Cette réunion a pour but de permettre à tous les membres du projet de connaitre les spécificités et fonctionnalités implémenté par chaque membre.
Cela va nous permettre de lever les incompréhensions.

Présentation :

Le projet est articulé en sous partie : une pour chaque datasets, ils possédent une architectures et des fichiers très similaires. La principal différence vient dans le pre-processing.
Le projet se présente sous beaucoup d'orienté objet (classe , objet python)
Les fonctions définits dans les fichiers unique sont ré-importé pour utilisation dans les classes. Les readme sont fait pour les classes. Ces classes permettent notamment d'importer les données en choisissant le type de dataset, la taille importé, l'encoding , le scaling.
Le fichier scripts contient les scripts : model, test et notamment le logger.
Le logger sert ici en particulier de logger mais il permert de sauvegarder dans un fichier à l'exécution les résultats. Il facilite également l'écriture de messages pour le debug, warning, error. On hiérarchise ainsi les messages et de trier les infos qui seront affiché ou non. Ici les messages de debug ne sont pas affiché. Pour la livraison on affichera à partir de "Warning" et "Error".
Dossier Analysis : Contient les codes d'analyses de résultat (Matrice de confusion , tableau de performances,etc). Les codes sont adaptés à tout type de modèles de prédiction.
Dossier models : les différents models de prédictions et leur implémentation, fonction train etc.
Le dossier pytorch pour les truc utilitaires en terme d'amélioration et d'optimisation.
Pour les test nous nous concentrerons sur les MLP.

Présentation des types d'attaques implémentées :

Attaque FGSM : Application de la loss entre la prédiction de la classe voulu, backward (calcul du gradient) pour savoir dans quel direction on modifis l'entrée pour que la prédiction soit cell voulu. On recherche ici une direction de déplacement pour atteindre la frontière de la nouvelle classe.
L'attaque se fait par itération, on décale de eps dans une direction jusqu'a atteindre la prédiction voulu
Amélioration : pouvoir connaitre le déplacement minimum pour chaque feature.
Attaque utilisé notamment pour l'attaque surrogate et substitut.

Attaque HopSkipJump: Attaque en "boite noire". Marche sur les modèles non différentiables. Utilise un algo pour estimer de manière non biaisé un "gradient" (ici direction vers la frontière de décisions). Pas trop concluant pour l'instant, on est à 8% de sucess rate (nombre d'échantillon faux négatif).

En terme de métrique d’analyse, on veut maximiser les faux négatifs. On ne modifera pas les attaques benigne.

Faisons des sous-datasets pour chaque datasets contenant surtout des attaques pour faciliter les tests.

Discussion sur le KNN: