

Détection d'activité non intrusive à partir de la courbe de charge

Pierre Dal Bianco
Paulin Kimpa
Pierrick Leroy
Julien Lair
Hugo Michel



Sommaire

- 1. Contexte et enjeux**
- 2. Présentation des données et labellisation**
- 3. Overview des approches**
- 4. Description de la pipeline**
- 5. Zoom sur les modèles**
 - a. Approches Supervisées**
 - b. Approche non supervisées**
- 6. Synthèse des performances**
- 7. Discussion des approches**
- 8. Les prochaines étapes : La parole à Baalbek**

Contexte et enjeux

Contexte et enjeux

Déploiement à grande échelle des compteurs intelligents Linky

- Accès sécurisé aux données de consommation d'énergie
- Meilleur gestion de l'énergie
- Nouveaux services aux usagers
- Sécurisation des logements notamment les logements secondaires ou les logements mis en location temporaire
- Prolonger l'autonomie des senior en leur permettant un maintien à domicile, en alertant les proches ou le personnel médical en cas d'absence anormal d'activité
- Travail sur des méthodes non-intrusives de détection d'activité à partir des courbes de charge de consommation électrique

Motivations

Les systèmes de détection d'activité actuelles utilisent une série de capteurs :

- Caméras, détecteur de mouvement, interrupteurs magnétiques, capteurs magnétiques....

OR ces systèmes sont :

- Très Intrusifs (par forcément accepté par le public)
- Lourd à installer
- Difficile à entretenir (changer les piles régulièrement)
- Pas très fiable dû à un manque de maintenance
- Très onéreux
- Courbe de charges agrégées : aucune information sur la consommation



=> Toutes ces raisons constituent les principales motivations de l'introduction des technologies de détection d'activité non intrusive tel que le NILM

Moyens

Moyens nécessaires

Découverte (phase 1) : formation de deux sous équipes

- 3 personnes : l'état de l'art
- 2 personnes : étiquetage des jeux de données (activité ou non)

Projet (phase 2) :

- répartition des différents blocs de travail
- mise en commun itérative

Encadrement du projet :

- Data Scientists de l'entreprise pour aiguiller nos choix / orienter pour répondre au besoin
- points d'avancement bi-mensuels avec l'entreprise

L'objectif du NILM

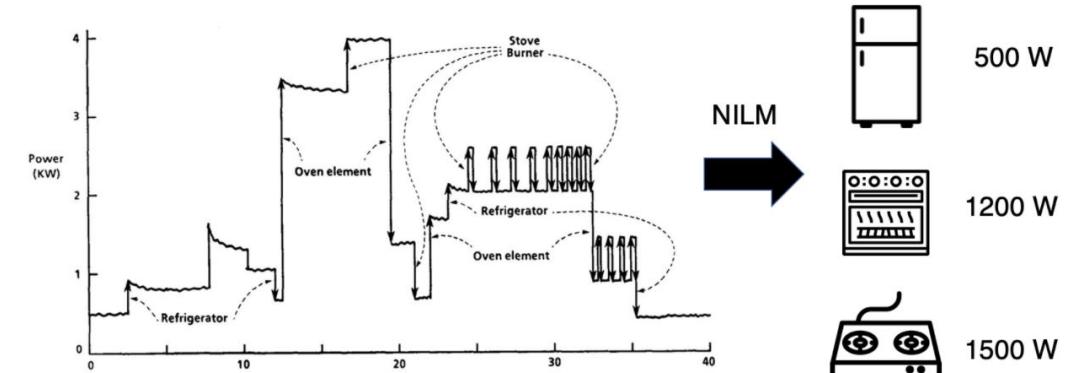
Le **NILM** comprend un ensemble de techniques pour la surveillance non intrusive de la charge.

Méthodes NILM :

- **Activity Detection**
- **Activity Modeling**

Activity Detection

- Déterminer les changements de tension et de courant entrant dans une maison et de déduire quels appareils sont utilisés dans la maison.
- S'appuient sur la détection d'événements, par exemple
 - Changement d'état de fonctionnement => ON/OFF



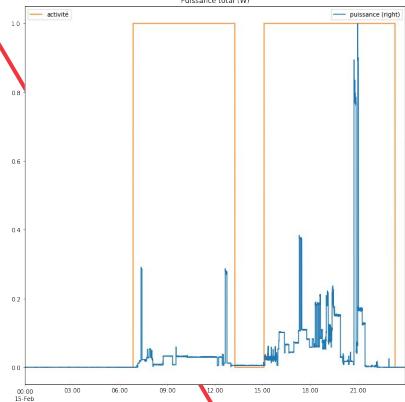
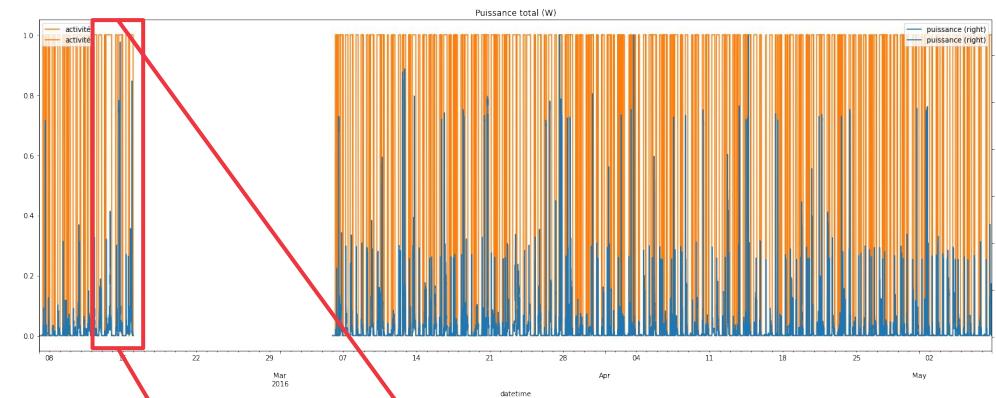
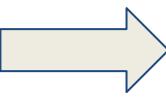
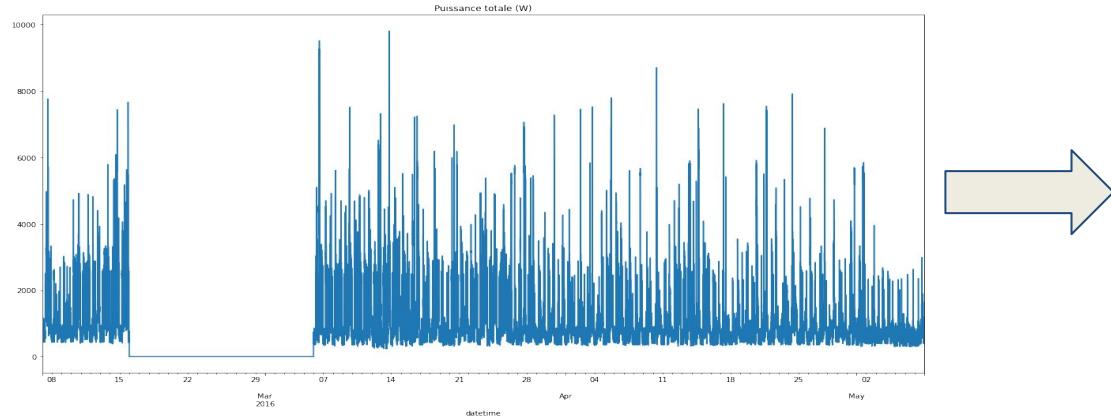
Méthodologie

1. Labellisation
2. Evaluation
3. Approches et pipeline

Objectifs poursuivis

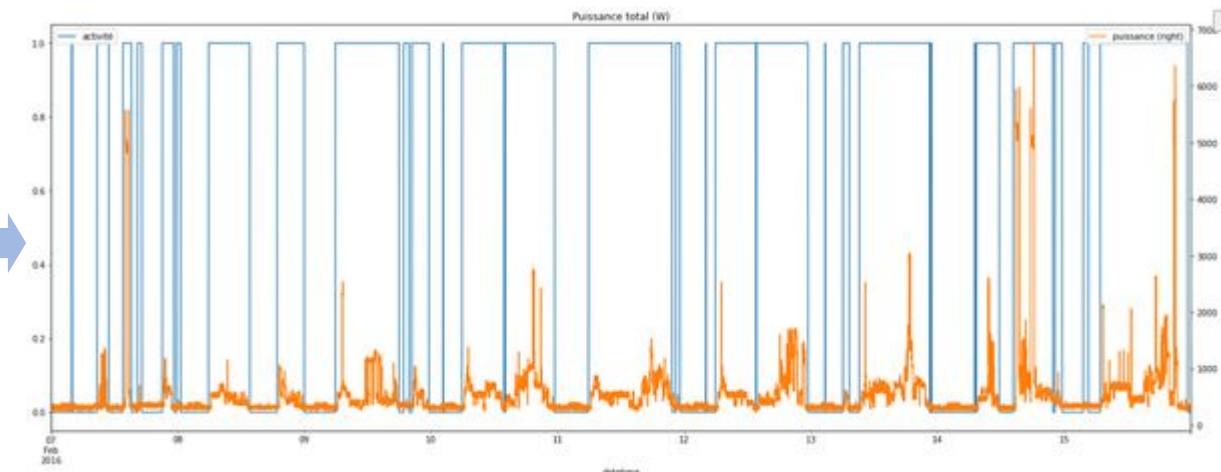
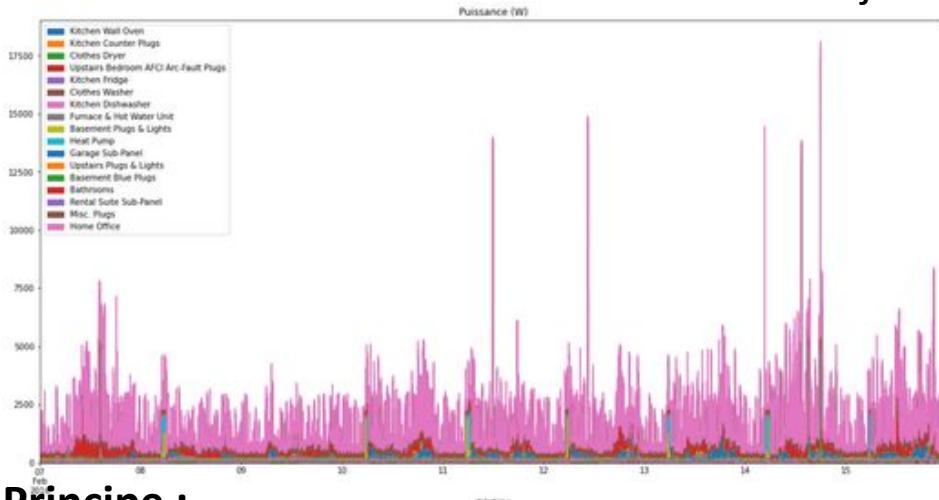
Détection de l'activité à partir de la mesure instantanée de la puissance électrique d'un logement :

- **Entrée** : courbe de charge agrégée, date/heure, [variables exogènes (semaine / week-end)]
- **Sortie** : classification binaire (activité ou pas)



Labellisation du jeu de données

- Jeux de données : *RAE - The Rainforest Automation Energy Dataset (Makonin & Stephen, 2017)*
 - Relevés de puissance par sous-compteur (1Hz) sur 2 logement pendant environ 60 jours
 - Données non labellisées => à rajouter en pré-traitement



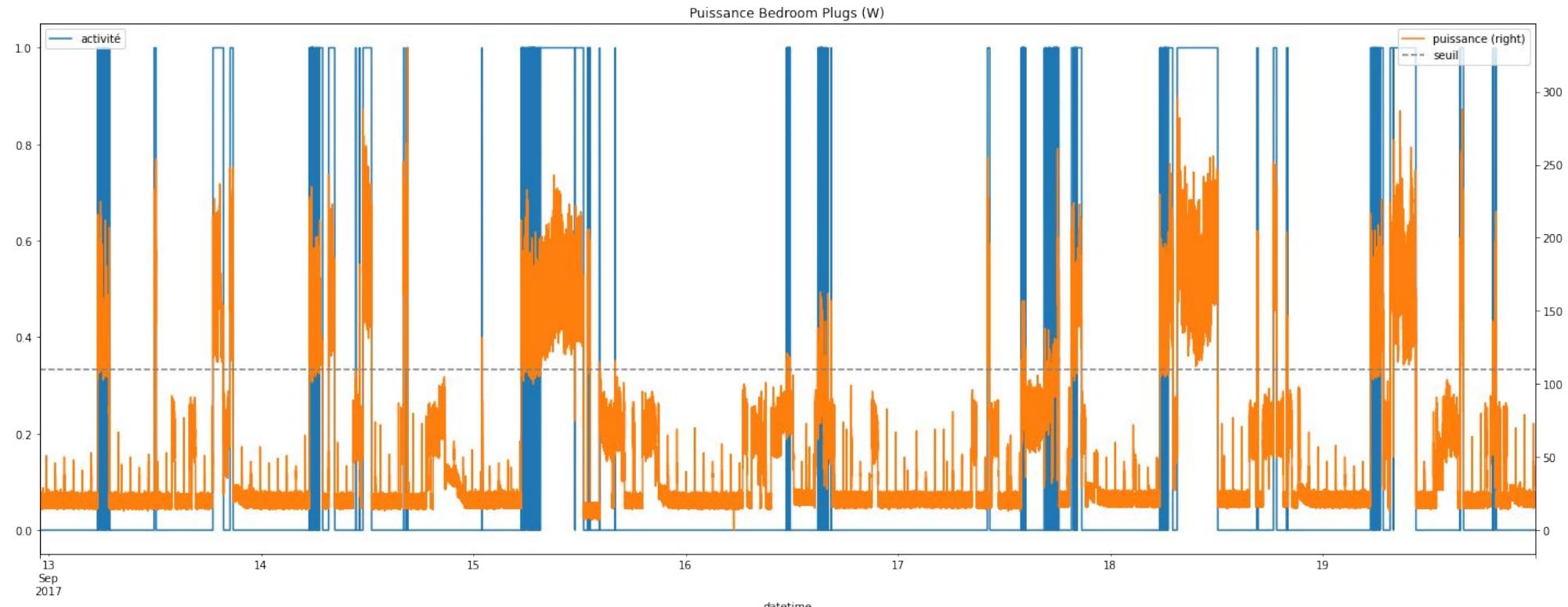
Principe :

1. Création d'un label activité par courbe de charge désagrégée
 - 1.1. Seuil
 - 1.2. Lissage
 - 1.3. Seuil
2. Sélection des courbes pertinentes
3. Agrégation pour obtenir la courbe finale

Labellisation du jeu de données

Etape 1.1 :

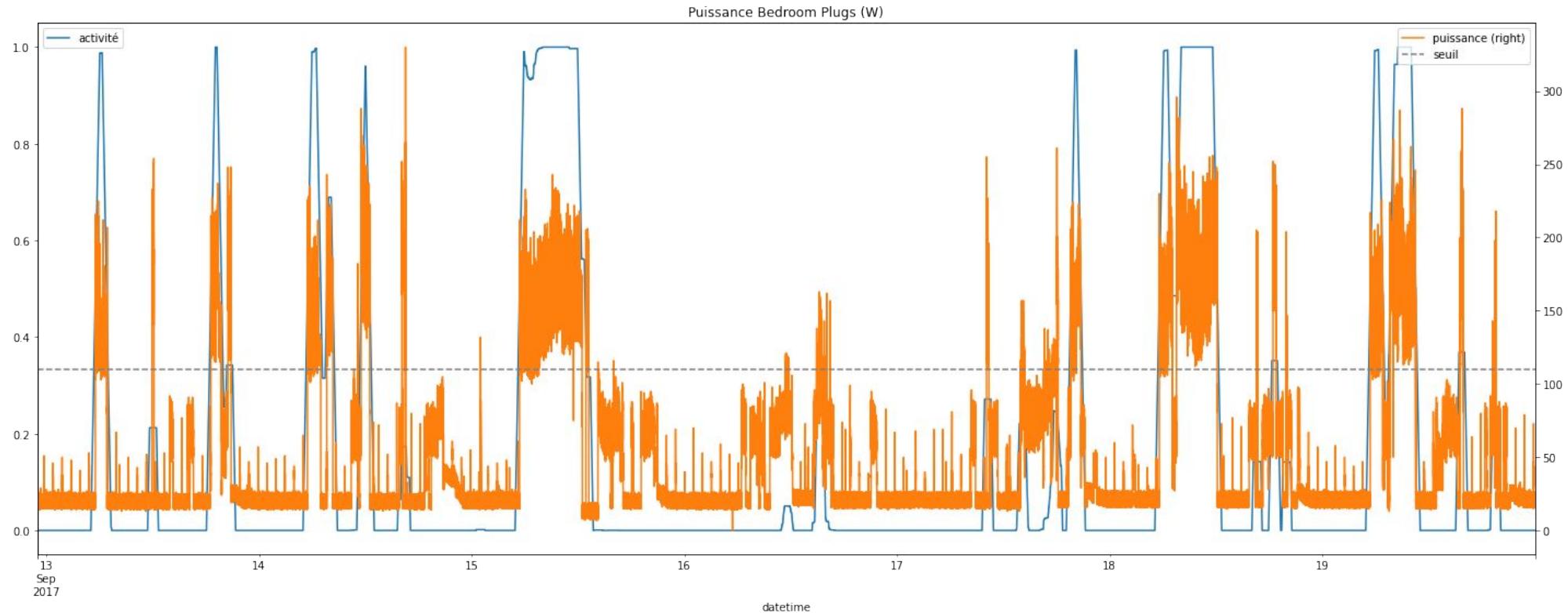
Seuil à la moyenne + écart-type x coefficient



Labellisation du jeu de données

Etape 1.2 :

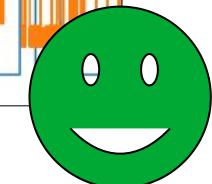
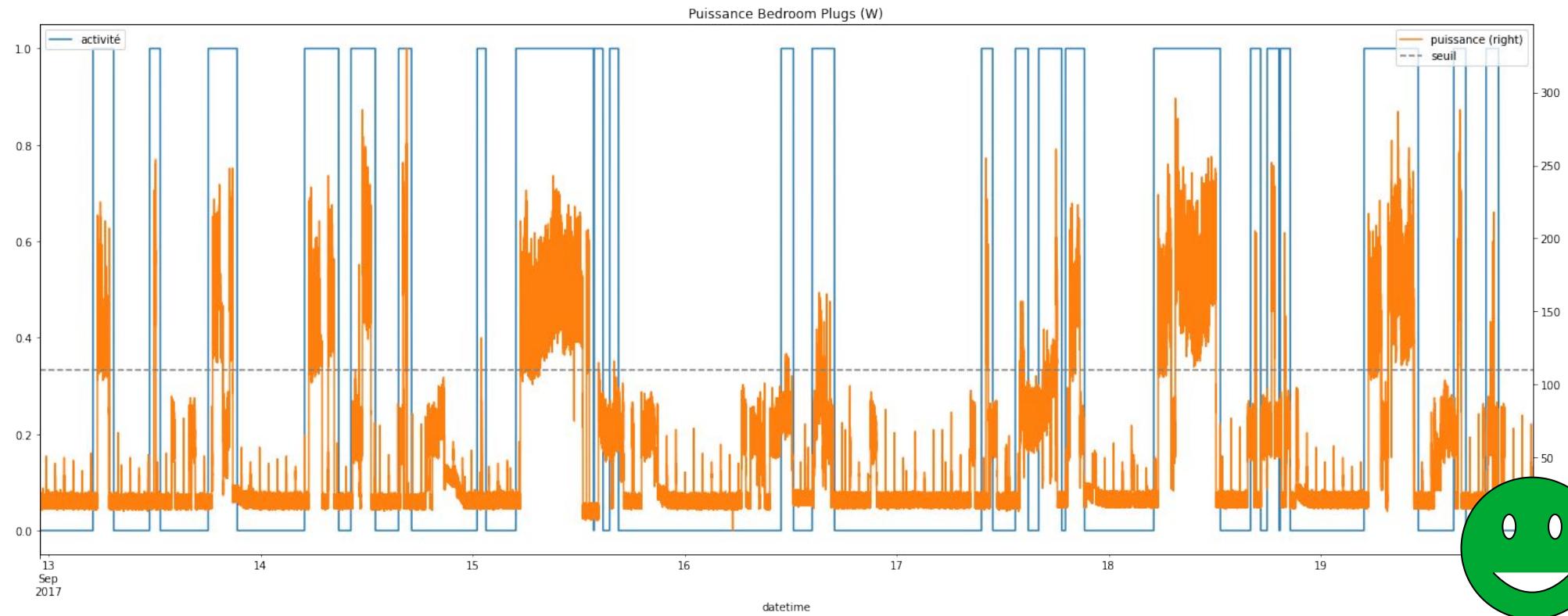
Lissage de la courbe activité (moyenne glissante, fenêtre à définir)



Labellisation du jeu de données

Etape 1.3 :

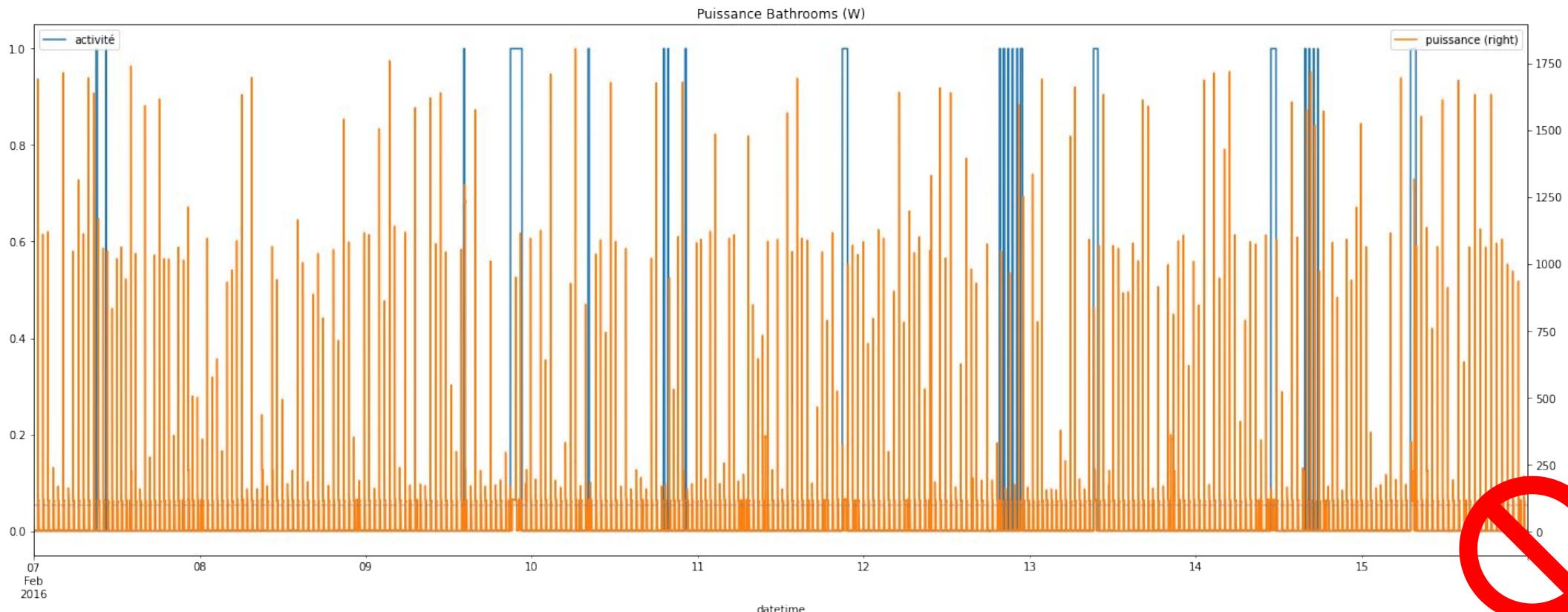
Seuil de la courbe activité



Labellisation du jeu de données

Etape 2 :

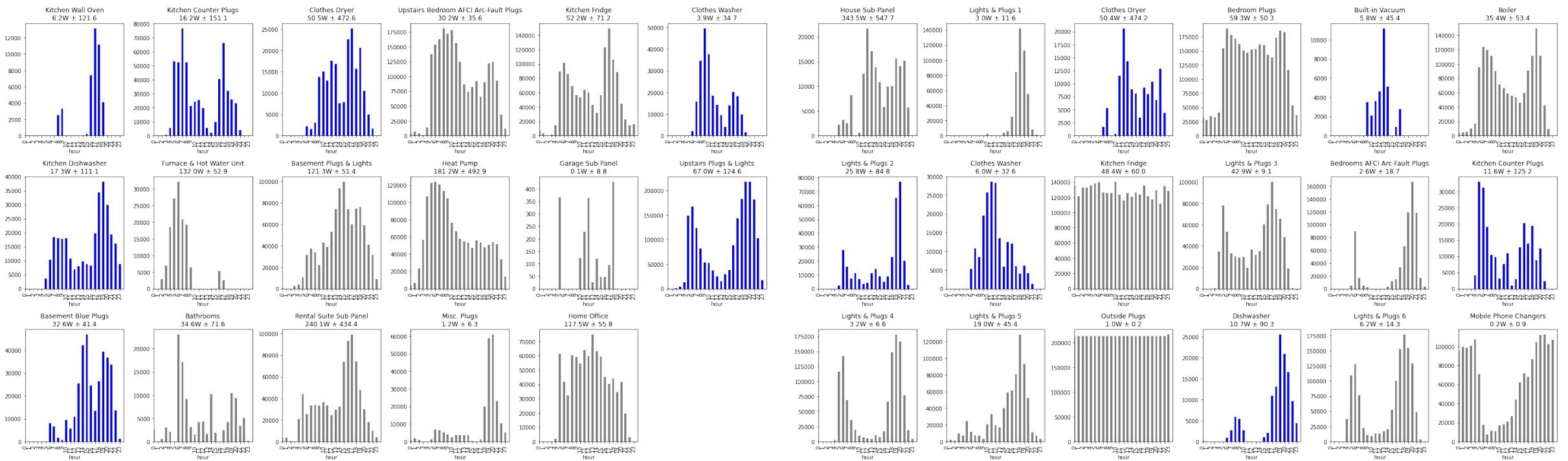
Sélection des courbes pertinentes (pas de signal faible ou bruité)



Labellisation du jeu de données

Etape 3 :

Agrégation sur toutes les courbes



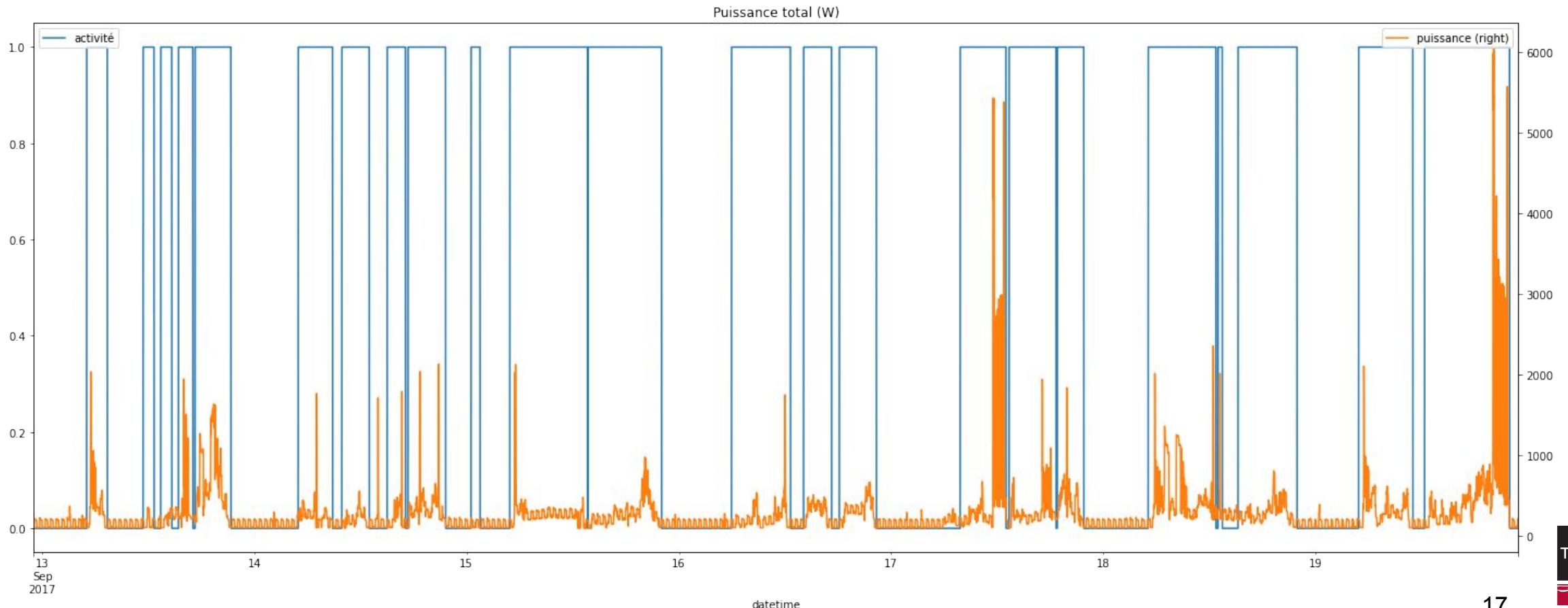
maison 1

maison 2

Labellisation du jeu de données

Etape 3 :

Agrégation sur toutes les courbes

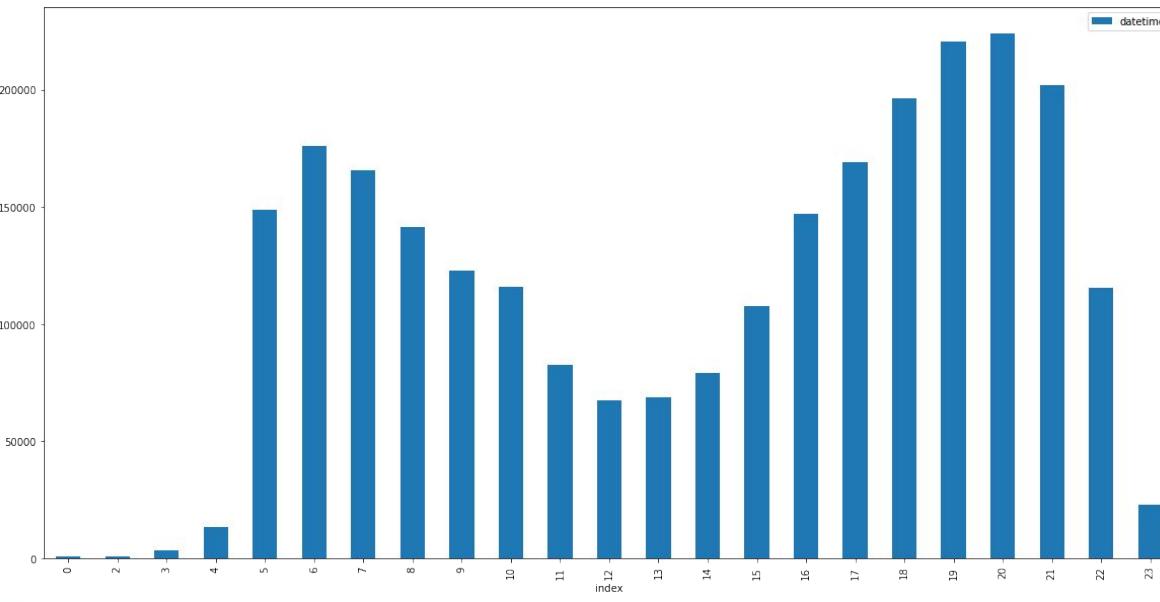


Labellisation : résultats et conclusion

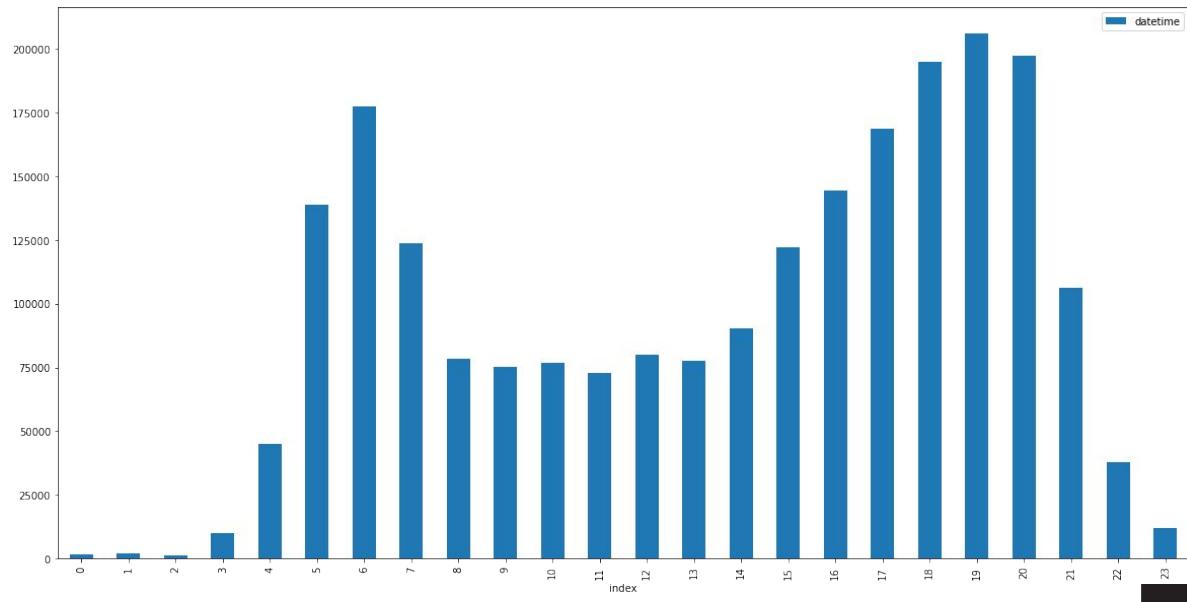
Résultats :

Observation des histogrammes d'activité par tranches horaires :

- pics d'activité vers 6h et 19h
- pas d'activité la nuit entre 23h et 4h du matin
- pas de jeux de données labellisées : meilleure approximation possible



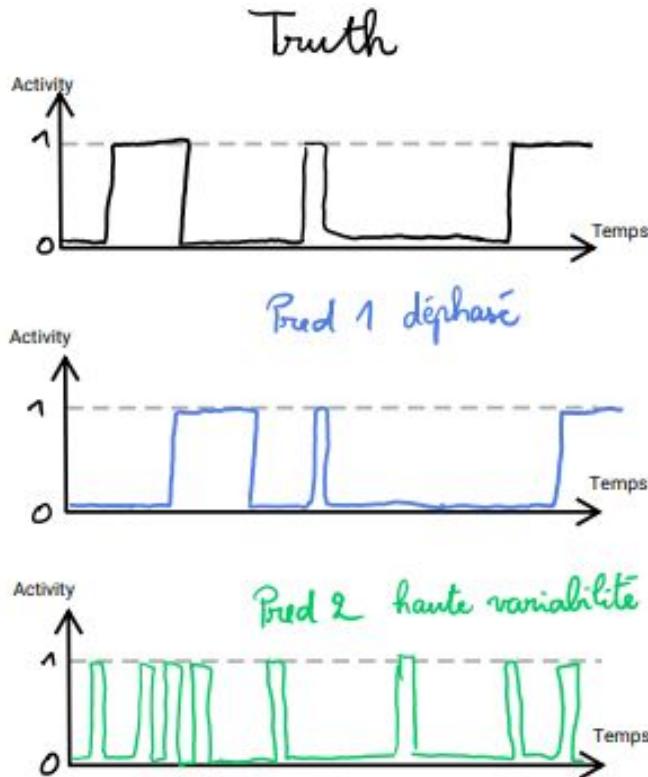
maison 1



maison 2

Evaluation

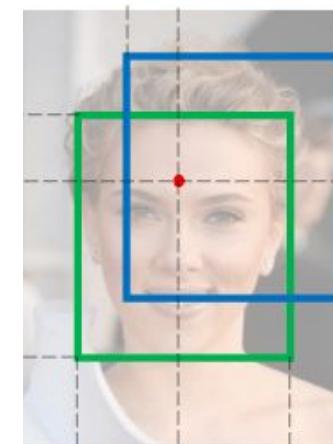
Evaluation du modèle



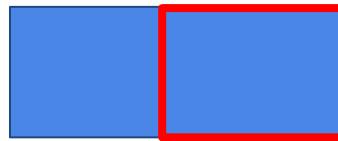
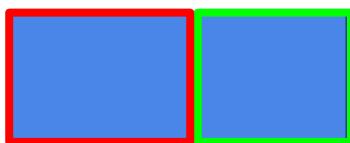
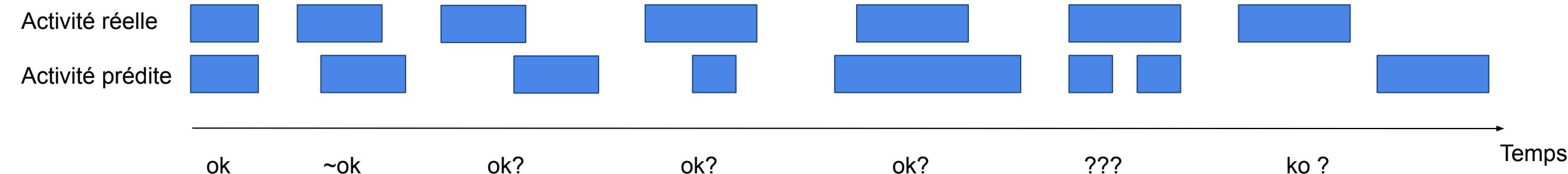
Idée 1 : classique

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Idée 2 : + adaptée
à notre tâche
Vision période



Evaluation du modèle



Principe d'évaluation : $mAP(\tau) = \frac{valid_{pred}(\tau)}{total_{pred}}$ $mAR(\tau) = \frac{valid_{true}(\tau)}{total_{true}}$

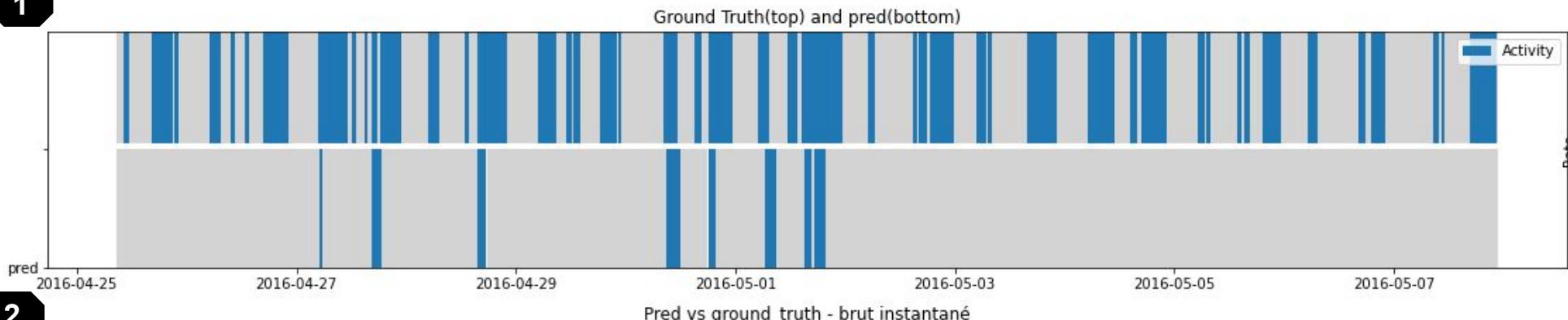
1. Choisir la référence : réelle pour mAR*, prédite pour mAP**
2. Pour chaque période considérée :
 - a. Mesurer IoU
 - b. Statuer sur la validité à un seuil donné
 - c. Répéter pour plusieurs seuils

$$IoU = \frac{\text{Intersection Area}}{\text{Union Area}}$$

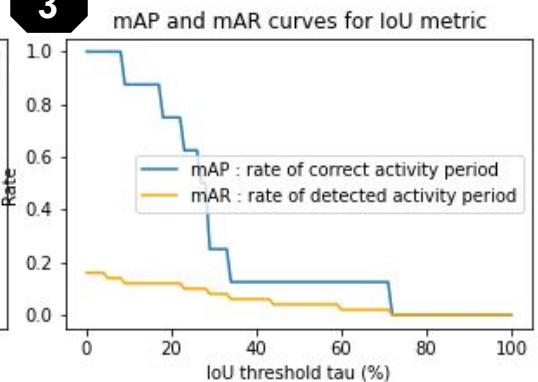
** mAP : mean Average Precision
* mAR : mean Average Recall

Evaluation du modèle - synthèse

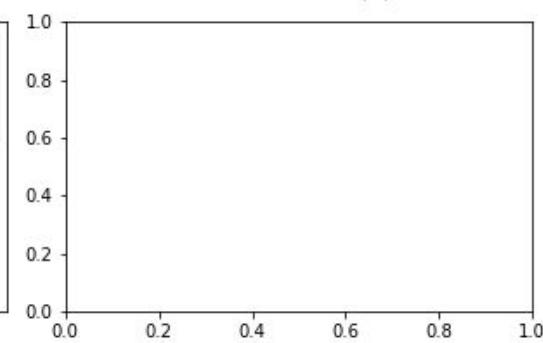
1



3



2



1 Activité réelle (haut) vs activité prédite (bas)

2 Superposition activité réelle/prédite et association TP / FP / TN / FN = matrice de confusion

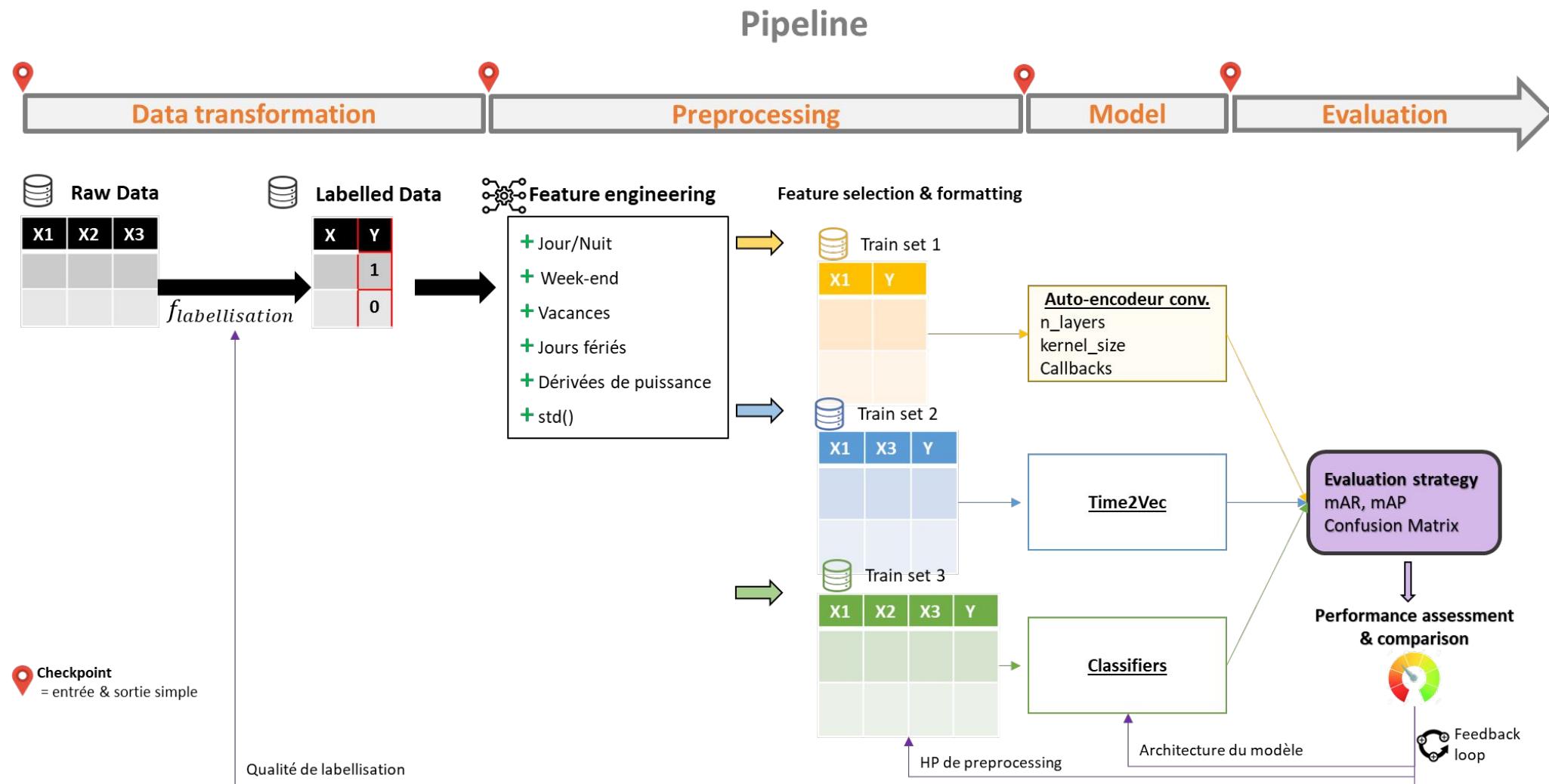
3 mAR et mAP pour différents seuils de la métrique de performance IoU

Overview des approches et pipeline

Overview des approches

- ❖ Première approche : Supervisée
 - Modèle Classiques : Ensemble Learning + classifieur
Classifieur traditionnel, qui ne prend vraiment pas en compte l'aspect temporel
XGboost , Catboost, Lightboost, Histboost, random Forest , KNN, voting
- ❖ Deuxième approche : Supervisée
 - Time2vec embeddings + classifieur
Approche plus temporelle, aspect time serie
- ❖ Troisième Approche : Non-supervisée
 - Auto-encodeur Convolutionnel pour la détection d'anomalie

Pipeline



Approche n°1 (supervisée)

Ensemble learning + Classifieur

Pre-processing : Création de features

Les 10 colonnes générées sont les suivantes :

- la moyenne glissante sur 1h
- la moyenne glissante sur 10 minutes
- l'écart type glissant sur 1h
- l'écart type glissant sur 10 minutes
- l'amplitude glissante (max - min) sur 1h
- l'amplitude glissante (max - min) sur 10 minutes
- le nombre de fois à la courbe de puissance agrégée coupe la moyenne glissante
- l'heure de la journée (un nombre à virgule flottante de 0 à 23),
- le fait que le jour soit un jour de semaine ou de weekend (encodé en one hot)
- la courbe de charge agrégé

Complément : gradient boosting classifier & Scalers

Stratégie :

- Suivre l'évolution de l'accuracy d'une liste de classifier en fonction d'une liste de Scaler. On entraînera nos classifieurs sur la maison 1 pour tester sur la maison 2 et vice versa. Choisir le meilleurs classifier et Scaler et fine-tuner ses hyperparamètres

liste des classifiers :

- Histogram-Based Gradient Boosting
- RandomForest
- Light Gradient Boosting Machine
- Categorical Boosting
- eXtreme Gradient Boosting

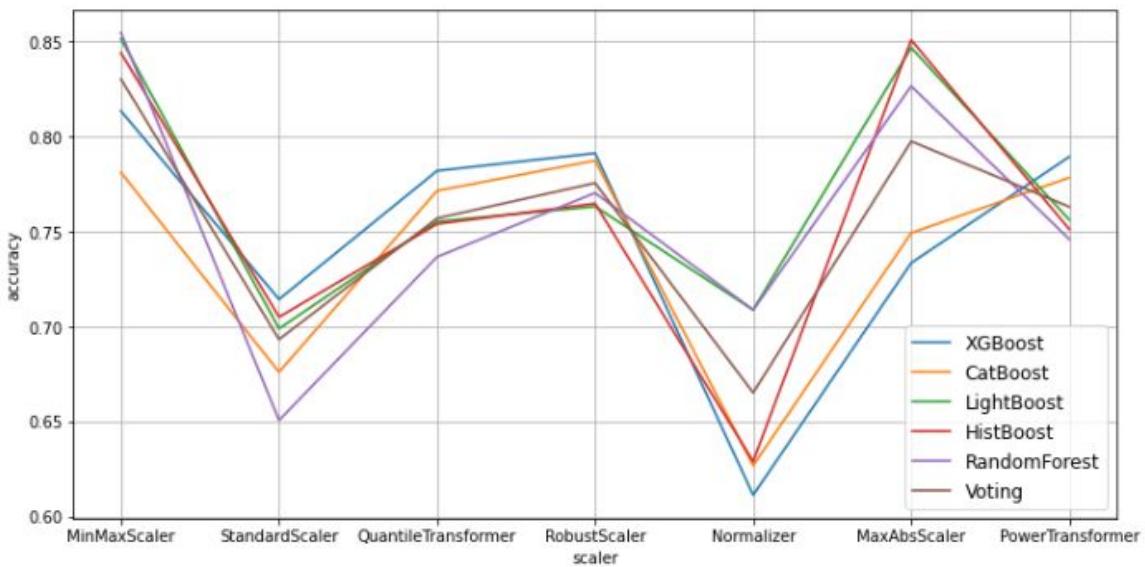
autre classifier :

- VotingClassifier

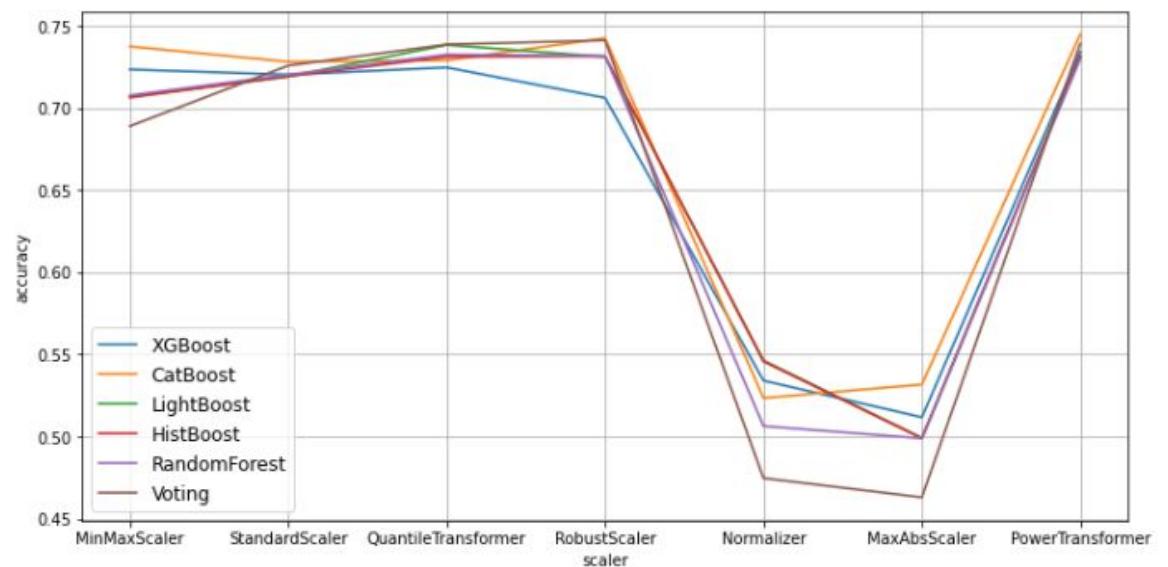
liste des scaler :

- MinMaxScaler
- StandardScaler
- QuantileTransformer
- RobustScaler
- Normalized
- MaxAbsScaler
- PowerTransformer

Entraînement des classifier en fonction des différents scaler

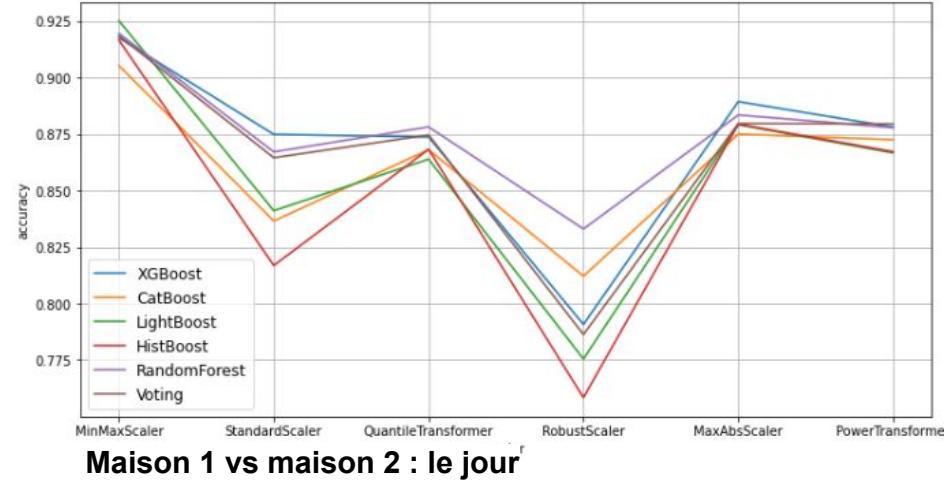


Évolution de l'accuracy en fonction des classifier, entraînement sur la maison 1 et test sur la maison 2

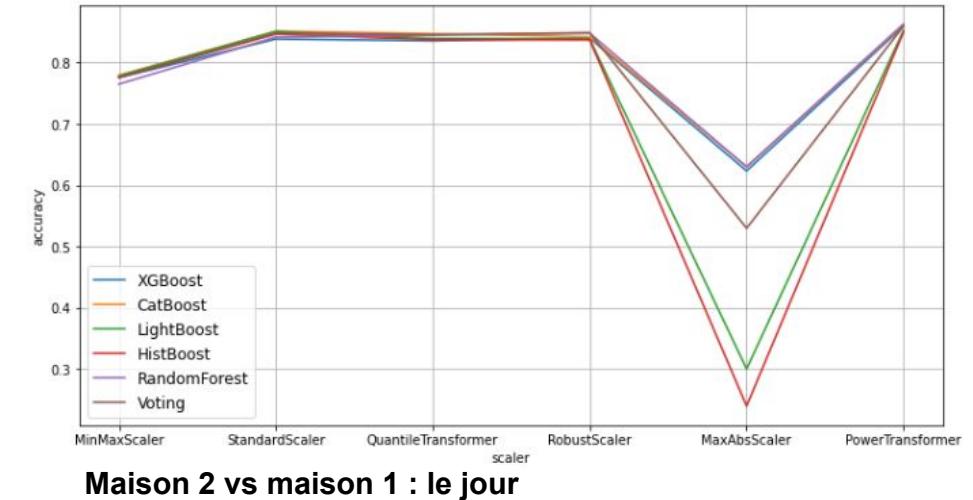


Évolution de l'accuracy en fonction des classifier, entraînement sur la maison 2 et test sur la maison 1

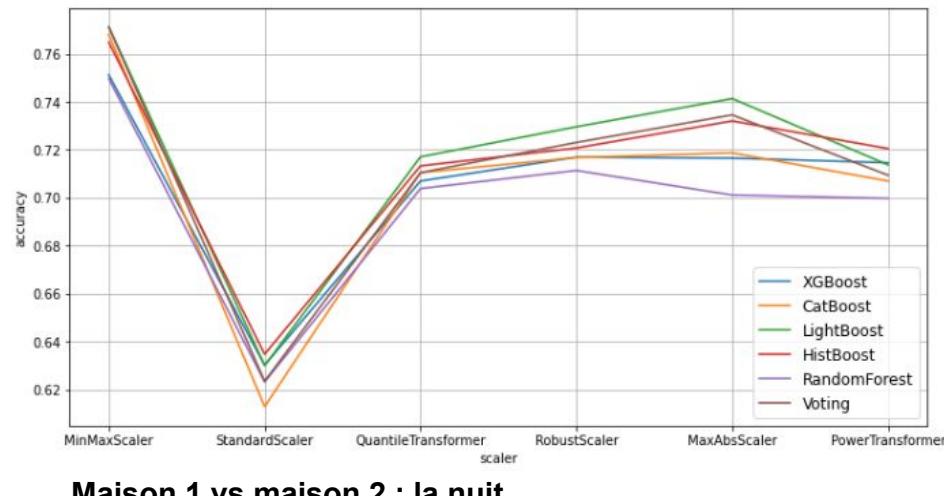
Entraînement des classifiers sur différentes périodes de la journée



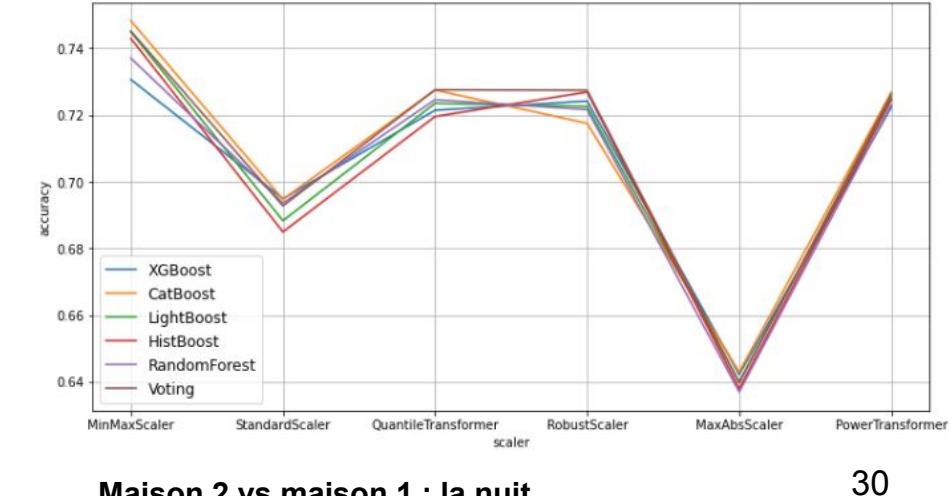
Maison 1 vs maison 2 : le jour



Maison 2 vs maison 1 : le jour



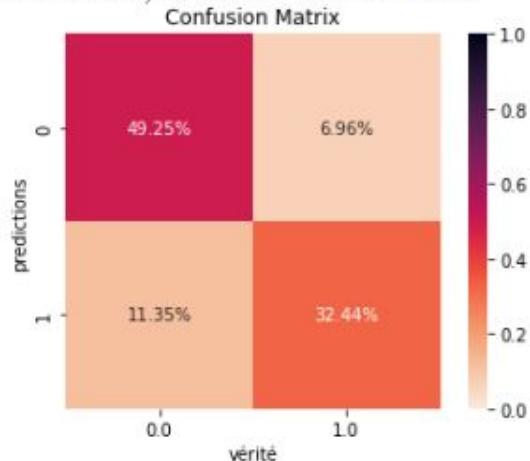
Maison 1 vs maison 2 : la nuit



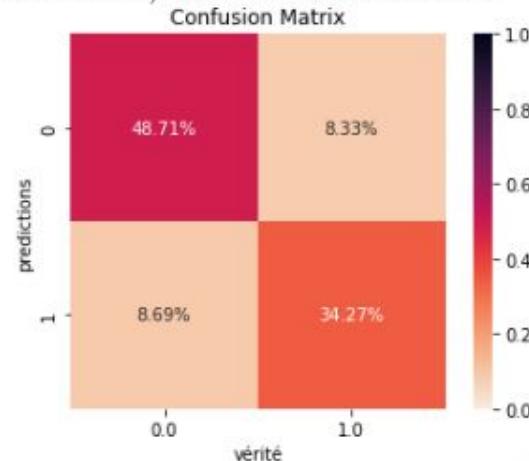
Maison 2 vs maison 1 : la nuit

Entraînement du LGBM Classifier et évaluation finale

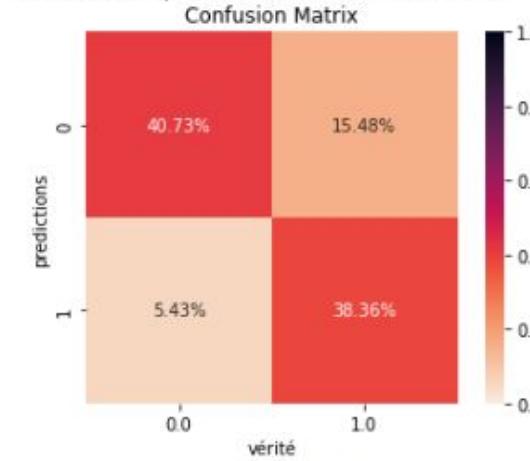
Entraînement classification sur maison 1, test sur la maison 1



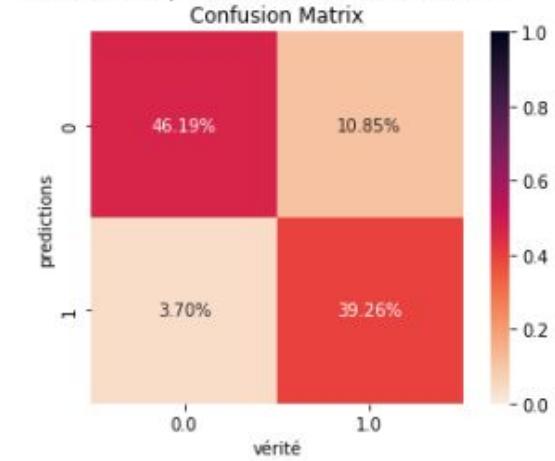
Entraînement classification sur maison 1, test sur la maison 2



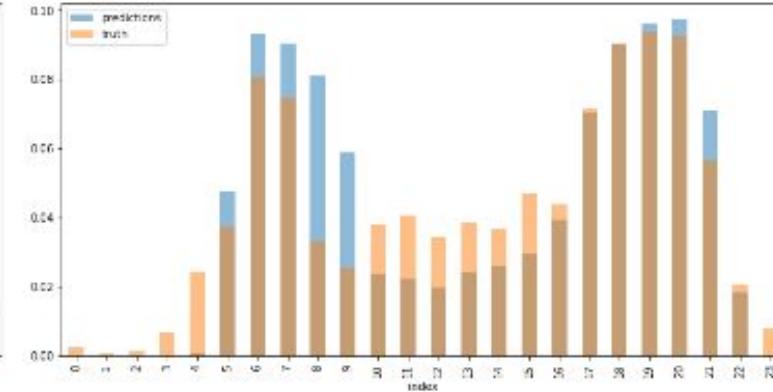
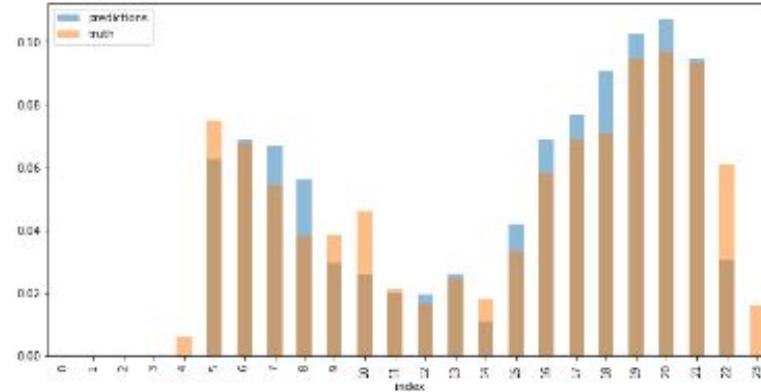
Entraînement classification sur maison 2, test sur la maison 1



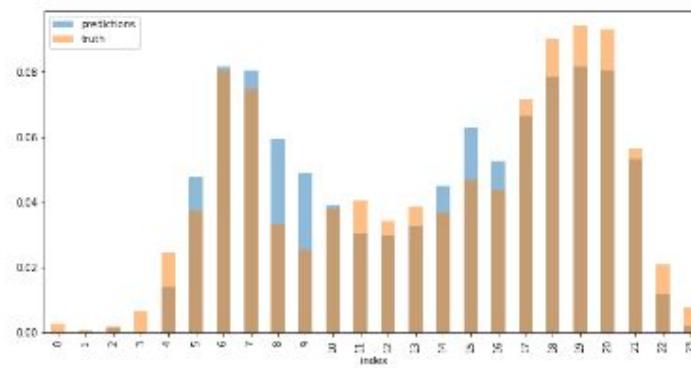
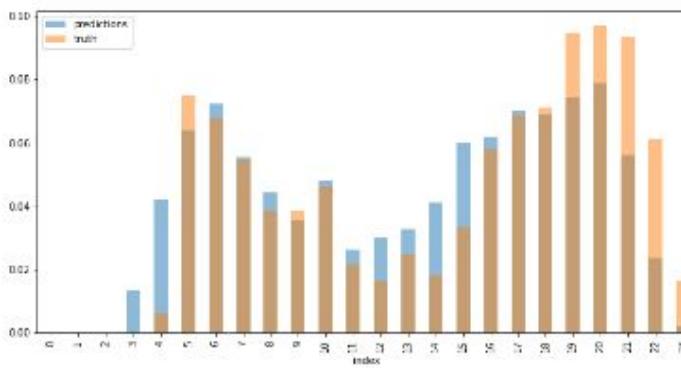
Entraînement classification sur maison 2, test sur la maison 2



LGBM Classifier - histogramme du label activité prédit par tranche horaire



– Histogramme du label d'activité prédict en fonction de l'heure de la journée par notre classifieur entraîné sur la maison 1 en testant sur la maison 1 (gauche) et la maison 2 (droite)



– Histogramme du label d'activité prédict en fonction de l'heure de la journée par notre classifieur entraîné sur la maison 2 en testant sur la maison 1 (gauche) et la maison 2 (droite)

Approche n°2 (supervisée)

Time2Vec Embeddings + Classifieur

Time2Vec : approche semi-supervisée

Time2Vec : Papier de 2019, par Seyed Mehran Kazemi et al.

L'idée : Avoir une **représentation simple et universelle des séries temporelles** entraînée sur une tâche prétexte.

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i, & \text{si } i = 0 \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{si } 1 \leq i \leq k \end{cases}$$

Avec :

- τ : scalaire représentant le temps
- $\mathbf{t2v}(\tau)[i]$: i ème élément du vecteur Time2Vec $\mathbf{t2v}(\tau)$
- \mathcal{F} : fonction d'activation périodique (nous utilisons la fonction sinus)
- ω_i et φ_i : paramètres appris par un algorithme de descente de gradient

Propriétés :

- Capacité à capturer les motifs périodiques et non-périodiques
- Invariance aux changements d'échelle

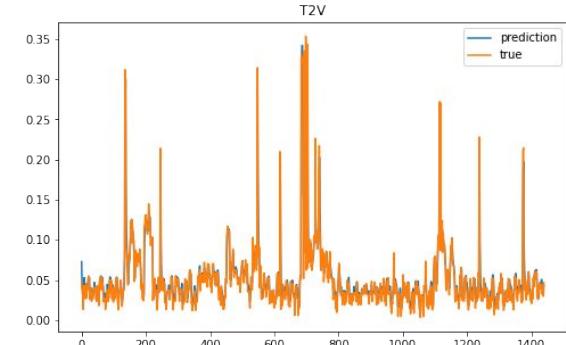
Time2Vec : méthodologie

Étape 1 : Couche Time2Vec entraînée sur une tâche de prédiction (apprentissage non supervisé)

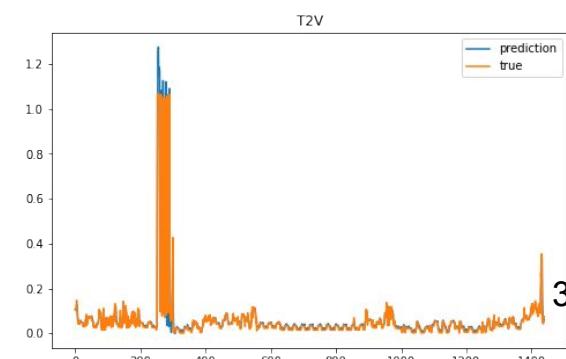
- Ré-échantillonnage des données à 1min30
- Entrée : séquences de 60 minutes
- Sortie : prédiction sur toutes les séquences
- Loss : MSE



Prédictions sur maison 1 avec modèle entraîné sur la maison 1



Prédictions sur maison 2 avec modèle entraîné sur la maison 1



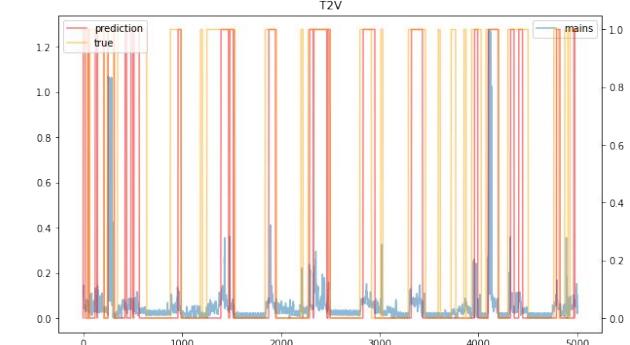
Time2Vec : méthodologie

Étape 2 : Réseau de classification binaire avec couche Time2Vec pré-entraînée en entrée (apprentissage supervisé)

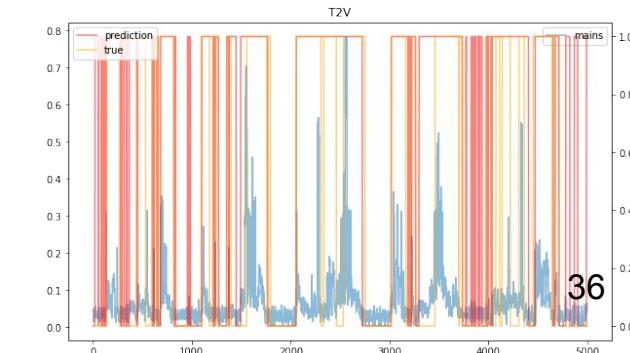
- Entrée : séquences de 60 minutes (séquence de longueur 40)
- Sortie : classification sur ces séquences : 40 labels activité / non activité
- Loss : BCE
- Activation sigmoïde en sortie de FC
- Choix : fine-tuning des embeddings ou non



Classification sur maison 1 avec modèle entraîné sur la maison 1



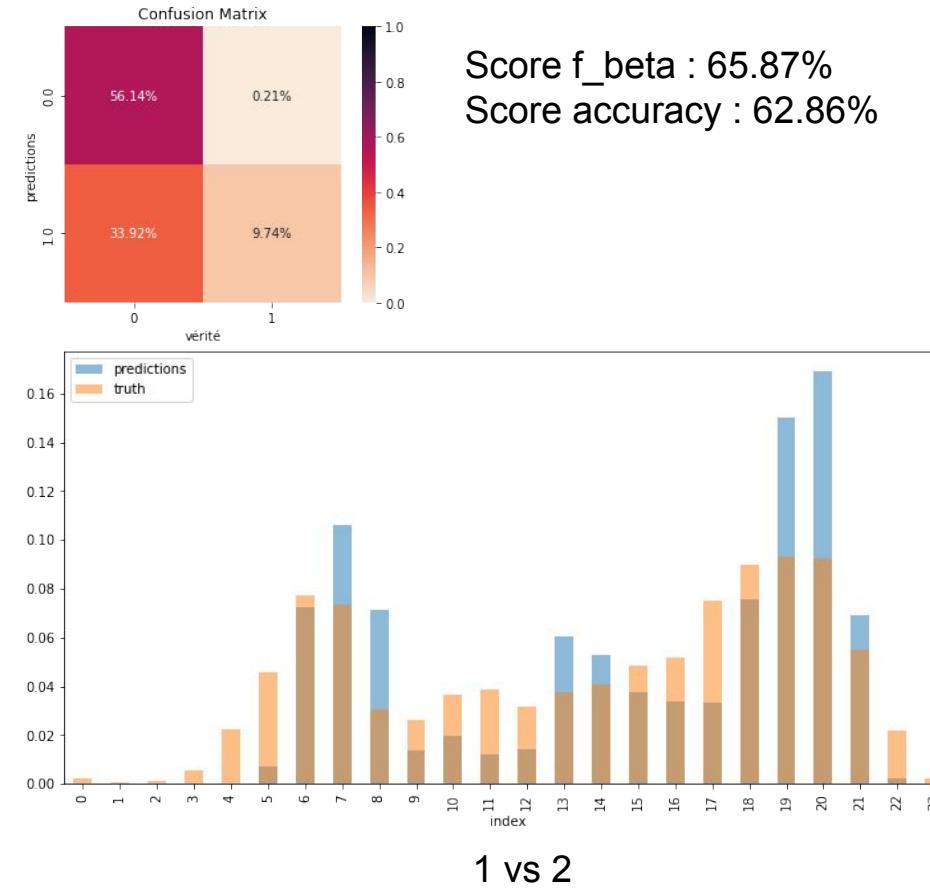
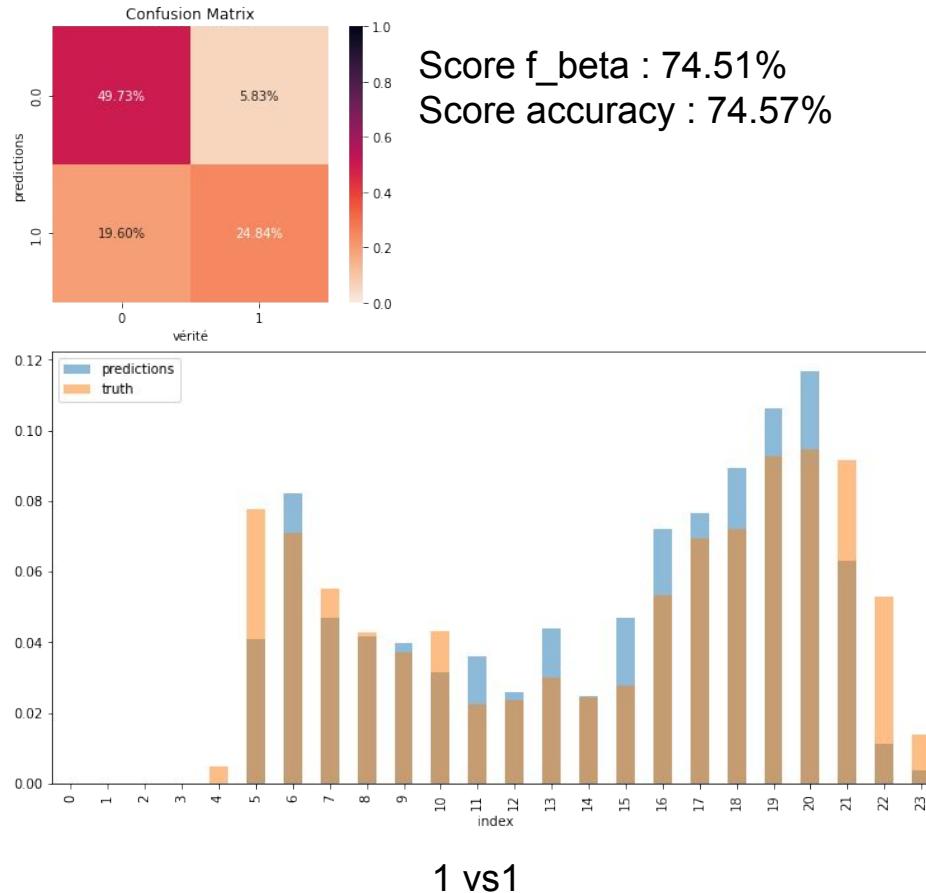
Classification sur maison 2 avec modèle entraîné sur la maison 1



Time2Vec : résultats

Méthode utilisée :

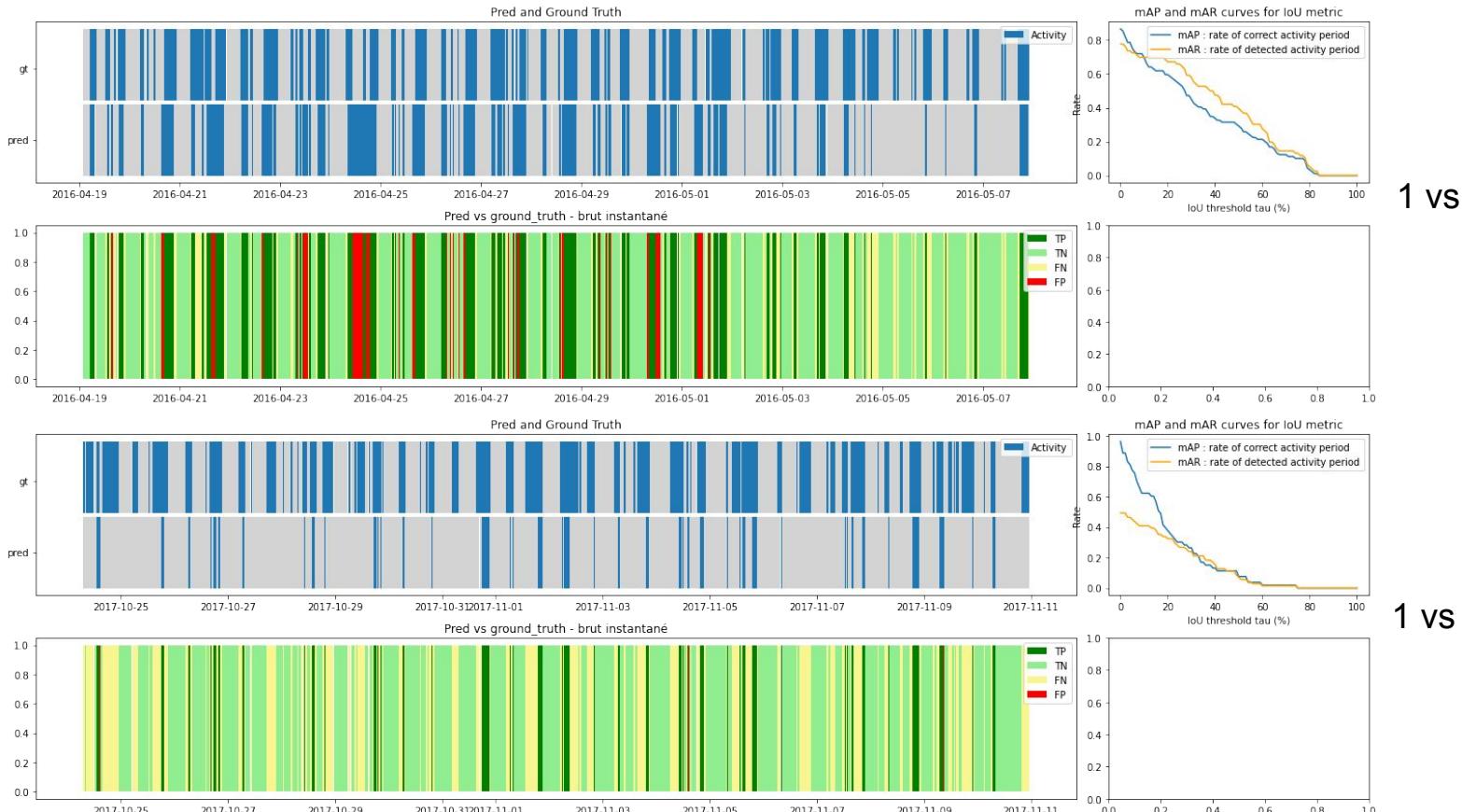
- Entraînement des embeddings sur la maison 1 (peu d'influence)
- Entraînement du classifieur (sans fine-tuning des embeddings) sur maison 1



Time2Vec : résultats

Méthode utilisée :

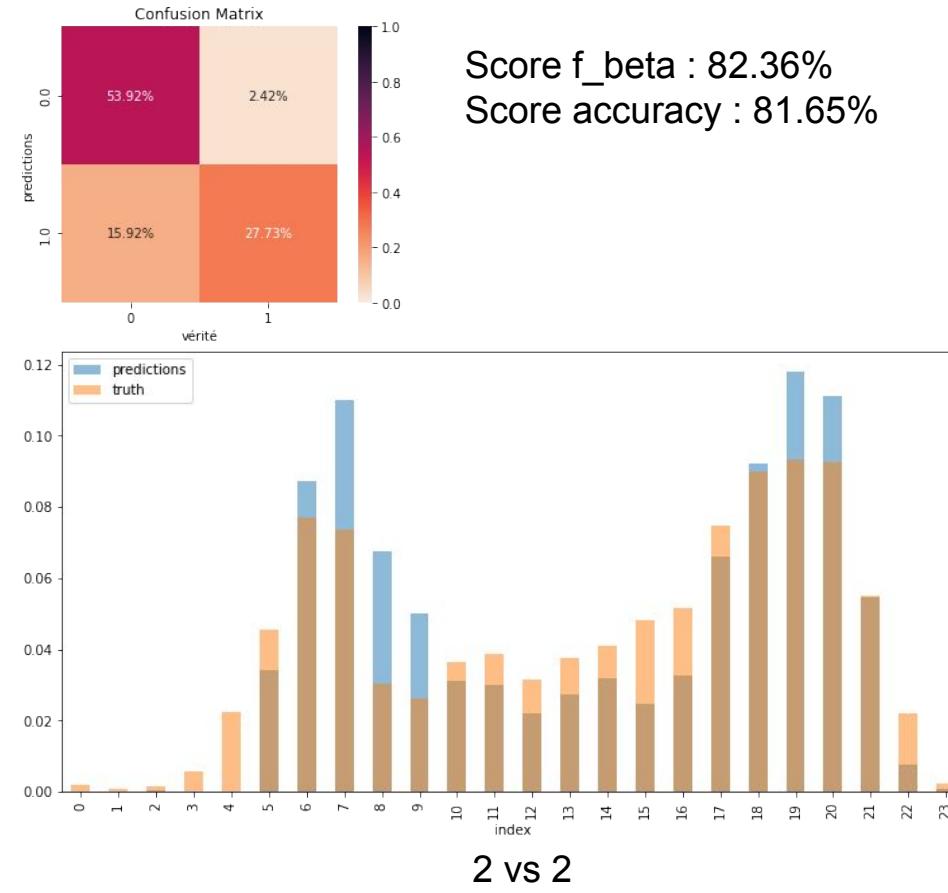
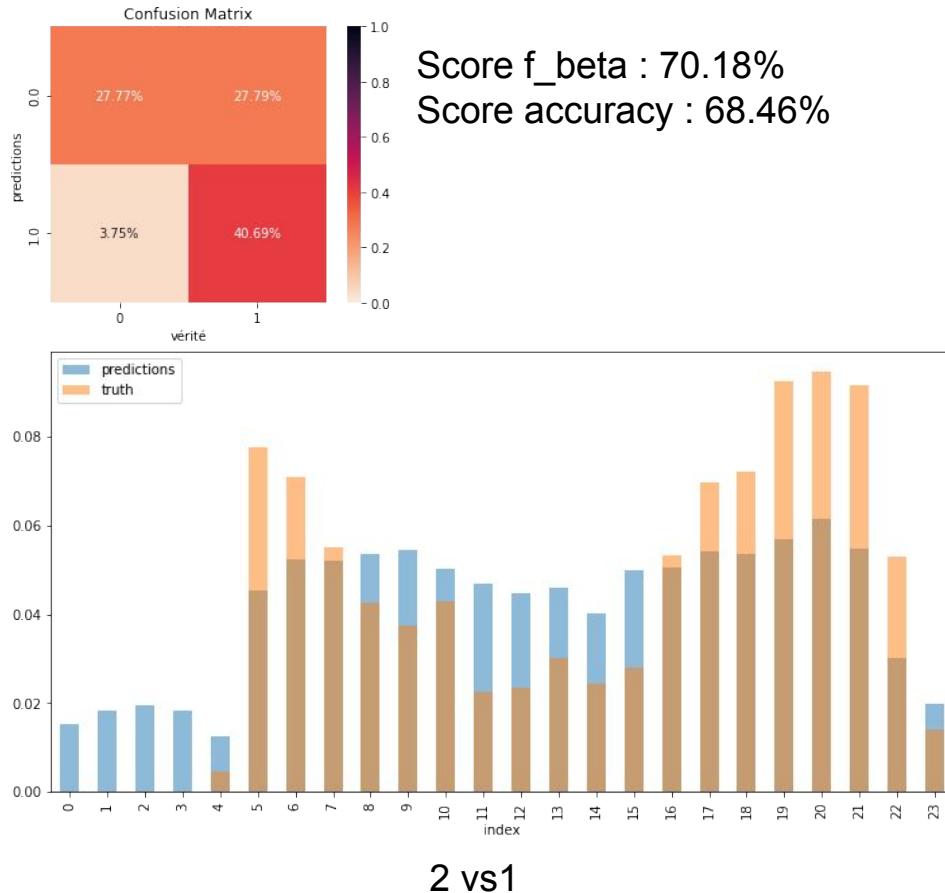
- Entraînement des embeddings sur la maison 1
- Entraînement du classifieur (sans fine-tuning des embeddings) sur maison 1



Time2Vec : résultats

Méthode utilisée :

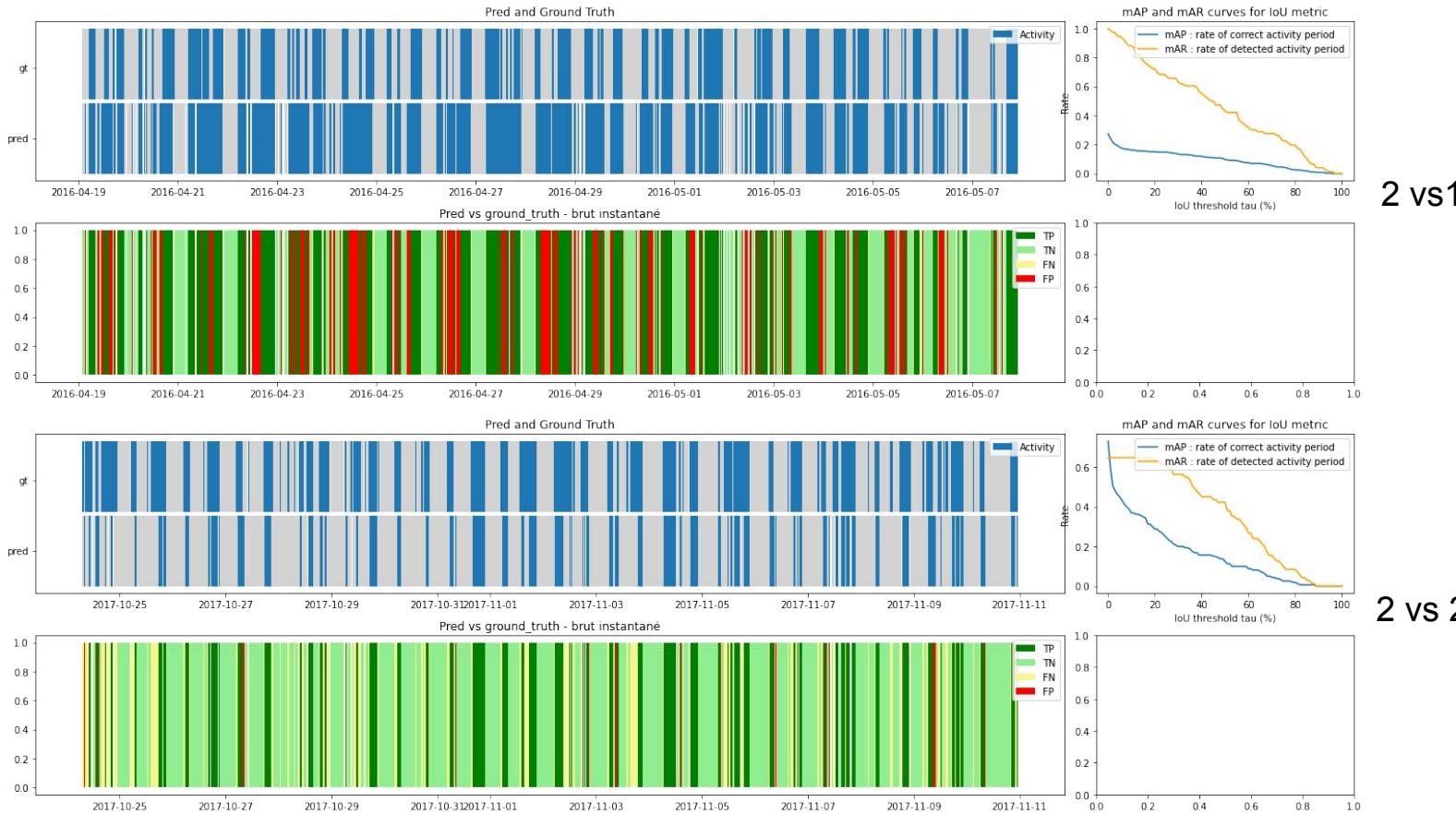
- Entraînement des embeddings sur la maison 1
- Entraînement du classifieur (sans fine-tuning des embeddings) sur maison 2



Time2Vec : résultats

Méthode utilisée :

- Entraînement des embeddings sur la maison 1
- Entraînement du classifieur (sans fine-tuning des embeddings) sur maison 2



2 vs 1

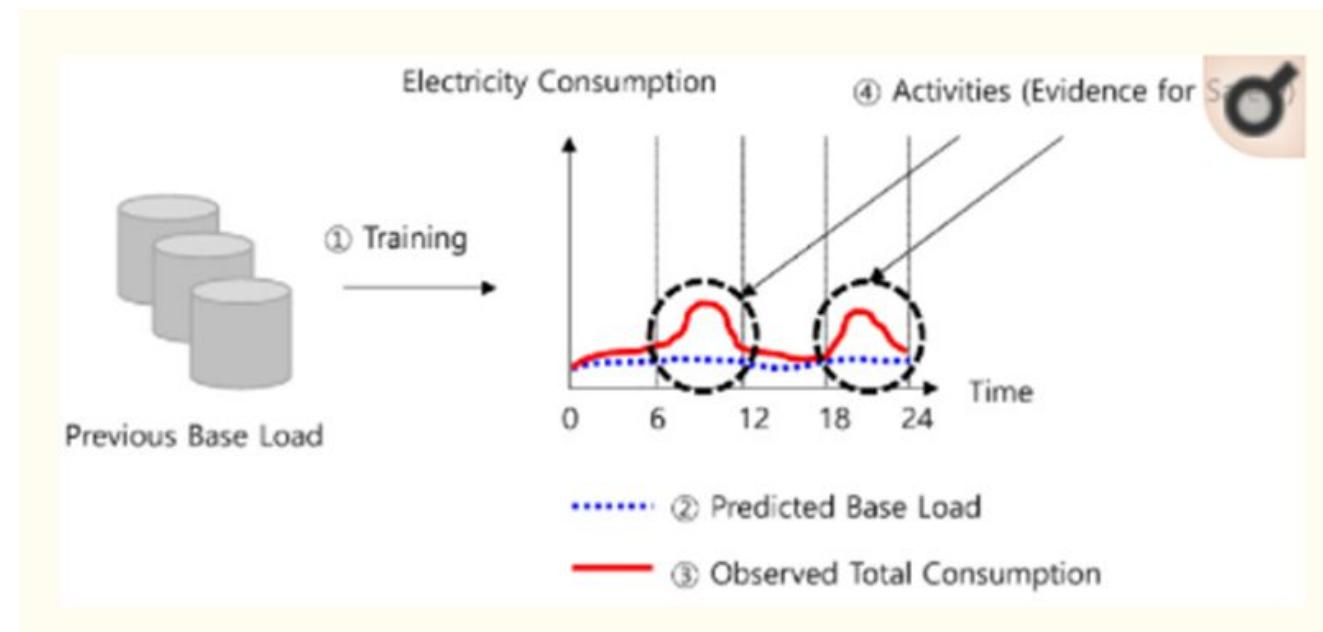
2 vs 2

Approche n°3 (non supervisée)

Auto-Encodeur Convolutionnel pour la détection
d'anomalie

L'approche Auto-Encodeur Convolutionnel

- Modéliser le comportement normal des activités quotidiennes d'un foyer
- Pour détecter une activité anormale, la consommation électrique de base doit être identifiée.
- Pour obtenir cette courbe de charge de consommation normale synonyme d'inactivité dans un foyer, deux possibilités s'offrent à nous :
 - **Stratégie n°1** : Se référer à la courbe de charge nocturne (heures creuses)
 - **Stratégie n°2** : Sélectionner individuellement la consommation électrique de chacun des appareils permettant de prédire une activité



Pre-processing (1)

TIME_STEP: durée d'une étape dans l'ensemble de données de re-échantillonnage (initialement 1 seconde)

DURATION_TIME: durée d'une séquence

OVERLAP_PERIOD_PERCENT: taux de chevauchement des données entre chaque séquence

TIMEFRAMES: plage horaire des heure creuses

SPLIT_METHOD = méthode pour construire la séparation du jeu d'entraînement et de test

Variables globales

ConvertToSequenceParameters()

*Convertit la période de chevauchement en entier
Choisi la longueur de la séquence optimale*

Preprocessing()

Créer les séquences

[samples, sequenceLength, features]

=> Retourne un 3D-Array

Pre-processing (2)

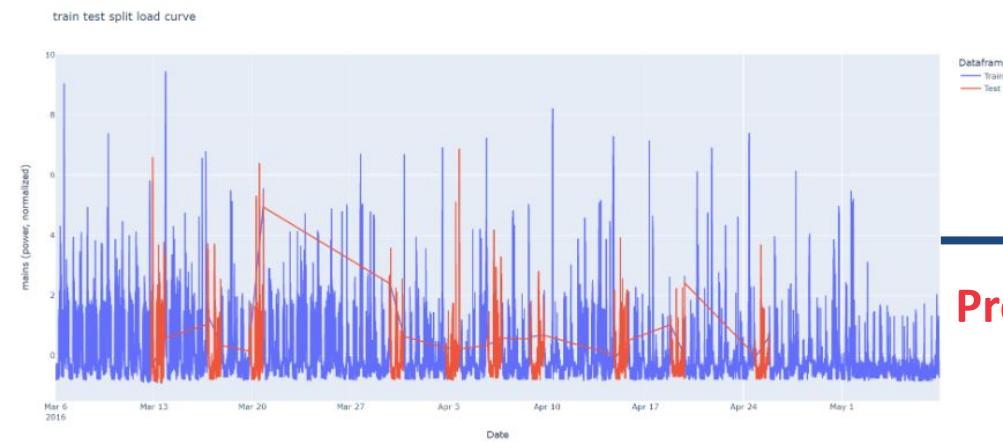


FIGURE 2.62 – Découpage de la courbe de charge en train, test

Pre-processing



FIGURE 2.49 – Séquence de la courbe de charge de base

[samples, sequenceLength, features]

---- X_train sequence shape ----
(561, 40, 1)

---- y_train sequence shape ----
(561, 40, 3)

---- X_test sequence shape ----
(1506, 40, 1)

---- y_test sequence shape ----
(1506, 40, 3)

[[-0.72298671],
[-0.55800157],
[-0.57759356],
...,
[-0.84363209],
[-0.8384763],
[-0.8240401]],

[[-0.70442588],
[-0.70751935],
[-0.70751935],
...,
[-0.44560545],
[44, -0.31877313],
[-0.45591702]],

Architecture AEC

Input :

=> **Input(sequenceLength, featuresLength)**

- *Conv1D (convolution)*
- *Couche Conv1D (convolution)*
- *Conv1D transpose (déconvolution)*
- *Conv1D transpose (déconvolution)*
- *Conv1D transpose (déconvolution)*

Output :

=> **[samples, sequenceLength, feature]**

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 20, 32)	256
dropout (Dropout)	(None, 20, 32)	0
conv1d_1 (Conv1D)	(None, 10, 16)	3600
conv1d_transpose (Conv1DTra nspose)	(None, 20, 16)	1808
dropout_1 (Dropout)	(None, 20, 16)	0
conv1d_transpose_1 (Conv1DT ranspose)	(None, 40, 32)	3616
conv1d_transpose_2 (Conv1DT ranspose)	(None, 40, 1)	129

Total params: 9,409

Trainable params: 9,409

Non-trainable params: 0

Entrainement AEC

Hyperparamètres :

- *epoch = 50*
- *batchSize = 128*
- *optimizer = Adam*
- *aeLoss = MSE*

MSE (Mean Squared Error)

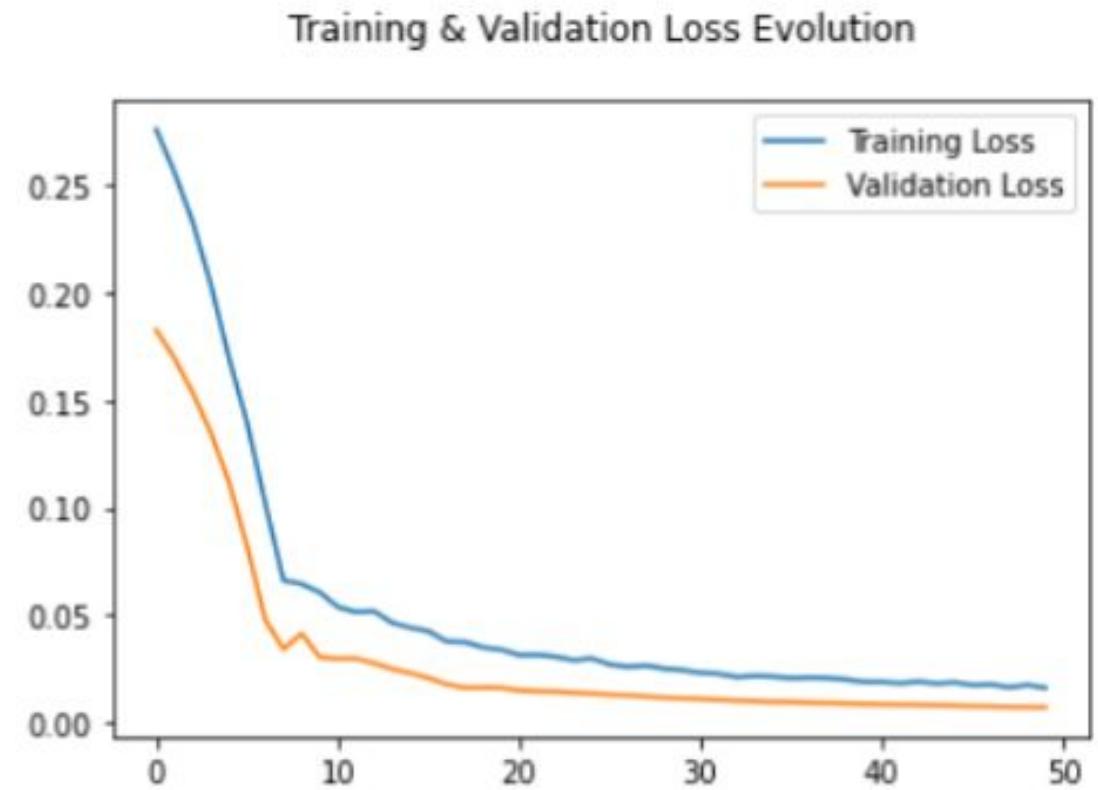
$$\sum_{i=1}^D (x_i - y_i)^2$$

Objectif :

Minimiser la fonction de coût de reconstruction (MSE)

Pourquoi MSE ?

Pénaliser les valeurs aberrantes étant donné que nous souhaitons apprendre la courbe de consommation de base



Utilisation d'un **early stopping** pour diminuer le temps d'entraînement lorsque la loss ne diminue pas

Visualisation de la reconstruction

Séquence reconstruite du **Jour 1** de notre jeu de données **d'entraînement**.

Pas de temps : **40** (longueur de la séquence)

=> L'AE parvient bien à reconstruire la courbe de charge de base

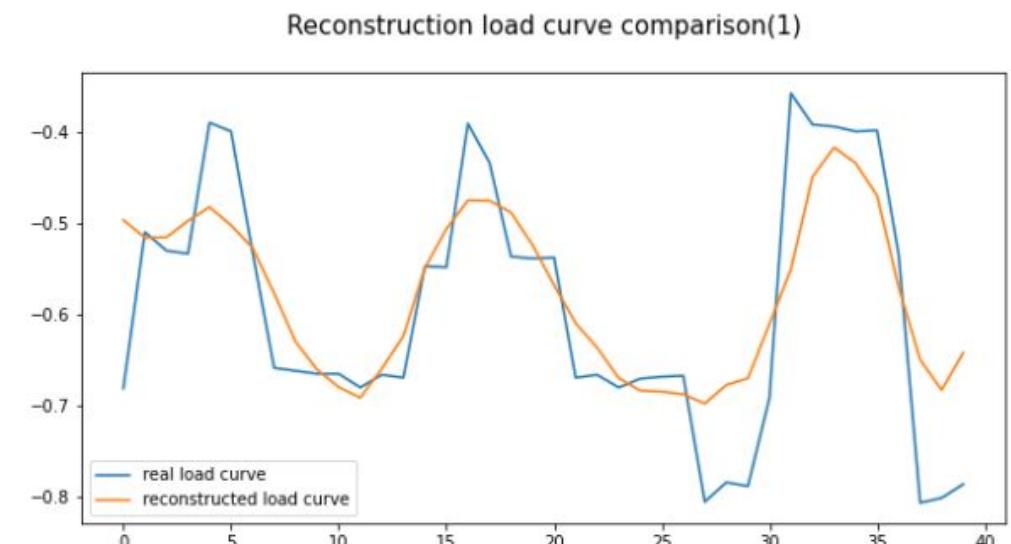


FIGURE 2.68 – Reconstruction de la courbe de charge de base par l'AE

Détection des anomalies

1. Calculer la perte **MAE** sur les échantillons d'entraînement.

MAE (Mean Absolute Error)

$$\sum_{i=1}^D |x_i - y_i|$$

2. Trouver la valeur **maximale** de perte **MAE** qui correspond à notre seuil (**threshold**)

3. Les séquences dont la valeur de puissance dépasse ce seuil sont des anomalies

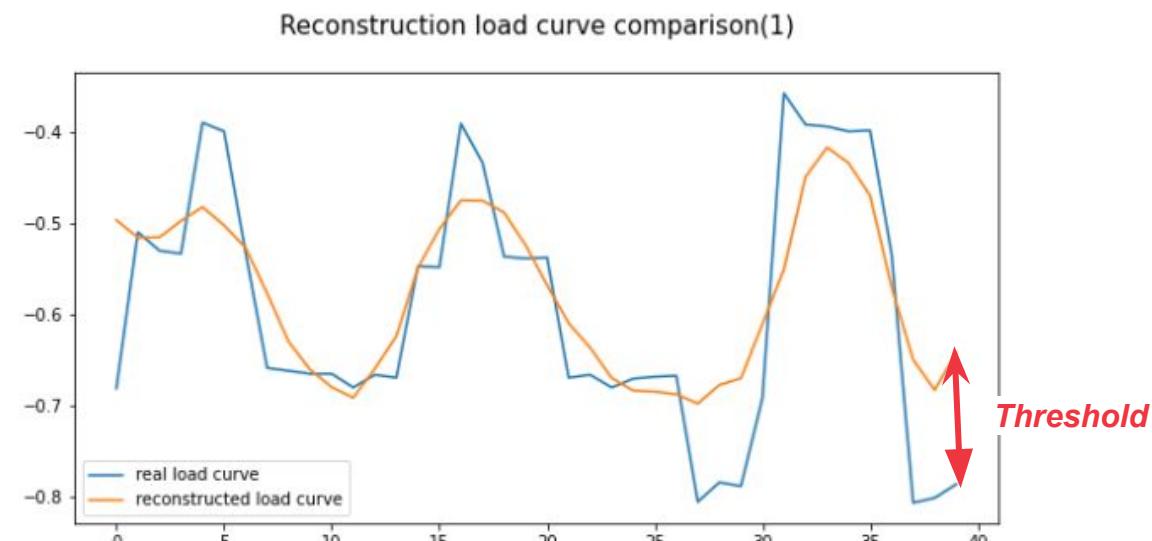


FIGURE 2.68 – Reconstruction de la courbe de charge de base par l'AE

Post-processing

- Affinage des prédictions
- Problèmes :
 - il arrive parfois qu'un point de donnée se trouve à la fois dans une séquence prédite par le modèle comme une anomalie et une séquence prédite par le modèle comme un fonctionnement normal

Solution

=> Réalisation d'un vote majoritaire

	Timestamp	list_idx_sequence_no_activity	list_idx_sequence_activity	nb_no_activity	nb_activity	total	method_prediction_1
868	2016-04-26 06:30:00	[]	[104, 105, 106, 107, 108]	0	5	5	1
869	2016-04-26 06:31:30	[]	[104, 105, 106, 107, 108]	0	5	5	1
870	2016-04-26 06:33:00	[]	[104, 105, 106, 107, 108]	0	5	5	1
871	2016-04-26 06:34:30	[]	[104, 105, 106, 107, 108]	0	5	5	1
872	2016-04-26 06:36:00	[109]	[105, 106, 107, 108]	1	4	5	1
873	2016-04-26 06:37:30	[109]	[105, 106, 107, 108]	1	4	5	1
874	2016-04-26 06:39:00	[109]	[105, 106, 107, 108]	1	4	5	1
875	2016-04-26 06:40:30	[109]	[105, 106, 107, 108]	1	4	5	1

Résultats (Après post-processing)

Detected anomalies with method prediction 1

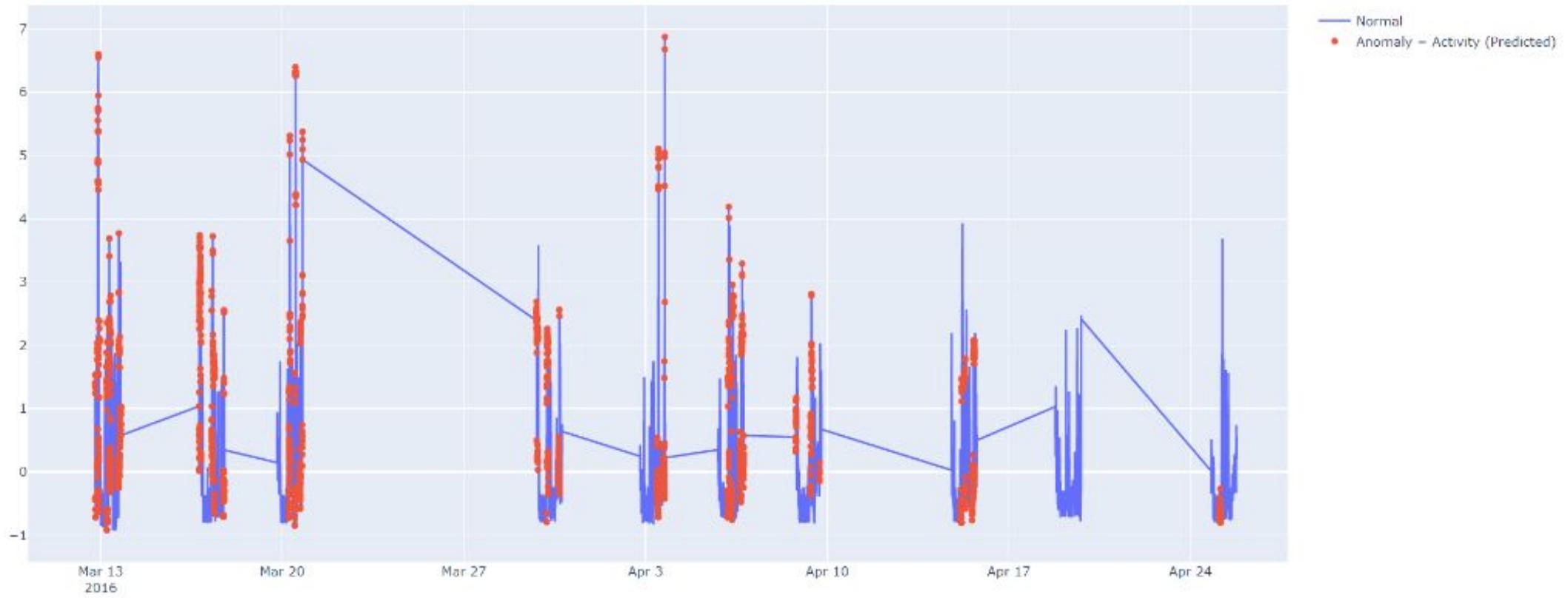


FIGURE 2.72 – Anomalies détectées sur la courbe de charge (Après post-processing)

Discussion des approches

Approches supervisées

{ BoostingClassifier , Time2Vec }

- Besoin de 15 jours de données
- + Rapide à entraîner
- + Bonnes performances
- + Simple à automatiser (time2vec)

- Peu généralisable (nécessite des labels)
- Explicabilité mauvaise (superposition de réseau de neurones) (t2v)
- Sensible aux hyperparamètres (t2v)
- Tendance à “sur-apprendre” (over-fitting) (t2v)

Approche non supervisée

{ Auto-Encodeur Convolutionnel }

- Besoin de 28 jours de données
- + Rapide à entraîner
- + Modèle ajustable (S'adapte à la composition du foyer)
- + Généralisable
- + Meilleure explicabilité (Connaissance du threshold de détection d'activité)

- Pas facile à déployer (beaucoup de paramètres à choisir)
- Résultats variables : dépend de l'initialisation des couches de l'AE
- Sensible aux hyperparamètres

Conclusion

- Sujet complexe, recherche active
- Plusieurs approches possibles : compromis à faire
- Enjeux importants sur les méthodes non intrusives

Merci à vous !!

Annexes

Pipeline “main.py”

- 1/ Load dataset
- 2/ Convert user input
- 3/ Preprocessing (build sequences)
- 4/ Build model
- 5/ Train model
- 6/ Plot Reconstructed curve
- 7/ Compute Threshold
- 8/ Make predictions
- 9/ Detecting activities
- 10/ Postprocessing
- 11/ Plot detected activities
- 12/ Evaluate model
- 13/ Plot Evaluation model (IoU threshold)

Execution time :
30 sec

Autre jeu de données : UK-DALE

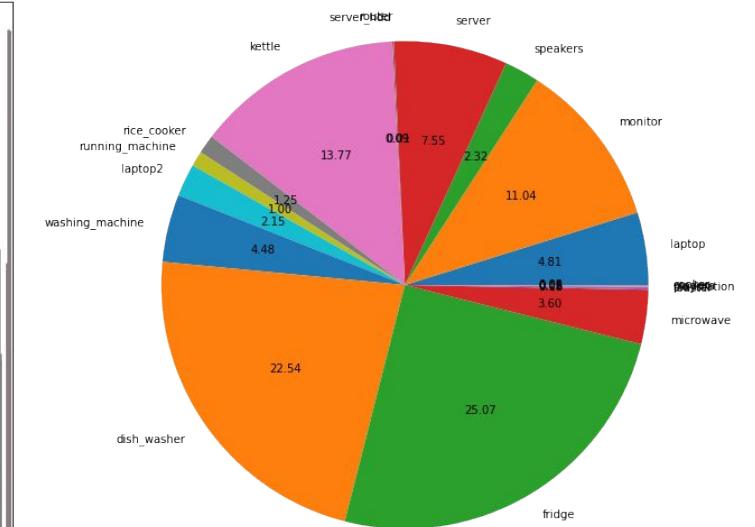
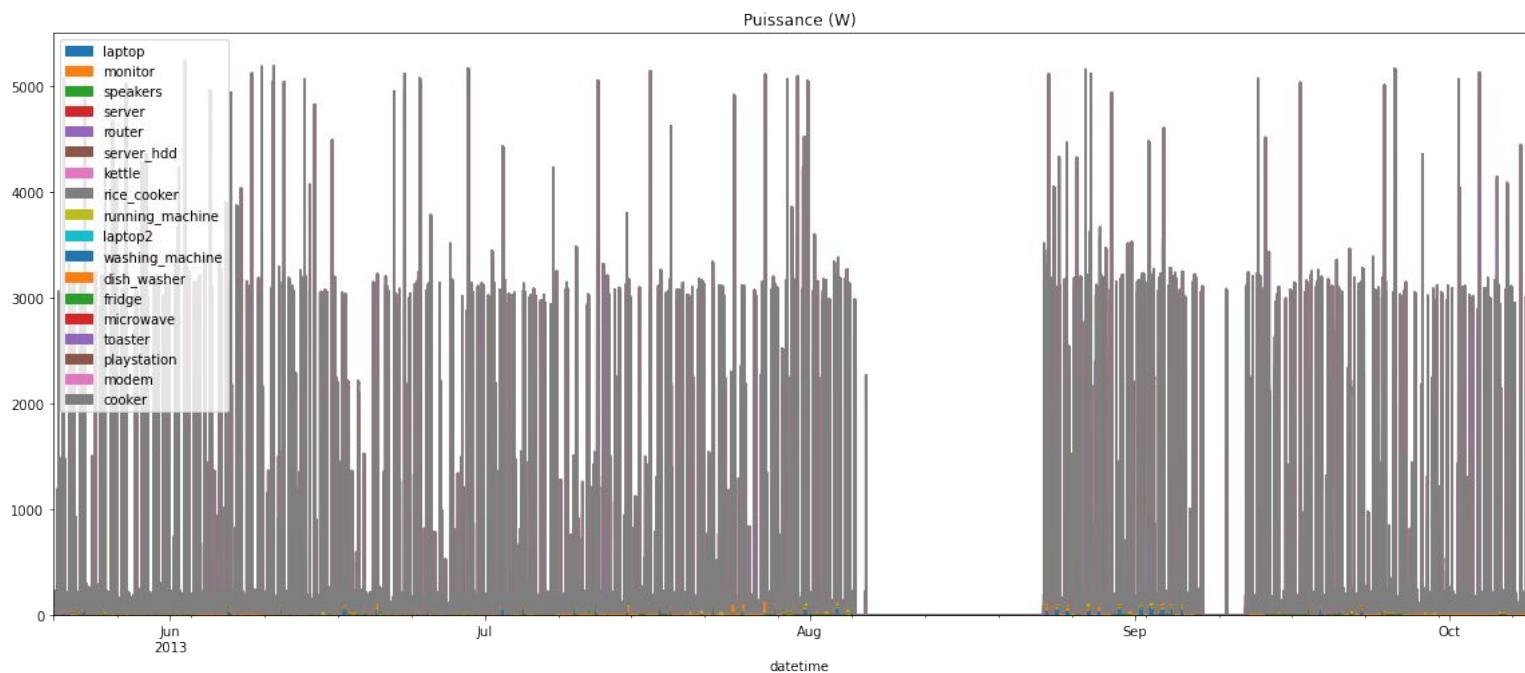
Provient du paper *Domestic appliance-level electricity demand and whole-house demand from five UK homes*

- 5 maisons
- Courbes de charge désagrégées (sur plusieurs appareils données)
- échantillonnage à 6 Hz

Autre jeu de données : UK-DALE

Choix maison 2

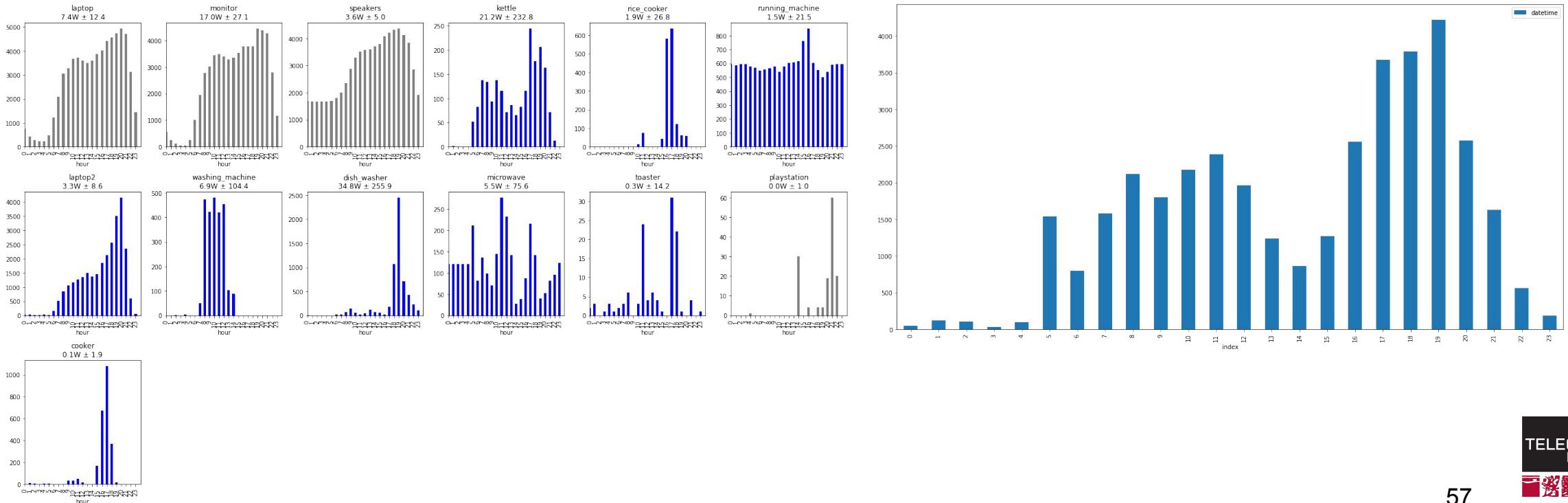
- Bonne durée (ni trop court ni trop long)
- 2 adults, 1 travaille en journée, l'autre parfois dans le foyer



Autre jeu de données : UK-DALE

Choix maison 2

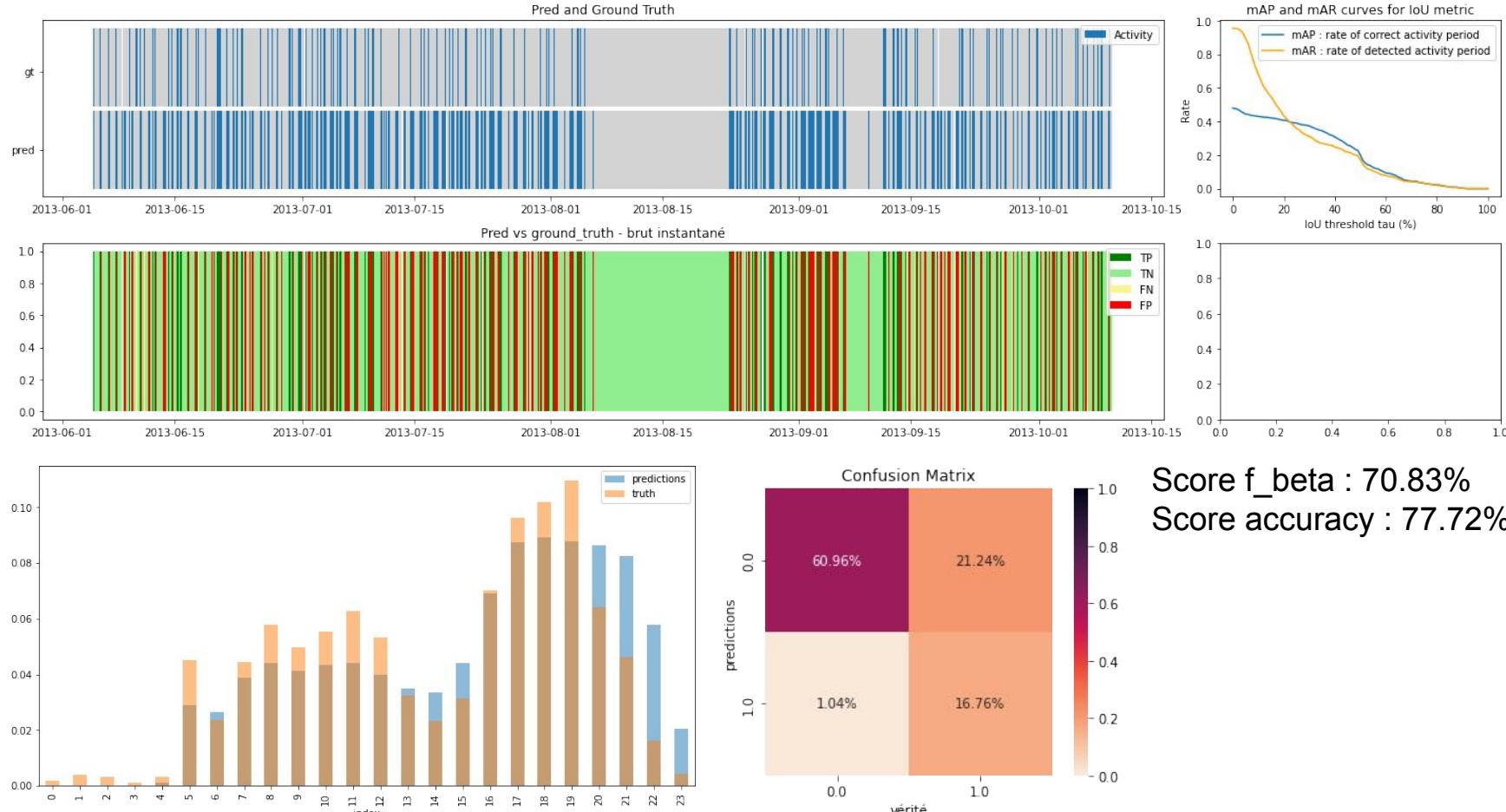
- Labellisation : analyse des courbe des appareils disponibles
- Choix des courbes de charge pertinentes
- Définition des seuils (même méthode que pour le jeu RAE)



Classifieur : résultats

Méthode utilisée :

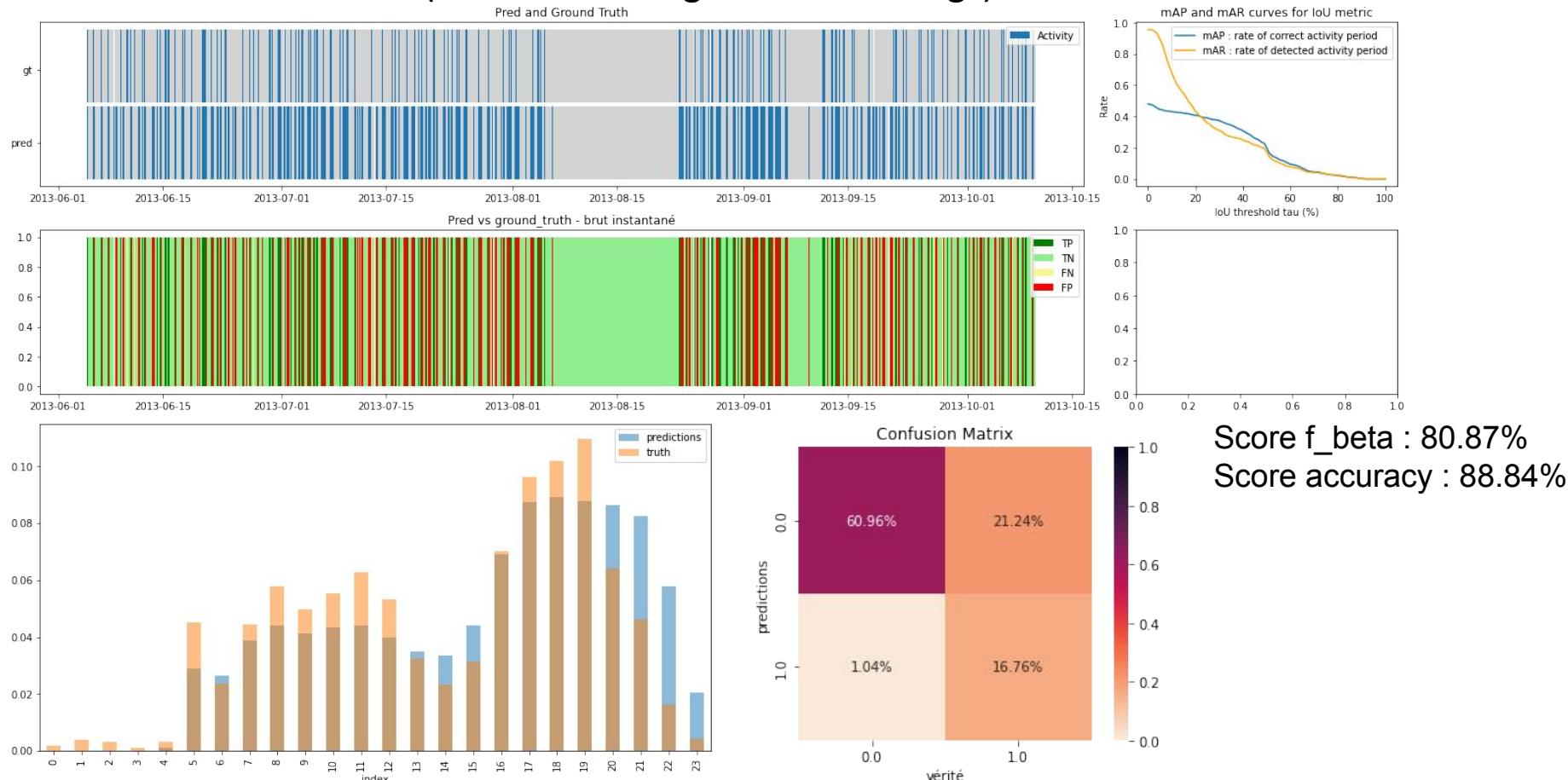
- Entraînement du classifieur sur la maison 1 & 2



Time2Vec : résultats

Méthode utilisée :

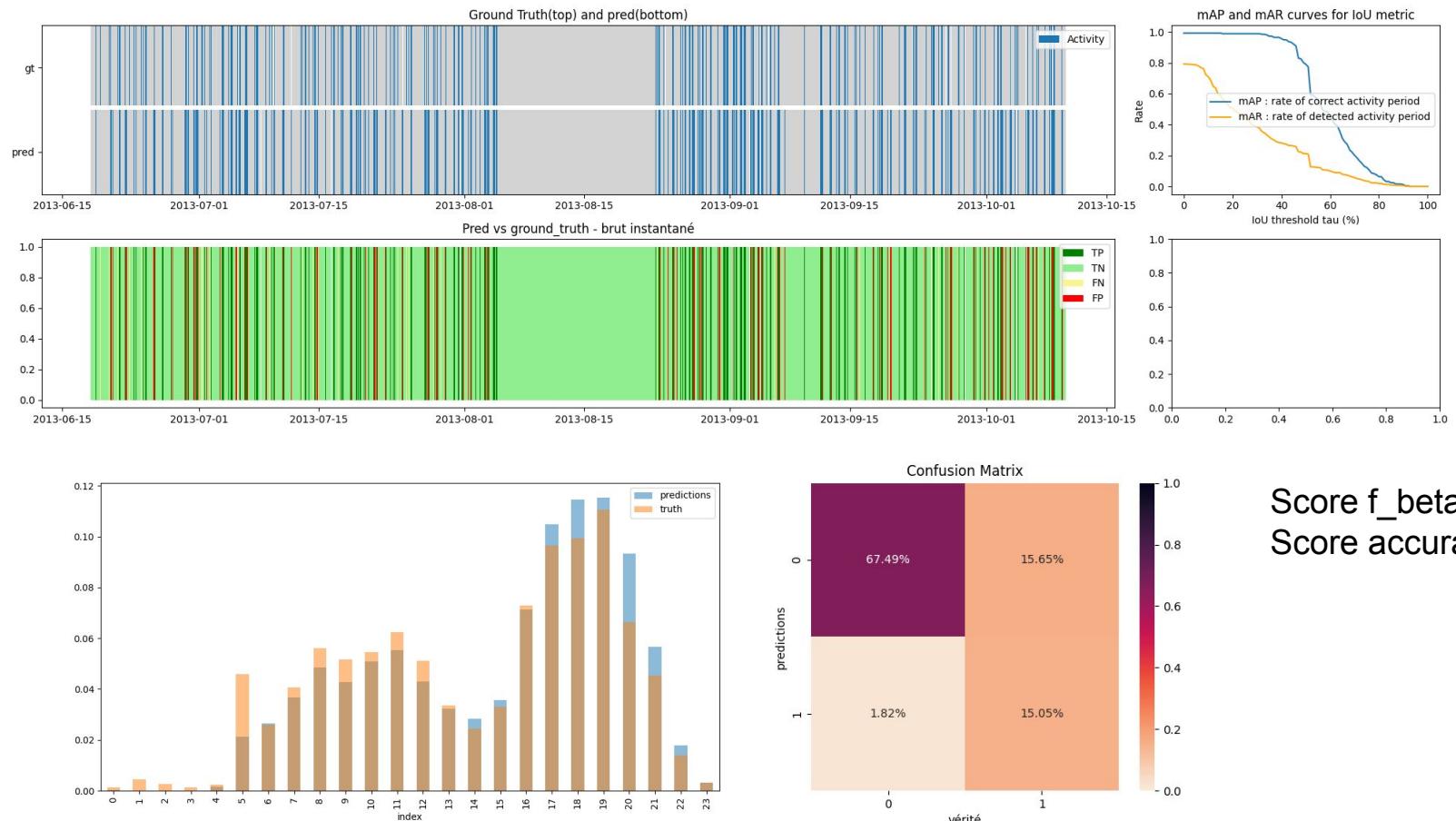
- Entraînement des embeddings sur la maison 1 & 2 + fine-tuning sur maison 2 UK-DALE
- Entraînement du classifieur (avec fine-tuning des embeddings) sur maison 1 & 2



Auto-Encodeur Convolutionnel : résultats

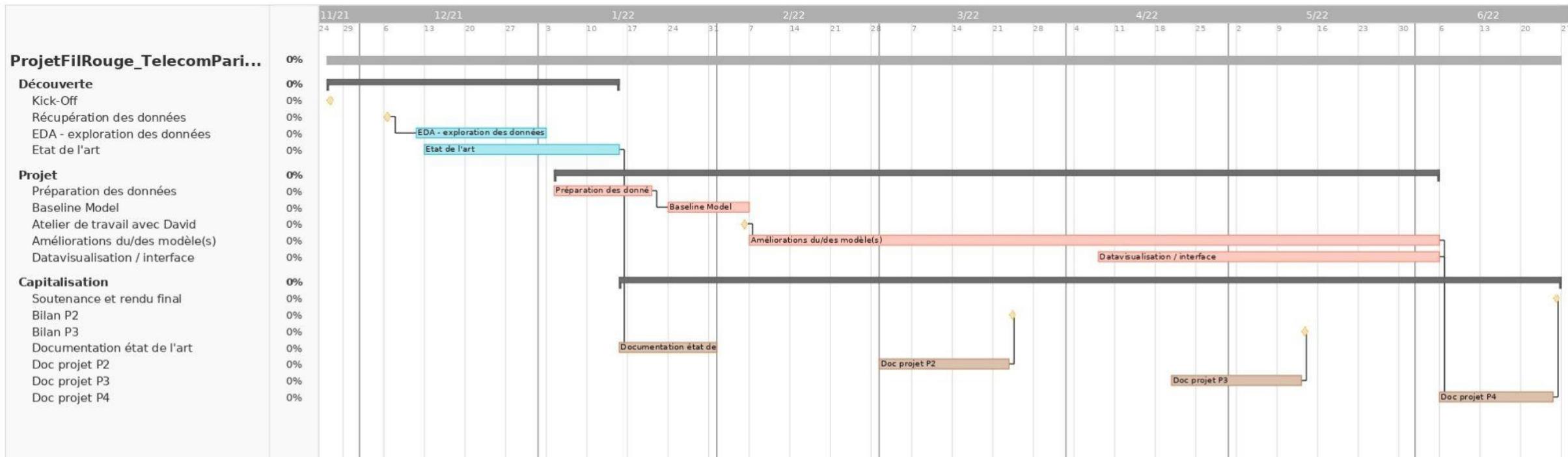
Méthode utilisée :

- Entraînement sur la maison 2 UK-DALE
- Définition du threshold and détection des anomalies



Planning initial

Created with Free Edition



Visualisation du threshold sur Xtrain et Xtest

MAE (Mean Absolute Error)

$$\sum_{i=1}^D |x_i - y_i|$$

Evolution de la perte MAE sur l'ensemble de jeu d'entraînement

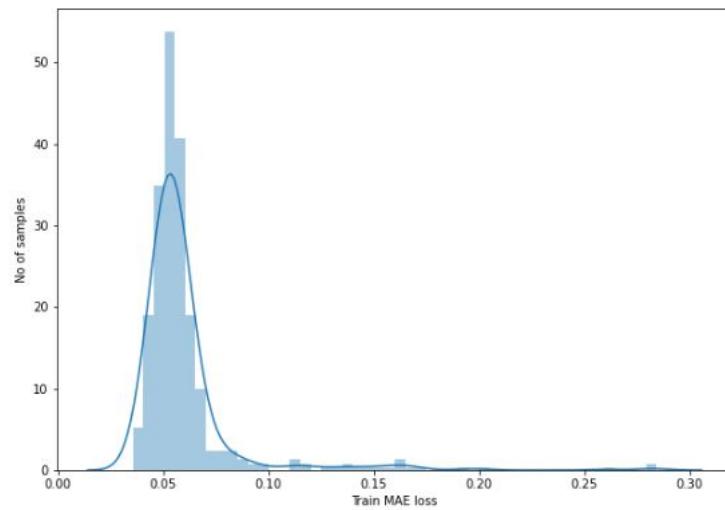


FIGURE 2.69 – Évolution de la fonction de perte MAE sur X_{train}

Évolution de la perte MAE sur l'ensemble de jeu de test

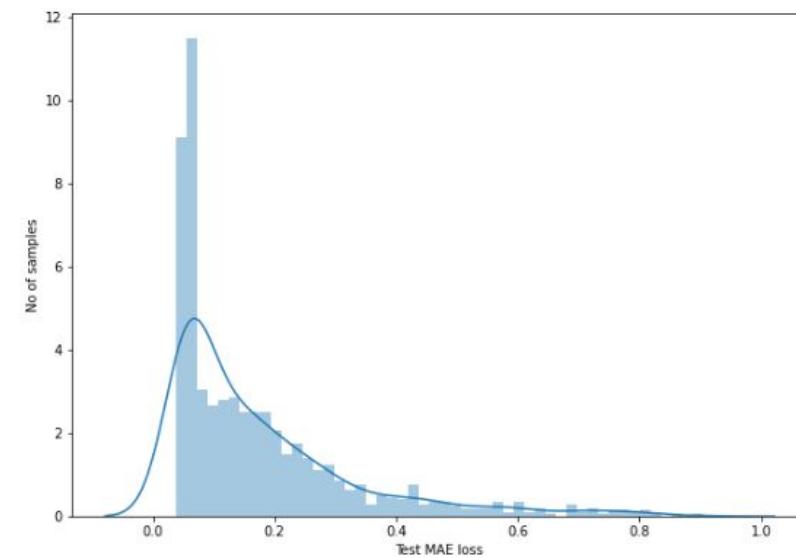


FIGURE 2.70 – Évolution de la fonction de perte MAE sur X_{test}