



ÉCOLE CENTRALE NANTES

INFO IA - PIIA MARÉES  
RAPPORT

---

## Rapport 3 - PIIA1 Marées

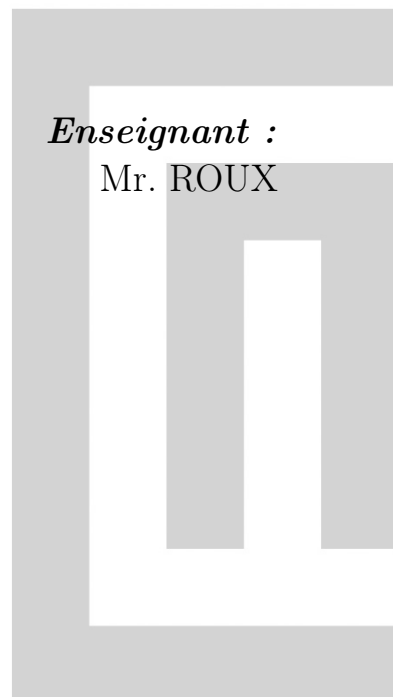
---

***Élèves :***

Daniel MACEDO GALEMBECK  
Nicolas CONTRERAS  
Alexandre THOMASSIN  
Liushuangfei XIE

***Enseignant :***  
Mr. ROUX

17 février 2023



# Table des matières

<b>1</b>	<b>Rapport des deux semaines passées</b>	<b>2</b>
1.1	Dataset . . . . .	2
1.2	Apprentissage Supervisé : Forecast Time-Series Data . . . . .	2
1.3	Modèle Multi-Step : Tensorflow . . . . .	4
1.3.1	Plage de test : . . . . .	4
1.3.2	Fonction d'entraînement : . . . . .	5
1.3.3	Modèle Linéaire : . . . . .	5
1.3.4	Modèle Dense : . . . . .	6
1.3.5	Modèle CNN (Convolutional Neural Network) : . . . . .	7
1.3.6	Modèle RNN (Recurrent Neural Network) : . . . . .	8
1.3.7	Conclusion : . . . . .	8
<b>2</b>	<b>Objectif des deux prochaines semaines</b>	<b>9</b>
2.0.1	Poursuite des recherches sur les modèles prédictifs . . . . .	9
2.0.2	Prédiction du futur . . . . .	9

# 1 Rapport des deux semaines passées

## 1.1 Dataset

Après avoir analysé le set de données, nous avons remarqué les particularités suivantes :

- **1846 - 1857** source 4 par heure
- **1858 et 1859** n'existent pas
- **1860 - 1914** source 4 par heure
- **1915** n'existe pas
- **1916 - 1937** source 4 par heure
- **1938** une seule ligne : 31/12/1938 23:00:00;5.56;4
- **1939 - 1944** source 4 par heure
- **[1942 - 1952]** n'existent pas
- **1953 - 1994** source 4 par heure
- **1995** : Le 10/08/1995 les données de la source 3 commencent à apparaître (Données validées temps différé). Il y a des valeurs entre heure (chaque dix minutes). La valeur de la source 3 est répétée avec la valeur de la source 4.
- **1996 - 2008** : source 3 et 4.
- **2009** : Le 01/12/2009 les données de la source 1 commencent à apparaître (Données brutes temps réel). Ses valeurs sont actualisées chaque minute. Il y a aussi la source 3 (chaque 10 minutes) et la source 4 (par heure).
- **2010** : Source 1, 3, et 4.
- **2011** : Le 31/05/2011 les données de la source 2 commencent à apparaître (Données brutes temps différé). Ses valeurs sont actualisées chaque 10 minutes. Il y a aussi la source 1, 3 et 4.
- **2012 - 2021** : Source 1, 2, 3, et 4.
- **2022** : Le 30/06/2022 est la dernière date à laquelle il y a des valeurs de source 3 et 4. Après il y a juste source 1 et 2.

Il est important de noter que toutes les données ne commencent pas le premier jour de janvier. Par exemple, la première donnée pour l'année 2020 est 02/01/2020 09:35:50;5.1717;1. Il en va de même pour les années 2000, 1993, 1860, entre autres.

## 1.2 Apprentissage Supervisé : Forecast Time-Series Data

Après enquête, nous avons réalisé que les données peuvent être traitées comme un problème de Time-Series forecasting. Pour cette raison, nous avons appliqué aux données un modèle SVM (Support vector machine) de apprentissage supervisé.

Pour l'entraînement du modèle, nous avons utilisé les données sources des années [1954 - 1957]. Nous pouvons regarder les resultats de la prediction pour l'année 1958 dans la image suivant :

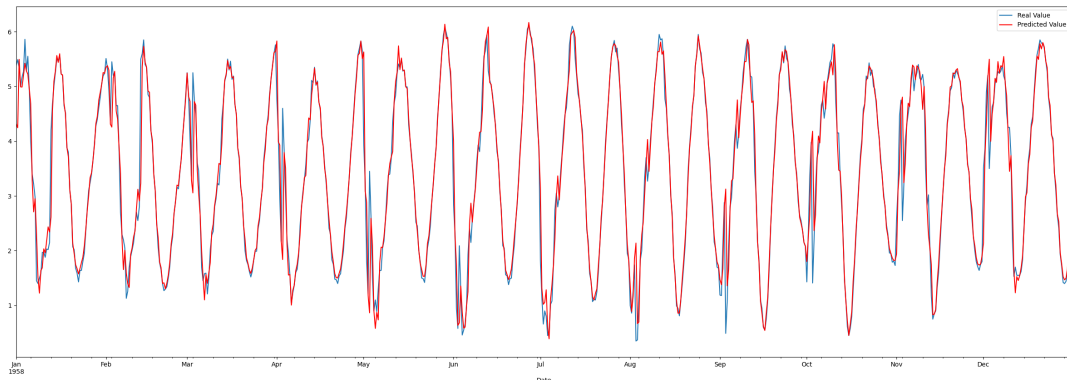


FIGURE 1 – Prediction 1958

Voici les métriques de la prediction des données de test :

	MSE	RMSE	R^2	MAE	MedAE	MaxErr
0	0.134127	0.366233	0.943784	0.161769	0.065375	2.771175

FIGURE 2 – Métriques de la prédiction 1958

Ayant constaté les bons résultats du modèle précédent, nous avons entraîné le même modèle, cette fois avec les données des années 2010 à 2020, et l'avons testé avec les années 2021 et 2022. Voici les résultats :

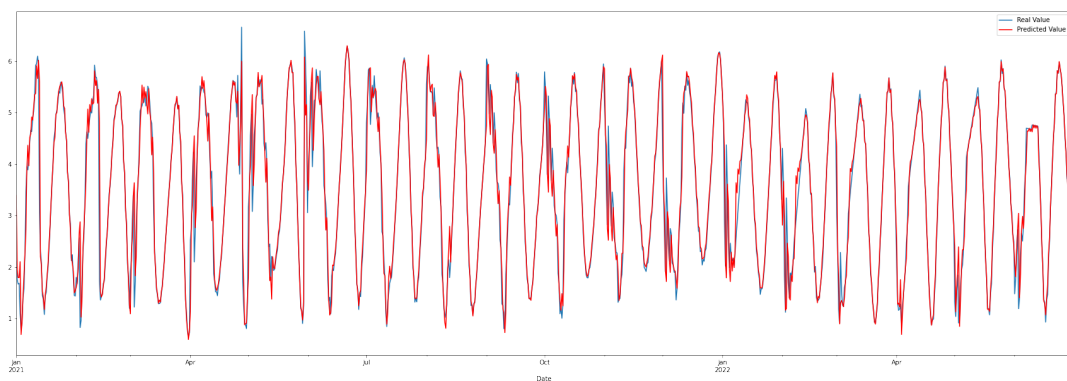


FIGURE 3 – Predictions des années 2021 et 2022

Voici les résultats de la prediction des données de test :

	MSE	RMSE	R^2	MAE	MedAE	MaxErr
0	0.130566	0.361339	0.942309	0.158102	0.052407	3.056119

FIGURE 4 – Métriques de la prédiction des années 2021 et 2022

En résumé, le modèle présente de bonnes métriques, mais nous n'avons pas encore été en mesure de prédire les valeurs futures des marées, c'est-à-dire pour les jours ou années à venir, comme 2023.

### 1.3 Modèle Multi-Step : Tensorflow

Nous avons aussi essayé d'entraîner différent modèle à l'aide de la bibliothèque Tensorflow.

#### 1.3.1 Plage de test :

Afin de réaliser les test on choisit des plages qui resteront les mêmes pour chaque test afin de pouvoir comparer les précisions des modèles.

On choisit les plages suivantes :

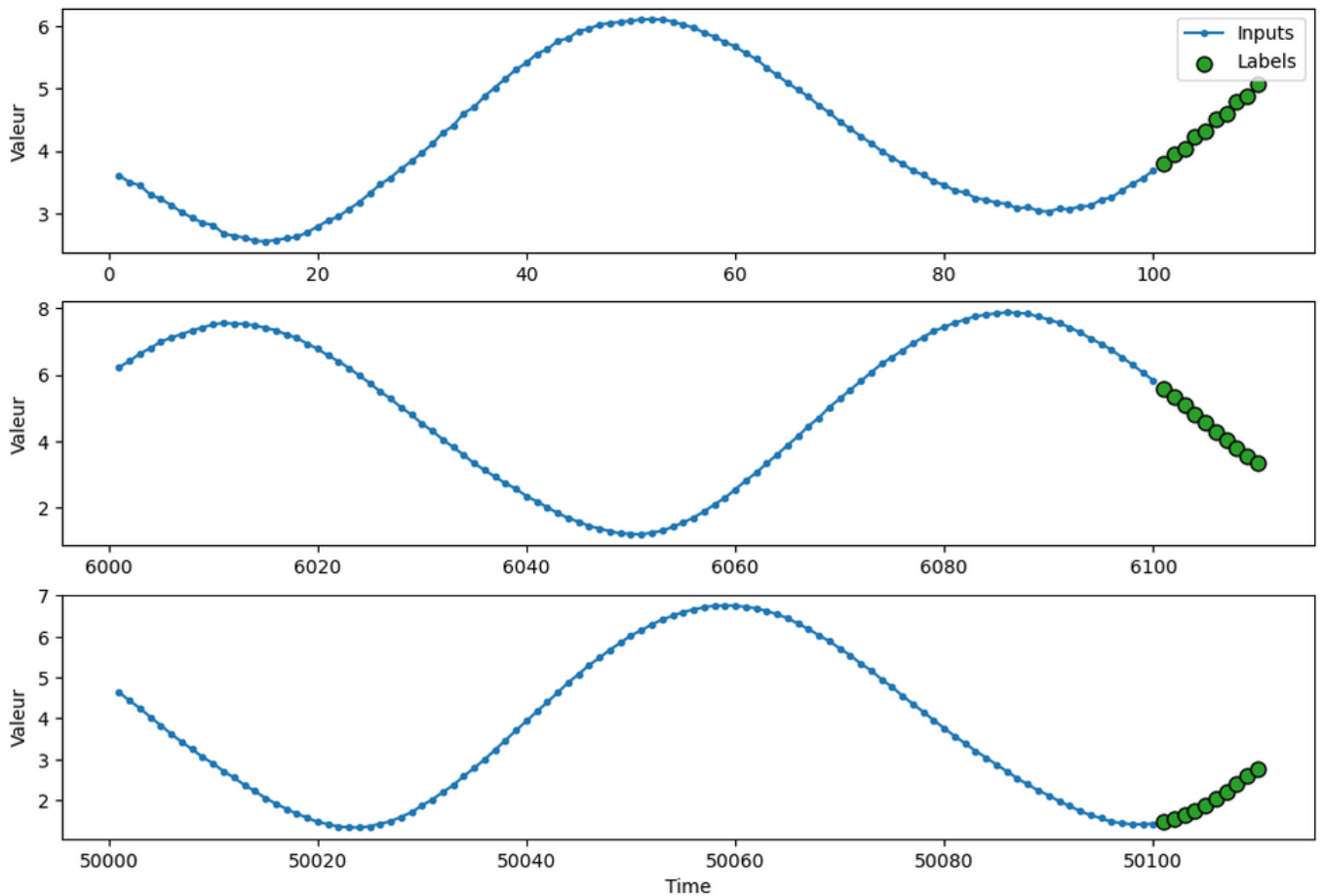


FIGURE 5 – Test windows

Les points bleus étant les données sur lesquelles le modèle va se baser afin de prédire les prochaines points, ici l'objectif de la prédiction est représentée par les points en vert.

### 1.3.2 Fonction d'entraînement :

On définit notre fonction d'entraînement du modèle comme ci-dessous :

```

1 MAX_EPOCHS = 50
2 def compile_and_fit(model, window, patience=2):
3     early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',
4                                                         patience=patience,
5                                                         mode='min')
6
7     model.compile(loss=tf.losses.MeanSquaredError(),
8                   optimizer=tf.optimizers.Adam(),
9                   metrics=[tf.metrics.MeanAbsoluteError()])
10
11     history = model.fit(window.train, epochs=MAX_EPOCHS,
12                         validation_data=window.val,
13                         callbacks=[early_stopping])
14     return history

```

On définit un objet de classe EarlyStopping afin d'arrêter l'entraînement si celui-ci n'est plus efficace afin de gagner du temps.

On se base sur la valeur MSE (mean squared error) afin de juger l'efficacité de l'entraînement.

### 1.3.3 Modèle Linéaire :

Si l'on essaye de prédire les données à l'aide d'un modèle linéaire on obtient les résultats suivant :

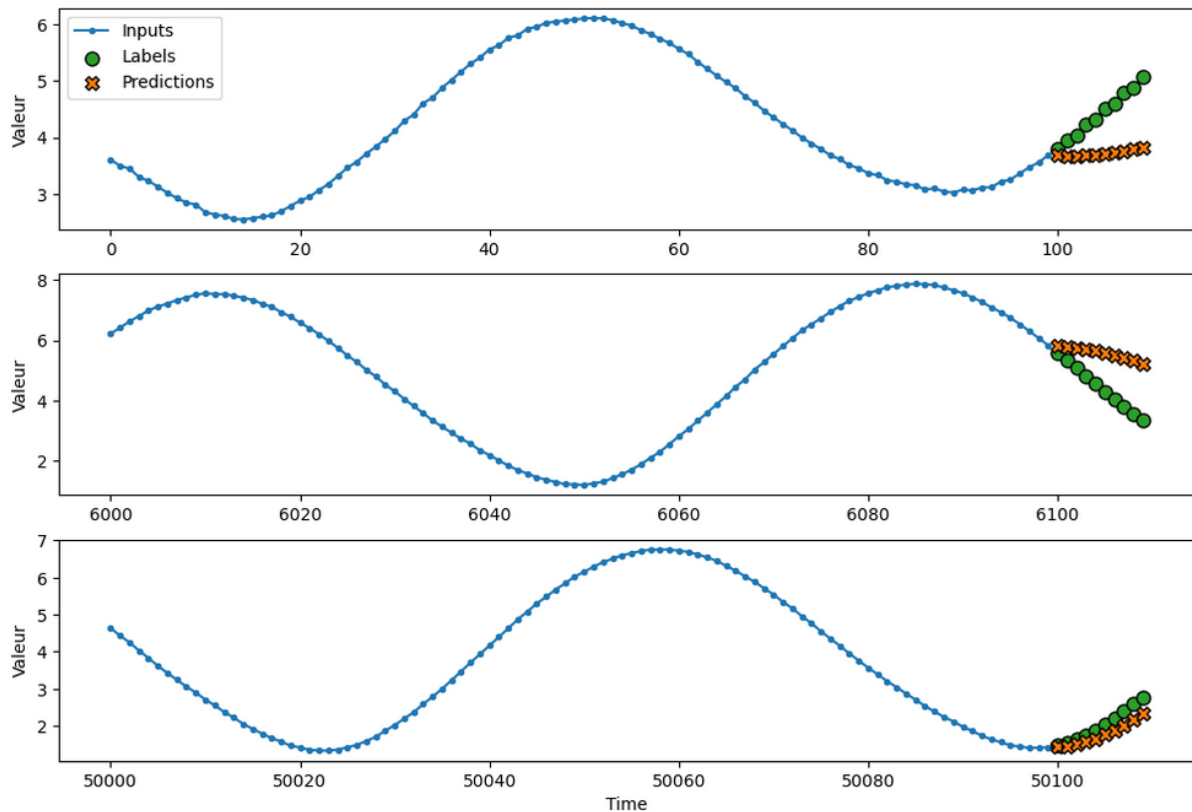


FIGURE 6 – Modèle Linéaire

On remarque que le modèle linéaire n'est pas du tout précis, en parti par le fait que les données ont un profil sinusoïdal. On pourrait augmenter la taille des données d'entrées mais cela augmenterait le temps d'entraînement.

### 1.3.4 Modèle Dense :

Le modèle Dense est basé sur la couche du même nom, elle est souvent utilisé en complément d'autre couche de neurones.

Avec ce modèle, on obtient les résultats suivant :

2111/2111 [=====] - 4s 2ms/step - loss: 0.6057 - mean\_absolute\_error: 0.6088

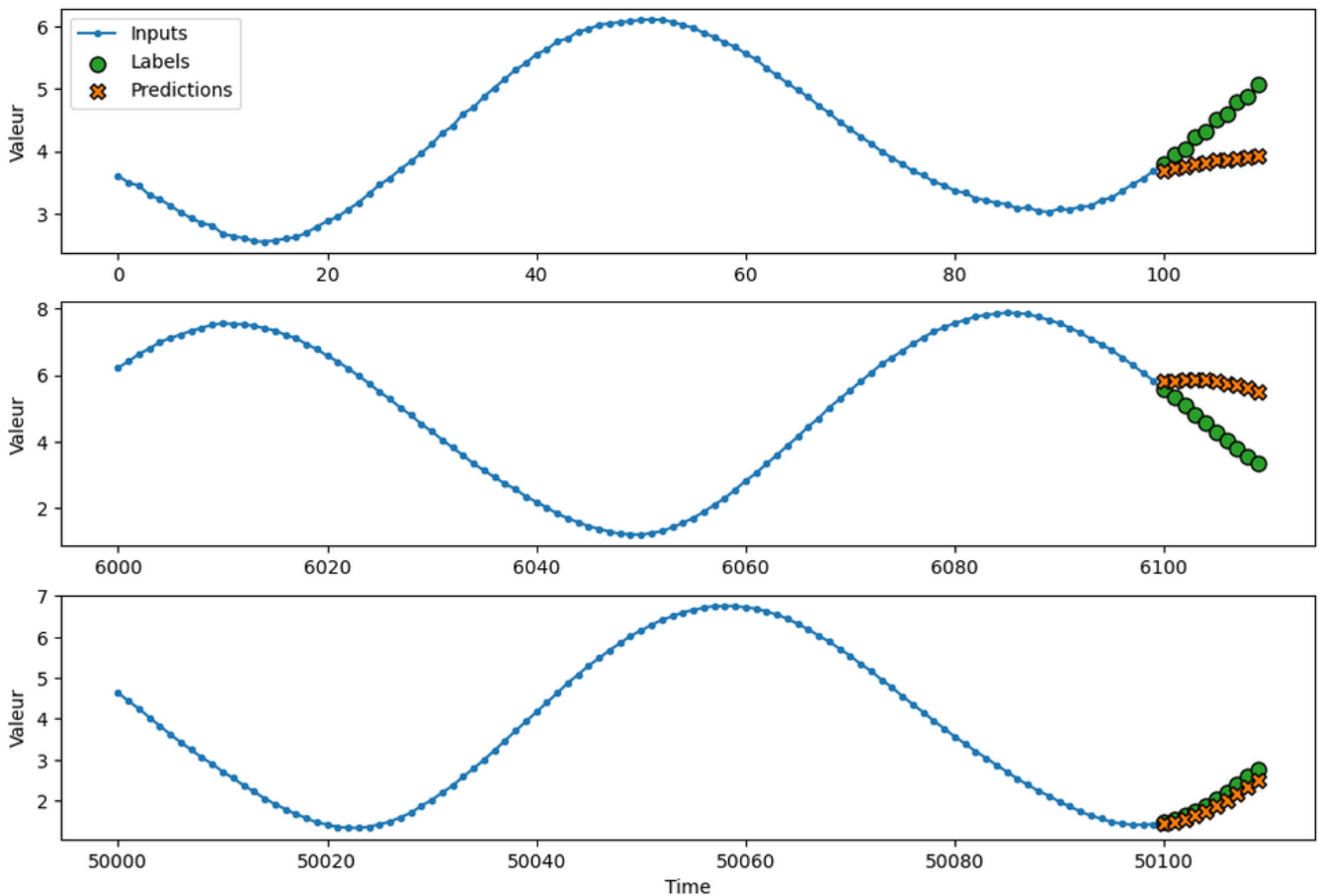


FIGURE 7 – Modèle Dense

Ce modèle étant peu complexe et le nombre de données étant assez faible, la précision est faible. Note : On peut augmenter le nombre de données d'entrées pour augmenter un peu la précision cependant les graphiques ne sont plus lisibles.

### 1.3.5 Modèle CNN (Convolutional Neural Network) :

Le modèle CNN est un modèle un peu plus complexe que les précédents. On obtient alors les résultats suivant :

2111/2111 [=====] - 3s 1ms/step - loss: 0.0086 - mean\_absolute\_error: 0.0504

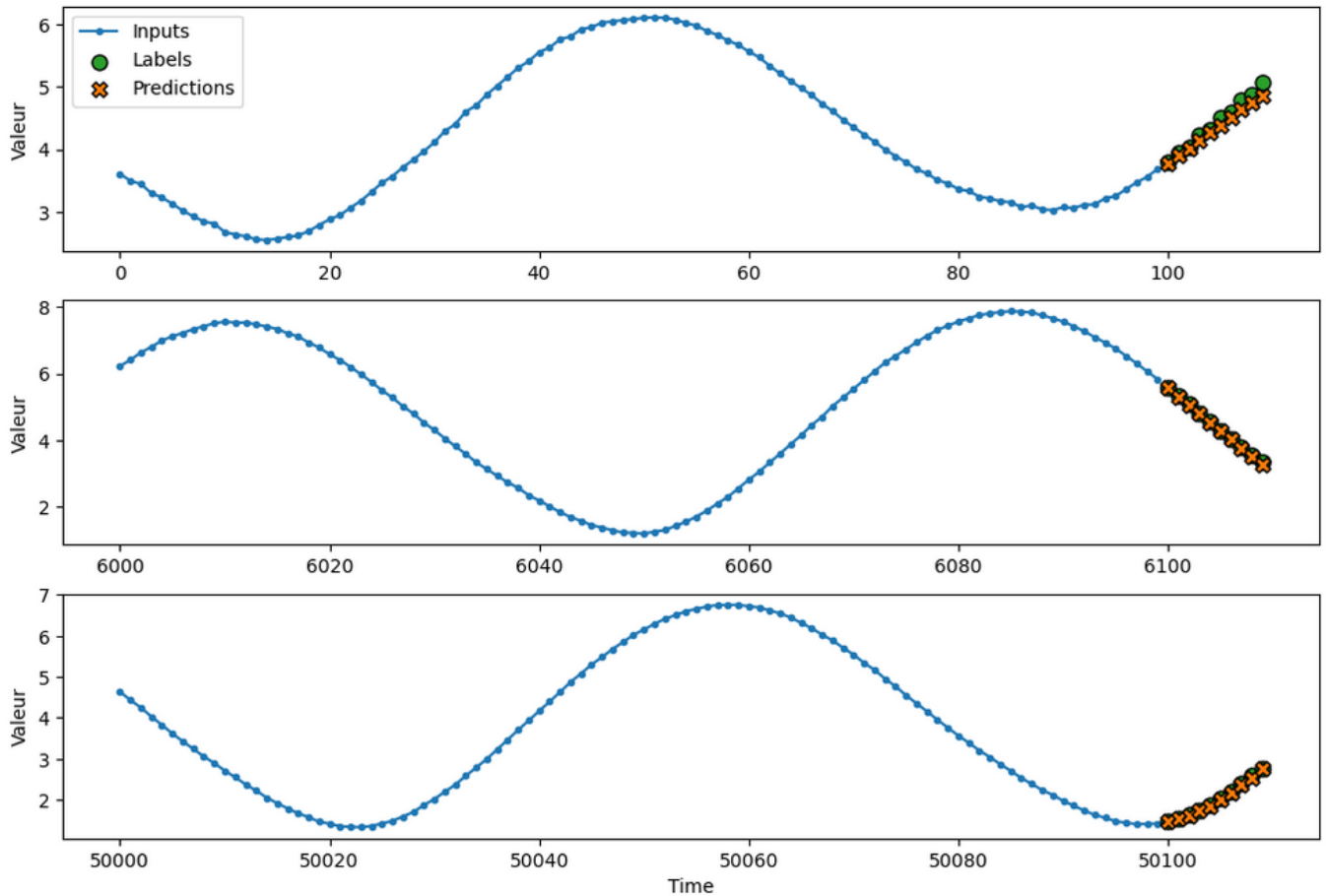


FIGURE 8 – Modèle CNN

Le modèle étant bien plus complexe, les résultats sont bien plus précis qu'avec les autres modèles.



### 1.3.6 Modèle RNN (Recurrent Neural Network) :

Le modèle RNN est encore un peu plus complexe que le précédent et utilise des couches LSTM (Long Short Time Memory).

On obtient les résultats suivant :

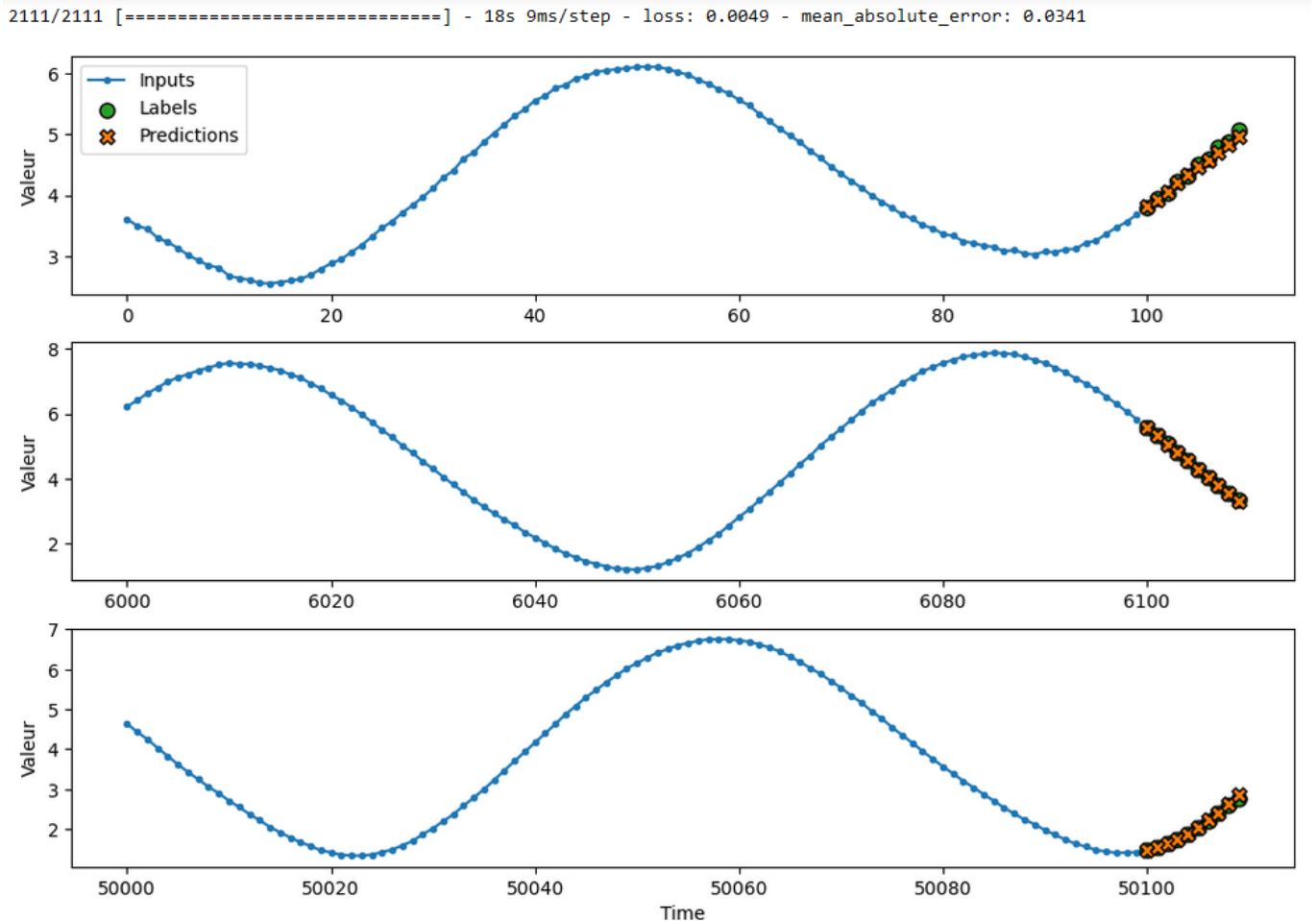


FIGURE 9 – Modèle RNN

Les résultats sont un peu plus précis que le modèle CNN, cependant l'entraînement a pris quasiment 45 minutes comparé à environ 2-3 minutes pour le modèle CNN. Malgré cela le modèle RNN ne sera pas autant limité sur des plus grand entraînement que le modèle CNN.

### 1.3.7 Conclusion :

On remarque que les modèles plus complexe obtiennent une bien meilleur précision lors des entraînement au prix d'un temps d'entraînement potentiellement plus important. Cependant, les modèles CNN et RNN n'ont pas des différence de précision si importante dans le cas de petit jeux de données, il peut donc être intéressant de sacrifier un peu de précision pour gagner un temps significatif.

## 2 Objectif des deux prochaines semaines

### 2.0.1 Poursuite des recherches sur les modèles prédictifs

Bien que nous ayons déjà obtenu de bons résultats avec les modèles que nous avons étudiés, nous allons continuer à chercher et à apprendre l'apprentissage automatique. L'objectif est de pouvoir comparer tous les modèles sélectionnés et de conserver ceux qui donnent les meilleurs résultats.

### 2.0.2 Prédiction du futur

Nous allons étudier les moyens de sauvegarder les modèles entraînés et de les mettre à jour avec les nouvelles données que nous recevons. De la même manière, nous allons essayer de prédire le temps futur, c'est-à-dire le temps pour lequel nous n'avons pas des données réelles.