Rapport 8INF972

8INF972 – Atelier pratique en IA

Présenté par

GUICHARD Tristan : GUIT13020005

LIZÉ Louise : LIZL16580007

BIZOT – PAGEOT Kyllian : BIZK17040001

Hiver 2023

Table des matières

1 Introduction

Dans un contexte où le marché des changes est l'un des plus grands et des plus liquides au monde, les investisseurs ont besoin d'outils pour les aider à prendre des décisions éclairées en matière d'investissement. C'est dans ce cadre que notre entreprise souhaite développer une application web pour aider les investisseurs à suivre de près le taux de change entre l'Euro (EUR) et le Dollar Canadien (CAD), tout en fournissant des conseils et des prévisions d'investissement basés sur des techniques d'intelligence artificielle.

Notre objectif est de développer une application web utilisant des technologies modernes telles que Python et des bibliothèques telles que Streamlit, Plotly et TensorFlow pour offrir une interface utilisateur conviviale et interactive. Les utilisateurs pourront ainsi visualiser des graphiques interactifs pour le taux de change EUR/CAD, recevoir des conseils d'investissement basés sur des algorithmes d'IA simples, et faire des prévisions de taux de change EUR/CAD à l'aide de modèles basés sur des données temporelles.

En outre, nous veillerons à garantir la sécurité de nos utilisateurs en développant une fonctionnalité de connexion pour accéder à l'application, ainsi qu'une section de contact pour répondre aux questions et aux besoins d'assistance des utilisateurs.

Enfin, nous sommes fiers de présenter les membres de notre équipe talentueuse qui ont travaillé sur ce projet et de partager les résultats de notre travail.

2 Contexte et besoin

Le marché des changes est l'un des marchés financiers les plus importants et les plus liquides au monde. Les investisseurs, les entreprises et les gouvernements achètent et vendent des devises pour différentes raisons, telles que le commerce international, les investissements étrangers et les voyages. La valeur d'une devise peut fluctuer rapidement en fonction de divers facteurs économiques, politiques et sociaux, ce qui rend la prise de décision difficile pour les investisseurs.

Dans ce contexte, notre entreprise souhaite développer une application web qui permettrait aux investisseurs de suivre de près le taux de change entre l'Euro (EUR) et le Dollar Canadien (CAD), ainsi que de fournir des conseils et des prévisions d'investissement basés sur des techniques d'intelligence artificielle. Les principales fonctionnalités de l'application seraient :

— Visualisation de graphiques interactifs pour le taux de change EUR/CAD.
— Génération de conseils d'investissement basés sur des algorithmes d'IA.
— Prévisions de taux de change EUR/CAD à l'aide de modèles LSTM ou GRU .
— Informations de contact pour les utilisateurs souhaitant poser des questions ou obtenir de l'aide.

L'objectif de cette application est d'aider les investisseurs à prendre des décisions plus éclairées en matière d'investissement sur le marché des changes, en utilisant des techniques d'IA pour fournir des conseils et des prévisions précises et fiables.

3 Implémentation de l'intelligence artificielle

3.1 Récupération des données

Pour nos données, on utilise la librairie yfinance, qui contient une API de Yahoo finance qui permet de récupérer le taux de change de l'Euro vers le Dollar Canadien. On récupère les données du premier janvier 2015 à aujourd'hui, ce qui a l'avantage d'avoir des données à jour quotidiennement. Les données récupérées sont la valeur du taux de change à l'ouverture, la fermeture, le maximum et le minimum et le volume (colonne à 0) pour chaque jour.

	Open float64	High float64	Low float64	Close float64	Adj Close float64
	1.31239998340606...	1.31793999671936...	1.30260002613067...	1.31270003318786...	1.31270003318786...
2015-01-01...	1.4049600362777 71	1.4049600362777 71	1.4049600362777 71	1.4049600362777 71	1.4049600362777 71
2015-01-02...	1.4043300151824 951	1.4105000495910 645	1.4021999835968 018	1.4044500589370 728	1.4044500589370 728
2015-01-05...	1.4094400405883 79	1.4112499952316 284	1.4020999670028 687	1.4084999561309 814	1.4084999561309 814
2015-01-06...	1.4033000469207 764	1.4061299562454 224	1.3996000289916 992	1.4033000469207 764	1.4033000469207 764
2015-01-07...	1.4053000211715 698	1.4089399576187 134	1.3976900577545 166	1.4055000543594 36	1.4055000543594 36
2015-01-08...	1.3982000350952 148	1.3993300199508 667	1.3881000280380 25	1.3983000516891 48	1.3983000516891 48
2015-01-09...	1.3942999839782 715	1.4031599760055 542	1.3940299749374 39	1.3947000503540 04	1.3947000503540 04
2015-01-12...	1.4056400060653 687	1.4114199876785 278	1.4002000093460 083	1.4059499502182 007	1.4059499502182 007
2015-01-13...	1.4157999753952 026	1.4187400341033 936	1.4036999940872 192	1.4157999753952 026	1.4157999753952 026
2015-01-14...	1.4076999425888 062	1.4160900115966 797	1.4055999517440 796	1.4076000452041 626	1.4076000452041 626

1872 rows, showing 10 per page Page 1 of 188

FIGURE 1 – Données collectées avec yfinance

3.2 Prétraitement des données

Afin de tirer un maximum de potentiel de nos données, nous avons réalisés une ACP sur nos variables (ouverture, fermeture, maximum, minimum, cours de clôture ajusté) afin de mieux les comprendre. Suite à celle-ci, nous avions vu que ces variables sont toutes fortement corrélées. Ce qui est assez cohérent car ces dernières sont très liées. A chaque donnée x construite ainsi, on associera une valeur y qui correspond à la valeur de taux de change à l'ouverture du jour suivant la dernière ligne de x, valeur que notre modèle devra prédire. Ainsi, une fois nos données construites, on a également choisi de les standardiser afin d'avoir des données compris dans l'intervalle [-1,1], ce qui nous a permis d'améliorer l'apprentissage

Nous avons fait également fait quelques analyses descriptives afin de nous familiariser avec notre jeu de données. Par exemple, nous nous sommes demandés si le cours de change pouvait être influencé par le jour de la semaine ou par le mois (figure 2). Ici, on ne peut pas relever une différence entre les jours de la semaine qui ont une boxlot plus que similaires. Cependant, au niveau des mois, on remarque une plus petite disparité en fin d'année, surtout en octobre, à l'opposé d'une plus grande disparité au printemps (mars, avril et mai).

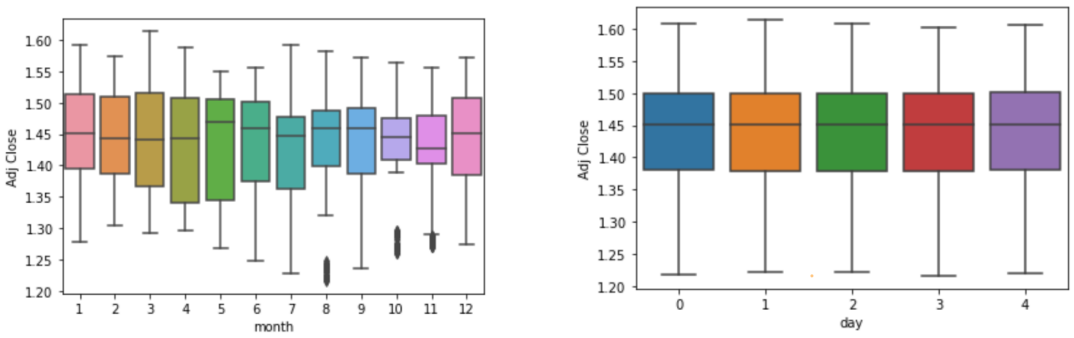


FIGURE 2 – Analyse descriptive

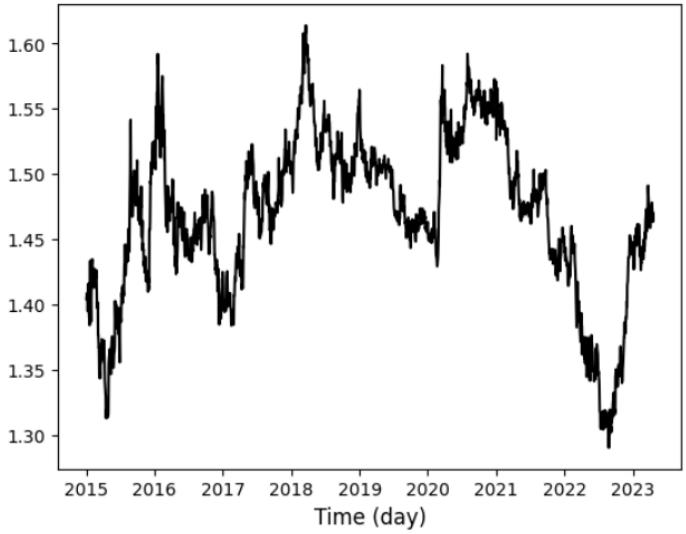


FIGURE 3 – Taux de change EUR-CAD en fonction du temps entre 2015 et 2023

3.3 Implémentation du modèle

Pour construire notre modèle, nous avons choisi d'utiliser un RNN, qui est un modèle adapté pour les données temporelles. Durant le projet, à l'aide de recherche sur le sujet et de tests, plusieurs modèles ont été retenus, particulièrement LSTM, BILSTM et GRU. Chacun de ces modèles ont été implémentés avec la librairie *tensorflow*.

RNN standard

Pour mieux comprendre nos modéles, nous allons faire une brève description des RNN.

Un réseau de neurones récurrents est un type de réseau de neurones artificiels utilisé pour traiter des données séquentielles. Contrairement aux réseaux de neurones classiques,

les RNN ont des connexions récurrentes, ce qui leur permet de prendre en compte les informations précédentes lors du traitement de nouvelles données.

Le fonctionnement d'un RNN peut être décrit en trois étapes :

1. L'entrée : le RNN prend en entrée une séquence de données. Dans notre cas, il s'agit d'une séquence de valeurs numériques représentant les taux de change des dernières années.
2. Le traitement : le RNN traite chaque élément de la séquence un par un, en utilisant des informations de l'état précédent du réseau. À chaque étape, le RNN calcule une sortie et une nouvelle représentation de l'état interne du réseau (cf. figure 4). En utilisant l'historique des taux de change, le RNN peut apprendre à détecter des tendances ou des motifs saisonniers dans les données, et à estimer le taux de change futur.
3. La sortie : à la fin du traitement de la séquence, le RNN peut produire une sortie finale basée sur l'état final du réseau. Par exemple, la sortie finale pourrait être une estimation du taux de change pour la prochaine heure ou le prochain jour.

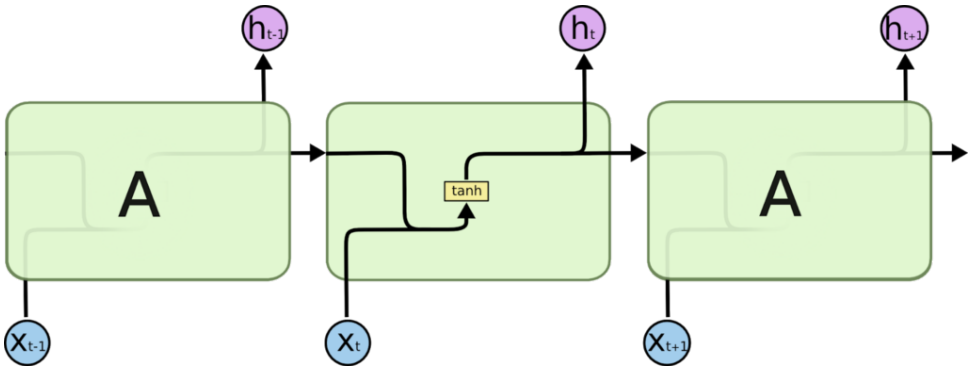


FIGURE 4 – Diagramme fonctionnement RNN

Le traitement des données dans un RNN est réalisé par des cellules RNN, qui sont des blocs de construction fondamentaux du réseau. Les cellules RNN contiennent une fonction d'activation non linéaire (dans le schéma ici, il s'agit de *tanh*) et une mémoire interne qui stocke des informations sur les états précédents du réseau.

LSTM

Les LSTM (Long Short-Term Memory) sont une variante de réseaux de neurones récurrents (RNN) qui ont été spécialement conçus pour traiter des séquences de données de longue durée en évitant les problèmes de vanishing gradient qui se posent souvent avec les RNN classiques.

Les LSTM ont un fonctionnement similaire aux RNN, mais leur architecture est plus complexe. Contrairement aux RNN standard, les LSTM ont des cellules de mémoire qui leur permettent de stocker et de rappeler des informations sur une séquence d'entrée. Les cellules de mémoire sont contrôlées par trois portes différentes : la porte d'entrée, la porte de sortie et la porte d'oubli. (cf. figure 7)

La porte d'entrée contrôle la quantité d'information qui doit être stockée dans la cellule de mémoire. La porte de sortie détermine la quantité d'information qui doit être extraite de la cellule de mémoire. Enfin, la porte d'oubli permet de supprimer des informations qui ne sont plus utiles ou qui peuvent perturber le modèle.

En combinant ces portes de manière appropriée, les LSTM sont capables de modéliser des relations à long terme entre les éléments d'une séquence d'entrée, tout en évitant les problèmes de vanishing gradient. C'est pourquoi les LSTM sont particulièrement bien adaptés à la reconnaissance de la parole, à la traduction automatique, à la prédiction de séries temporelles et à d'autres tâches impliquant des données de séquence à long terme.

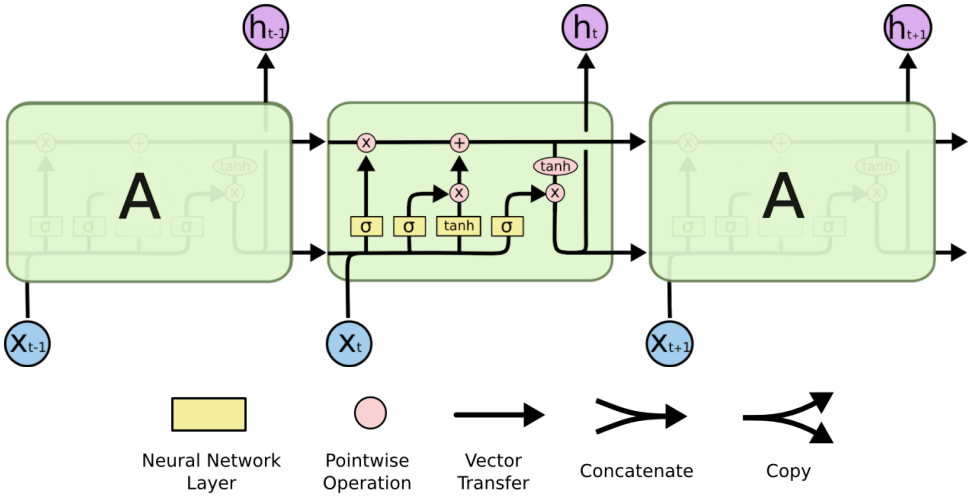


FIGURE 5 – Diagramme de fonctionnement du LSTM

Source : analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes

BILSTM

Le BiLSTM (Bidirectional LSTM) est une variante du LSTM qui permet de prendre en compte l'information contextuelle à la fois en avant et en arrière d'une séquence, contrairement au LSTM qui ne considère que l'information en avant.

Le BiLSTM fonctionne en utilisant deux LSTM, l'un qui traite la séquence dans l'ordre des éléments et l'autre qui traite la séquence dans l'ordre inverse (cf. figure 6). Les sorties des deux LSTM sont ensuite concaténées pour former la représentation finale de la séquence.

Cette approche permet au BiLSTM de prendre en compte l'information contextuelle à la fois en avant et en arrière d'une séquence, ce qui peut être très utile pour les tâches qui nécessitent une compréhension globale de la séquence. Par exemple, dans le cas de la reconnaissance de la parole, le BiLSTM peut prendre en compte le contexte précédent et suivant pour mieux comprendre le mot en cours de prononciation.

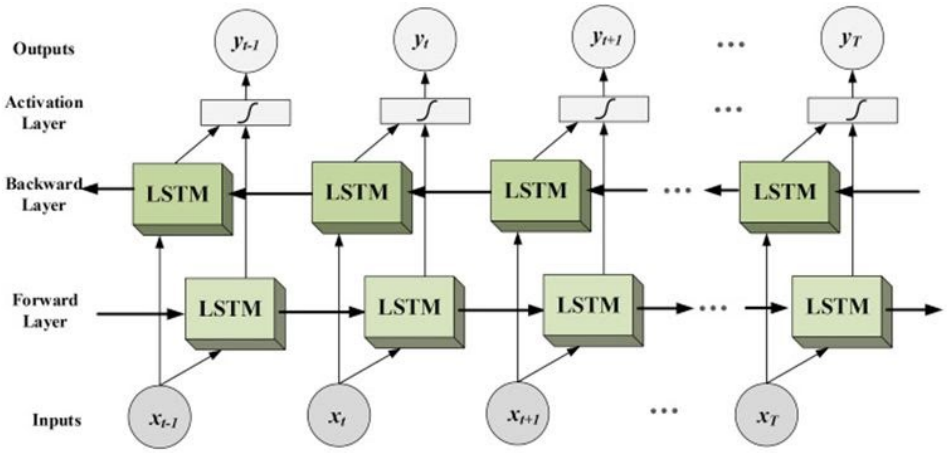


FIGURE 6 – Diagramme fonctionnement BILSTM

Source : analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes

GRU

Les GRU (Gated Recurrent Units) sont similaires aux LSTM. Cependant, les GRU ont une architecture plus simple que les LSTM. Ils utilisent moins de portes (deux portes dans les GRU contre trois dans les LSTM) (cf. figure 7) et ils ont moins de paramètres à entraîner. Cela peut rendre les GRU plus faciles à entraîner et plus rapides à exécuter que les LSTM, tout en obtenant des performances similaires sur de nombreuses tâches de traitement du langage naturel.

Les GRU ont également une connexion de réinitialisation, qui permet au réseau d'oublier plus facilement les informations inutiles ou redondantes. Cette connexion de réinitialisation permet aux GRU de mieux gérer les dépendances à long terme dans les séquences, tout en évitant la sur-apprentissage.

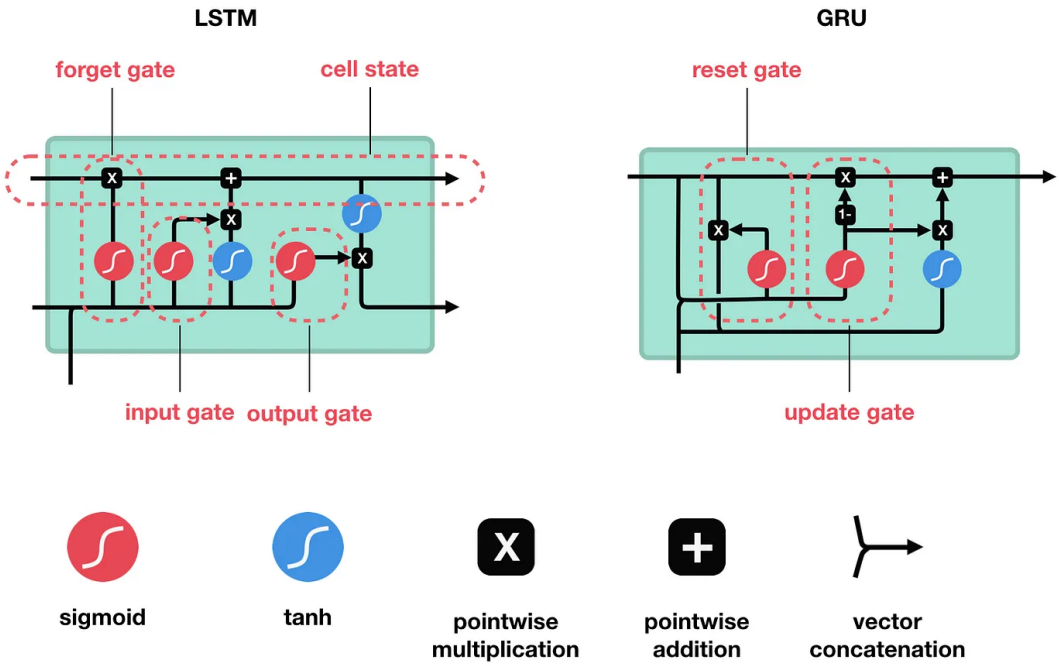


FIGURE 7 – Diagramme comparaison LSTM et GRU

Source : https ://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Le code

L'explication du code suivant est assez similaire pour nos 3 modèles. Nous allons juste expliquer brièvement celui du BILSTM.

Les paramètres de la fonction de création du modèle BILSTM sont les suivants :

- units : le nombre de neurones dans chaque couche LSTM, fixé à 64 par défaut
- activation : la fonction d'activation à utiliser pour la couche de sortie Dense. Par défaut, elle est définie sur 'sigmoid'
- optimizer : l'algorithme d'optimisation à utiliser pour entraîner le modèle. Par défaut, il est fixé sur 'adam'

Le modèle créé par cette fonction contient une couche LSTM bidirectionnelle avec units neurones, suivie d'une autre couche LSTM bidirectionnelle avec units neurones, et enfin une couche de sortie Dense avec une seule sortie et une fonction d'activation définie par le paramètre activation.

Le modèle est compilé en utilisant l'erreur quadratique moyenne, le MSE, comme fonction de perte et l'algorithme d'optimisation spécifié par optimizer.

```
# Create BiLSTM model
def create_model_bilstm(units = 64, activation = 'sigmoid', optimizer =
    'adam'):
```

```
    model = Sequential()
    model.add(Bidirectional(LSTM(units = units,
              return_sequences=True),
              input_shape=(X_train.shape[1], X_train.shape[2])))
    model.add(Bidirectional(LSTM(units = units)))
    model.add(Dense(1,activation = activation))
    #Compile model
    model.compile(loss='mse', optimizer=optimizer)
    return model

# Create GRU model
def create_model_gru(units = 64, activation = 'sigmoid', optimizer = 'adam'):
    model = Sequential()
    model.add(GRU(units = units, return_sequences = True,
                input_shape = [X_train.shape[1], X_train.shape[2]]))
    model.add(Dropout(0.2))
    model.add(GRU(units = units))
    model.add(Dropout(0.2))
    model.add(Dense(units = 1, activation = activation))
    #Compile model
    model.compile(loss='mse', optimizer=optimizer)
    return model

# Create lstm model
def create_model_lstm(units = 64, activation = 'sigmoid', optimizer = 'adam'):
    model = Sequential()
    model.add(LSTM(units = units, return_sequences = True,
                input_shape = [X_train.shape[1], X_train.shape[2]]))
    model.add(Dropout(0.2))
    model.add(LSTM(units = units))
    model.add(Dropout(0.2))
    model.add(Dense(units = 1, activation = activation))
    #Compile model
    model.compile(loss='mse', optimizer=optimizer)
    return model
```

3.4 Validation du modèle

Pour évaluer la performance de nos entraînements sur le modèle, nous avons voulu mettre en place une validation croisée. La validation croisée standard consiste à diviser les données en un certain nombre de groupes de tailles égales et à répéter le processus d'entraînement et de test en utilisant chaque groupe comme ensemble de test et tous les autres groupes comme ensemble d'entraînement. Cependant, cette approche ne convient pas aux données temporelles car elle ne respecte pas l'ordre temporel des données. Pour résoudre ce problème, on sépare les données temporelles de manière progressive en sous-ensembles (cf. figure 8). On divise les données en plusieurs ensembles de test, chacun étant plus récent que le précédent, ce qui garantit que les données de test sont toujours

plus récentes que les données d'entraînement.

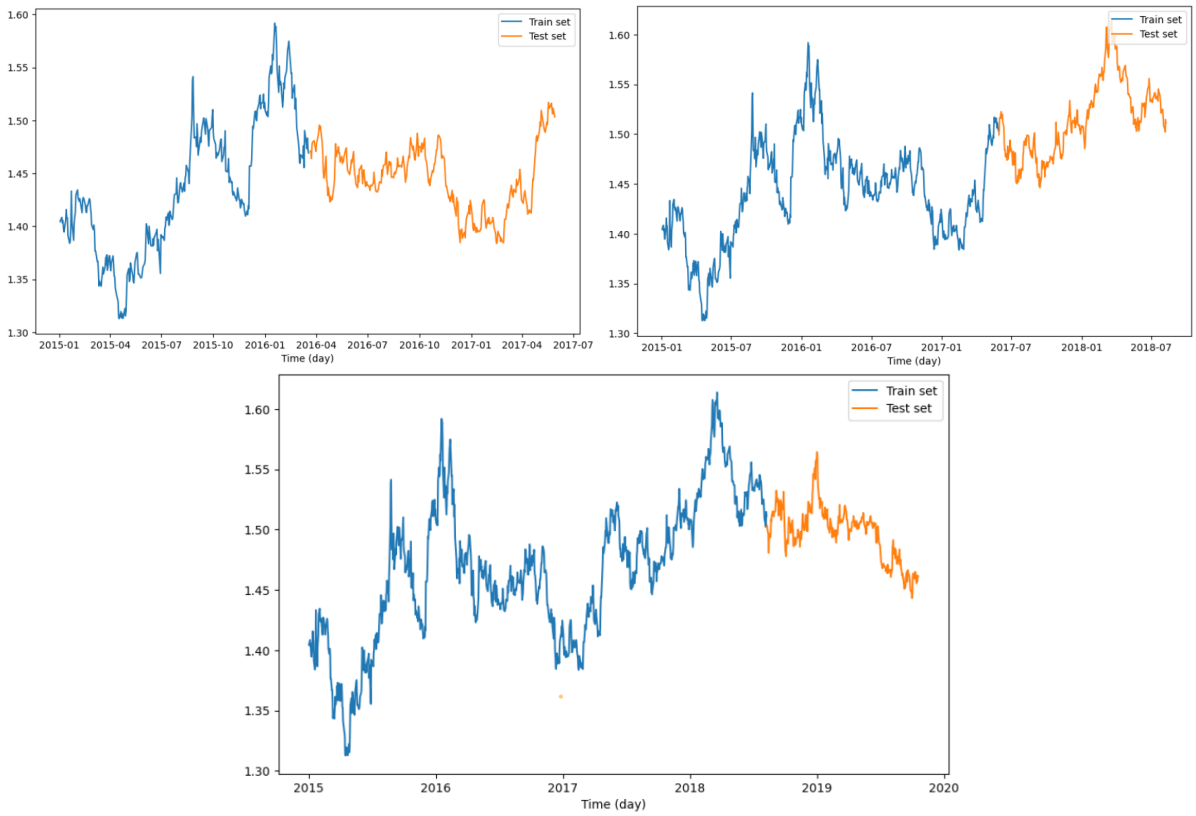


FIGURE 8 – Création de plusieurs plis

Pour faire cela nous avons utilisé la classe *TimeSeriesSplit* est disponible dans la librairie *scikit-learn*.

```
tscv = TimeSeriesSplit(n_splits=5)
for train_index, test_index in tscv.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    ...
```

3.5 Choix des hyperparamètres et du modèle

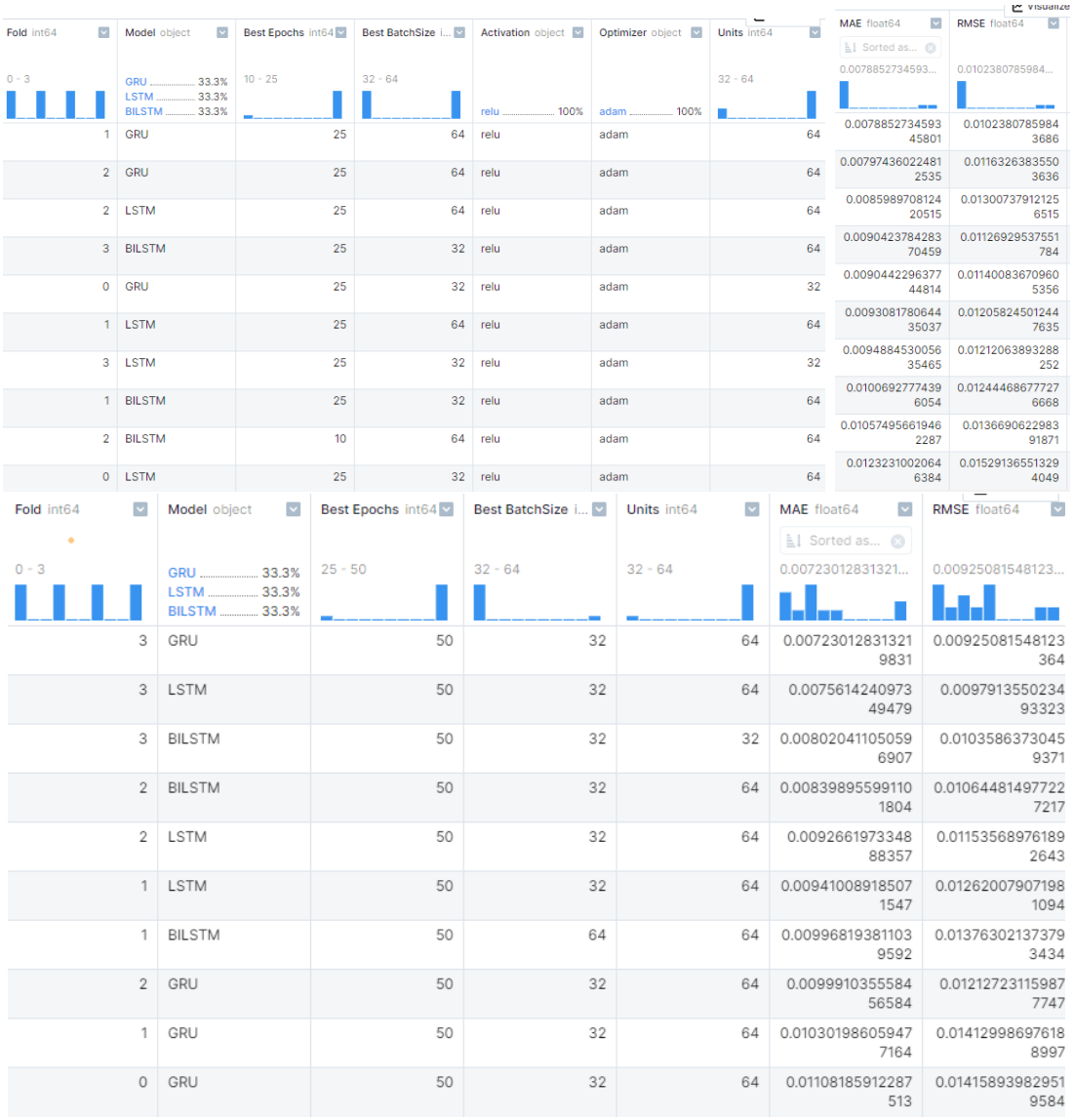
Nos modèles sont faits de plusieurs hyperparamètres : nombre d'epoch, batchsize, fonction d'activation, optimizer et nombre d'units.

Pour tenter de savoir quelle combinaison de ces paramètres était la meilleure nous avons utilisé *GridSearchCV*, une fonctionnalité de la librairie *Scikit-Learn*. Cette dernière est une technique d'optimisation de paramètres pour les modèles d'apprentissage automatique. Cette méthode permet de trouver les meilleurs paramètres pour un modèle donné en testant différentes combinaisons de paramètres à partir d'une grille de valeurs prédéfinie.

Voici un exemple de grille que nous avons testée :

```
'batch_size': [32, 64],
    'epochs': [50, 100, 200],
    'units': [32, 64],
    'activation' : ['relu','sigmoid'],
    'optimizer' : ['adam','sgd'],
```

Nous avons stocké ces résultats dans un tableau pour pouvoir comparer. Pour chaque ligne, nous avons la meilleure combinaison d'hyperparamètres pour un pli et un type de modèle (LSTM,BILSTM,GRU) donné.



Fold int64	Model object	Best Epochs int64	Best BatchSize i...	Activation object	Optimizer object	Units int64	MAE float64	RMSE float64
0 - 3	GRU 33.3% LSTM 33.3% BILSTM 33.3%	10 - 25	32 - 64	relu 100%	adam 100%	32 - 64	Sorted as...	
1	GRU	25	64	relu	adam	64	0.007885273459345801	0.010238078598436686
2	GRU	25	64	relu	adam	64	0.007974360224812535	0.011632638355036636
2	LSTM	25	64	relu	adam	64	0.008598970812420515	0.013007379121256515
3	BILSTM	25	32	relu	adam	64	0.009042378428370459	0.011262953755178 4
0	GRU	25	32	relu	adam	32	0.009044229637744814	0.011400836709605356
1	LSTM	25	64	relu	adam	64	0.009308178064435037	0.012058245012447635
3	LSTM	25	32	relu	adam	32	0.009488453005635465	0.012120638932882 52
1	BILSTM	25	32	relu	adam	64	0.010069277743960 54	0.012444686777276668
2	BILSTM	10	64	relu	adam	64	0.010574956619462287	0.013669062298391871
0	LSTM	25	32	relu	adam	64	0.012323100206463384	0.015291365513294049

Fold int64	Model object	Best Epochs int64	Best BatchSize i...	Units int64	MAE float64	RMSE float64
0 - 3	GRU 33.3% LSTM 33.3% BILSTM 33.3%	25 - 50	32 - 64	32 - 64	Sorted as...	
3	GRU	50	32	64	0.007230128313219831	0.009250815481233 64
3	LSTM	50	32	64	0.007561424097349479	0.009791355023493323
3	BILSTM	50	32	32	0.008020411050596907	0.010358637304593 71
2	BILSTM	50	32	64	0.008398955991101804	0.010644814977227217
2	LSTM	50	32	64	0.009266197334888357	0.011535689761892643
1	LSTM	50	32	64	0.009410089185071547	0.012620079071981094
1	BILSTM	50	64	64	0.009968193811039592	0.013763021373793434
2	GRU	50	32	64	0.009991035558456584	0.012127231159877747
1	GRU	50	32	64	0.010301986059477164	0.014129986976188997
0	GRU	50	32	64	0.011081859122875 13	0.014158939829519584

FIGURE 9 – Essai de différents hyperparamètres

C'est ainsi que nous avons construit le tableau de la première image. Cependant, le défaut de notre algorithme est qu'il est très chronophage : comme il y a trois modèles

RNN à tester avec plusieurs plis (cross validation) et une assez grosse combinaison d'hyperparamètres, il prenait beaucoup d'heures à tourner. Nous avons décidé de réduire le nombre d'hyperparamètres pour nos tests à venir. Nous avons par exemple supprimer la mutlimodalité de la fonction d'activation et de l'optimizer car notre premier tableau nous renvoi toujours Adam et reLU.

Avec ces comparaisons, on peut donc sélectionner le modéle GRU, avec 50 epochs, 32 lots, 64 unit au troisième pli.

3.6 Perte, prédiction et résultats

Notre algorithme nous permet également d'afficher la fonction loss.

Ainsi pour notre "meilleur" modèle on obtient les courbes suivantes :

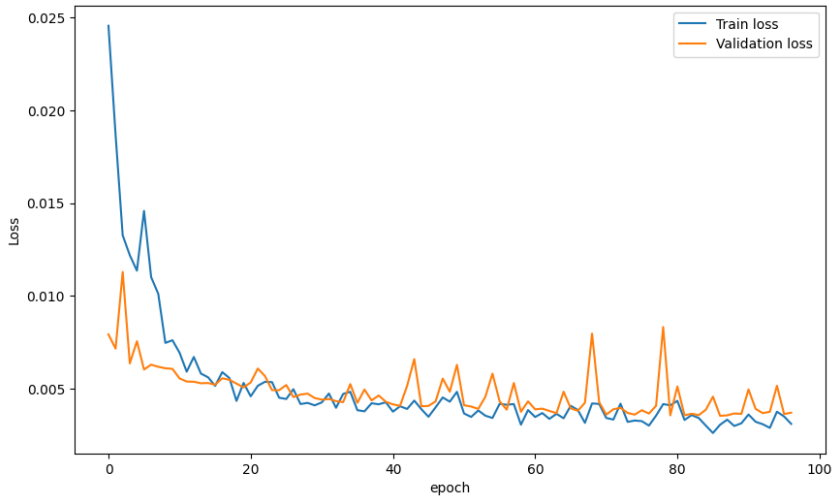


FIGURE 10 – Fonction de perte du "meilleur modèle"

Ce modèle nous a renvoyé la prédiction sur notre ensemble de tests suivants :

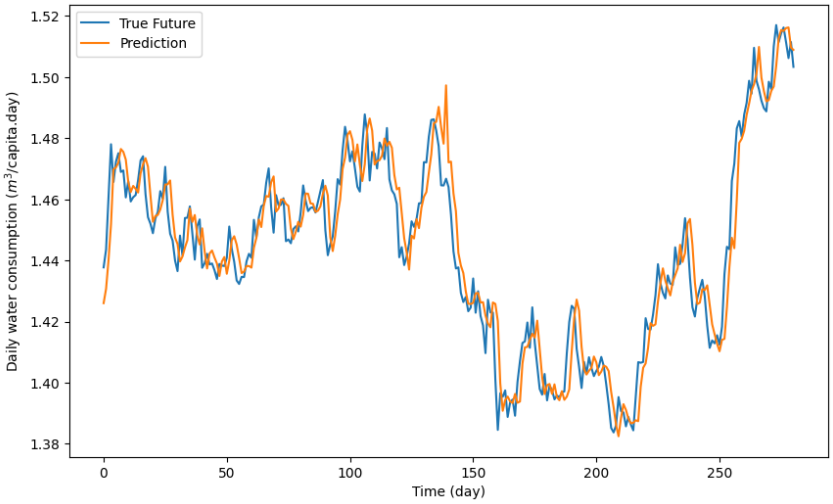


FIGURE 11 – Prédiction du "meilleur modèle"

Notre algorithme est désormais complet, et prêt à être utilisé. Nous donnons en entrée X les données d'aujourd'hui.

3.7 Prédictions pour la semaine

Nous sommes ainsi capables d'entrainer notre algorithme à prédire le tauxde change de demain, grâce aux décalages de "1" fait à notre jeu de données. Notre application visait la prédiction sur la semaine. Pour cela, nous avons étudié deux façons de faire :

- Utiliser la prédiction de demain pour faire celle d'après demain et ainsi de suite jusqu'au vendredi
- Décaler le y d'une ligne pour un jour. Exemple, nous voulons prédire dans deux jours, nous modifions le dataset pour que y soit à t+2 de x.

Nous avons ainsi fait quelques tests et nous avons opté pour la deuxième option. Nous pensons que l'erreur de prédiction semble se propager fortement avec la première.

4 Interface web

4.0.1 Structure du code

En termes de structure de code, l'application est organisée en plusieurs sections qui correspondent à ces différentes fonctionnalités. Les principales sections sont les suivantes :

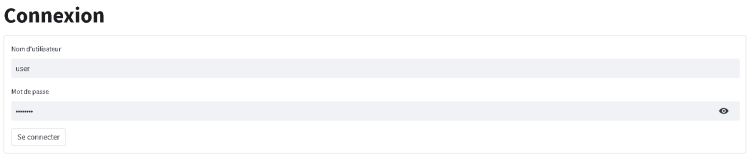
- Configuration de Streamlit : cette section initialise l'application en définissant son titre, son icône et son style, ainsi que les bibliothèques nécessaires.
- Fonction pour récupérer les données financières : cette section définit une fonction pour récupérer les données financières en utilisant la bibliothèque yfinance.
- Fonction pour créer un graphique interactif : cette section définit une fonction pour créer un graphique interactif des données financières en utilisant la bibliothèque Plotly.

— Fonction pour donner des conseils d'investissement : cette section définit une fonction simple pour donner des conseils d'investissement en comparant le cours de clôture à la moyenne mobile sur 30 jours.
— Fonction pour entraîner un modèle de régression linéaire et effectuer des prédictions : cette section définit une fonction pour entraîner un modèle de régression linéaire sur les données financières et effectuer des prédictions pour une période donnée.
— Fonction pour vérifier les identifiants de connexion : cette section définit une fonction simple pour vérifier les identifiants de connexion de l'utilisateur.
— Sections Contact et Notre Équipe : ces sections affichent des informations sur le contact et l'équipe de l'application.
— Fonction principale "main" : cette fonction principale affiche l'interface utilisateur de l'application. Elle commence par vérifier si l'utilisateur est connecté, si ce n'est pas le cas, elle affiche un formulaire de connexion. Sinon, elle affiche l'interface utilisateur de l'application avec un menu burger pour naviguer entre les différentes sections.

4.0.2 Organisation de l'Interface

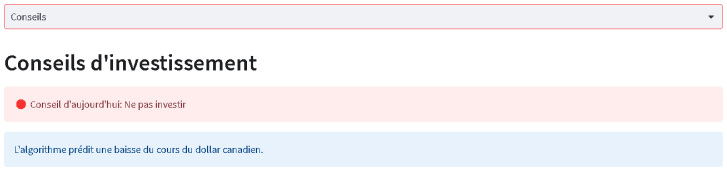
L'application "MonnAI" est une interface web développée en Python avec l'utilisation de la bibliothèque Streamlit. Elle permet de visualiser des données financières, de donner des conseils d'investissement et de faire des prédictions pour le taux de change EUR/CAD. Voici une présentation de chaque section de l'interface :

— Connexion : l'utilisateur doit se connecter avec un nom d'utilisateur et un mot de passe pour accéder à l'application.



— Accueil : l'accueil présente l'interface principale de l'application, qui comporte plusieurs fonctionnalités.
— Graphiques : cette section permet de visualiser des graphiques interactifs pour le taux de change EUR/CAD, en choisissant une période de temps avec des dates de début et de fin.

— Conseils : cette section propose des conseils d'investissement basés sur une fonction simple d'algorithme d'IA, qui compare le cours de clôture à la moyenne mobile sur 30 jours.



— Prédictions : cette section permet de faire des prédictions pour le taux de change EUR/CAD en utilisant un modèle de régression linéaire, avec la possibilité de choisir une période de temps pour les prédictions.

— Contact : la section contact donne les informations de contact pour l'application.
— Notre Équipe : la section notre équipe présente les membres de l'équipe qui ont développé l'application.

5 Conclusion

En conclusion, notre application web pour le suivi du taux de change EUR/CAD offre une solution pratique et efficace pour les investisseurs et les étudiants qui cherchent à prendre des décisions éclairées en matière d'investissement. En combinant des techniques d'intelligence artificielle, des graphiques interactifs et des modèles de prévision de taux de change, notre application fournit des informations précieuses pour aider les utilisateurs à naviguer sur le marché des changes.

Bien que notre application soit déjà très fonctionnelle, nous sommes conscients qu'il y a toujours place à l'amélioration et à l'innovation dans le domaine de l'investissement sur le marché des changes. Dans l'avenir, nous envisageons d'ajouter de nouvelles fonctionnalités telles que des alertes personnalisées pour les événements économiques clés, des rapports d'analyse de marché en temps réel, et des outils de trading automatisé basés sur des modèles d'IA plus avancés. Nous sommes également engagés dans la protection et la confidentialité des données de nos utilisateurs, et nous continuerons à travailler sur ces aspects pour assurer la sécurité de leurs informations personnelles.

En somme, notre objectif est de continuer à offrir une solution fiable et efficace pour les investisseurs et les étudiants sur le marché des changes, et nous sommes impatients de travailler sur ces améliorations futures pour notre application.