deuxième prototype « <i>Version 1</i> ».....	26
6.4.3	Deuxième version du logiciel d'acquisition.....	35
6.5	Mesures.....	36
6.5.1	Script de visualisation.....	36
6.5.2	Mesures en laboratoire.....	36
6.5.3	Mesures sur le terrain.....	38
6.6	Quantification du débit à l'aide d'algorithmes de <i>machine learning</i>	42
6.6.1	Pourquoi le <i>machine learning</i> ?.....	42
6.6.2	Calcul de la performance d'un modèle.....	42
6.6.3	Méthodes sans réseau de neurones.....	43
6.6.4	Méthode avec réseau de neurones artificiels.....	45
6.6.5	Discussion des résultats.....	49
6.7	Améliorations possibles.....	50
6.8	Conclusion de la tâche.....	50
7	Bibliographie.....	51
7.1	Machine Learning avec Scikit-Learn.....	51
7.2	Deep Learning avec TensorFlow.....	51
7.3	Hands-On Data Science and Python Machine Learning.....	51
8	Remerciements.....	51
9	Conclusion personnelle.....	52
10	Table des figures.....	53

1 MOTIVATION

Ce projet a été proposé comme thèse de bachelor par le Dr. Pierre-André Mudry, professeur à la Haute Ecole d'ingénierie du Valais. Ayant toujours eu un attrait pour l'informatique et plus particulièrement les nouvelles technologies, c'est tout naturellement que ce projet a figuré comme mon premier choix pour ce travail.

Il s'agit là d'une belle occasion de mettre en pratique les connaissances acquises tout au long de mes études à la HEI et de les combiner avec quelque chose qui m'est personnellement nouveau : le *machine learning*.

De nos jours, le *machine learning* se retrouve dans bon nombre de nos applications¹ quotidiennes sans que nous nous rendions compte. Nous pouvons par exemple citer la détection de spam, la publicité ciblée sur internet, ou encore la reconnaissance vocale que nous trouvons sur chacun de nos smartphones.

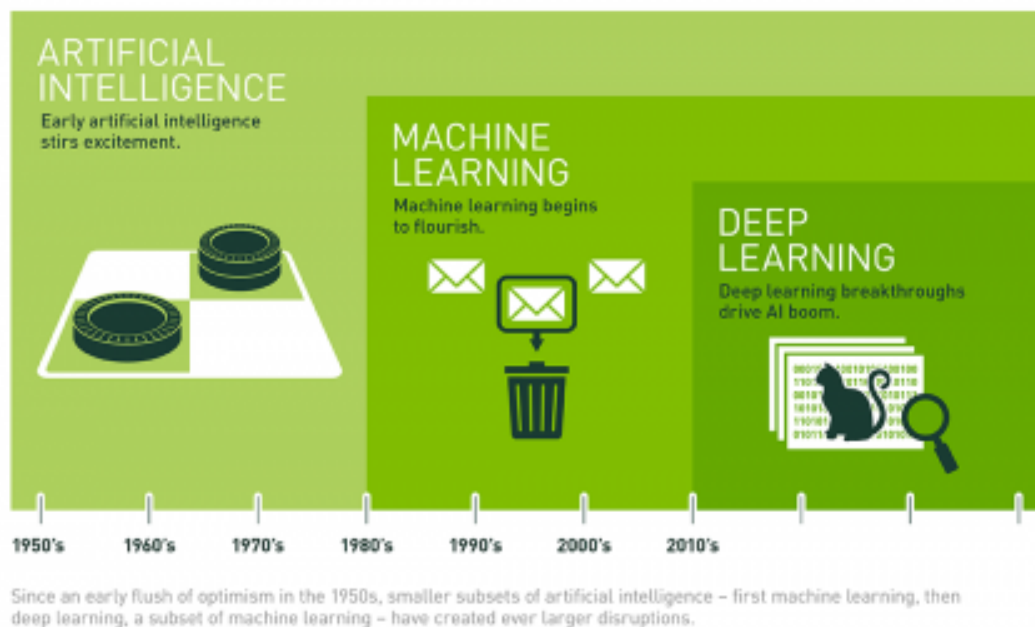


Figure 1 : Evolution du Machine Learning.

Source : <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>

1.1 UN MARCHÉ EN PLEINE EXPANSION

200 millions de dollars, c'est la valeur du marché de l'intelligence artificielle en 2015, d'après le cabinet d'analyse Tractica cette valeur devrait atteindre 11.1 milliards de dollars d'ici 2024. Bon nombre d'analystes prédisent un avenir radieux pour cette technologie qui compte plus de 1550 start-up actives et prédisent une augmentation de 40% de la productivité mondiale d'ici 2035 grâce à celle-ci².

¹ 10 cas d'usage pratiques du Machine Learning : <https://mrmint.fr/machine-learning-applications>

² Les 5 chiffres à absolument connaître sur l'IA : <https://experiences.microsoft.fr/business/intelligence-artificielle-ia-business/ia-chiffres-cles/>

2 BUT DU PROJET

Ce projet qui provient d'une collaboration entre la HEI et les services industriels de Bagnes, a pour but de réaliser un appareil à bas coût permettant de mesurer, avec une précision suffisante, le débit d'eau qui traverse une conduite. Dans le but de minimiser les coûts d'installation, le capteur doit être basé sur une méthode non-intrusive (l'appareil doit venir se poser autour du tuyau).

2.1 OBJECTIFS DU PROJET

2.1.1 Prise en main de la littérature et des techniques de *machine learning*

Cette prise en main est effectuée en se documentant sur le sujet et en réalisant un petit projet employant des outils d'apprentissage automatique. Ce projet a pour but de déterminer l'emplacement d'une personne dans le bâtiment de la HEI en fonction des réseaux Wi-Fi que son smartphone peut détecter

2.1.2 Réalisation d'un prototype pour l'acquisition du débit d'un tuyau

Pour réaliser ce prototype, la piste étudiée est de mesurer les vibrations induites sur un tuyau par un fluide qui le traverse. Pour mesurer ces vibrations, les méthodes suivantes sont étudiées : capteur piézoélectrique, microphone, accéléromètre.

2.1.3 Analyse et préparation des données mesurées

Cette étape est cruciale pour permettre à un modèle de *machine learning* de s'entraîner sur les différentes données mesurées. Dans cette étape, des points importants sont analysés tel que les corrélations entre les vibrations et le débit ainsi que la gestion des valeurs aberrantes.

2.1.4 Etude du résultat, précision du système, améliorations possibles.

Cette partie correspond principalement à la mise en place d'un modèle d'apprentissage automatique pour la détection de débit. Une fois le modèle généré, celui-ci est testé et permet de déterminer les différentes améliorations qui permettraient une meilleure précision de celui-ci.

2.1.5 Comparer une méthode de résolution classique à une résolution par intelligence artificielle.

Si le temps le permet, il serait intéressant de comparer une résolution classique avec une résolution par des outils de *machine learning*. Cette comparaison permettrait de donner un meilleur aperçu sur les limites de l'un ou de l'autre.

3 PLANIFICATION INITIALE

Du	16.05	21.05	28.05	04.06	11.06	18.06	25.06	02.07	09.07	30.07	06.08	13.08	20.08	27.08
Au	18.05	25.05	01.06	08.06	15.06	22.06	29.06	06.07	13.07	03.08	10.08	17.08	24.08	31.08
Numéro de semaine	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Prise en main des outils de ML														
Documentation sur le sujet														
Réalisation d'un premier exemple de ML														
Approfondissement des connaissances														
Réalisation d'un prototype pour l'acquisition du débit														
Etude du phénomène														
Choix des composants														
Montage du prototype														
Programmation du prototype														
Test du prototype														
Réalisation des mesures														
Analyse et préparation des mesures														
Etude des variables														
Visualisation des mesures														
Etude de la corrélation														
Identifier les transformations														
Mise en place d'un modèle d'IA														
Analyse de modèles prometteurs														
Choix du modèle														
Implémentation														
Affinage du réglage														
Analyse du résultat														
Précision du système														
Améliorations possibles														
Comparaison de l'IA avec une méthode classique														
Documentation														
Jours Fériés		Lu	Je							Ma		Ma		

4 OUTILS EMPLOYÉS

4.1 LOGICIELS

- Python 3.6.5 : Interpréteur pour les scripts Python.
- PyCharm Community 2018.1 : IDE pour le Python.
- Arduino Genuino 1.8.5 : IDE pour l'Arduino du prototype.
- Office 365 Pro Plus : Outils de bureautique pour la documentation.
- Visual Studio 2017 : IDE utilisé dans ce projet pour le C#.

4.2 LIBRAIRIES / API POUR PYTHON

- NumPy 1.14.2 : Extension pour l'utilisation de matrices.
- SciPy 1.0.1 : Extension à usage scientifique.
- Matplotlib 2.2.2 : Extension pour la création de graphiques.
- Pandas 0.22.0 : Extension pour la manipulation de données.
- Scikit-Learn 0.19.1 : Extension dédiée à l'apprentissage automatique.
- TensorFlow 1.8.0 : Outil d'apprentissage automatique.

5 TÂCHE 1 – PRISE EN MAIN DES OUTILS DE MACHINE LEARNING

5.1 MOTIVATION

Dans le but de prendre en main les outils d'apprentissage automatique, notamment la librairie « *Scikit-learn* ». Le premier objectif qui a été donné est de réaliser un exercice de classification automatique.

5.1.1 La classification automatique

La classification automatique est une branche du *machine learning* (*supervisé*) et plus spécialement des statistiques qui permet d'attribuer à un groupe de données une étiquette. Un exemple très répandu de classification est la détection de spam dans votre boîte mail (en fonction des données présentes dans le mail, l'algorithme peut attribuer l'étiquette « Spam » ou « Clean » à un mail).

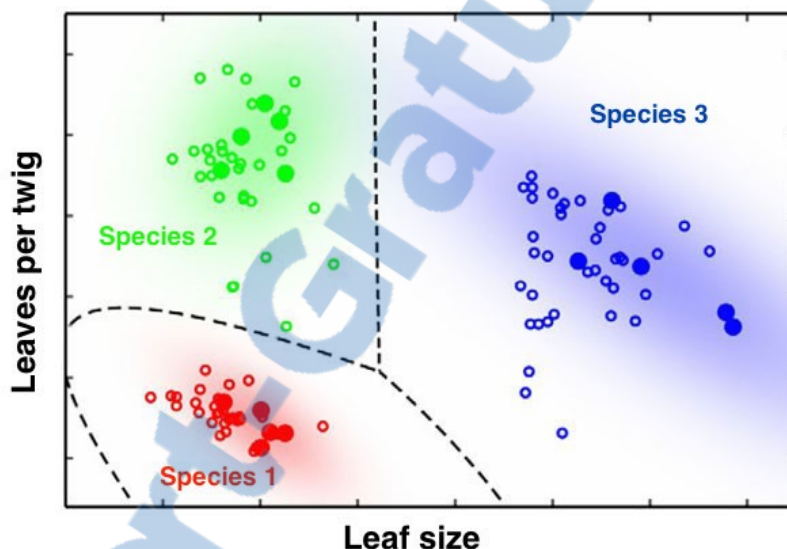


Figure 2 : Un exemple visuel de classification. Des arbres sont triés par deux paramètres, la taille de leurs feuilles et le nombre de feuilles par brindille. Une fois le modèle de classification entraîné, l'algorithme peut classer un nouveau jeu de donnée dans une des 3 espèces connues par le modèle. Source : <http://nanotechweb.org/cws/article/lab/46619>

5.2 ENONCÉ DU PROBLÈME

Sur le campus de la HEVs, plusieurs réseau TCP/IP sont disponibles. Afin de permettre un accès en Wi-Fi à ces différents réseaux dans tous les bâtiments, des répéteurs ou des points d'accès sont installés à plusieurs endroits dans ceux-ci. Voici quelques exemples des différents réseaux Wi-Fi qui peuvent être détecté sur le campus :

Tableau 1 : Echantillon des différents réseaux Wi-Fi disponible sur le campus. Source : <https://sinf.hevs.ch/Ressources/Wifi> :

Réseau / SSID	Utilité
Secure-hevs	Permet d'accéder au réseau de l'école et à internet.
Install-hevs	Destiné uniquement à l'installation initiale des laptops privés.
Eduroam	Permet aux personnes d'autres institutions (suisses et mondiales) de se connecter à internet sans restriction.

L'idée de cette prise en main est de réaliser une application de *machine learning* permettant à un utilisateur de savoir dans quelle salle il se trouve uniquement en scannant les différents Wi-Fi disponibles (adresse MAC, SSID, puissance du signal).

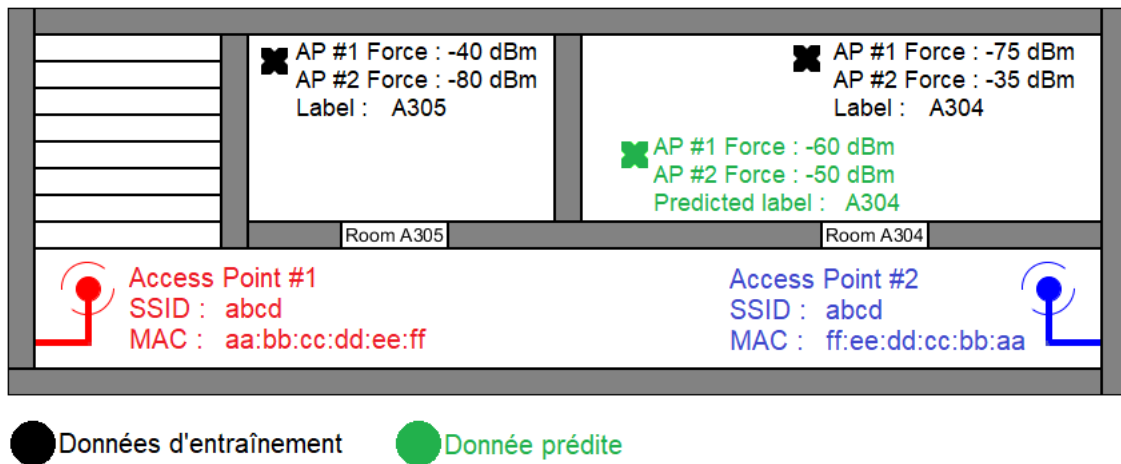


Figure 3 : Illustration du problème.

5.3 OBTENTION DES DONNÉES

Pour permettre au modèle de s'entraîner correctement, il est primordial de lui fournir des données qui font sens pour la prédiction désirée. Sur ce fait, les données suivantes ont été sélectionnées pour chaque réseau Wi-Fi détecté par l'appareil de mesure (un smartphone dans ce cas) :

- SSID : Nom commun du Wi-Fi.
- MAC : L'adresse MAC, Unique à l'interface de l'*Access Point*.
- Power : Puissance de réception du signal Wi-Fi, appelé RSSI.
- Classe : Le label de la classe, primordial pour l'entraînement du modèle.

5.3.1 Application de mesure

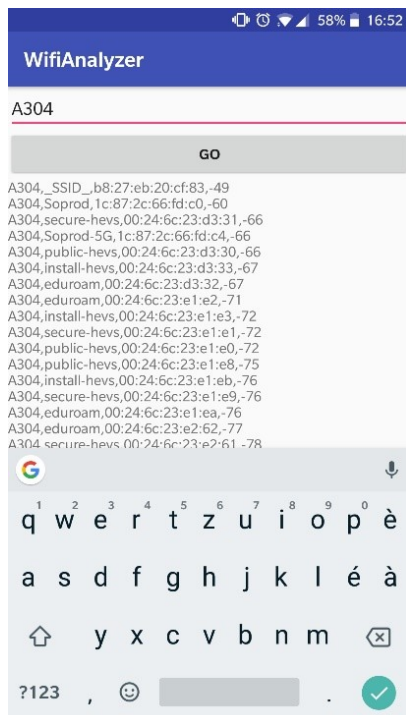


Figure 4 : Activité de l'application de mesure.

Dans le but d'automatiser au maximum la récupération des données, une application Android a été développée. L'application consiste en une seule activité qui, lorsque on clique sur le bouton « **GO** », va scanner les réseaux Wi-Fi disponibles, ajouter le label de la classe saisi précédemment et enregistrer le résultat dans un fichier sur la mémoire principale du téléphone.

L'architecture de cette application est fortement inspirée d'un tutoriel du site **Developpez.com** sur le Wi-Fi sous android.

Source :

<https://acesyde.developpez.com/tutoriels/android/wifi-android/>

5.3.2 Format des données

Pour permettre un passage facilité des données de l'application de mesure au script Python s'occupant de générer le classificateur de cette tâche, le format de fichier CSV (*Comma-separated values*) a été choisi. Ce format à l'avantage de nécessiter aucune librairie pour l'exportation et de plus il est facilement visualisable depuis Excel et son importation dans le script Python est très simple à l'aide de la librairie *Pandas*³.

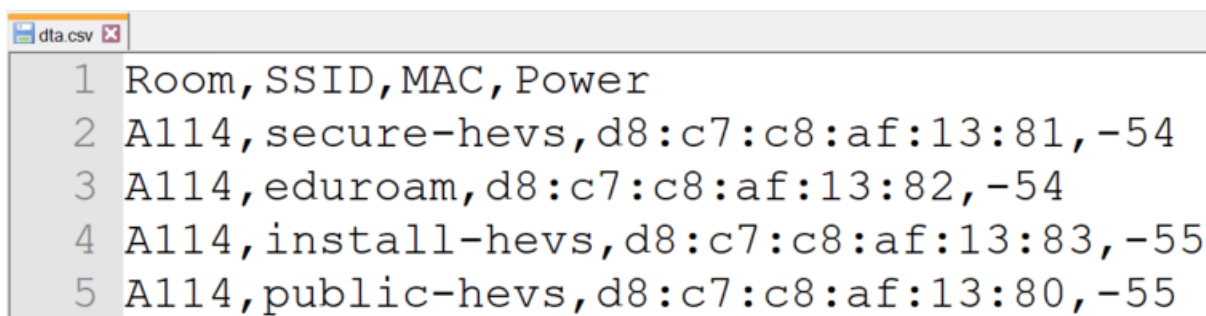


Figure 5 : Extrait du fichier CSV généré.

³ Plus d'informations sur la librairie *Pandas* : <https://pandas.pydata.org/>

5.4 EXPLORATION DES DONNÉES

L'application de mesure a été exécutée à plusieurs reprises dans les salles suivantes du bâtiment A:

- A112, A113, A114, A115, A115B, A201, A203, A204b, A304, A309.

Pour permettre de caractériser au mieux la réception des signaux Wi-Fi dans chacune de ses salles, les mesures ont été effectuées dans les 4 coins et le centre de chacune de celles-ci à différentes hauteurs.

Une fois toutes les mesures effectuées, voici les informations du jeu de données :

Tableau 2 : Statistique des valeurs du jeu de données :

Classe	Nombre de données
A112	425
A113	98
A114	769
A115	320
A115b	609
A201	149
A203	619
A204b	501
A304	373
A309	126
Total	3989

Nombre d'adresse MAC : 163
 RSSI Minimum : -92 [dBm]
 RSSI Maximum : -39 [dBm]
 RSSI Médian : -77 [dBm]

5.4.1 Caractéristique des variables

5.4.1.1 Variable « Room »

Cette variable correspond à la donnée que le modèle cherche à prédire. Elle est cependant importante dans le jeu de données pour entraîner et tester le modèle de classification.

5.4.1.2 Variable « SSID »

La variable SSID n'apporte que peu d'informations au modèle étant donné que à quelques exceptions près, les mêmes SSID se retrouvent dans toutes les salles. Elle apporte cependant une aide précieuse pour se représenter les adresses MAC ainsi que pour éliminer certaines valeurs aberrantes tel que des points d'accès non-désirable généré par un smartphone par exemple.

Si le SSID présente des valeurs aberrantes, cela indique que l'adresse MAC et le RSSI correspondant sont eux aussi aberrants.

5.4.1.3 Variable « MAC »

La variable MAC est primordiale du fait de sa définition (unique pour chaque interface d'un Access Point). Couplée avec le RSSI, ils vont donner les informations dont le modèle aura besoin.

5.4.1.4 Variable « Power »

Comme expliqué plus haut, la variable *Power* couplée avec la variable MAC est primordiale pour le modèle de la classification. En effet, cette mesure donne un indice sur la distance entre un *Access Point* et un utilisateur.

La puissance d'un signal Wi-Fi peut être mesurée à une précision de ± 0.5 [dBm], autrement dit, l'application de mesure donne uniquement des valeurs entières pour le RSSI.

En linéarisant le RSSI d'un signal Wi-Fi en un endroit donné en fonction de la distance qui sépare ce point de la source du signal (en s'intéressant uniquement à des distances qui pourraient caractériser une salle de classe, c-à-d de 0 à 15 mètres). Une atténuation d'au minimum 1.3 [dBm] par mètre peut être espérée (dans l'air, graphique de gauche). Ce qui représente une variation suffisante pour permettre une représentation sur plusieurs points des réseaux Wi-Fi d'une salle de classe.

Au contraire, le graphique de droite indique que sur les Access Point les plus distants d'une salle de classe une atténuation d'au minimum 0.17 [dBm] par mètre peut être espérée. Ce qui peut être un problème car une trop faible atténuation sur les longues distances risque d'ajouter du bruit rendant difficile la classification selon des salles de classes.

En se basant sur ces constatations il serait donc intéressant de filtrer les puissances inférieures à un certain seuil.

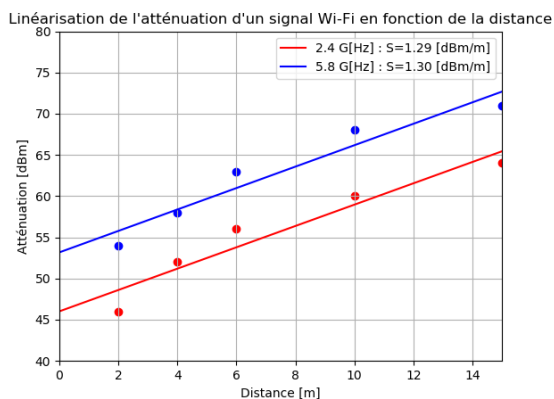


Figure 6 : Linéarisation dans une salle de classe.

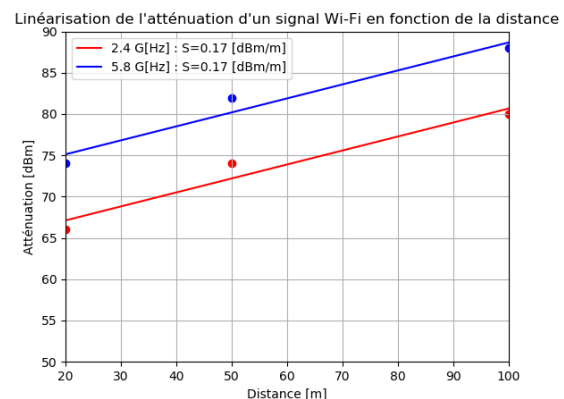


Figure 7 : Linéarisation sur des longues distances.

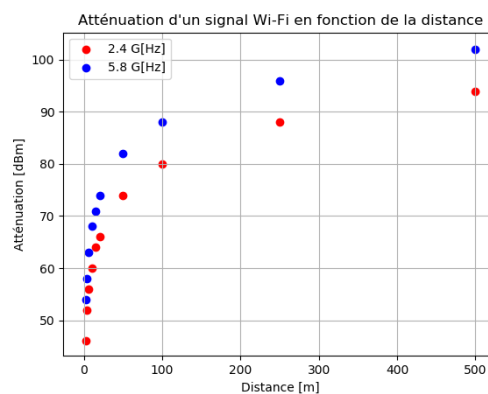


Figure 8 : Atténuation d'un signal Wi-Fi dans l'air.⁴

⁴ Données source selon : https://uniwan.be/project/calcul_attenuation_wifi/

5.4.1.5 Récapitulatif

Tableau 3 : Récapitulatif des différentes variables :

Nom	Type	Données aberrantes ?	Utilité
Room	Qualitative	Non	Variable à prédire
SSID	Qualitative	Oui	Moindre
Mac	Qualitative		Forte
Power	Quantitative		Forte

5.4.2 Visualisation des données

En visualisant la puissance de réception en fonction des différentes adresses MAC, sans les différentes couleurs des salles de classe sur le graphique ci-dessous, il serait pratiquement impossible de pouvoir déterminer avec suffisamment de précision celles-ci.

Il est cependant intéressant de noter pour la suite que pratiquement chaque salle de classe obtient un RSSI maximum avec une adresse MAC unique aux autres salles.

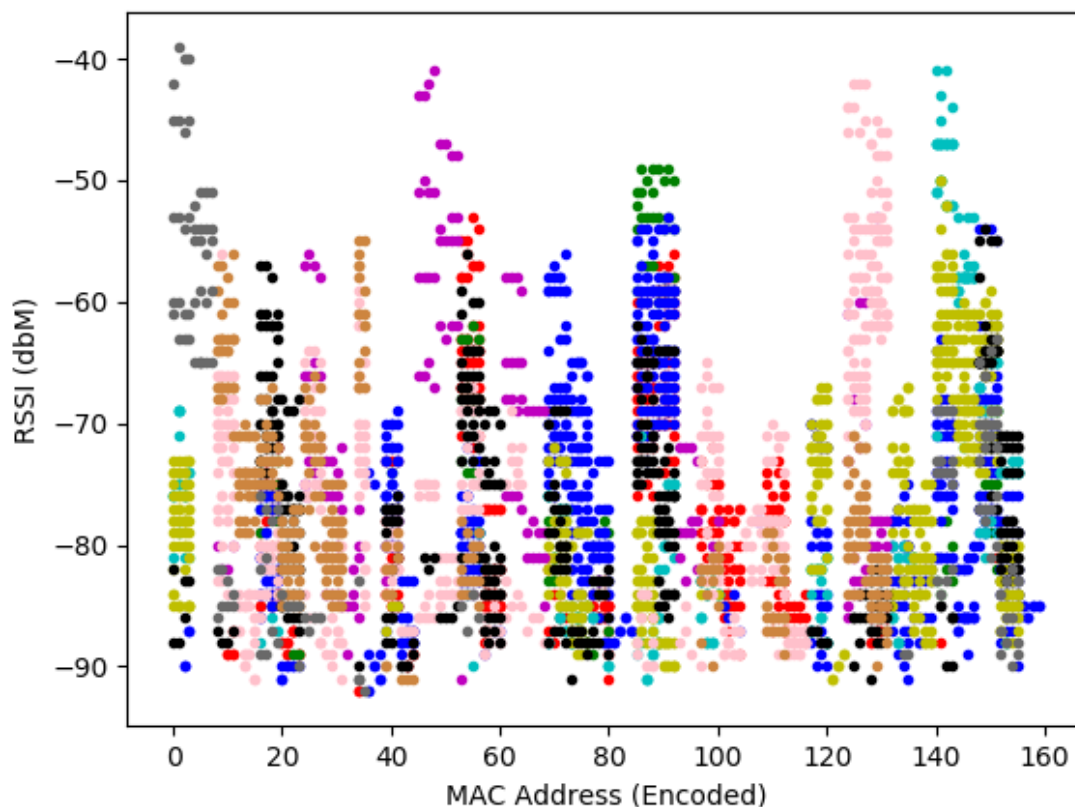


Figure 9 : RSSI en fonction des adresses MAC. Chaque couleur représente une salle de classe.

Un autre point intéressant est que l'hypothèse effectuée au point Variable « Power » 5.4.1.4 de ce rapport selon laquelle les *Access Point* à grande distance des salles de classes rajouteraient des valeurs aberrantes au modèle peut être vérifiée par le graphique ci-dessous :

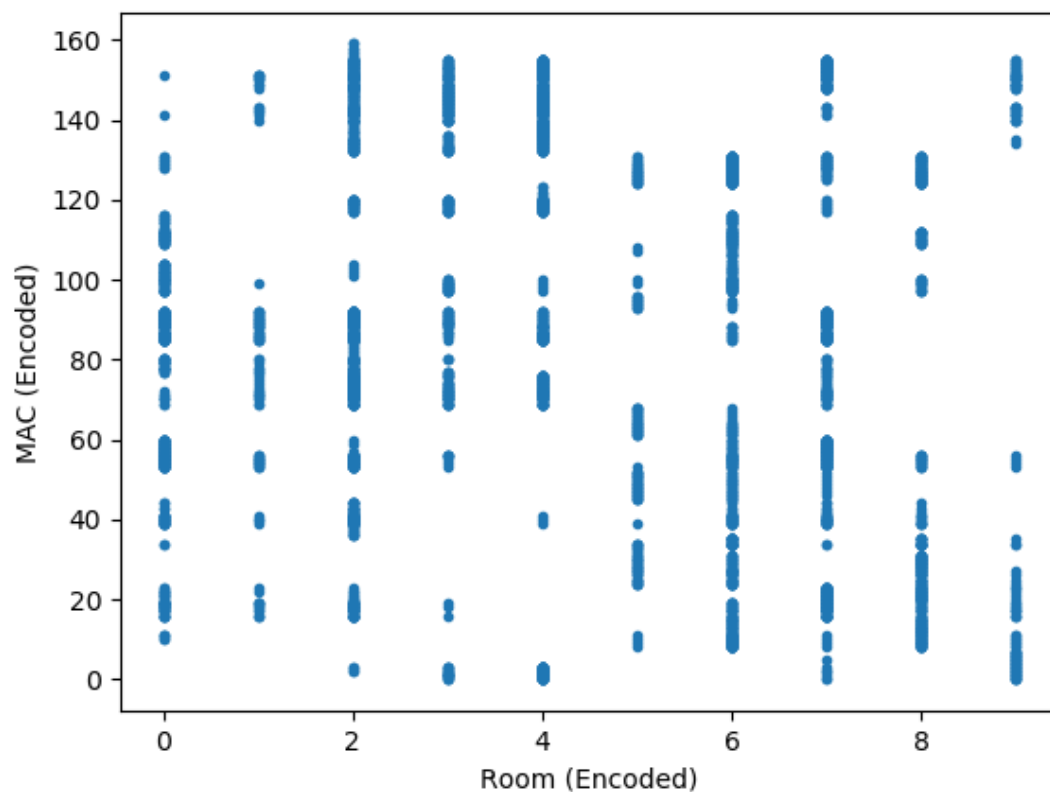


Figure 10 : Adresses MAC disponibles en fonction de la salle de classe.

En effet, il est assez étonnant de constater que dans chaque classe du bâtiment A, une grande partie des différentes adresses MAC peuvent être captée par l'appareil de mesure. Un filtre sur la puissance minimum de la réception sera donc très intéressant sur le jeu de données !

5.5 PRÉPARATION DES DONNÉES

5.5.1 Est-il nécessaire de supprimer les doublons ?

Comme spécifié au point 5.4.1.4, l'atténuation du RSSI d'un *Access Point* est moins significative lorsque l'appareil de mesure se trouve à une distance plus grande de l'origine du signal. De ce fait de nombreux doublons sont présents dans le jeu de mesures, en effet en raison de la précision de la mesure et de la puissance du signal il est courant de constater dans le jeu de mesure des doublons sur le triplet « *Mac, SSID, Power* » pour une salle de classe donnée.

Se pose alors la question de que faire avec ces doublons. Les supprimer ou les garder ?

Cependant les doublons donnent un poids à un élément. Par exemple, si on supprime les doublons, un trèfle à quatre feuilles devient autant signifiant qu'un trèfle à trois feuilles, alors que dans la vraie vie, il y a un trèfle à quatre feuilles pour 10'000 trèfles normaux.

En se basant sur cette expérience de pensée, les doublons ont été conservés dans le jeu de données.

5.5.2 Suppression des SSID « Outliers »

Comme expliqué précédemment, il est important de supprimer les données en provenance d'Access Point non significatif tel que les points d'accès mobile. Pour déterminer si un réseaux Wi-Fi provient d'un téléphone portable, le SSID va être analysé manuellement.

Nombre de SSID aberrant : 4

Nombre de SSID utile : 8

Nombre de données aberrantes supprimées : 68

Une fois que les SSID aberrants ont été supprimés, la colonne SSID est supprimée pour ne garder que les valeurs utiles au modèle.

5.5.3 Gestion des données qualitatives

Dans le but de permettre aux algorithmes de classification de travailler correctement, il est préférable de les faire travailler avec des nombres et non avec des chaînes de caractères. Etant donné que l'attribut « SSID » a été supprimé, il reste deux attributs de type texte à traiter dans le modèle :

- L'adresse MAC.
- La salle de classe.

Afin de mener à bien cette opération, deux algorithmes sont possibles :

- L'encodage de type « Label » qui consiste à donner un identifiant unique pour chaque valeur possible de l'attribut. Le problème de cet algorithme est que les algorithmes de *machine learning* vont considérer que deux valeurs voisines sont plus semblables que deux valeurs éloignées.
- L'encodage de type « One Hot » qui consiste à transformer toutes les valeurs uniques d'un attribut en attribut et lui donner la valeur 1 ou 0 en fonction de s'il caractérise l'élément ou non. Le problème de cet algorithme c'est qu'en fonction du nombre de valeurs uniques d'un attribut, le jeu de données est considérablement alourdi.

Tableau 4 : Exemple de l'encodage "Label" et de l'encodage "One Hot" :

Brute

Room	RSSI
A304	-60
A203	-55
A115b	-67

Encodage Label

Room	RSSI
0	-60
1	-55
2	-67

Encodage One Hot

A304	A203	A115b	RSSI
1	0	0	-60
0	1	0	-55
0	0	1	-67

Concernant les attributs qualitatifs du jeu de donnée un encodage « Label » a été préféré :

- Pour les salles de classes, car il est utile dans ce cas d'avoir un indice de proximité, car en effet, la salle A115 est semblable à la salle A115b et plus différente de la salle A304.
- Pour les adresses MAC, étant donné que 160 adresses MAC différentes sont présentes dans le jeu de données, un encodage « One Hot » alourdirait trop le modèle de classification.

5.5.4 Filtre sur le RSSI minimum

Comme expliqué au point 5.4.1.4, la faible variation du RSSI sur des grandes distances va ajouter du bruit sur le jeu de données, il est donc nécessaire de filtrer les valeurs qui sont qualifiées d'aberrantes. Le filtre a été effectué sur la valeur médiane du RSSI soit -77 [dBm].

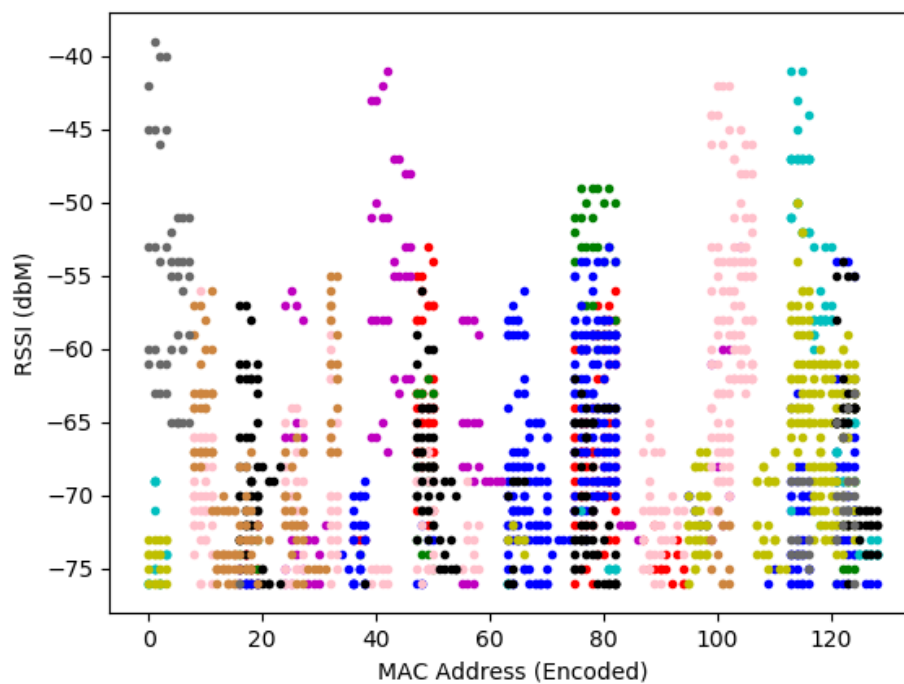


Figure 11 : RSSI filtré en fonction des adresses MAC.

Il est intéressant de constater que les différentes salles de classe peuvent maintenant être visualisée sur ce graphique bien mieux que sur le graphique non filtré.

5.6 ENTRAÎNEMENT DU MODÈLE

5.6.1 Sélection du jeu d'entraînement et du jeu de test

Pour juger de la qualité du modèle de classification, deux jeux de données vont être créé :

- Le jeu d'entraînement : comme son nom l'indique, celui-ci va servir à entraîner le modèle. Il se compose de 80% du jeu de données sélectionné aléatoirement.
- Le jeu de test : Une fois le modèle entraîné, ce jeu de données permettra la validation du modèle. Il se compose des 20% restant du jeu de données.

5.6.2 Sélection des modèles prometteurs

Les algorithmes de classification⁵ suivant ont été sélectionnés pour être étudiés plus en profondeur :

- **Linear Discriminant Analysis (LDA)**
- **K Nearest Neighbors (KNN)**
- **Decision Tree Classifier (CART)**
- **Random Forest Classifier (FOREST)**
- **Stochastic Gradient Descent (SGD)**

Ces différents algorithmes ont été entraînés sur le jeu d'entraînement, pour permettre de les classer par leur précision.

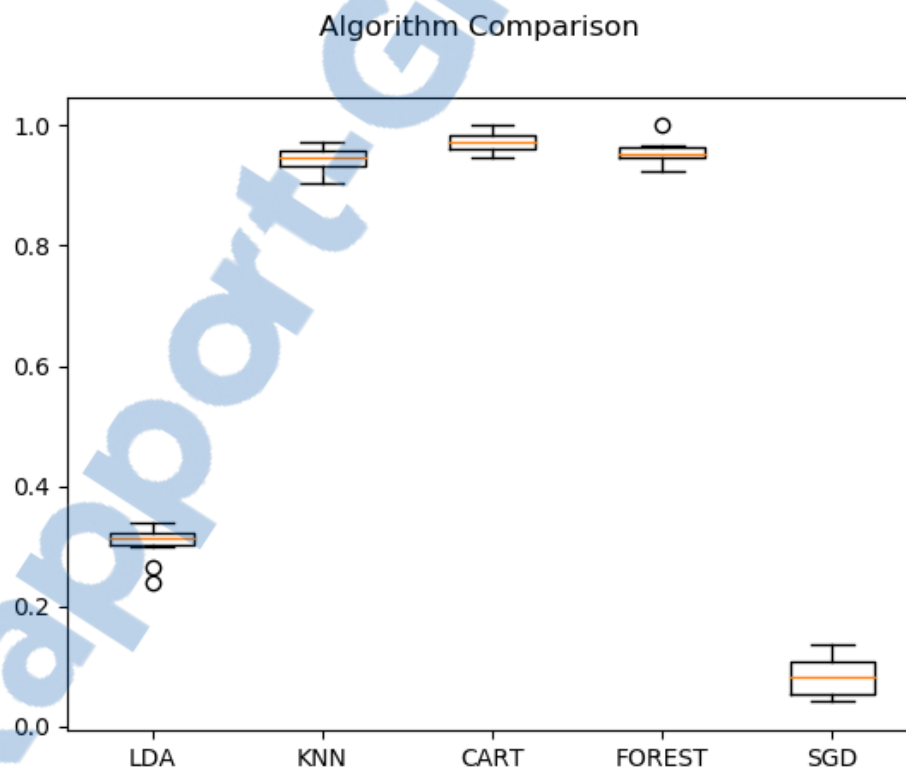


Figure 12 : Comparaison de la précision des différents algorithmes de classification.

Ce test permet d'établir une liste restreinte de modèles prometteurs qui vont également être testés sur le jeu de test : *KNN*, *CART*, *FOREST*.

⁵ Des détails sur les différents algorithmes de classification sont disponibles sur : https://en.wikipedia.org/wiki/Category:Classification_algorithms

5.7 VALIDATION DU MODÈLE

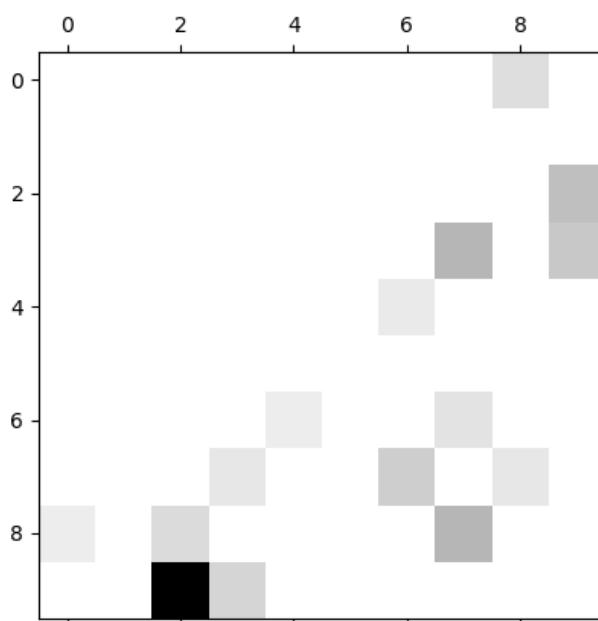
Les trois modèles prometteurs sélectionnés précédemment ont été exécutés sur le jeu de test et donnent la précision suivante :

	KNN	CART	FOREST
Entraînement	81.60%	96.80%	93.50%
Test	55.10%	85.40%	77.80%

La classification la plus performante est la classification par arbre de décision qui donne une meilleure précision sur le jeu de test. A noter que le hasard donnerait une précision de 10% (1 sur le nombre total de salles de classes).

5.7.1 Matrice de confusion

Il est intéressant de constater où les erreurs sont faites par le modèle. Pour cela la matrice de confusion ci-dessous indique les erreurs du modèle (plus une case est sombre, plus le nombre de confusion est nombreux) :



Classe	Index
A112	0
A113	1
A114	2
A115	3
A115b	4
A201	5
A203	6
A204b	7
A304	8
A309	9

Figure 13 : Matrice de confusion.

Il est étrange de constater que la plus grande confusion se trouve entre la salle A309 (9) et la salle A114 (2). Pour expliquer cette confusion, la seule hypothèse est que ces deux salles ne captent aucun Wi-Fi caractéristique à une certaine zone (comme le Wi-Fi propre à certaines classes). Cette hypothèse a pu être confirmée en analysant le jeu de donnée.

5.8 CONCLUSION DE LA TÂCHE

Ainsi se termine l'exercice de prise en main des outils de Machine Learning. Dans le but d'améliorer la précision du modèle, il serait intéressant de disposer d'un outil plus précis pour mesurer le RSSI. En effet, un appareil de mesure impliquerait une meilleure visualisation de l'évolution de celui-ci ce qui impliquerait directement une précision bien meilleure pour le modèle.

Pour améliorer l'utilité du projet, celui-ci pourrait être transformé en outil de géolocalisation dans un bâtiment. Mais cette transformation impliquerait un changement de domaine dans le Machine Learning car il s'agirait là d'une régression et non d'une classification.

Cependant cet exercice était fort intéressant pour assimiler les différents termes du Machine Learning pour être préparé au mieux pour la suite du projet.

6 TÂCHE 2 – MESURE DU DÉBIT PAR LES VIBRATIONS D'UN TUYAU

6.1 MOTIVATION

La commune de Bagnes par sa superficie (3^{ème} plus grande de Suisse) dispose d'un réseau d'irrigation important (environ 3'000 cannes d'arrosage), leur récent désir d'avoir une instrumentation de ce réseau pose les bases de ce travail de Bachelor.

Installer un débitmètre classique sur chacune des cannes serait trop couteux en appareil (un débitmètre standard coutant dans les 3'000CHF) et en manœuvre. Le but est donc de développer un débitmètre à bas coût en se basant sur les vibrations induites par un fluide sur un tuyau. Le débitmètre doit pouvoir détecter un débit allant jusqu'à 400 [l/min], débit maximum autorisé par la commune aux utilisateurs du réseau.



Figure 14 : Canne d'arrosage dans la nature.

6.2 LA PHYSIQUE⁶ DERRIÈRE CE PROJET

La vitesse d'un fluide dans un tuyau peut être décrite comme la somme entre la vitesse moyenne ($\bar{u}$) et une perturbation de la vitesse (u'). En partant du principe que la vitesse verticale moyenne d'un fluide dans un tuyau est égale à zéro, il en résulte :

$$\vec{V} = \begin{pmatrix} \bar{u} + u' \\ \bar{v} + v' \end{pmatrix} = \begin{pmatrix} \bar{u} + u' \\ 0 + v' \end{pmatrix}$$

Lorsqu'une molécule d'eau approche la paroi d'un tuyau, celle-ci présente de l'énergie cinétique (juste avant le choc). Lors du choc une partie de cette énergie sera dispersée en chaleur ainsi qu'en vibration alors que le reste de cette énergie sera employer pour « rebondir ».

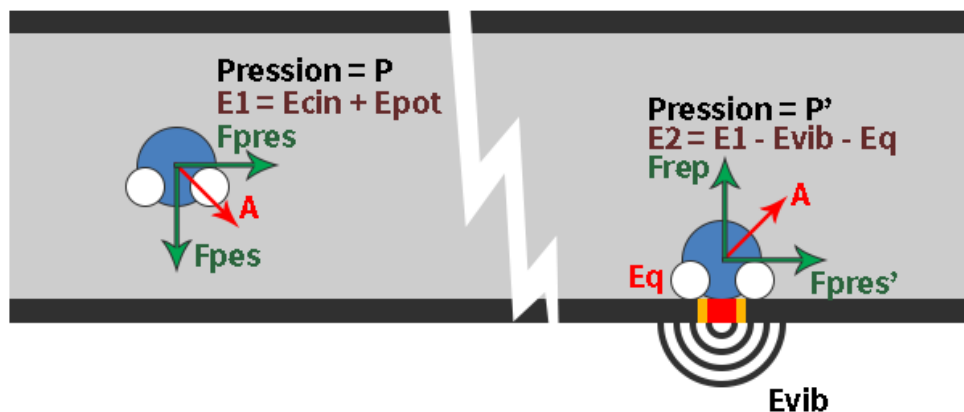


Figure 15 : Schéma de principe.

La vibration produite par fluide qui s'écoule sur un tube est reliée à la fluctuation de pression à l'intérieur du tube par l'équation suivante⁶ :

$$\frac{\partial^2 y}{\partial t^2} = - \frac{g}{A\gamma} EI \frac{\partial^4 y}{\partial x^4}$$

$$\text{avec } \frac{dp}{dx} = EI \frac{\partial^4 y}{\partial x^4}$$

Où :

- A est la section du tuyau.
- γ est le poids volumique du tuyau.
- g est l'accélération terrestre.
- EI est la rigidité à la flexion.
- p est la pression à l'intérieur du tuyau.
- y est l'axe radial du tuyau et x l'axe axiale.

⁶ R.P. Evans, J.D. Blotter, A.G. Stephens. Flow Rate Measurements Using Flow-Induced Pipe Vibration, 2004, Journal of Fluids Engineering p. 280-281

Il reste encore à expliquer la relation entre les vibrations d'un tube et la valeur moyenne du débit qui le traverse.

Pour ce faire, une nouvelle relation doit être ajoutée. Il s'agit de l'intensité de turbulence :

$$\frac{\sqrt{\overline{m}}}{\bar{u}} = \text{Intensité de turbulence, avec } m = u'^2$$

u' correspond à la perturbation de la vitesse axiale du fluide alors que $\bar{u}$ représente la vitesse axiale moyenne du fluide.

Depuis la définition de la vitesse d'un fluide expliquée au début de ce chapitre, l'intensité de turbulence peut être présentée de la manière suivante :

$$\frac{\sqrt{\overline{m}}}{\bar{u}} = \frac{\bar{m}}{\bar{u}^2} = \frac{\frac{1}{N} \sum_{i=1}^N [u_i(t) - \bar{u}]^2}{\bar{u}^2} = \frac{g}{A\gamma} = C$$

En multipliant les deux côtés par N , le nombre de points échantillonnés, et $\bar{u}^2$ et en divisant le tout par $N - 1$ on obtient :

$$\frac{1}{N-1} \sum_{i=1}^N [u_i(t) - \bar{u}]^2 = \frac{NC}{N-1} \bar{u}^2$$

Avec la partie de gauche qui correspond à la définition de l'écart-type au carré (variance).

Ces équations permettent de montrer qu'une fluctuation du débit est proportionnelle à une fluctuation de pression et que celle-ci est proportionnelle aux vibrations du tuyau. Il en résulte que l'écart-type des vibrations d'un tuyau est proportionnel à la moyenne du débit d'eau qui le traverse.

Cette tentative d'explication du phénomène physique est un résumé condensé d'un article universitaire sur le sujet dont les informations sur celui-ci peuvent être trouvées sur la note en bas de page numéro 6.

Cela permet tout de même de définir l'information principale que le modèle aura besoin pour être précis (l'écart-type) de plus l'offset, le spectre et l'amplitude du signal seront également analysés par la suite.

Bien que les vibrations d'un tuyau soient influencées par le fluide qui le traverse, il ne faut pas oublier que l'amplitude de ces vibrations va être fortement influencée par le diamètre intérieur et la matière du tuyau. Par chance, après une discussion avec les services industriels de Bagnes, il s'est avéré que l'entier des cannes d'arrosages présentes sur leur commune ont un diamètre de deux pouces et qu'elles sont toutes en acier galvanisé.

6.3 BANCS DE TEST HYDRAULIQUE

6.3.1 Petit banc de test

Dans le but de tester les différents prototypes de manière facilitée, un petit banc de test a été mis en place dans le laboratoire d'hydraulique.

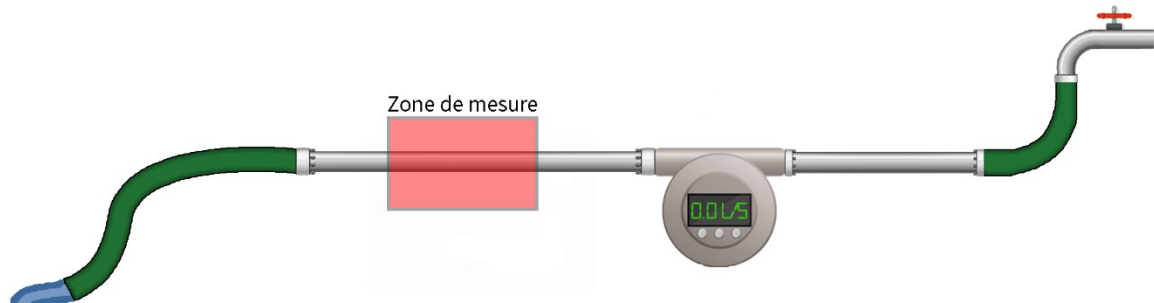


Figure 16 : Schéma du banc de test.

Ce banc de test se présente sous la forme d'un long tube en inox (1 pouce de diamètre) où au milieu se trouve un débitmètre électromagnétique qui présente l'avantage de ne perturber ni l'installation, ni l'écoulement de l'eau.

Aux deux extrémités du tube, un tuyau d'arrosage est ajouté pour atténuer les perturbations qui pourraient être engendrées par la source ou par la sortie du robinet.

La mesure s'effectue à un endroit suffisamment éloigné de l'entrée et de la sortie d'eau pour pouvoir négliger toutes perturbations engendrées par celles-ci (zone en rouge ci-dessus).

Ce banc de test, bien que facile de mise en place et d'utilisation a le désavantage de fournir uniquement un petit débit de 0 à 42 [l/min], ce qui est bien éloigné du débit réel attendu sur le terrain.

6.3.2 Grand banc de test



Figure 17 : Schéma du grand banc de test.

Le laboratoire d'hydraulique de la HES-SO Valais-Wallis dispose d'un banc de test hydraulique⁷ de plus grande ampleur qui est principalement utilisé pour l'évaluation des performances de turbines ou de pompes.

Il est composé d'un circuit en boucle fermée alimenté par trois pompes qui peuvent fournir un débit maximal de 100 m³/h.

Pour ce projet, il est possible de venir brancher une canne d'arrosage sur le circuit pour pouvoir y effectuer les différentes mesures.

Bien que ce banc de test puisse fournir un débit suffisant, le pompage de l'eau fournit une perturbation vibratoire importante sur l'installation qui peut compromettre l'emploi du prototype de mesure.

⁷ De plus amples informations sur le banc de test : <https://www.hevs.ch/fr/mini-sites/projets-produits/hydroelectricite/pages-minisites/banc-d-essai-hydraulique-921>

6.4 PROTOTYPES DE MESURE

Pour pouvoir étudier le phénomène, plusieurs versions d'un prototype de mesure ont été réalisées. Le principal objectif de ces prototypes est de donner une mesure relevante de la vibration d'un tuyau. En effet, la série de mesure d'un prototype doit donner de manière précise l'état vibratoire d'un tuyau pour un débit saisi par l'utilisateur.

Une application d'acquisition doit également s'occuper de communiquer avec l'électronique et de générer les fichiers de sorties.

6.4.1 Premier prototype : « Piézoélectrique »

Comme première étude du phénomène, une série de mesures a été effectuée sur le petit banc de test à l'aide d'un capteur piézoélectrique. L'objectif étant de capter avec une précision suffisante la vibration d'un tuyau pour permettre une analyse du résultat.

Tableau 5 : Matériel employé

Intitulé	Fournisseur	Modèle	Prix
Capteur piézoélectrique	Récupération		
Carte d'acquisition	National Instrument	USB6003	Prêté par la HES

6.4.1.1 Acquisition des données

Pour l'acquisition des données, une application LabView a été réalisée. Pour chaque débit, rentré manuellement dans l'application, celle-ci effectue 20 fois 10'000 échantillonnages à 100 [kHz] qu'elle exporte dans un fichier CSV où chaque ligne correspond à une des 20 séries de mesures.

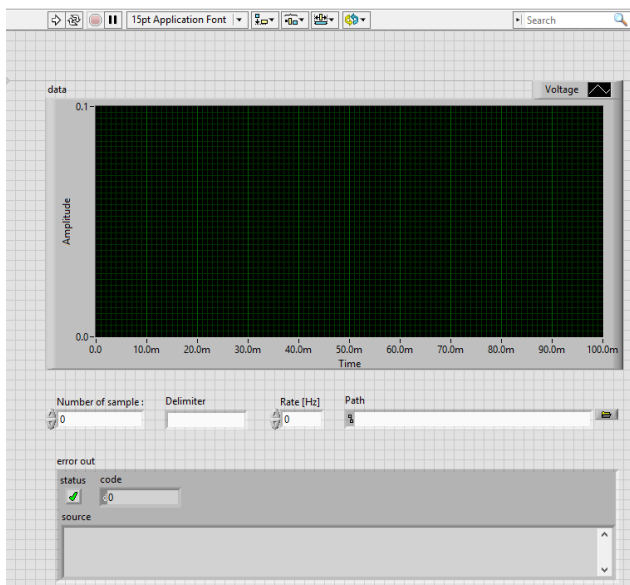


Figure 18 : GUI de l'application.

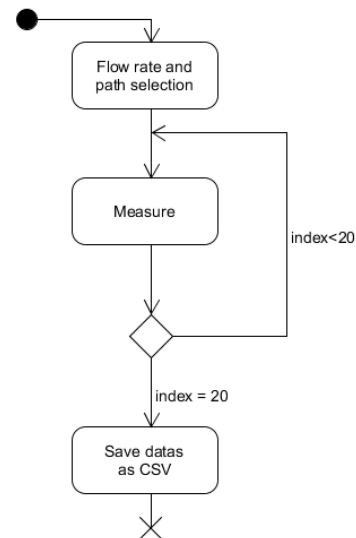


Figure 19 : Schéma bloc de l'application.

6.4.1.2 Précision de l'appareil de mesure

La carte d'acquisition fournit un convertisseur analogique digital avec une résolution de 16 bits. La précision

$$V_{LSB} = \frac{V_{CC} - V_{SS}}{2^{BitNb}} = \frac{10 + 10}{2^{16}} = 305.2 [\mu V]$$

6.4.1.3 Analyse des données

En analysant les données, un problème a été constaté, le signal ressemble à un sinus à 50 [Hz]. L'origine de ce problème provient du fait que le tuyau était relié à la terre du réseau électrique qui était sans aucun doute perturbée. Les électrodes en cuivre du capteur ont donc ajouté cette perturbation à la mesure. Après plusieurs essais infructueux de filtrer ce signal perturbateur, la décision a été prise de l'analyser tel quel en raison du bruit de filtrage qui est également ajouté.

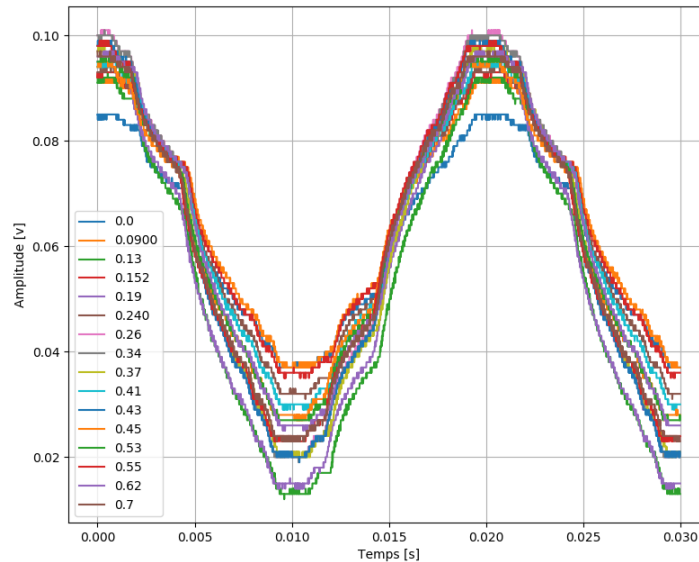


Figure 20 : Résultat temporel des mesures.

Pour une étude plus poussée du résultat, les corrélations entre le débit et l'écart-type, la différence des amplitudes maximum et minimum, la valeur moyenne et l'amplitude de la première harmonique ont été analysées. Cette analyse a permis de mettre en avant certaines valeurs aberrantes (encadrées en rouge dans les graphiques ci-dessous) qui apparaissent certainement en raison de la perturbation expliquée plus haut.

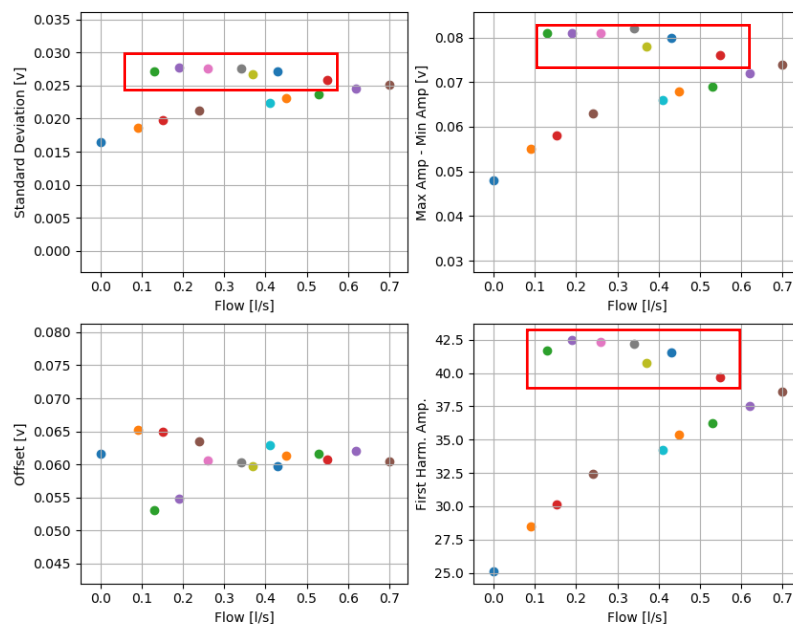


Figure 21 : Analyse des premières mesures.

Il est tout de même intéressant de constater une évolution à première vue logarithmique de la différence d'amplitude aux extremums et de l'amplitude de la première harmonique en fonction du débit et une évolution linéaire de la déviation standard en fonction du débit. La valeur moyenne du signal semble donner aucune information sur le débit.

6.4.1.4 *Avantages / Inconvénients du prototype*

Bien que simple de mise en place ce prototype d'acquisition présente quelques problèmes :

- En plus d'être perturbé par les vibrations extérieures au système, ce capteur est également perturbé par les tensions présentes sur un tuyau si celui-ci est relié à la terre (Ce qui est assez fréquent). Il est toutefois possible d'isoler les électrodes pour résoudre ce problème.
- Que trop peu d'informations sont disponibles sur l'emploi d'éléments piézoélectrique comme capteur de vibration. Le datasheet ne donne malheureusement aucune information sur les caractéristiques de déformation du capteur.
- Un élément piézoélectrique n'a qu'un seul signal en sortie. Difficile donc savoir si les 3 axes sont correctement représentés.

6.4.1.5 *Décision*

Après discussion avec les différents responsables, le choix de laisser tomber la mesure de vibration par capteur piézoélectrique a été fait. Il a été jugé préférable de se concentrer sur un prototype plus professionnel et plus propre pour permettre d'étudier le phénomène et non juste de le constater.

6.4.2 *Deuxième prototype « Version 1 »*

A la vue des résultats obtenus avec le premier prototype, la décision de réaliser un nouveau capteur en partant de zéro a été prise.

Pour répondre aux objectifs du projet, un microphone a d'abord été envisagé car à première vue celui-ci était moins sujet (du moins plus facilement isolable) aux perturbations qu'un accéléromètre. Mais celui-ci a été mis de côté en raison de la limite basse de sa bande passante (à partir de 20 [Hz]), jugée trop élevée (cf. chapitre 6.2). Il existe cependant des microphones avec une bande passante inférieure plus basse mais le prix est beaucoup plus élevé (environ 50 CHF).

Une fois le microphone écarté, un accéléromètre a logiquement été retenu pour réaliser le prototype. En plus de celui-ci, le choix a été fait de prendre également un capteur de température dans le but d'éliminer les faux positifs. Vu que le réseau d'irrigation est enterré à environ deux mètres de profondeurs le passage de l'eau dans le tube refroidira celui-ci, ce qui évitera au prototype de mesurer un débit alors qu'aucun écoulement n'est présent.

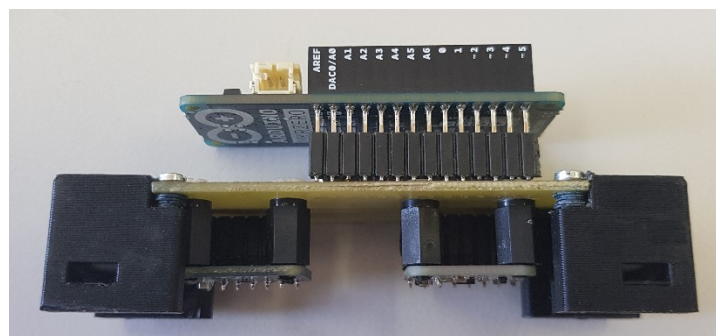


Figure 22 : Deuxième prototype.

6.4.2.1 Choix des capteurs

Avant de choisir les capteurs, il a été choisi d'utiliser le protocole I²C pour la communication entre les capteurs et le microcontrôleur.

6.4.2.1.1 Capteur de température

Le choix du capteur de température s'est porté sur le TMP006 de Texas Instrument car celui-ci est un capteur de température sans contact ce qui simplifie la mise en place. De plus un « breakout » a été créé par Adafruit⁸ pour faciliter le prototypage avec celui-ci.

Caractéristiques principales :

- Tension d'alimentation : 3 à 5 [V]
- Température de fonctionnement : de -40 à 125 [°C]
- Adresse I²C : 40₁₆ (configurable)
- Dimensions : 20 x 20 x 2 [mm]



Figure 23 : TMP006 avec sa carte d'adaptation.

6.4.2.1.2 Accéléromètre

Pour le capteur d'accélération, un MMA8451⁹ de chez Xtrinsic a été choisi. Il dispose également d'un « breakout » réalisé par Adafruit.

Caractéristiques principales :

- Tension d'alimentation : 3 à 5 [V]
- Température de fonctionnement : de -40 à 85 [°C]
- Résolution du convertisseur A/D : 14 bits
- Plage d'utilisation : de ±2 à ±8 [g]
- Adresse I²C : 1D₁₆ (configurable)
- Fréquence d'échantillonnage : jusqu'à 850 [Hz]
- Dimensions : 21 x 18 x 2 [mm]

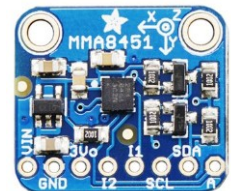


Figure 24 : MMA8451 avec sa carte d'adaptation.

6.4.2.1.3 Choix du microcontrôleur

Dans le but d'accélérer la mise en place du prototype, une carte de développement a été préférée.

En raison des capteurs choisis et des différentes contraintes d'installation, la carte de développement doit répondre aux caractéristiques suivantes :

- Une carte de petite taille serait préférable pour une installation directement sur le tuyau.
- Offrir une alimentation entre 3 et 5 [V] pour les capteurs.
- Le microcontrôleur présent sur la carte doit répondre aux caractéristiques suivantes :
 - Dispose d'une interface I²C.
 - Dispose d'une interface UART pour la communication entre le PC et le capteur.

En se basant sur les prérequis cités plus haut. La carte de développement Arduino MKRZERO¹⁰ a été choisie.

Caractéristiques principales :

- Microcontrôleur : ARM Cortex M0+
- Tension d'alimentation : De 3.7[V] (batterie) à 5 [V] (par USB)
- Interfaces : 1x UART, 1x I²C
- Dimensions : 65 x 25 [mm]

⁸ Plus d'informations sur le capteur : <https://www.adafruit.com/product/1296>

⁹ Plus d'informations sur l'accéléromètre : <https://www.adafruit.com/product/2019>

¹⁰ Plus d'informations sur la carte de développement : <https://store.arduino.cc/usa/arduino-mkrzero>



Figure 25 : Carte de développement Arduino MKR Zero.

6.4.2.1.4 Coûts du matériel du prototype

Tableau 6 : Récapitulatif des coûts Hardware du prototype.

Désignation	Modèle	Fournisseur	N° fournisseur	Prix
Kit de développement	ARDUINO MKR 0	Digikey	1050-1137-ND	24.64 CHF
Accéléromètre	MMA8451	Digikey	1528-1041-ND	7.95 CHF
Capteur de température	TMP006	Digikey	1528-1808-ND	12.50 CHF

6.4.2.2 PCB de raccordement

Dans le but de faciliter le branchement des différents éléments du capteur, un PCB qui s'occupe de relier les pattes SDA, SCL, GND, VCC de chaque élément correctement a été réalisé. Grâce à celui-ci, il suffit de brancher chaque élément dans les barrettes qui lui correspondent.

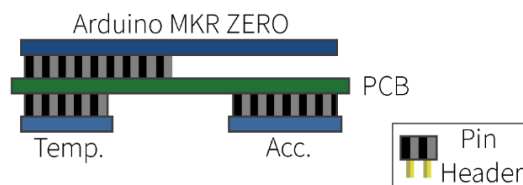


Figure 26 : Schéma du PCB de raccordement.

De plus des entretoises ont été rajoutées pour éviter aux pattes des capteurs de plier en cas d'accident.

6.4.2.3 Support pour fixation sur un tuyau

Pour faciliter la fixation du capteur sur un tuyau, une pièce imprimée en 3D a été réalisée. Cette pièce vient se poser sur la largeur du PCB de raccordement. Une excavation sur le bas de la pièce donne un effet rail pour permettre au capteur d'être correctement posé sur le tuyau. La fixation sur le tuyau se fait par des brides qui peuvent être enfilées dans la pièce.

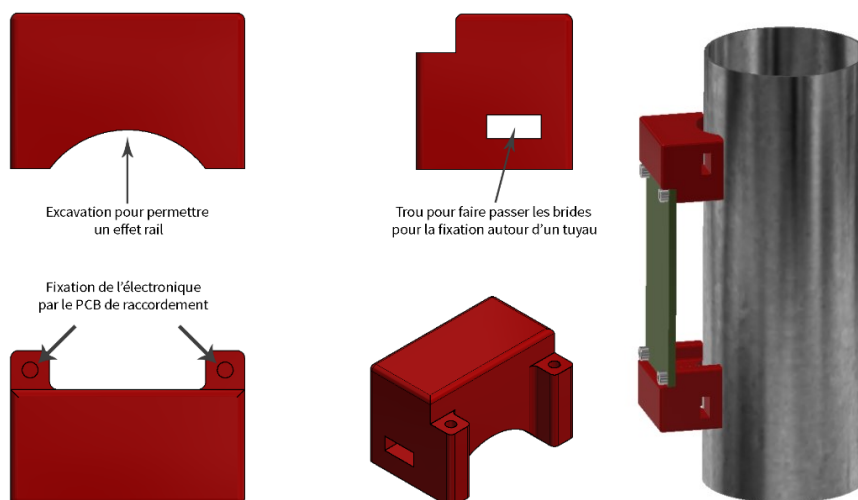


Figure 27 : Présentation du support de fixation.

6.4.2.4 Partie software (logicielle)

L'architecture logicielle du prototype est séparée en deux parties :

- Une première partie qui est un script d'acquisition python exécuté sur un ordinateur portable. Celui-ci a comme missions :
 - Faire le pont entre l'utilisateur et le capteur.
 - Générer les fichiers de résultats pour qu'ils puissent être analysés par la suite pour les insérer dans un algorithme de Machine Learning.
- La deuxième partie est le logiciel embarqué sur le capteur, celui-ci a comme mission d'attendre les ordres de la première partie et de répondre à ceux-ci. Il permet de lire et de paramétrer les capteurs.

Les deux logiciels communiquent entre eux par UART à travers de l'USB qui fournit également l'alimentation de la carte de développement. La vitesse de transmission est de 115'200 bauds.

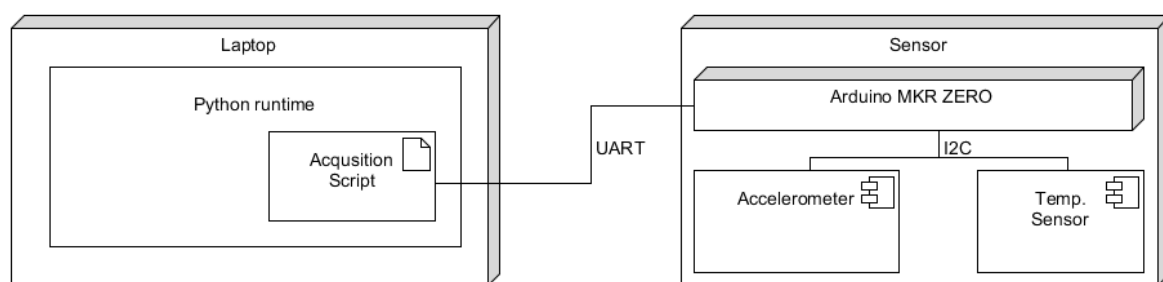


Figure 28 : Diagramme physique du capteur.

6.4.2.4.1 Logiciel embarqué dans le capteur

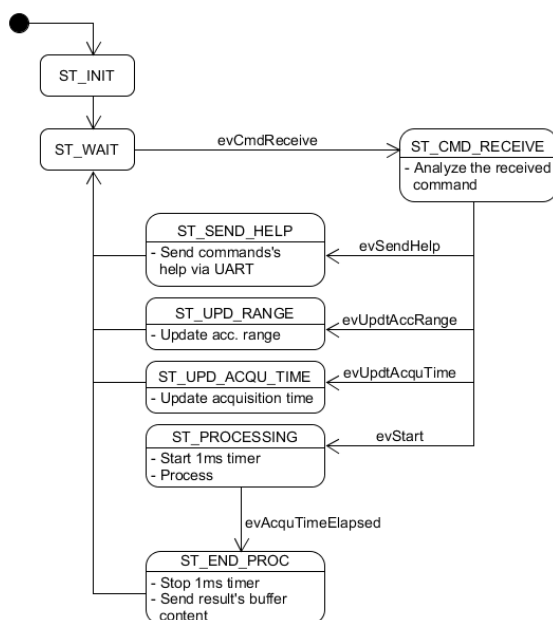


Figure 29 : Machine d'état du capteur.

Le capteur a été programmé pour répondre à 4 commandes :

- **GHELP**, qui envoie au destinataire des informations sur les commandes disponibles.
- **START**, qui lance une acquisition de données.
- **RANGE <2|4|8>**, qui modifie l'échelle de l'accéléromètre.
- **STIME <Time [ms]>**, qui modifie le temps d'acquisition des vibrations. Ce temps peut être compris entre 10 et 1000 [ms]

Les commandes ont été exprimées explicitement dans un anglais compréhensible pour permettre l'emploi du capteur sans logiciel d'acquisition.

Lors du lancement d'une acquisition, le capteur démarre un timer sera échu lorsque le temps d'acquisition aura été atteint.

Pendant que le timer est enclenché, le capteur s'efforce de faire le plus de mesures de vibration possible et les stocke dans un tampon.

Une fois le timer échu, le contenu du tampon est envoyé au destinataire ainsi que la température de l'objet.

Le capteur donne ses réponses au demandant par format JSON alors que les requêtes sont au format texte brut.

```

1 {
2   "type": "setTime",
3   "time": 999,
4   "range": 2,
5   "acc": {
6     "x": 0,
7     "y": 0,
8     "z": 0,
9     "t": -1
10  },
11  "temp": {
12    "objtemp": 0,
13    "ambtemp": 0
14  }
15 }

```

Figure 30 : Format de réponse du capteur.

Le format de réponse est toujours le même et c'est au demandant de ressortir les informations qui l'intéressent.

L'attribut *type* indique l'action qui est à l'origine de cette réponse. Dans ce cas, c'est la modification du temps d'acquisition qui en est l'origine (d'où le *setTime*).

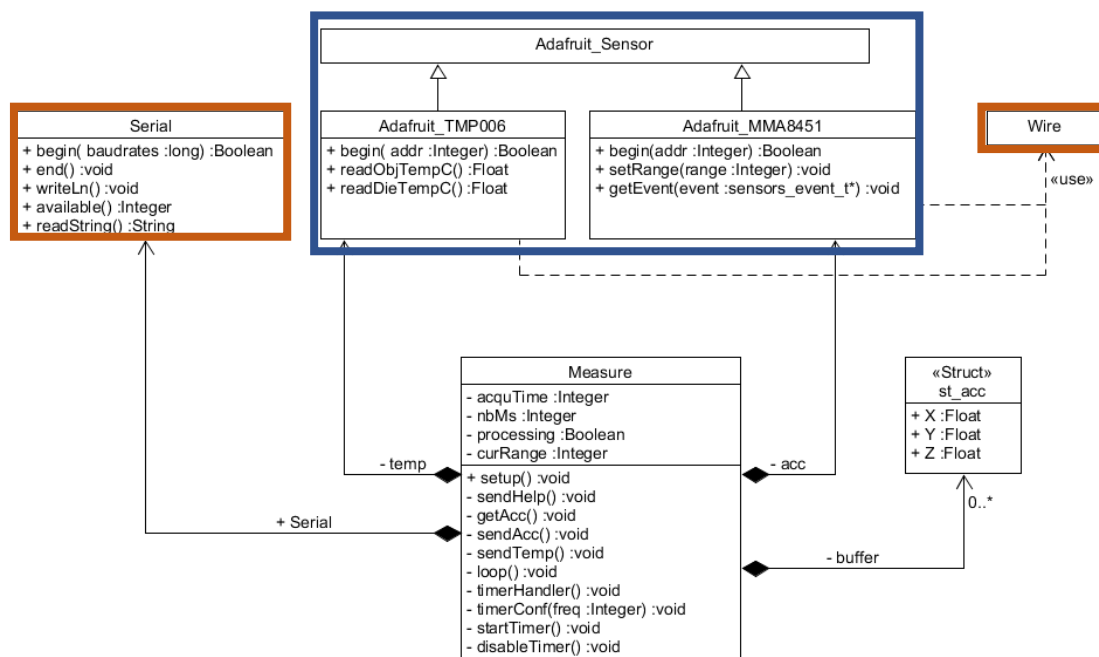


Figure 31 : Diagramme de classe de l'application embarquée.

 Librairie fournie par Adafruit.

 Librairie fournie par Arduino.

La plateforme Arduino ne propose pas de Timer matériel en natif. Les délais étant non relevant dans ce cas, un timer matériel a été rajouté¹¹ en copiant les éléments liés aux timers dans la HAL du fabricant du microprocesseur (Atmel, µC : SAM D21J).

¹¹ Workaround pour le timer : <https://gist.github.com/nonsintetic/ad13e70f164801325f5f552f84306d6f>

6.4.2.4.2 Script d'acquisition Python

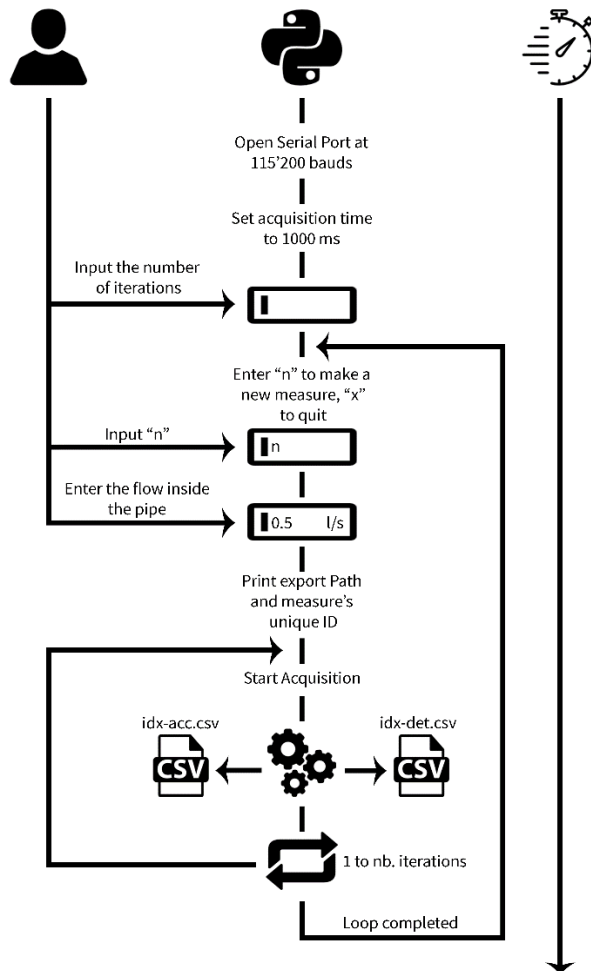


Figure 32 : Diagramme de flux du script d'acquisition.

Le script d'acquisition a été réalisé en Python. Celui-ci s'exécute dans la console et permet à un utilisateur d'effectuer une série de mesure sur le capteur. Il ne donne cependant pas la possibilité de définir la résolution de l'accéléromètre et le temps d'acquisition. Il permet cependant de définir le nombre de fois qu'une mesure doit être prise pour un même débit. Ce paramètre peut être utile pour éliminer les valeurs aberrantes sur une série de mesures. Pour chaque mesure de vibration pour un débit rentré par l'utilisateur, deux fichiers de sortie sont générés :

- Le premier contient les informations vibratoires des 3 axes pendant la période d'acquisition du capteur.
- Le deuxième contient les informations sur la température ambiante et celle de la conduite. De plus il dispose aussi de toutes les informations sur le capteur et la conduite (Résolution, débit, diamètre de la conduite, matériel du tuyau, la position du capteur par rapport au flux d'eau).

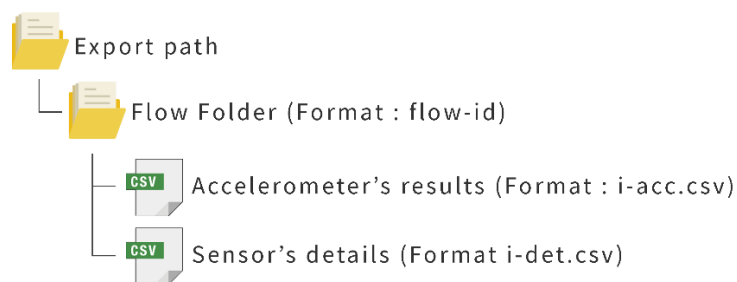


Figure 33 : Chemin d'exportation des fichiers de sortie.

6.4.2.4.3 Tests des logiciels du prototype

Description du test	Fonctionne ?
Un Timer s'enclenche chaque 1 [ms] pour permettre la gestion du temps d'acquisition (cf. Figure 34).	OUI
Lors de l'envoi d'une commande GHELP par UART, le capteur renvoie l'aide aux commandes (cf. Figure 35).	OUI
Lors de l'envoi d'une commande STIME + temps par UART, le capteur modifie son temps d'acquisition (cf. Figure 35).	OUI
Lors de l'envoi d'une commande RANGE + résolution par UART, le capteur modifie la résolution de l'accéléromètre (cf. Figure 35).	OUI
Lors de l'envoi d'une commande START par UART, le capteur, indique dans l'ordre : Le début d'une acquisition, la température ambiante et la température de l'objet, les valeurs de l'accéléromètre pendant le temps d'acquisition, la fin de l'acquisition (cf. Figure 36).	OUI
Le script d'acquisition respecte le nombre d'itération demandé pour chaque débit ainsi que la nomenclature décrite précédemment (cf. Figure 37).	OUI
Pour chaque itération d'une mesure, un fichier contenant les détails de l'accéléromètre et un fichier contenant les détails du capteur et la température est créé (cf. Figure 37 et Figure 38).	OUI

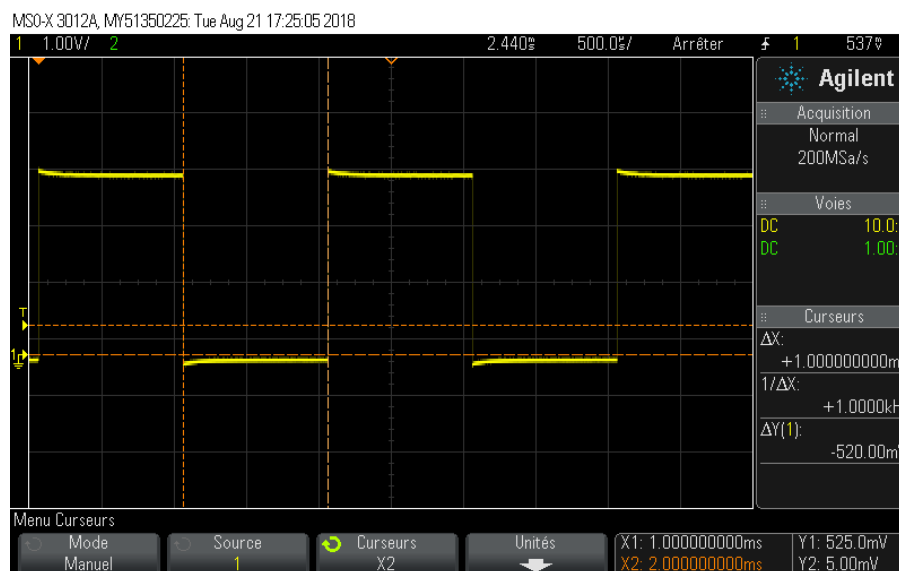


Figure 34 : Timer à 1 [kHz].

```

COM5 - PuTTY
GHELP
FLOW METER
HES-SO Valais
Syntax :
STIME <Time [ms]>      Set the acquisition time in ms from 10 to 1000
RANGE <2|4|8>          Set the accelerometer's resolution from +- 2 to +-8 g
START                  Start a new measure
GHELP                  Get the syntax help
=====
STIME 850
[{"type":"setTime","time":850,"range":2,"acc":{"x":0,"y":0,"z":0,"t":-1},
"temp":{"objtemp":0,"ambtemp":0}}]
RANGE 4
[{"type":"setRange","time":850,"range":4,"acc":{"x":0,"y":0,"z":0,"t":-1},
"temp":{"objtemp":0,"ambtemp":0}}]
^S

```

Figure 35 : Commandes de base du capteur.


```

COM5 - PuTTY
{ "type": "setTime", "time": 10, "range": 4, "acc": { "x": 0, "y": 0, "z": 0, "t": -1 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
START
{ "type": "startProc", "time": 10, "range": 4, "acc": { "x": 0, "y": 0, "z": 0, "t": -1 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "temp", "time": 10, "range": 4, "acc": { "x": 0, "y": 0, "z": 0, "t": -1 }, "temp": { "objtemp": 30.83, "ambtemp": 29.53 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.21, "y": 0.74, "z": -9.79, "t": 0 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.22, "y": 0.70, "z": -9.76, "t": 1 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.23, "y": 0.69, "z": -9.76, "t": 2 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.23, "y": 0.71, "z": -9.80, "t": 3 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.21, "y": 0.71, "z": -9.80, "t": 5 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.21, "y": 0.69, "z": -9.81, "t": 6 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.20, "y": 0.71, "z": -9.79, "t": 7 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "acc", "time": 10, "range": 4, "acc": { "x": -0.23, "y": 0.68, "z": -9.80, "t": 9 }, "temp": { "objtemp": 0, "ambtemp": 0 } }
{ "type": "endProc", "time": 10, "range": 4, "acc": { "x": 0, "y": 0, "z": 0, "t": -1 }, "temp": { "objtemp": 0, "ambtemp": 0 } }

```

Figure 36 : Exécution d'une commande START.

Windows (C:) > temp > Measure > Cannelabo > 0.00-338

Dossier d'exportation

Dossier pour un débit (débit-id)

Modifié le

Nom	Modifié le
0-acc.csv	16.07.2018 11:31
0-det.csv	16.07.2018 11:31
1-acc.csv	16.07.2018 11:31
1-det.csv	16.07.2018 11:31
2-acc.csv	16.07.2018 11:31
2-det.csv	16.07.2018 11:31
3-acc.csv	16.07.2018 11:31
3-det.csv	16.07.2018 11:31
4-acc.csv	16.07.2018 11:31
4-det.csv	16.07.2018 11:31

Pour chaque itération d'une mesure, un fichier contenant les détails de l'accéléromètre et un fichier contenant les détails du capteur.

Figure 37 : Nomenclature pour les résultats.

	A	B	C	D
1	x	y	z	t
2	-6.38	11.73	-8.4	0
3	13.46	-10.2	-3.41	1
4	2.86	-7.35	-11.75	2
5	4.8	6.27	-6.58	3

	A	B	C	D	E	F	G
1	Débit	Temp. Ob	Temp. Amb.	Diam.	Matériel	Résolutio	Position
2	0	22.72	23.74	51	Galvanized steel (old)	2	Horizontal (X+)

Figure 38 : Aperçu des fichiers du résultat.

6.4.2.5 Tests du prototype sur le terrain

Durant la réalisation de ce prototype, l'occasion d'aller directement tester ce prototype sur le terrain s'est présentée. Ces tests sur le terrain ont été l'occasion de s'intéresser particulièrement à l'application concrète de ce prototype et aux problèmes du terrain.



Figure 39 : Test du prototype sur le terrain.

Ces différents tests ont pu détecter certains défauts du prototype tel que son imperméabilité et le confort de prise de mesure. De plus en raison des débits importants mesurés (400 [l/s]), il arrive que le capteur sature (résolution fixée à ± 2 [g]).

Pour pallier ses différents problèmes, les mesures différentes ont été décidées :

- Lors de la prise de mesure un tuyau doit être installé à la sortie de la canne d'arrosage (ce qui est toujours le cas en application réelle).
- L'application d'acquisition va être réalisée de manière graphique au lieu de l'application console de manière à pouvoir lancer une mesure d'un simple clic sur un bouton.
- Pour éviter la saturation, l'application d'acquisition doit la détecter et régler la résolution de l'accéléromètre en conséquence (de ± 2 [g] à ± 8 [g]).

Pour donner suite à ces différentes modifications, la décision de repenser complètement l'application d'acquisition a été prise.

6.4.3 Deuxième version du logiciel d'acquisition

Pour résoudre les problèmes cités plus haut, une nouvelle version du logiciel d'acquisition a été réalisée. Après avoir longtemps étudié la piste d'une application graphique en Python (avec Tkinter¹²), celle-ci a été abandonnée en raison de la complexité de la structure et du manque d'ergonomie pour la création d'une interface graphique et la gestion d'événements.

Une application en C# WPF¹³ a été préférée, grâce notamment à la simplicité d'utilisation de Visual Studio.

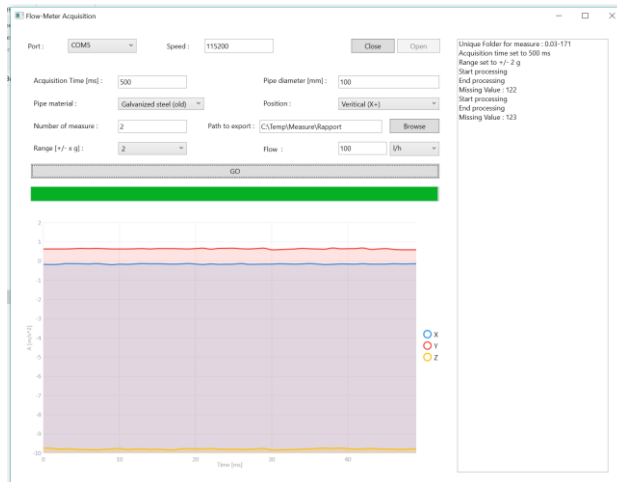


Figure 40 : Interface graphique de l'application d'acquisition.

L'interface graphique a été séparée en 3 parties : Les paramètres à régler, les logs et la visualisation des mesures qui est utile pour détecter manuellement des perturbations sur les mesures. Les signaux ont été affichés par le biais de la librairie LiveCharts¹⁴.

Au démarrage de l'application, il est uniquement possible de sélectionner le port COM et la vitesse pour se connecter au capteur. Une fois la connexion établie, le reste de l'application devient opérationnel. Les fichiers de sorties restent de même format que dans la première version du logiciel.

De plus, lorsque le capteur sature, une fenêtre vient informer l'utilisateur et propose de relancer la mesure avec une résolution plus haute.

Cette détection se fait en analysant l'accélération mesurée et en la comparant à la résolution choisie du capteur. Lorsque celle-ci est trop proche à plusieurs reprises de la limite, l'utilisateur est notifié.

L'unité du débit peut être spécifiée pour permettre de s'adapter facilement aux différents formats des débitmètres, celui-ci sera converti en litre par seconde dans le fichier de sortie.

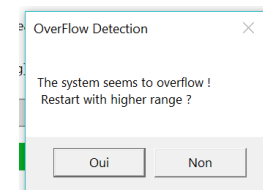


Figure 41 : Avertissement de saturation du capteur.

Une autre amélioration effectuée sur cette version a été la compensation des valeurs manquantes. Etant donné que l'accéléromètre dispose d'une fréquence d'échantillonnage plus petite que 1 [kHz] (la fréquence du timer sur le capteur), il est fréquent que les données vibratoires pour une certaine milliseconde est manquante. Pour corriger ce problème le logiciel effectue la moyenne de la valeur suivante et précédente.

$$A_{t(\text{manquante})} = \frac{A_{t-1} + A_{t+1}}{2}$$

Cette application a l'avantage de permettre une modification directe de tous les paramètres présents dans les fichiers de sortie alors que dans la version précédente, ceux-ci étaient présent en dur dans le code.

¹² Plus d'informations sur Tkinter : <https://docs.python.org/2/library/tkinter.html>

¹³ Plus d'informations sur WPF : https://fr.wikipedia.org/wiki/Windows_Presentation_Foundation

¹⁴ Librairie pour les graphiques : <https://lvcharts.net>

6.5 MESURES

Une fois le prototype réalisé, différentes mesures ont été effectuées dans un premier temps sur les deux bancs de test et par la suite sur le terrain directement à Bagnes.

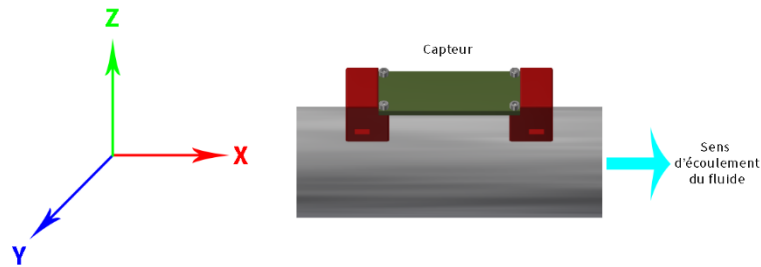


Figure 42 : Position des axes de l'accéléromètre.

6.5.1 Script de visualisation

Un script de visualisation des données a été effectué en Python. Ce script a pour objectif principal de déterminer à quel point les données mesurées pour un débit sont fiables et exploitables. Ce script, affiche la déviation standard, l'offset, l'amplitude de la première harmonique et le delta entre l'amplitude maximale et minimale en fonction du débit pour les 3 axes.

6.5.2 Mesures en laboratoire

6.5.2.1 Petit banc de test

La première série de mesure a été effectuée sur le petit banc de test, 135 mesures de vibrations ont été effectuées pour 27 débits différents de 0 [l/s] à 0.8 [l/s]. Le temps d'acquisition pour chaque vibration était de 500 [ms].

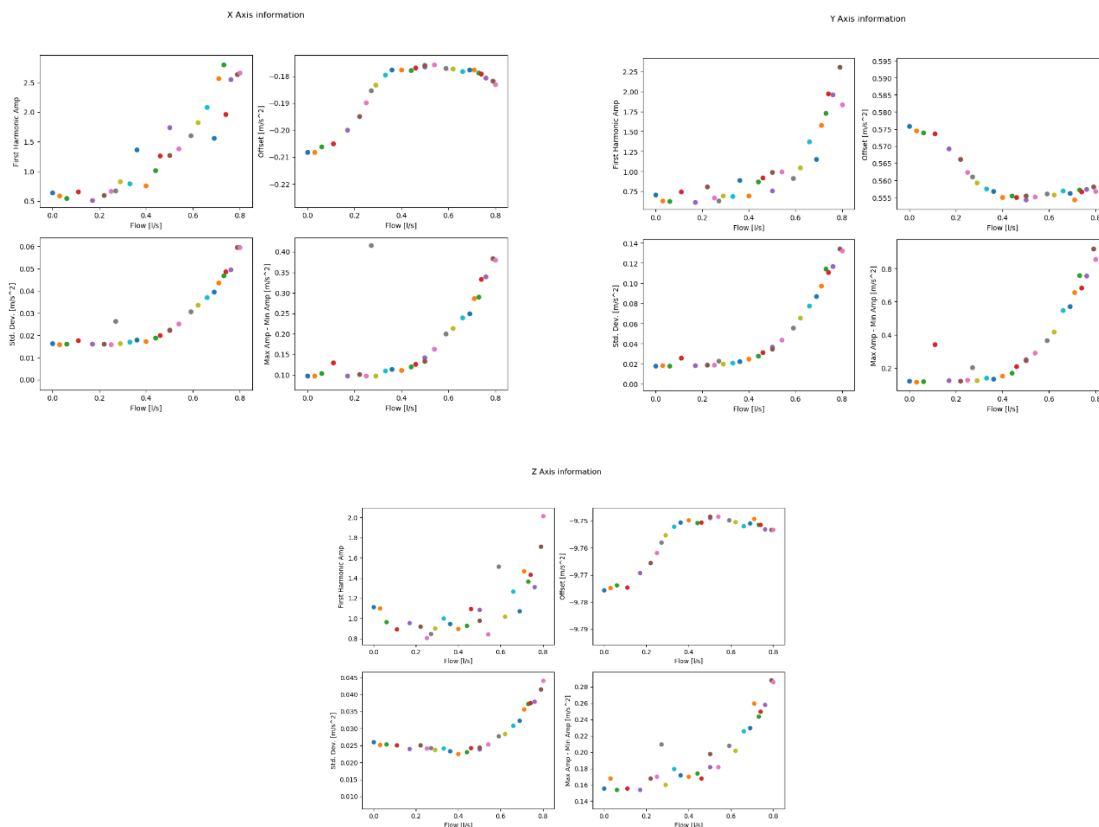


Figure 43 : Résultats du script de visualisation sur le petit banc de test.

Il est intéressant de constater que les axes X et Y donnent de bonnes informations sur la relation entre les vibrations et le débit qui en découle. L'axe Z quant à lui semble donner des informations moins précises, bien que les 3 axes donnent les mêmes tendances de courbes. Avec une courbe qui pourrait être qualifiée de quadratique pour la déviation standard, l'amplitude de la première harmonique et le delta des amplitudes extrêmes. Alors que l'offset donne une courbe logarithmique qui est particulièrement intéressante pour les petits débits.

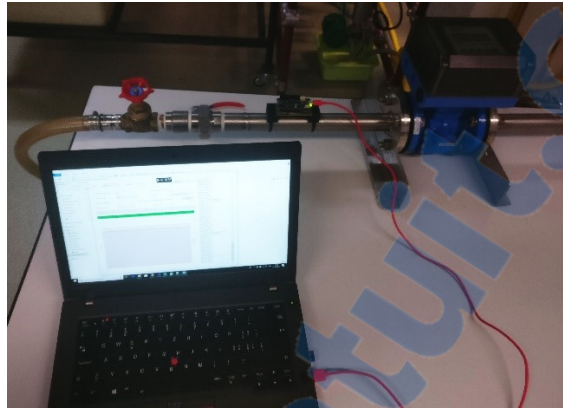


Figure 44 : Système monté sur le petit banc de test.

6.5.2.2 Grand banc de test

Une fois les mesures sur le petit banc de test analysées, le prototype a été testé sur le grand banc de test hydraulique de la HEI, 130 mesures de vibrations ont été effectuées pour 26 débits différents de 0 [l/s] à 6.88 [l/s]. Le temps d'acquisition pour chaque vibration était de 1'000 [ms].

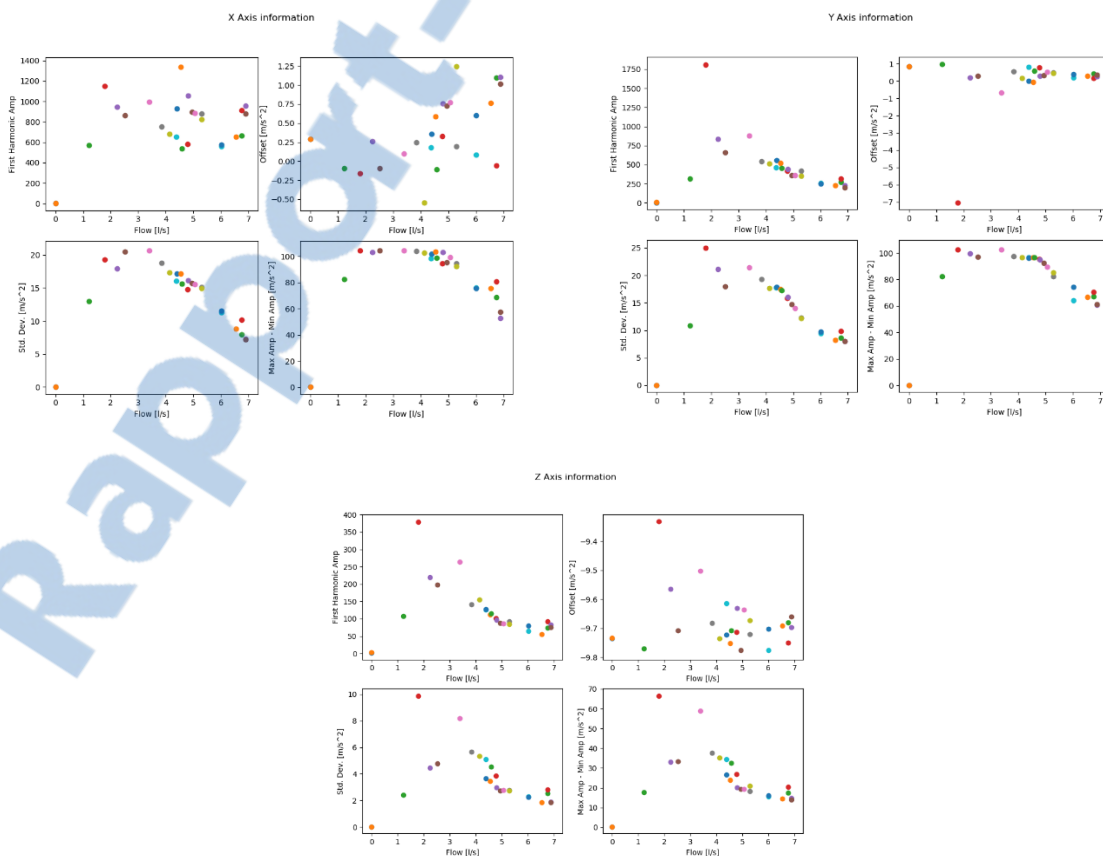


Figure 45 : Résultats du script de visualisation sur le grand banc de test.

Malheureusement, les vibrations sont trop perturbées par les pompes (cf. chapitre 6.3.2) en fonctionnement pour donner des courbes précises. La perturbation en elle-même de la pompe pourrait être filtrée, mais dû à l'environnement (notamment un plancher en bois qui sépare le banc de test des pompes) cette vibration devient impossible à filtrer vu qu'elle occupe l'entier du spectre des données. Une tendance peut tout de même être observée sur la courbe de la déviation standard ainsi que la courbe de l'offset.

Il est également intéressant de constater que certaines courbes présentent un maximum vers la moitié du débit maximal possible sur le banc de test. Une hypothèse concernant ce maximum serait que la vanne de réglage du débit perturbe le flux d'eau et que ce maximum correspondrait au moment où le débit et la surface du boisseau encore à l'intérieur de la conduite seraient maximaux (vanne à moitié ouverte). Pour confirmer cette hypothèse, il est nécessaire d'aller effectuer des tests directement sur le terrain où aucune pompe ou machine ne sont présentes pour perturber le capteur.

6.5.3 Mesures sur le terrain

Comme expliqué précédemment, des tests sur le terrain sont nécessaires pour déterminer les perturbations de la vanne et si ce prototype peut être fiable en pratique. Pour réaliser ces tests, une journée complète a été dédiée à des mesures sur le terrain. Un débitmètre à ultrasons portable a été emprunté et le prototype a été testé sur deux cannes différentes pour permettre de déterminer la manière dont une pression de conduite différente interfère sur les différentes courbes.

Pour ces séries de mesures, un tuyau a été installé au bout de la canne pour éviter d'endommager l'électronique avec des éclaboussures d'eau.



Figure 46 : Montage du prototype sur une canne d'arrosage.

6.5.3.1 Canne numéro 1

Pour cette canne, 175 mesures ont été effectuées pour 35 débits différents de 0 [l/s] à 9.17 [l/s]. Le temps d'acquisition pour chaque vibration était de 1'000 [ms].

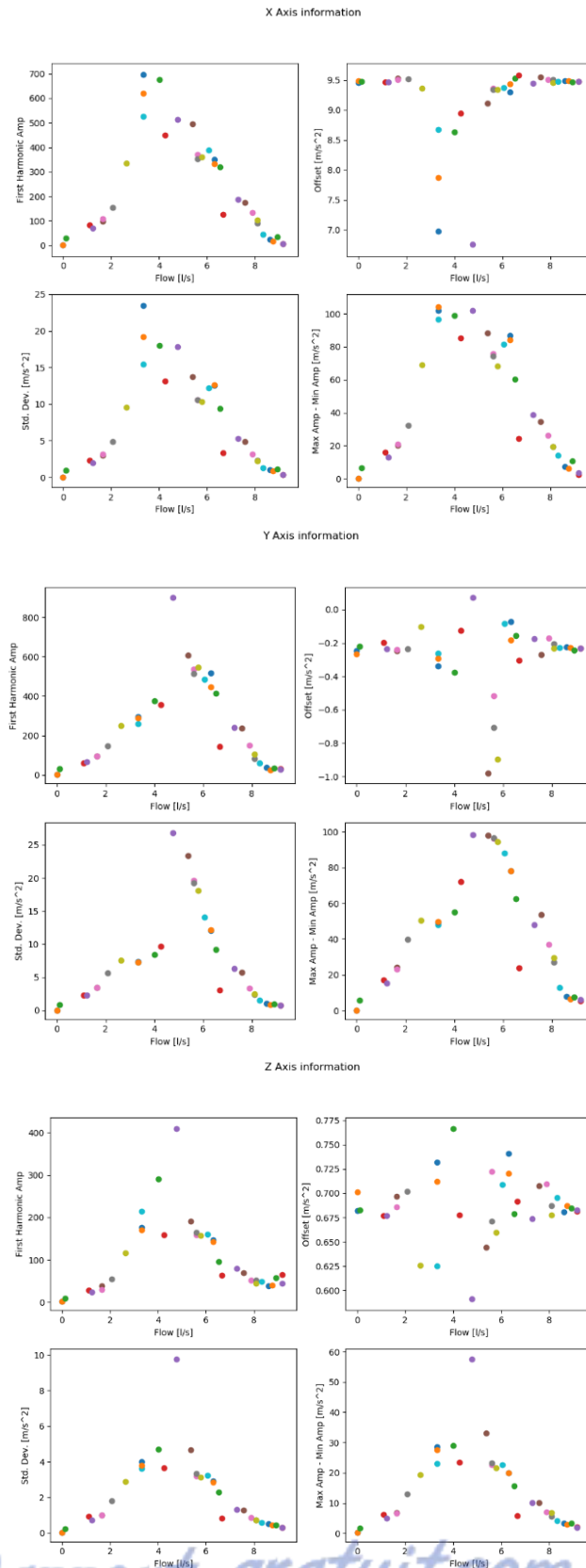


Figure 47 : Résultats du script de visualisation sur la canne d'arrosage n°1.

6.5.3.2 Canne numéro 2

Pour cette canne, 125 mesures ont été effectuées pour 25 débits différents de 0 [l/s] à 6.13 [l/s]. Le temps d'acquisition pour chaque vibration était de 1'000 [ms].

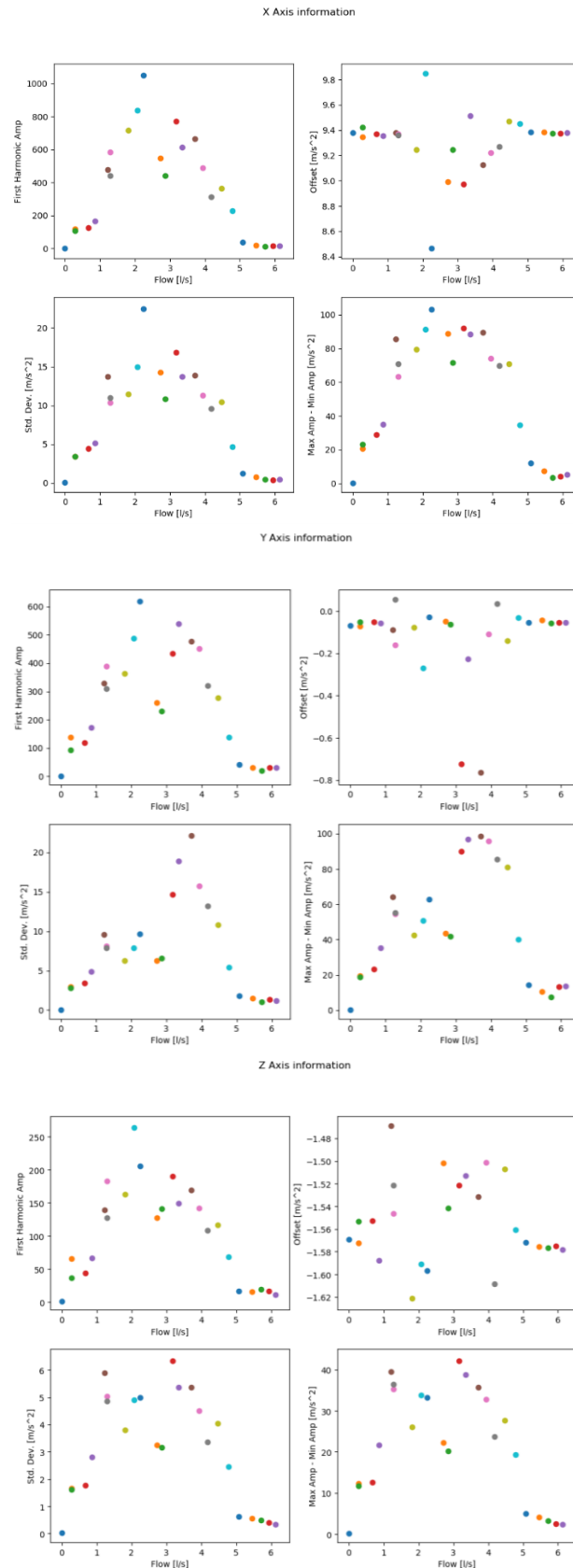


Figure 48 : Résultats du script de visualisation sur la canne d'arrosage n°2.

6.5.3.3 Comparaison entre les deux cannes

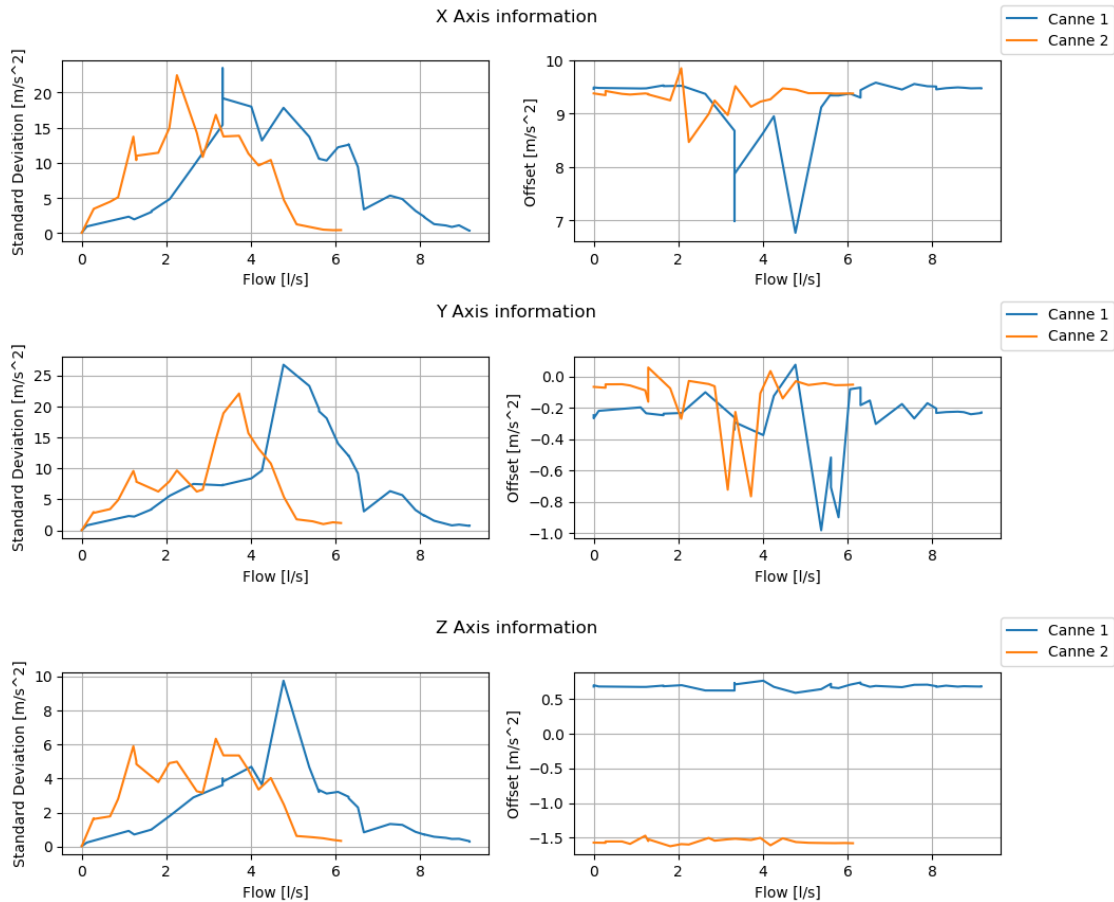


Figure 49 : Comparaison entre les deux cannes testées sur le terrain.

La présence des différents maximums sur les données du terrain, indique que l'hypothèse sur la perturbation de la vanne sur la mesure est correcte (cf. chapitre 6.5.2.2). Il est également intéressant de constater la similitude des données entre les deux cannes d'arrosage, il peut donc être constaté qu'une pression différente (=débit maximal différent) ne change en rien à la tendance de la courbe, celle-ci sera juste élargie sur son axe X (débit) et Y.

Cependant, la suite du projet n'est pas facilitée par la perturbation liée à la vanne, car de savoir de quel côté de l'extrema une vibration se trouve ne va pas être tâche facile pour les algorithmes de *machine learning* vu qu'aucune des caractéristiques principales du modèle est injective.

En analysant les différents résultats des tests sur le terrain, la pertinence de l'emplacement du prototype sur la canne d'arrosage peut être remise en cause. La proximité de la vanne perturbe fortement le système.

6.5.3.4 Bilan des différentes mesures

Ces différentes mesures, ont montrées que le prototype peut être performant sur des écoulements libres (sans pompes), mais elles ont également permis de montrer la plus grande faiblesse de ce prototype, à savoir qu'il est très facilement perturbé par l'environnement extérieur.

6.6 QUANTIFICATION DU DÉBIT À L'AIDE D'ALGORITHMES DE *MACHINE LEARNING*

A la vue de la complexité du système, il a été choisi plutôt que de comparer une résolution classique à une résolution par *machine learning*, de comparer une résolution avec un réseau de neurones à une résolution par *machine learning* sans réseau de neurones.

6.6.1 Pourquoi le *machine learning* ?

L'analyse des mesures a permis de mettre en avant une caractéristique prometteuse (l'écart-type). Malheureusement celle-ci ne donne que de faibles informations sur les petits débits. Par chance d'autres caractéristiques donnent de plus amples informations sur les plus petits débits comme le montre la figure ci-dessous. De plus, les mesures du terrain sont fortement perturbées, c'est pour ça que toutes les caractéristiques disponibles sont nécessaires pour espérer un bon résultat.

C'est dans ce genre de situation expliquée ci-dessus que le *machine learning* peut s'avérer particulièrement efficace, lorsqu'il s'agit de prédire des modèles complexes. En effet, une résolution du problème de manière classique serait bien trop couteuse en temps et certainement moins performante.

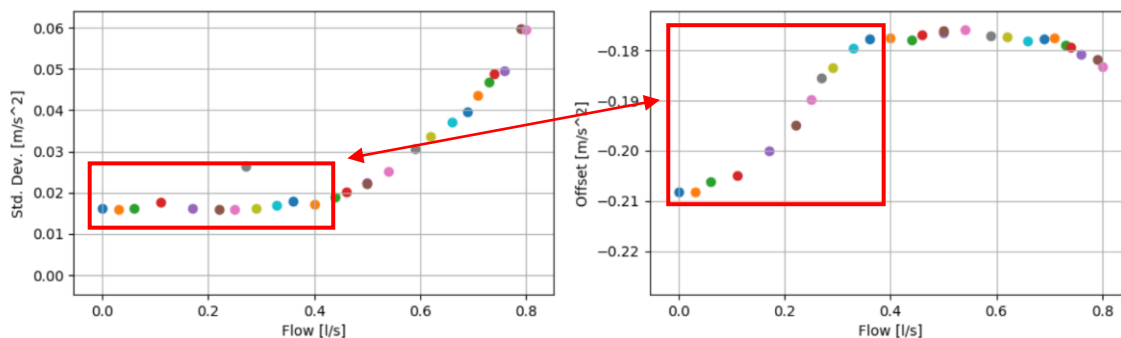


Figure 50 : Analyse des données du petit banc de test.

6.6.2 Calcul de la performance d'un modèle

Pour pouvoir comparer les différents modèles générés, il est nécessaire d'avoir une caractéristique qui permet d'indiquer la précision de ceux-ci. Lorsqu'il s'agit d'un problème de régression (une valeur doit être prédite et non classée), la mesure de performance utilisée est le RMSE (*Root Mean Square Error*).

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(X_i) - y_i)^2}$$

Avec :

X , la matrice d'entrée.

h , le modèle de prédiction.

m , le nombre d'entrée.

X_i , l'entrée à l'index i .

y_i , la sortie voulue pour l'entrée X_i .

De part son équation, le RMSE à l'avantage de donner une valeur dans la même unité que la sortie recherchée (ce qui permet de se visualiser facilement l'erreur).

6.6.3 Méthodes sans réseau de neurones

6.6.3.1 Optimisation d'un modèle

L'optimisation d'un modèle, se fait de manière générale en modifiant des paramètres qui lui sont propres. Cette optimisation pourrait se faire manuellement mais elle prendrait beaucoup trop de temps. A la place, *Scikit-Learn* met à disposition un outil qui permet de faire ça de manière automatique : *GridSearchCV*. En lui passant le modèle avec les paramètres qui doivent être réglé, il ressort les réglages optimaux du modèle.

6.6.3.2 Entrée des modèles

Les paramètres analysés dans la partie mesure (cf. chapitre 6.5.1) ont été mis en entrée des modèles. Les cannes d'arrosage du terrain ont été analysées deux fois (une fois avec toutes les données et une fois uniquement avec la partie injective de celles-ci).

6.6.3.3 Entraînement d'un modèle par la méthode des K plus proches voisins (KNN)

L'algorithme des K plus proches voisins est une méthode d'apprentissage supervisé qui consiste lorsqu'une entrée X est présentée, à prendre en compte ses K plus proches voisins connus pour définir la sortie Y.

6.6.3.3.1 Paramètres d'un KNN

Dans le but d'optimiser le modèle, deux paramètres ont été identifié :

- Le nombre de voisins à analyser.
- L'algorithme pour déterminer les voisins les plus proches.
 - *Ball_tree*.
 - *Kd_tree*.
 - *Brute*, une recherche par *brute-force*.

6.6.3.3.2 Résultats obtenus

Petit banc de test	
Regressor	KNN
Nb. Neighbors	7
Algorithm	ball_tree
RMSE [l/s] (TEST) :	0.06

Canne 1	
Regressor	KNN
Nb. Neighbors	4
Algorithm	ball_tree
RMSE [l/s] (TEST) :	1.83

Canne 2	
Regressor	KNN
Nb. Neighbors	1
Algorithm	ball_tree
RMSE [l/s] (TEST) :	1.01

Canne 1 (Injectif)	
Regressor	KNN
Nb. Neighbors	4
Algorithm	brute
RMSE [l/s] (TEST) :	1.83

Canne 2 (Injectif)	
Regressor	KNN
Nb. Neighbors	1
Algorithm	ball_tree
RMSE [l/s] (TEST) :	1.01

Canne 1 & 2 (Injectif)	
Regressor	KNN
Nb. Neighbors	3
Algorithm	brute
RMSE [l/s] (TEST) :	1.73

6.6.3.4 Entraînement d'un modèle de forêts aléatoire (Random Forest)

L'algorithme de forêts aléatoires est un outil de *machine learning* simple d'emploi qui produit en général de bons résultats sur les prédictions. Il s'agit d'un algorithme d'apprentissage supervisé qui peut être autant bien employé sur une classification que sur une régression.

Cet algorithme se présente sous la forme d'un ensemble d'arbres de décisions qui sont entraînés par une méthode de *bagging*. C'est-à-dire que le jeu de données est séparé en plusieurs parties de manière aléatoire et que pour chacune de ces parties, un arbre de décision est entraîné.

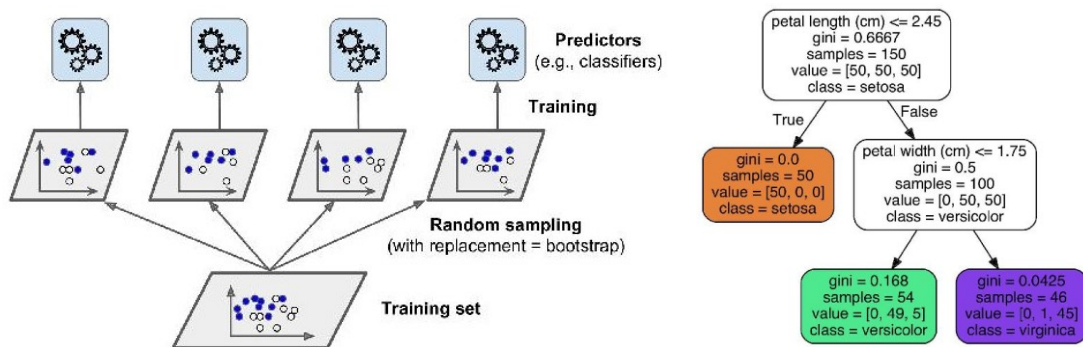


Figure 51 : Représentation du "bagging" & exemple d'un arbre de décision.

Source : A. Géron, *Deep Learning avec TensorFlow*, chap. 6 & 7

6.6.3.4.1 Paramètres d'une Random Forest

Dans le but d'optimiser le modèle, deux paramètres ont été identifiés :

- Le nombre de propriété par nœuds (*max_features*). Qui indique le nombre de propriétés qui sont étudiée sur chaque nœud.
- La profondeur maximale de l'arbre (*max_depth*).

6.6.3.4.2 Résultats obtenus

Petit banc de test	
Regressor	Random Forest
Max Features	7
Max depth	10
RMSE [l/s] (TEST) :	0.026

Canne 1	
Regressor	Random Forest
Max Features	10
Max depth	10
RMSE [l/s] (TEST) :	1.33

Canne 2	
Regressor	Random Forest
Max Features	5
Max depth	Pas de limite
RMSE [l/s] (TEST) :	0.67

Canne 1 (Injectif)	
Regressor	Random Forest
Max Features	4
Max depth	15
RMSE [l/s] (TEST) :	0.95

Canne 2 (Injectif)	
Regressor	Random Forest
Max Features	9
Max depth	10
RMSE [l/s] (TEST) :	0.44

Canne 1 & 2 (Injectif)	
Regressor	Random Forest
Max Features	9
Max depth	10
RMSE [l/s] (TEST) :	0.65

6.6.4 Méthode avec réseau de neurones artificiels

6.6.4.1 TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Son principe d'utilisation est simple : Il faut tout d'abord définir un graphique de calcul en Python, Java, C ou Go et par la suite, il prend se graphique et l'exécute de manière optimisée.

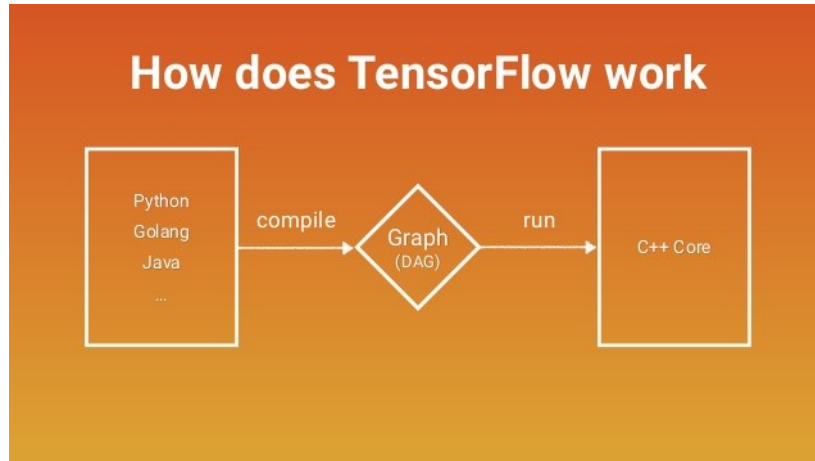


Figure 52 : Comment fonctionne TensorFlow ?
Source : www.tensorflow.org

Dans ce projet, TensorFlow a été utilisé pour la génération d'un réseau de neurone.

6.6.4.2 Introduction au réseau de neurones artificiels

Un réseau de neurone artificiel est un système dont la conception est inspirée du fonctionnement de neurones biologiques. Pour ce projet, le type de réseau de neurone qui a été implémenté est un réseau de neurone par Perceptron multicouche.

Celui-ci est composé d'une succession de couches dont les entrées de celles-ci sont les sorties de la couche précédente. Chaque entrée est associée à un poids qui va définir l'importance de ce paramètre dans le réseau. La fonction de combinaison définit la manière dont les différentes entrées seront regroupées en une seule fonction. Pour ce projet, la fonction de combinaison sera linéaire. La fonction d'activation permet de gérer l'activation du neurone.

Celle-ci présente généralement 3 intervalles :

- En dessous du seuil, le neurone est inactif.
- A proximité du seuil, une phase de transition est observable.
- Au-dessus du seuil, le neurone est actif.

Cette fonction d'activation peut prendre différentes formes en pratique.

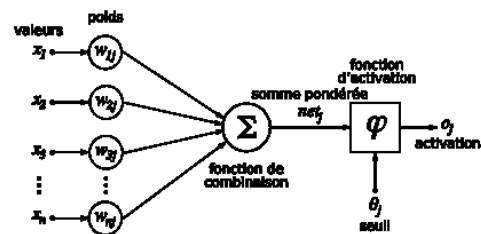


Figure 53 : Structure d'un neurone artificiel.
Source : www.wikipedia.org

6.6.4.2.1 Fonction d'activation Heaviside

$$f(x) = 1, \text{ pour } x \geq 0$$

Il s'agit de la fonction d'activation la plus simple possible. Elle peut être particulièrement utile pour la création d'un classificateur binaire (uniquement deux classes). Cette fonction d'activation donne uniquement deux états possibles : neurone actif ou non.

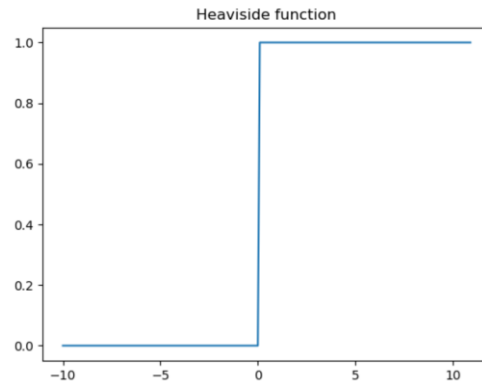


Figure 54 : Fonction Heaviside.

6.6.4.2.2 Fonction d'activation ReLU

La fonction d'activation ReLU (Rectified Linear Unit) est certainement la plus largement utilisée. Elle se présente sous la forme :

$$f(x) = \max(0, x)$$

L'avantage principal d'une activation par ReLU est qu'elle n'active pas tous les neurones en même temps. Car si l'entrée du neurone est négative, celui-ci ne sera pas activé. Ce qui rend le réseau efficace et facile à calculer.

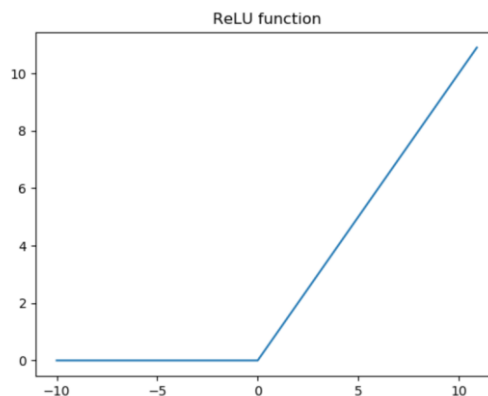


Figure 55 : Fonction ReLU.

Le principal désavantage d'un ReLU est que son gradient (= dérivée de la courbe) peut valoir 0, ce qui signifie que dans ces cas les poids ne sont pas mis à jour lors de la propagation arrière et ça peut donc créer des neurones « morts ». Pour résoudre ce problème, une version améliorée de la fonction ReLU existe : le Leaky ReLU qui rajoute une pente infime sur la partie négative de l'axe des X.

6.6.4.2.3 Fonction d'activation sigmoïde

La fonction sigmoïde est une fonction d'activation largement utilisée. Elle se présente sous la forme :

$$f(x) = \frac{1}{1 + e^{-x}}$$

Son plus grand avantage est qu'elle est non-linéaire. Cela signifie que lorsque plusieurs neurones ont une fonction sigmoïde comme fonction d'activation, la sortie est également non linéaire. De plus elle a l'avantage de proposer des valeurs comprises entre 0 et 1 ce qui peut être très pratique dans le cas où une probabilité doit être prédite.

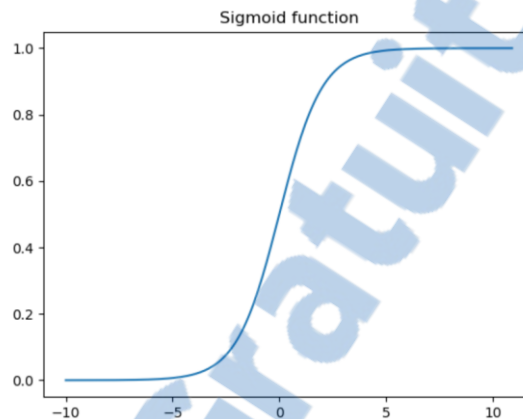


Figure 56 : Fonction sigmoïde.

6.6.4.2.4 Apprentissage d'un réseau de neurones Perceptron multicouche

Pour l'apprentissage d'un réseau de neurones, il est fortement conseillé de standardiser chaque paramètre (suppression de la moyenne et mise à l'échelle) pour qu'ils soient tous sur la même plage de valeur.

L'apprentissage d'un réseau de neurones se passe de la manière suivante :

1. Insertion d'un jeu de données d'entraînement à l'entrée de ce réseau.
2. Comparaison entre la sortie réelle et la sortie souhaitée.
3. Calcul de l'erreur de sortie sur chacun des neurones du réseau.
4. Calcul pour chacun des neurones de la valeur de sortie correcte.
5. Ajustement du poids de toutes les connexions.
6. Attribution d'un blâme à tous les neurones précédents.
7. Recommencer à partir de l'étape 4, sur les neurones précédents un utilisant le blâme comme erreur.

6.6.4.2.5 Ajustement du poids de toutes les connexions

L'ajustement du poids est régi par l'équation suivante :

$$W'_i = W_i + \alpha(Y_t - Y)X_i$$

Avec :

W'_i , le poids corrigé.

W_i , le poids actuel.

α , le taux d'apprentissage.

Y_t , la sortie attendue.

Y , la sortie obtenue.

X_i , l'entrée pour la sortie attendue.

6.6.4.2.6 Explication sur le taux d'apprentissage

Le taux d'apprentissage est un paramètre de la fonction de coût du modèle qui a pour but d'être minimisée pour obtenir le modèle le plus fiable possible.

Une explication simple du taux d'apprentissage serait : « La taille de saut minimum effectuée sur l'optimisation des poids du réseau ». Ce qui signifie qu'un taux d'apprentissage trop faible donnerait un apprentissage trop long pour le modèle alors qu'un taux trop grand ferait diverger le modèle.

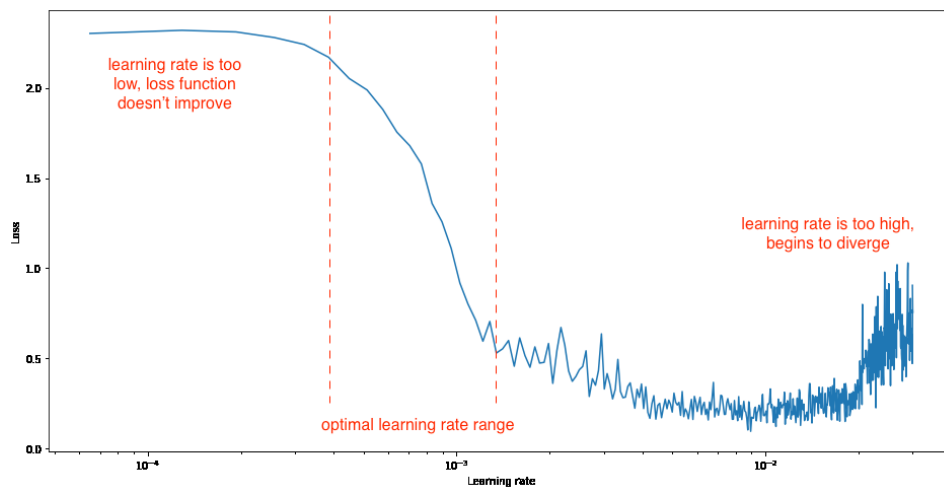


Figure 57 : Choix d'un taux d'apprentissage. Source : www.jeremyjordan.me

6.6.4.3 Mise en place d'un réseau de Perceptron multicouche pour la détection du débit

Etant donné les bons résultats obtenus avec des outils de *machine learning* classique, la décision a été prise de modifier les données d'entrée par rapport à la méthode précédente. Plutôt que de donner en entrée l'écart-type, l'offset, l'amplitude et l'amplitude de la première harmonique, il a été décidé de donner en entrée les 30 premiers éléments du spectre du signal pour les 3 axes.

Ce changement a été motivé par le fait que lorsqu'on entraîne un réseau de neurone profond (réseau de neurone avec plus d'une couche intermédiaire), l'abstraction des données d'entrée est généralement plus grande. Dans le but de pouvoir réaliser une véritable comparaison entre ces deux méthodes. Un réseau de neurone a également été effectué avec les mêmes entrées qu'une méthode classique.

6.6.4.3.1 Résultats avec le spectre du signal en entrée

Résultats du petit banc de test (Spectre)				
Learning Rate	0.001	0.001	0.001	0.001
Nb. Neurones couche 1	60	60	70	70
Nb. Neurones couche 2	30	30	50	50
Nb. Neurones couche 3			30	30
RMSE [I/s] (TEST)	0.1093	0.0610	0.1005	0.0605
Fonction d'activation	ReLU	Sigmoïde	ReLU	Sigmoïde

Résultats de la canne numéro 1 (Spectre)				
Learning Rate	0.001	0.001	0.001	0.001
Nb. Neurones couche 1	60	60	70	70
Nb. Neurones couche 2	30	30	50	50
Nb. Neurones couche 3			30	30
RMSE [I/s] (TEST)	11.14	8.24	10.62	8.23
Fonction d'activation	ReLU	Sigmoïde	ReLU	Sigmoïde

Résultats de la canne numéro 2 (Spectre)				
Learning Rate	0.001	0.001	0.001	0.001
Nb. Neurones couche 1	60	60	70	70
Nb. Neurones couche 2	30	30	50	50
Nb. Neurones couche 3			30	30
RMSE [I/s] (TEST)	4.413	3.464	4.35	3.462
Fonction d'activation	ReLU	Sigmoïde	ReLU	Sigmoïde

Résultats des deux cannes (Spectre)				
Learning Rate	0.001	0.001	0.001	0.001
Nb. Neurones couche 1	60	60	70	70
Nb. Neurones couche 2	30	30	50	50
Nb. Neurones couche 3			30	30
RMSE [I/s] (TEST)	8.84	7.65	8.74	7.62
Fonction d'activation	ReLU	Sigmoïde	ReLU	Sigmoïde

6.6.4.3.2 Résultats avec les mêmes entrées qu'une méthode classique

Résultats du petit banc de test		
Learning Rate	0.001	0.001
Nb. Neurones couche 1	6	6
RMSE [I/s] (TEST)	0.1188	0.1150
Fonction d'activation	ReLU	Sigmoïde

Résultats de la canne 1 (Partie injective)		
Learning Rate	0.001	0.001
Nb. Neurones couche 1	6	6
RMSE [I/s] (TEST)	9.01	8.27
Fonction d'activation	ReLU	Sigmoïde

Résultats de la canne 2 (Partie injective)		
Learning Rate	0.001	0.001
Nb. Neurones couche 1	6	6
RMSE [I/s] (TEST)	4.62	3.60
Fonction d'activation	ReLU	Sigmoïde

Résultats des deux cannes (Partie injective)		
Learning Rate	0.001	0.001
Nb. Neurones couche 1	6	6
RMSE [I/s] (TEST)	8.3	7.5300
Fonction d'activation	ReLU	Sigmoïde

6.6.5 Discussion des résultats

Comme suspecté à la fin de la prise de mesures, la perturbation causée par la vanne, a donné du fil à retordre aux outils de *machine learning* pour passer outre celle-ci. Cependant, le résultat du petit banc de test est très encourageant et indique qu'il est théoriquement possible de déterminer un débit en fonction des vibrations.

D'un point de vue purement technique, il est intéressant de constater que des algorithmes de *machine learning* classique sont beaucoup plus performant qu'un réseau de neurones dans ce cas. Cela peut naturellement s'expliquer par le fait qu'un réseau de neurone aura besoin de manière général de beaucoup plus de données d'entraînement qu'un algorithme classique d'apprentissage automatique et que dans ce cas le nombre de données n'était certainement pas suffisant pour entraîner de manière optimum un réseau.

Ce nombre de données était encore moins suffisant pour tous les modèles injectifs des mesures du terrain ont été prises. Vu que en rendant les caractéristiques d'entrée injective, environ la moitié du jeu de donnée a été supprimée.

De plus pour la partie non-neuronale, une forêt aléatoire donne dans tous les cas une meilleure précision qu'une régression par KNN. Cela s'explique par le fait que de nos jours l'algorithme *Random Forest* et certainement l'algorithme d'apprentissage automatique le plus performant et surtout le plus polyvalent.

6.7 AMÉLIORATIONS POSSIBLES

Bien que fonctionnel sur le petit banc de test, ce projet aurait besoin de quelques améliorations pour fonctionner correctement sur le terrain. La tâche principale à effectuée serait de passer outre la perturbation causée par la vanne. Pour passer outre cette perturbation plusieurs améliorations ont été imaginées :

- La solution la plus simple serait de filtrer cette perturbation en rajoutant de la distance entre la vanne et le capteur. Le problème est qu'une distance convenable se trouverait certainement à un endroit où la conduite est sous terre ce qui isolerait bien le capteur de l'environnement extérieur mais rajouterait des problèmes liés à l'environnement (humidité, transmission des données, recharge de l'appareil, ...).
- Une autre solution qui pourrait être imaginée serait de garder le capteur au même endroit mais de rajouter un élément permettant de définir l'ouverture de la vanne. De cette manière la perturbation pourrait être gardée vu qu'une information nous donnerait directement de quel côté de celle-ci le débit se trouve.
- A l'heure actuelle, le capteur de température n'a pas été employé dans le modèle d'apprentissage automatique. Son but premier serait d'aider à déterminer si une vibration captée provient ou non d'un débit. Il n'a pas été employé car pour intégrer cette tâche, il faudrait avoir une série de mesures à 0 [l/s] où l'évolution de la température pourrait être constatée. Ce qui n'a pas été fait.

6.8 CONCLUSION DE LA TÂCHE

Ainsi se termine la tâche principale de ce travail de Bachelor qui consistait à prédire un débit par les vibrations qui sont créées par celui-ci sur un tuyau.

Le choix a été porté dès le début sur une résolution par des outils de *machine learning* qui à la vue de la complexité du modèle et des différentes perturbations, et au nombre de ces caractéristiques s'est avéré être une bonne idée.

Bien qu'à l'heure actuelle le prototype réalisé ne puisse pas fonctionner sur le terrain, celui-ci a montré des résultats excellents sur un flux d'eau laminaire (ou faiblement perturbé, petit banc de test). Ce qui laisse un champ d'application ouvert (bien que très petit) pour ce procédé de mesure du débit.

Il pourrait très bien être installé sur des écoulements libres tel que les tuyaux de captation de sources pour l'eau potable où même dans certaines usines où les débitmètres sont généralement présents en nombre pour tous les appareils hydrauliques.

L'avantage principal de ce capteur est son coût très faible (pour rappel, tout le prototype coûte moins de 50.-) qui ensemble avec une précision acceptable, arrivera très certainement à intéresser du monde dans des domaines très spécifiques.

7 BIBLIOGRAPHIE

7.1 MACHINE LEARNING AVEC SCIKIT-LEARN

ISBN : 978-2-10-076540-9

Auteur : Aurélien Géron Maison d'édition : O'REILLY

Explication des concepts fondamentaux du Machine Learning et apprentissage des outils permettant l'implémentation d'algorithmes de Machine Learning.

7.2 DEEP LEARNING AVEC TENSORFLOW

ISBN : 978-2-10-075993-4

Auteur : Aurélien Géron Maison d'édition : O'REILLY

Initiation au Deep Learning avec l'outil TensorFlow de Google. Guide avancé du Machine Learning.

7.3 HANDS-ON DATA SCIENCE AND PYTHON MACHINE LEARNING

ISBN : 978-1-78-728074-8

Auteur : Frank Kane Maison d'édition : Packt

Couvre les fondamentaux du Machine Learning avec une introduction sur Python et un point de vue plus mathématique que les deux précédents livres.

8 REMERCIEMENTS

Je souhaite exprimer ma gratitude aux personnes qui m'ont soutenu dans la réalisation de ce travail de diplôme :

- Monsieur Pierre-André Mudry, professeur à la HES-SO Valais, pour le suivi du projet et ses différents conseils techniques.
- Monsieur César Papilloud, assistant à la HES-SO Valais, pour son aide concernant les prises de mesures et ses conseils.
- Monsieur Vlad Hasmatuchi, professeur à la HES-SO Valais, pour la mise à disposition du laboratoire d'hydraulique.

Je remercie également toutes les personnes qui m'ont encouragé, de près ou de loin, tout au long de ce travail.

9 CONCLUSION PERSONNELLE

Ainsi se termine ces 14 semaines de travail de Bachelor, tout a commencé par l'étude du *machine learning* avec une petite mise en pratique pour par la suite étudier une idée un peu folle au début de ce projet : « *Ecouter* » un tuyau pour déterminer le débit qui le traverse.

C'est le fait de partir d'une idée et d'arriver à quelque chose de concret qui a été particulièrement intéressant dans ce projet, de plus avoir eu l'opportunité de se rendre sur le terrain pour constater les différences entre le laboratoire et celui-ci a été très enrichissant et m'a permis de me rendre compte de l'immense fossé qui sépare ces deux mondes.

J'étais au début inquiet par le fait d'apprendre le *machine learning* pour ce travail. Mais j'ai été extrêmement intéressé par ce domaine de l'informatique qui m'était encore inconnu au moment de commencer ce travail.

D'un point de vue purement personnel, je suis satisfait par ce travail de Bachelor qui m'a énormément intéressé, et dans le but de conclure sur une note un peu plus technique, il peut être dit que dans sa version actuelle le prototype démontre qu'il est possible de prédire un débit en se basant sur les vibrations d'un tuyau, mais que du travail est encore nécessaire pour une application sur le terrain.

Sion, le 3 septembre 2018

Jérémy Maceiras

10 TABLE DES FIGURES

Figure 1 : Evolution du Machine Learning.....	3
Figure 2 : Un exemple visuel de classification. Des arbres sont triés par deux paramètres, la taille de leurs feuilles et le nombre de feuilles par brindille. Une fois le modèle de classification entraîné, l'algorithme peut classer un nouveau jeu de donnée dans une des 3 espèces connues par le modèle. Source : http://nanotechweb.org/cws/article/lab/46619	7
Figure 3 : Illustration du problème.	8
Figure 4 : Activité de l'application de mesure.....	9
Figure 5 : Extrait du fichier CSV généré.....	9
Figure 6 : Linéarisation dans une salle de classe.	11
Figure 7 : Linéarisation sur des longues distances.	11
Figure 8 : Atténuation d'un signal Wi-Fi dans l'air.	11
Figure 9 : RSSI en fonction des adresses MAC. Chaque couleur représente une salle de classe.....	12
Figure 10 : Adresses MAC disponibles en fonction de la salle de classe.	13
Figure 11 : RSSI filtré en fonction des adresses MAC.	16
Figure 12 : Comparaison de la précision des différents algorithmes de classification.	17
Figure 13 : Matrice de confusion.	18
Figure 14 : Canne d'arrosage dans la nature.....	20
Figure 15 : Schéma de principe.	21
Figure 16 : Schéma du banc de test.	23
Figure 17 : Schéma du grand banc de test.	23
Figure 18 : GUI de l'application.....	24
Figure 19 : Schéma bloc de l'application.	24
Figure 20 : Résultat temporel des mesures.	25
Figure 21 : Analyse des premières mesures.....	25
Figure 22 : Deuxième prototype.	26
Figure 23 : TMP006 avec sa carte d'adaptation.....	27
Figure 24 : MMA8451 avec sa carte d'adaptation.....	27
Figure 25 : Carte de développement Arduino MKR Zero.	28
Figure 26 : Schéma du PCB de raccordement.....	28
Figure 27 : Présentation du support de fixation.....	28
Figure 28 : Diagramme physique du capteur.	29
Figure 29 : Machine d'état du capteur.....	29
Figure 30 : Format de réponse du capteur.....	30
Figure 31 : Diagramme de classe de l'application embarquée.....	30
Figure 32 : Diagramme de flux du script d'acquisition.	31
Figure 33 : Chemin d'exportation des fichiers de sortie.	31
Figure 34 : Timer à 1 [khz].....	32
Figure 35 : Commandes de base du capteur.....	32
Figure 36 : Exécution d'une commande START.	33
Figure 37 : Nomenclature pour les résultats.	33
Figure 38 : Aperçu des fichiers du résultat.....	33
Figure 39 : Test du prototype sur le terrain.	34
Figure 40 : Interface graphique de l'application d'acquisition.	35
Figure 41 : Avertissement de saturation du capteur.....	35
Figure 42 : Position des axes de l'accéléromètre.....	36
Figure 43 : Résultats du script de visualisation sur le petit banc de test.	36
Figure 44 : Système monté sur le petit banc de test.	37

Figure 45 : Résultats du script de visualisation sur le grand banc de test.	37
Figure 46 : Montage du prototype sur une canne d'arrosage.	38
Figure 47 : Résultats du script de visualisation sur la canne d'arrosage n°1.	39
Figure 48 : Résultats du script de visualisation sur la canne d'arrosage n°2.	40
Figure 49 : Comparaison entre les deux cannes testées sur le terrain.	41
Figure 50 : Analyse des données du petit banc de test.	42
Figure 51 : Représentation du "bagging" & exemple d'un arbre de décision. Source : A. Géron, Deep Learning avec TensorFlow, chap. 6 & 7	44
Figure 52 : Comment fonctionne TensorFlow ? Source : www.tensorflow.org	45
Figure 53 : Structure d'un neurone artificiel. Source : www.wikipedia.org	45
Figure 54 : Fonction Heaviside.	46
Figure 55 : Fonction ReLU.	46
Figure 56 : Fonction sigmoïde.	47
Figure 57 : Choix d'un taux d'apprentissage. Source : www.jeremyjordan.me	48