



---

# Rapport d'avancement

Tentative d'apprentissage des corrélations entre phases de la lune et du soleil et coefficients des marées

---

*Auteurs :*

Otmane EL ALOI, Abdelhafid SAOUD, Abdelhadi ZIANE

*Encadrant :*

Mr. Olivier ROUX

Option :  
Informatique

## Table des matières

1	Bibliographie des modèles de prédiction des séries temporelles univariées	2
1.1	Définition formelle d'une serie temporelle . . . . .	2
1.2	ARIMA . . . . .	2
1.3	Lissage exponentiel Exponential Smoothing) . . . . .	2
1.4	Prophet . . . . .	3
1.5	SVM . . . . .	3
1.6	LSTM . . . . .	4
2	Structuration des fichiers . . . . .	5
3	Etablissement du premier modèle : "Baseline model" . . . . .	5
3.1	Préparation des données . . . . .	6
3.2	Entraînement . . . . .	6
4	Planning . . . . .	9
4.1	Planning de la semaine en cours . . . . .	9
4.2	Planning de la semaine à venir . . . . .	9

## Introduction

Nous avons consacré cette semaine à la recherche bibliographique sur les modèles de prédiction, qui s'intègre dans la phase de « Data understanding » de notre approche CRISP DM de management de projet. De plus, nous avons commencé l'implémentation de quelques modèles sur les données disponibles.

## 1 Bibliographie des modèles de prédiction des séries temporelles univariées

### 1.1 Définition formelle d'une serie temporelle

Une série temporelle est une séquence de points de données, mesurés généralement à des moments successifs, espacés d'un intervalle de temps uniforme. Elle est généralement modélisée par un processus stochastique  $Y(t)$ , c'est-à-dire une séquence de variables aléatoires. Dans le cadre d'une prévision, nous nous trouvons au temps  $t$  et nous sommes intéressés par l'estimation de  $Y(t+h)$ , en utilisant uniquement les informations disponibles au temps  $t$ .

### 1.2 ARIMA

ARIMA (Auto-Regressive Integrated Moving Average) est une approche pour modéliser la variable observée  $x_t$  et suppose que  $x_t$  peut être décomposée en composantes tendancielle, saisonnière et irrégulière, composantes saisonnière et irrégulière. Au lieu de modéliser ces composantes séparément, Box et Jenkins ont eu l'idée de différencier la série temporelle  $x_t$  afin d'éliminer la tendance et la saisonnalité. La série résultante est traitée comme une série de données de série temporelle stationnaire et est modélisée en utilisant une combinaison de ses valeurs de séries temporelles retardées ("AR") et de la moyenne mobile des erreurs de prévision décalées ("MA"). Un modèle ARIMA est généralement spécifié par un tuple  $(p; d; q)$ , où  $p$  et  $q$  définissent les ordres de AR et MA, et  $d$  spécifie l'ordre de l'opération de différenciation[1].

### 1.3 Lissage exponentiel Exponential Smoothing)

Le lissage exponentiel est une méthode de prévision des séries temporelles pour les données univariées. Les méthodes de séries temporelles telles que la famille de méthodes ARIMA, développent un modèle dans lequel la prédiction est une somme linéaire pondérée d'observations ou de décalages passés récents. Les méthodes de prévision de lissage exponentiel sont similaires en ce qu'une prédiction est une somme pondérée d'observations passées, mais le modèle utilise explicitement un poids décroissant de façon exponentielle pour les observations passées. Plus précisément, les observations passées

sont pondérées avec un rapport géométriquement décroissant. Les méthodes de lissage exponentiel peuvent être considérées comme des pairs et une alternative à la classe de méthodes Box-Jenkins ARIMA populaire pour la prévision des séries temporelles[2].

## 1.4 Prophet

Prophet est un modèle de prévision qui permet de traiter de multiples saisonnalités. Il s'agit d'un logiciel open source publié par l'équipe Core Data Science de Facebook. Le modèle Prophet part du principe que la série chronologique peut être décomposée comme suit :  $y(t) = g(t) + s(t) + h(t) + \epsilon(t)$ .

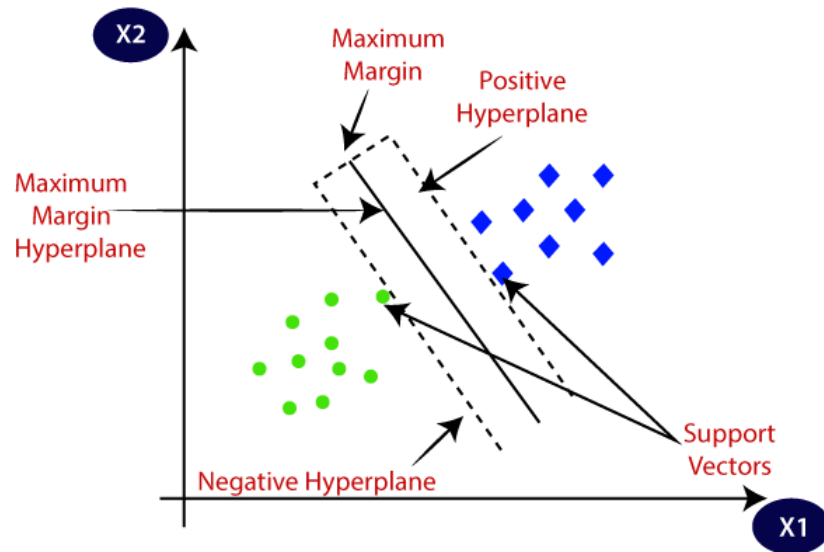
Les trois termes  $g(t)$ ,  $s(t)$  et  $h(t)$  correspondent respectivement à la tendance, à la saisonnalité et aux vacances. Le dernier terme est le terme d'erreur. L'ajustement du modèle est conçu comme un exercice d'ajustement de courbe, il ne prend donc pas explicitement en compte la structure de dépendance temporelle dans les données. Cela permet également d'avoir des observations irrégulièrement espacées. Il existe deux options pour les séries chronologiques à tendance : un modèle de croissance saturant et un modèle linéaire par morceaux. Le modèle de saisonnalité multi-période s'appuie sur les séries de Fourier. L'effet des jours fériés connus et personnalisés peut être facilement incorporé au modèle. Le modèle prophète est inséré dans un cadre bayésien et il permet de faire une inférence a posteriori complète pour inclure l'incertitude des paramètres du modèle dans l'incertitude des prévisions[3].

## 1.5 SVM

Support Vector Machine ou SVM est l'un des algorithmes d'apprentissage supervisé les plus populaires, qui est utilisé pour les problèmes de classification ainsi que de régression. Cependant, il est principalement utilisé pour les problèmes de classification en apprentissage automatique.

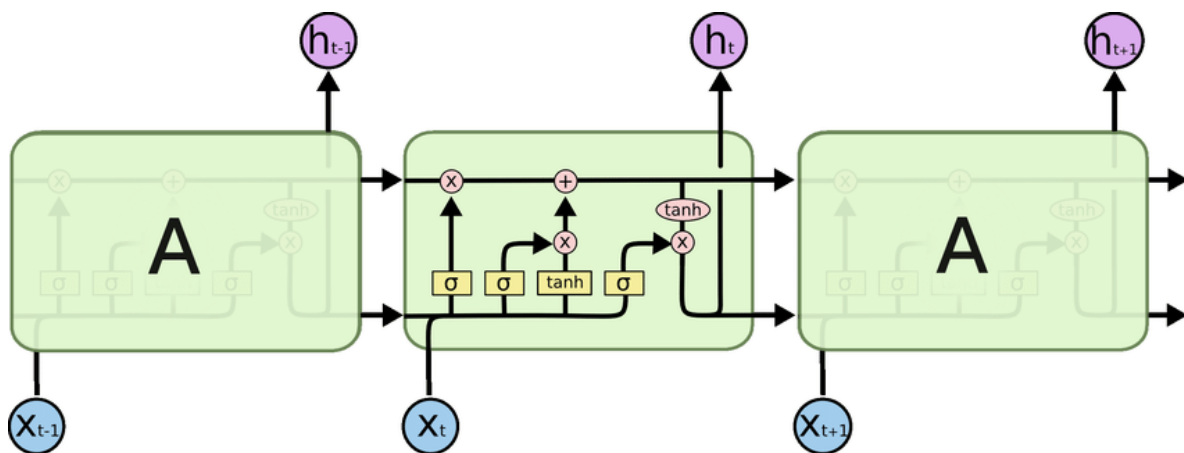
L'objectif de l'algorithme SVM est de créer la meilleure ligne ou limite de décision qui puisse séparer l'espace à  $n$  dimensions en classes afin que nous puissions facilement placer le nouveau point de données dans la bonne catégorie à l'avenir. Cette limite de meilleure décision est appelée hyperplan.

SVM choisit les points/vecteurs extrêmes qui aident à créer l'hyperplan. Ces cas extrêmes sont appelés vecteurs de support et, par conséquent, l'algorithme est appelé machine à vecteurs de support. Considérez le diagramme ci-dessous dans lequel il existe deux catégories différentes qui sont classées à l'aide d'une limite de décision ou d'un hyperplan :



## 1.6 LSTM

Les modèles LSTM peuvent être utilisés pour prévoir les séries temporelles (ainsi que d'autres réseaux neuronaux récurrents). LSTM est un acronyme qui signifie "Long-Short Term Memories". L'état d'un réseau LSTM est représenté par un vecteur d'espace d'état. Cette technique permet de garder la trace des dépendances des nouvelles observations avec les observations passées (même très lointaines). En général, les LSTM sont des modèles complexes et ils sont rarement utilisés pour prédire une seule série temporelle, car ils nécessitent une grande quantité de données pour être estimés. Cependant, ils sont couramment utilisés lorsque des prédictions sont nécessaires pour un grand nombre de séries temporelles[4].



## 2 Structuration des fichiers

L'ensemble de ces fichiers va nous permettre de bien organiser notre travail en tant qu'équipe et aussi d'avoir une flexibilité et une agilité au cours de développement de notre modèle de prédiction des coefficients de marées.

📁 configs	OOP paradigm	37 minutes ago
📁 data	project structure	7 days ago
📁 dataLoader	OOP paradigm	37 minutes ago
📁 evaluations	OOP paradigm	37 minutes ago
📁 models	delete otmane personal analysis from main	11 minutes ago
📁 papers	OOP paradigm	37 minutes ago
📁 trainedModels	OOP paradigm	37 minutes ago
📁 utils	OOP paradigm	37 minutes ago
📄 .gitignore	OOP paradigm	37 minutes ago
📄 LICENSE	Initial commit	16 days ago
📄 README.md	Update README.md	16 days ago

FIGURE 1 – Structuration des fichiers

Description du rôle de chaque fichier :

- **configs** : Paramètres de configuration
- **data** : Contenant les données brutes récupérer auprès de l'encadrant.
- **dataLoader** : Contient les fonctions principales permettant l'import et l'export des données.
- **evaluations** : Contient les outils d'évaluation des modèles développés.
- **models** : Regroupe l'ensemble des modèles développés au cours du travail sur le projet.
- **papers** : Contient les sources du projet.
- **trainedModels** : Dédié à l'entraînement des modèles et leurs résultats.
- **utils** : Concerne tout autre outil qui pourra être utiles pour notre projet.

## 3 Etablissement du premier modèle : "Baseline model"

Pour évaluer la faisabilité de la prédiction, nous avons construit un premier modèle basé sur des forêts aléatoires. Nous avons gardé volontairement pour ce "Baseline" les paramètres par défaut.

### 3.1 Préparation des données

un modèle de machine learning consiste à estimer une fonction  $f$  tel que :

$$Y = f(X)$$

avec  $X$  les données et  $Y$  la(les) valeur(s) cible(s). Dans notre cas, nous disposons seulement d'une longue séquence des valeurs. Il fallait d'abord les mettre en forme pour les modèles de machine learning. Pour ce faire, nous avons utilisé l'approche de la fenêtre glissante illustrée dans la figure 2 :

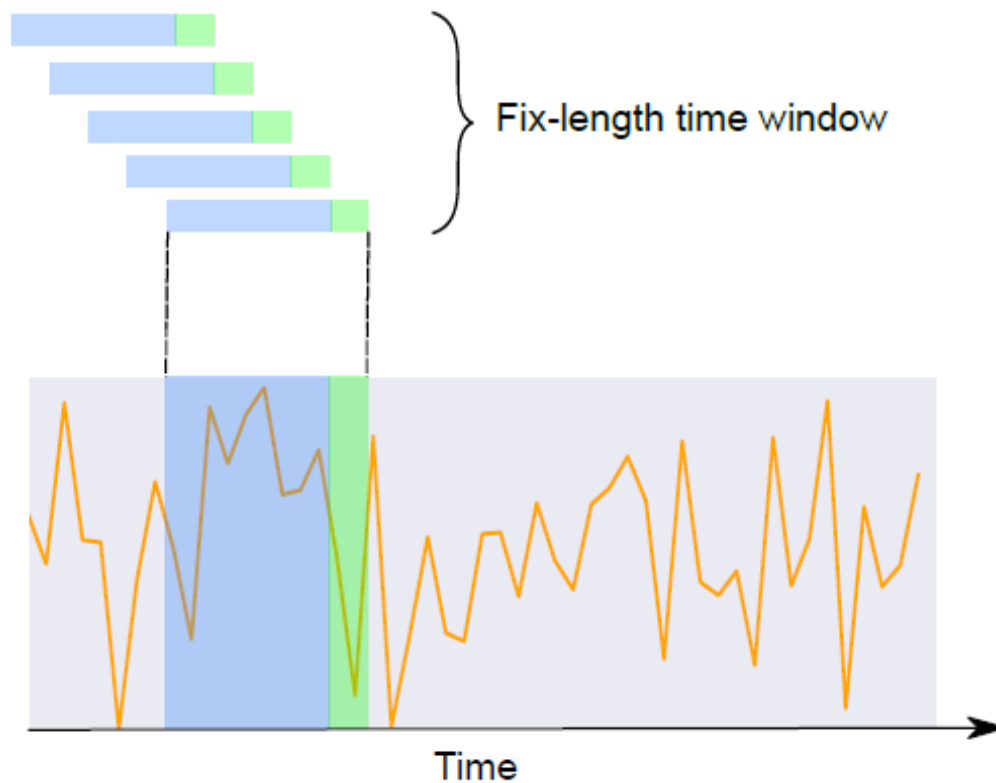


FIGURE 2 – Illustration de la méthode de la fenêtre glissante

Les données

### 3.2 Entraînement

Comme mentionné précédemment, ce modèle sert seulement comme "baseline". La figure suivante montre le résultat de la prédiction :

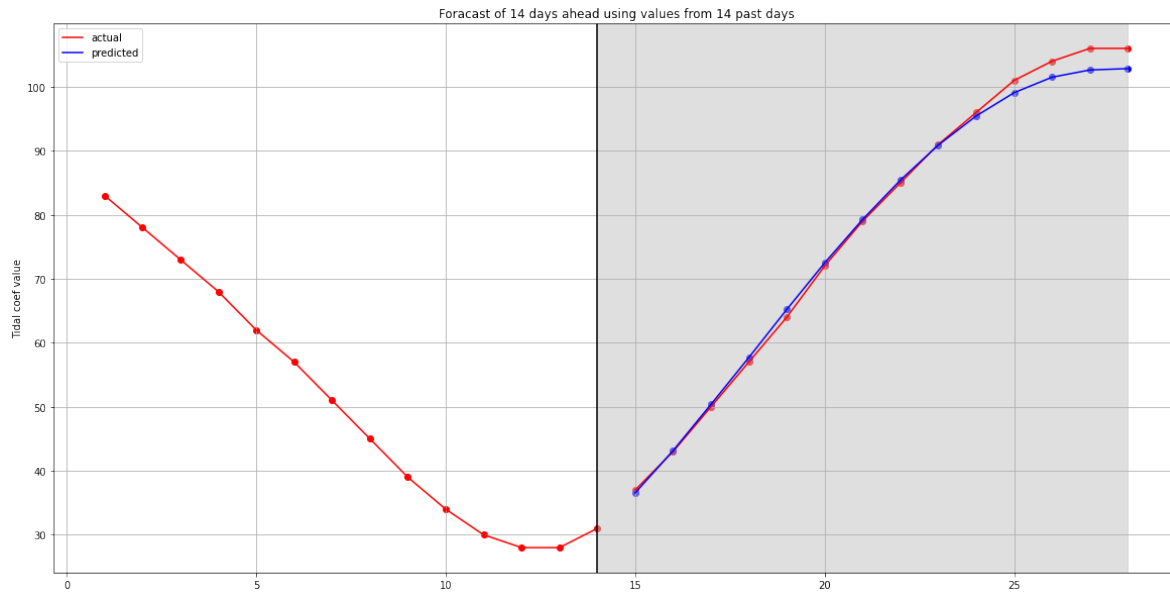


FIGURE 3 – Résultat de la prédiction d'une semaine dans le future

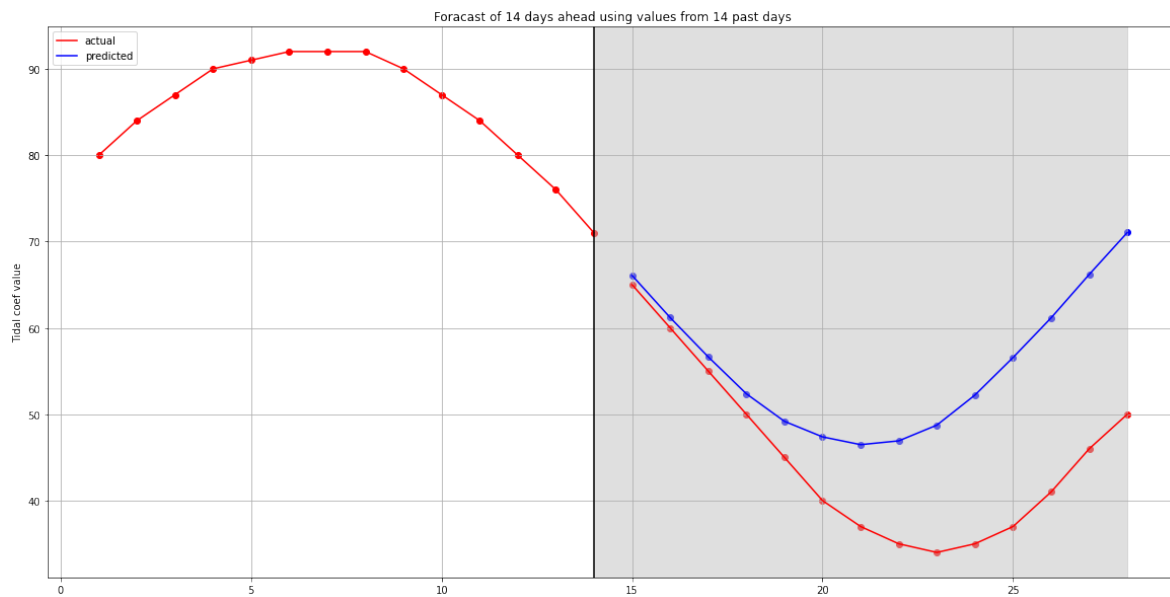


FIGURE 4 – Résultat de la prédiction d'une semaine dans le future

Pour évaluer les modèles, nous travaillons avec la racine de l'erreur moyenne quadratique. Elle à le même échelle que les valeurs des coefficients, elle donne ainsi une approximation directe de combien d'unité en moyenne diffère la valeur prédite de la valeur réelle.



Ci-dessous la courbe des erreurs en fonction du jours de la prédiction. Les erreurs deviennent de plus en plus grandes lorsqu'on veut prédire des valeurs plus loin dans le future, ce qui est tout à fait normale.

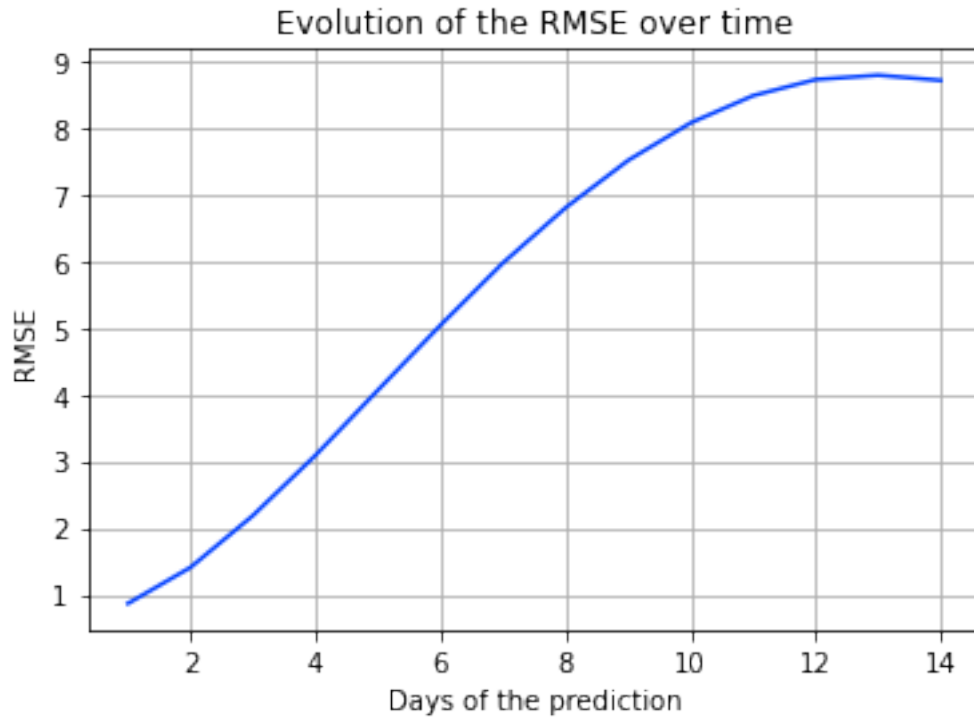


FIGURE 5 – RMSE en fonction de la chronologie de la valeur prédite.

## 4 Planning

### 4.1 Planning de la semaine en cours

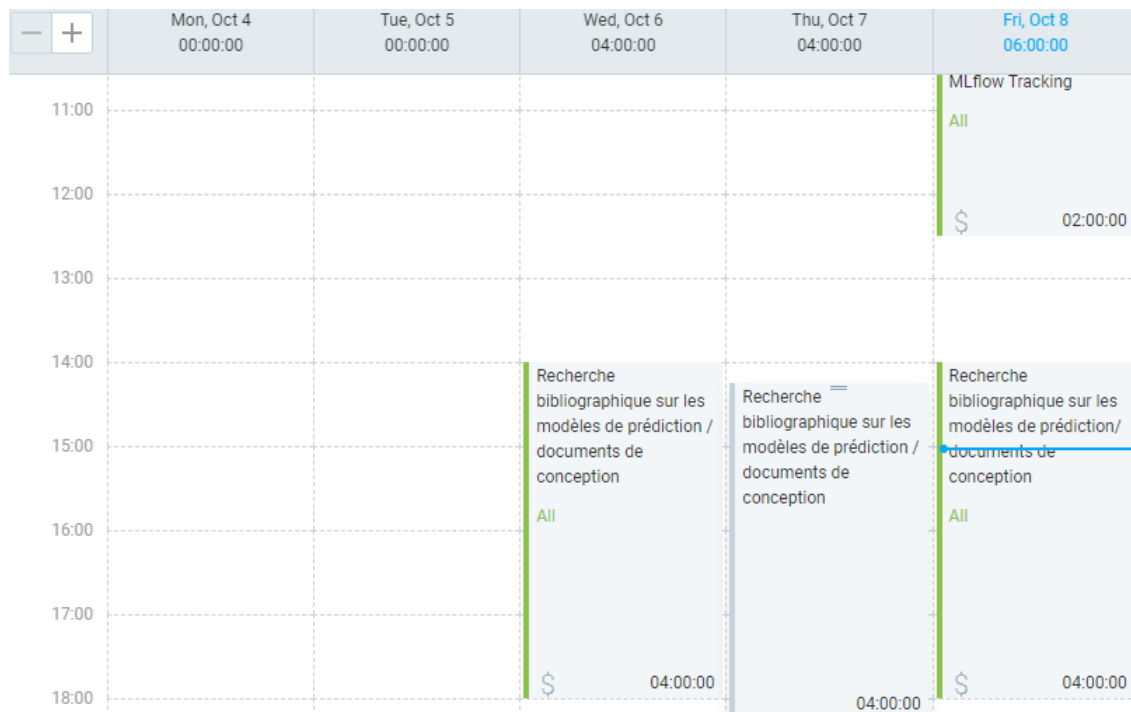


FIGURE 6 – Planning de la semaine en cours

### 4.2 Planning de la semaine à venir

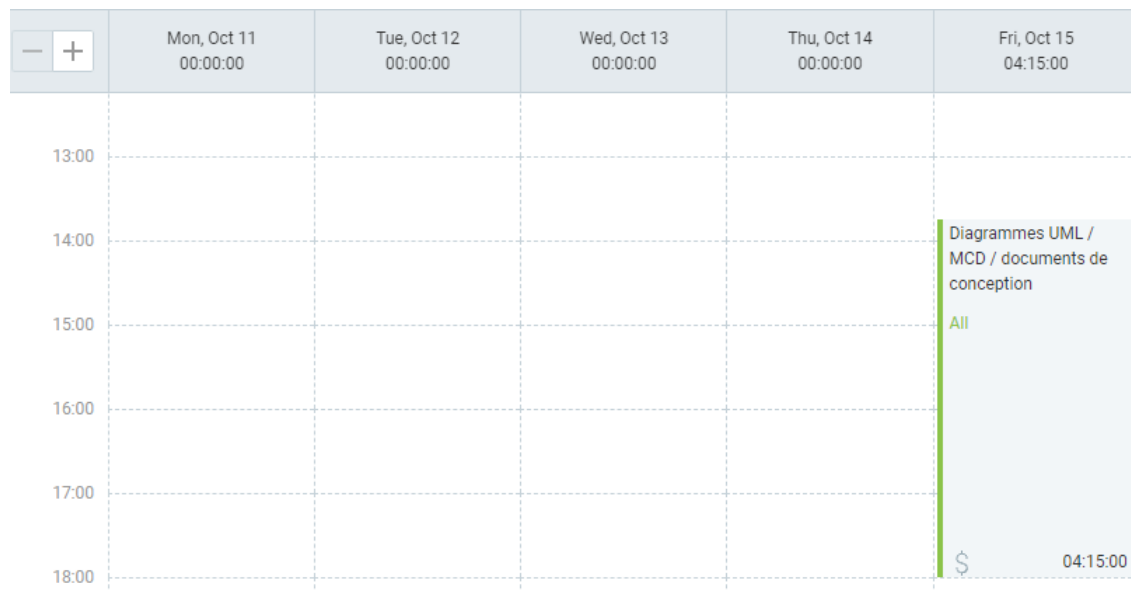


FIGURE 7 – Planning de la semaine à venir

## Références

- [1] MACHINELEARNINGMASTERY, « How to Create an ARIMA Model for Time Series Forecasting in Python, » adresse : <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>.
- [2] —, « A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python, » adresse : <https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/>.
- [3] O. HOME, « Time Series Forecasting using Facebook Prophet library in Python, » adresse : <https://www.analyticsvidhya.com/blog/2020/10/time-series-forecasting-using-facebook-prophet-library-in-python/>.
- [4] colah's BLOG, « Understanding LSTM Networks, » adresse : <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.