

Introduction. En parallèle à l'étape 3, nous avons ajouté des extensions au projet. En particulier, l'architecture des modèles d'apprentissage a été enrichie de nouveaux réseaux, tels que le réseau récurrent GRU, les réseaux convolutifs, pleinement connectés, les réseaux basés sur l'attention. A titre d'exploration et de comparaison avec les stratégies précédentes, nous avons appliqués GRU aux modèles d'apprentissage PPO et TDQN donnant un nouveau type de modèle TDRQN (TDQN à réseau récurrent), de même pour les réseaux convolutifs, pleinement connecté et basé sur l'attention, donnant les types de modèle TDCQN (TDQN à réseau convolutif). De plus, une fonctionnalité d'introduction de variables d'observations supplémentaire a été implémentée, ceci permet l'ajout de contexte supplémentaire qui aide la prise de décision de l'agent. En effet, lors de l'apprentissage au lieu d'avoir que les observations des cours boursiers, l'agent a également de la visibilité sur d'autres informations du marché tels que le prix de l'or, du pétrole, le taux de change, etc.

L'état de l'art. Les GRU (Gated Recurrent Units) sont une variante des réseaux récurrents comme les LSTM. Ils ont été introduits en 2014 par Cho et al. pour améliorer l'apprentissage à long terme des réseaux de neurones récurrents tout en réduisant le nombre de paramètres. Comme les LSTM, les GRU ont des portes qui contrôlent l'information qui doit être conservée et celle qui doit être oubliée. Cependant, les GRU ont deux portes au lieu de trois dans les LSTM : une porte de réinitialisation et une porte de mise à jour.

La porte de réinitialisation permet aux GRU de décider à quel moment ils doivent oublier l'état précédent et réinitialiser l'état actuel. Cela peut aider à éviter le problème du vanishing gradient dans les réseaux de neurones récurrents en permettant une régénération régulière de l'état caché.

La porte de mise à jour permet aux GRU de décider quelles informations doivent être conservées dans l'état caché. Cela permet de supprimer ou d'ajouter des informations de manière dynamique dans l'état caché.

En ce qui concerne le trading, les GRU ont également été utilisés pour prédire les tendances du marché et détecter les anomalies. Les techniques de prétraitement et de normalisation sont similaires à celles utilisées pour les LSTM, et les GRU peuvent être combinées avec d'autres techniques d'apprentissage automatique pour améliorer les prévisions.

Les CNN ont montré des résultats prometteurs dans l'analyse des données financières et la réalisation de prévisions basées sur les tendances et les modèles passés. Dans le contexte du trading, les CNN ont été utilisés pour diverses tâches telles que la prévision du cours des actions, l'analyse des sentiments et la détection d'anomalies. Plusieurs études ont exploré l'utilisation de ces modèles pour le trading. Par exemple, un [article](#) en a utilisé un pour prédire les mouvements du cours des actions sur la base des données historiques de prix et de volume, obtenant de bonnes performances. Un bloc convolutif peut être utilisé pour extraire des caractéristiques des données d'entrée et identifier des modèles qui peuvent être pertinents pour prendre des décisions commerciales. Par exemple, un bloc convolutif peut être utilisé pour identifier les tendances, la saisonnalité ou les anomalies dans les données d'entrée qui peuvent indiquer des changements dans les conditions du marché ou les opportunités commerciales. En incorporant un bloc convolutif dans un TDQN, le modèle peut apprendre à extraire les caractéristiques pertinentes des données d'entrée et à les utiliser pour informer le processus d'apprentissage Q.

Le bloc linéaire est un bloc commun utilisé dans les réseaux de neurones, qui applique une transformation linéaire aux données d'entrée. Il est souvent utilisé dans la première couche d'un réseau de neurones pour transformer les données d'entrée en une représentation d'entités pouvant être apprise par les couches suivantes. L'utilisation d'un bloc linéaire avec TDQN peut aider à réduire la dimensionnalité des données d'entrée et à extraire des caractéristiques utiles pour l'apprentissage de la fonction Q. En réduisant la dimensionnalité des données d'entrée, le bloc linéaire peut également réduire le coût de calcul de la formation du réseau neuronal. De plus, l'utilisation d'un bloc linéaire avec TDQN peut aider à améliorer la stabilité du processus d'apprentissage. L'algorithme TDQN est connu pour être instable pendant l'entraînement, ce qui peut entraîner des valeurs Q divergentes ou oscillantes. Cependant, l'utilisation d'un bloc linéaire peut aider à stabiliser le processus d'apprentissage en contraignant les valeurs Q à un espace linéaire.

Un bloc 'attention' est un modèle qui apprend à se concentrer sur des parties spécifiques de l'entrée qui sont pertinentes pour la tâche à accomplir, tout en ignorant les autres parties qui sont moins pertinentes. Le mécanisme 'attention' calcule un ensemble de pondérations qui déterminent la quantité d'attention à attribuer à chaque partie de

l'entrée. Les blocs d'attention se composent généralement de trois parties : les matrices de requête, de clé et de valeur. La matrice de requête est utilisée pour déterminer les parties de la séquence d'entrée à prendre en compte, en fonction de leur similarité avec la matrice de clé. La matrice de valeurs fournit les informations réelles à prendre en compte. La sortie d'un bloc d'attention est une somme pondérée des valeurs, où les poids sont déterminés par la similarité entre la requête et les matrices de clé. Les mécanismes 'attention' sont devenus un outil populaire dans le traitement du langage naturel, la vision par ordinateur et la reconnaissance vocale. Ils permettent aux modèles de se concentrer sur les parties les plus pertinentes des données d'entrée, ce qui peut améliorer leur précision et leur efficacité. Dans le contexte du trading, des mécanismes d'attention ont été utilisés pour analyser et interpréter les données financières, telles que les cours des actions et les volumes de transactions. Les modèles basés sur l'attention peuvent apprendre à traiter de manière sélective différentes parties des données financières, telles que des actions spécifiques ou des tendances de marché, ce qui peut conduire à des prévisions plus précises et à de meilleures stratégies de trading. Une approche récente de pointe dans le commerce basé sur 'attention' est l'utilisation de réseaux 'auto-attention', tels que le modèle [Transformer](#). Ces modèles peuvent traiter des séquences de données financières, telles que des données de séries chronologiques sur les cours des actions, et apprendre à s'occuper de manière sélective des parties les plus pertinentes de la séquence à chaque pas de temps.

Méthodologie. Dans le but de simplifier l'exécution des simulations et l'affichage des résultats, nous avons produit un JupyterNotebook pour chaque type d'apprentissage dans la branche principale *main* notre [dépôt git de projet](#), en particulier, l'application de méthode classique, de TDQN, de PPO, de TDRQN, de TDCQN, respectivement avec et sans contexte.

A partir de la restructuration du code réalisée durant l'étape 3 du projet, l'intégration de nouvelles architectures devient moins compliquée. Nous avons donc profité pour ajouter les extensions suivantes :

- Bloc GRU, un réseau récurrent, dans DQN
- Bloc Linéaire, un réseau linéaire pleinement connecté, dans DQN
- Bloc Attention, un réseau basé sur l'attention, dans DQN
- Bloc GRU, un réseau récurrent dans PPO
- Contexte supplémentaire (par exemple : l'ajout de prix de l'or, de ressources premières, etc.)

Des réseaux de type GRU ont été ajoutés au début du modèle PPO et à la fin du modèle DQN dans le but d'introduire une propriété de mémorisation temporelle, elle nous permettra notamment de comparer avec les résultats obtenus par un réseau LSTM. Comme le modèle TDQN performe déjà correctement, nous essayons d'améliorer celui-ci en faisant le choix d'introduire à la fin, au contraire, le modèle PPO avait des résultats peu satisfaisants, nous essayons de modifier l'extraction des caractéristiques en introduisant le réseau au début.

Nous avons ajouté des réseaux convolutionnels dans le modèle DQN pour permettre de nuancer le choix de notre architecture avec d'autres types de réseaux, comme Linéaire et Attention, cela nous permet de voir l'influence que pourrait avoir ces types de réseau dans l'apprentissage. La procédure d'ajout reste similaire aux réseaux précédents.

Afin de permettre au *trading* bot d'avoir plus d'informations à prendre en compte lors des simulations, un contexte supplémentaire a été ajouté pour introduire des informations environnementales diverses en plus des états actuels (prix du cours boursier selon une période d'observation). Ces informations additionnelles, tels que le prix de l'or, le prix des ressources premières, le montant des *currencies* permettent d'avoir de la visibilité sur des phénomènes qui peuvent arriver, comme prévoir une chute, une explosion, des changements brusques, instabilité, volatilité, etc. Ceci rendrait globalement le bot plus performant et il est attendu que le bot de *trading* fasse des choix plus optimaux.

Expérimentations. Le *bot* a été lancé dans les mêmes conditions que l'étape précédente, sur les actions d'Apple et de Tesla en considérant 30 jours pour la taille des états sur 6 ans d'apprentissage et 2 ans de simulation en phase test. Le contexte pris en compte pour les simulations sont les informations suivantes : prix de l'or, prix du pétrole, le taux de change entre l'euro (EUR) et le dollar américain (USD), le cours boursiers des entreprises membre du S&P500, l'indice de volatilité VIX ainsi que le rendement du Trésor américain pour les périodes de 5, 10 et 30 ans.

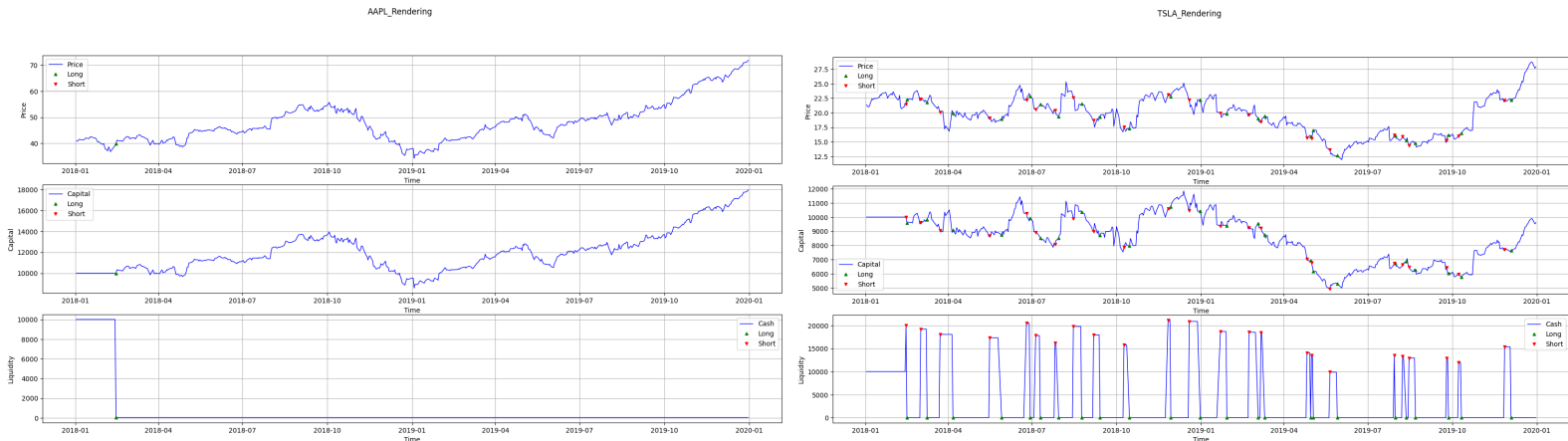


Figure 1 - Apprentissage de l'agent RL (TDQN + GRU) entre 2012 et 2017 sur les cours boursiers d'Apple (gauche) et de Tesla (droite)

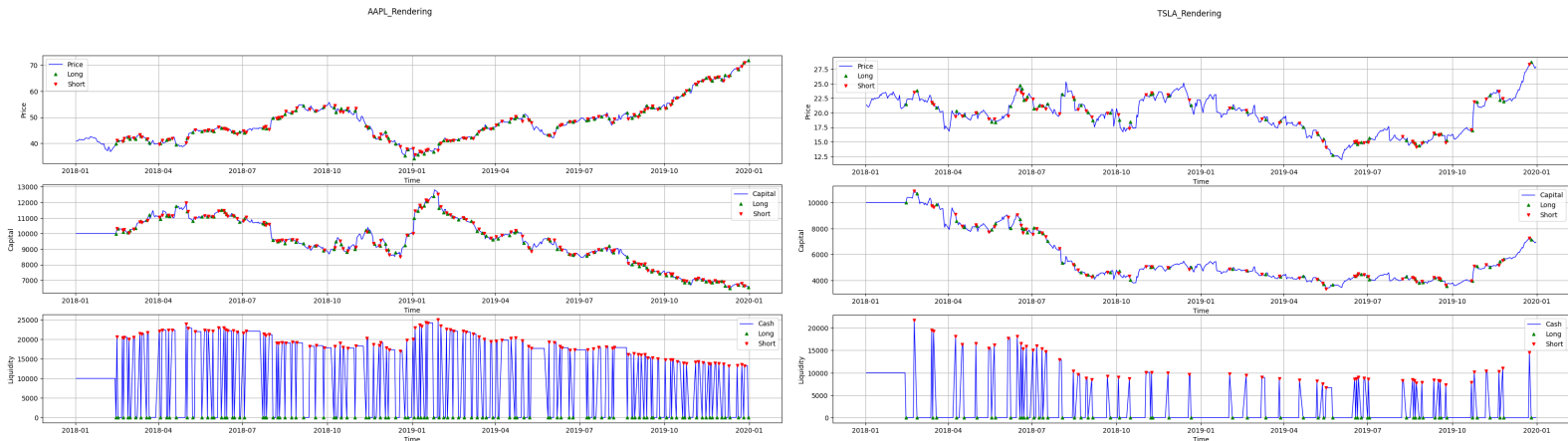


Figure 2 - Apprentissage de l'agent RL (PPO + GRU) entre 2012 et 2017 sur les cours boursiers d'Apple (gauche) et de Tesla (droite)

En figures 1 et 2, nous avons respectivement les courbes d'apprentissages de LSTM dans TDRQN et LSTM dans PPO du *bot* sur les actions d'Apple et de Tesla. On remarque que pour Apple, le modèle TDQN + GRU, je prends qu'une décision, comme pour TDQN + LSTM à l'étape 3, mais pour Tesla, le modèle semble revenir à la normal, et prend des décision fréquemment. En figure 3 et 4, nous avons des applications du bot en test en utilisant le réseau GRU dans les modèles TDQN et PPO, respectivement sans contexte et avec contexte. Nous remarquons que pour Tesla, l'induction du contexte améliore les performances du bot, ceci peut être expliqué par le fait que Tesla est un cours volatile qui dépend facilement de perturbation extérieure. De ce fait, le bot en ayant plus d'informations sur le marché, donc plus de visibilité, le bot prend de meilleures décisions en se projetant dans l'avenir. De plus, nous remarquons que le GRU dans TDQN sans contexte donne de résultats semblable au LSTM dans TDQN.

Performance Indicator	TDRQN_GRU (Testing)	Performance Indicator	PPO_GRU (Testing)	Performance Indicator	TDRQN_GRU (Testing)	Performance Indicator	PPO_GRU (Testing)
Profit & Loss (P&L)	7971	Profit & Loss (P&L)	-3452	Profit & Loss (P&L)	-396	Profit & Loss (P&L)	-3048
Annualized Return	28.82%	Annualized Return	-19.64%	Annualized Return	11.01%	Annualized Return	-4.16%
Annualized Volatility	26.58%	Annualized Volatility	26.49%	Annualized Volatility	52.54%	Annualized Volatility	53.16%
Sharpe Ratio	1.239	Sharpe Ratio	-0.668	Sharpe Ratio	0.221	Sharpe Ratio	-0.077
Sortino Ratio	1.558	Sortino Ratio	-0.981	Sortino Ratio	0.333	Sortino Ratio	-0.104
Maximum Drawdown	38.46%	Maximum Drawdown	49.23%	Maximum Drawdown	58.25%	Maximum Drawdown	69.17%
Maximum Drawdown Duration	62 days	Maximum Drawdown Duration	222 days	Maximum Drawdown Duration	108 days	Maximum Drawdown Duration	309 days
Profitability	100.00%	Profitability	45.81%	Profitability	41.67%	Profitability	39.13%
Ratio Average Profit/Loss	inf	Ratio Average Profit/Loss	0.977	Ratio Average Profit/Loss	1.361	Ratio Average Profit/Loss	1.243
Skewness	-0.475	Skewness	0.404	Skewness	0.662	Skewness	0.110

Figure 3 - Tableaux des métriques de performance en phase de test du *bot* (GRU dans TDQN et PPO sans contexte) sur les *stocks* d'Apple (gauche) et de Tesla (droite)

Performance Indicator	TDRQN_GRU (Testing)	Performance Indicator	PPO_GRU (Testing)	Performance Indicator	TDRQN_GRU (Testing)	Performance Indicator	PPO_GRU (Testing)
Profit & Loss (P&L)	-65	Profit & Loss (P&L)	-1679	Profit & Loss (P&L)	3990	Profit & Loss (P&L)	2287
Annualized Return	3.43%	Annualized Return	-5.78%	Annualized Return	28.34%	Annualized Return	21.92%
Annualized Volatility	27.66%	Annualized Volatility	26.71%	Annualized Volatility	55.76%	Annualized Volatility	53.03%
Sharpe Ratio	0.126	Sharpe Ratio	-0.210	Sharpe Ratio	0.579	Sharpe Ratio	0.458
Sortino Ratio	0.183	Sortino Ratio	-0.254	Sortino Ratio	0.830	Sortino Ratio	0.655
Maximum Drawdown	44.35%	Maximum Drawdown	46.32%	Maximum Drawdown	53.99%	Maximum Drawdown	62.87%
Maximum Drawdown Duration	206 days	Maximum Drawdown Duration	196 days	Maximum Drawdown Duration	91 days	Maximum Drawdown Duration	59 days
Profitability	41.79%	Profitability	46.67%	Profitability	52.42%	Profitability	47.57%
Ratio Average Profit/Loss	1.386	Ratio Average Profit/Loss	0.971	Ratio Average Profit/Loss	1.000	Ratio Average Profit/Loss	1.269
Skewness	0.068	Skewness	-0.671	Skewness	0.275	Skewness	0.371

Figure 4 - Tableaux des métriques de performance en phase de test du *bot* (GRU dans TDQN et PPO avec contexte) sur les *stocks* d’Apple (gauche) et de Tesla (droite)

Performance Indicator	TDRQN_LSTM (Testing)	Performance Indicator	TDCQN_LINEAR (Testing)	Performance Indicator	TDRQN_LSTM (Testing)	Performance Indicator	TDCQN_LINEAR (Testing)
Profit & Loss (P&L)	5157	Profit & Loss (P&L)	5910	Profit & Loss (P&L)	-2980	Profit & Loss (P&L)	-6156
Annualized Return	21.97%	Annualized Return	23.57%	Annualized Return	-7.38%	Annualized Return	-41.46%
Annualized Volatility	26.43%	Annualized Volatility	24.64%	Annualized Volatility	46.04%	Annualized Volatility	54.43%
Sharpe Ratio	0.921	Sharpe Ratio	1.068	Sharpe Ratio	-0.154	Sharpe Ratio	-0.604
Sortino Ratio	1.155	Sortino Ratio	1.487	Sortino Ratio	-0.205	Sortino Ratio	-0.784
Maximum Drawdown	35.49%	Maximum Drawdown	29.30%	Maximum Drawdown	54.04%	Maximum Drawdown	75.94%
Maximum Drawdown Duration	62 days	Maximum Drawdown Duration	90 days	Maximum Drawdown Duration	144 days	Maximum Drawdown Duration	301 days
Profitability	44.44%	Profitability	61.76%	Profitability	0.00%	Profitability	41.38%
Ratio Average Profit/Loss	3.288	Ratio Average Profit/Loss	1.179	Ratio Average Profit/Loss	0.000	Ratio Average Profit/Loss	0.995
Skewness	-0.574	Skewness	-0.082	Skewness	-0.024	Skewness	-0.294

Figure 5 - Tableaux des métriques de performance en test du *bot* (LSTM et Linear dans TDQN avec contexte) sur les *stocks* d’Apple (gauche) et de Tesla (droite)

En figure 5, nous avons les applications d’un LSTM et réseau Linéaire dans TDQN avec contexte. Nous remarquons qu’avec le contexte les résultats restent satisfaisants pour Apple mais pour Tesla, les résultats sont assez mauvais. Il est compliqué de savoir si c’est la volatilité qui impacte la décision ou bien la connaissance d’informations supplémentaires qui va ajouter du bruit au lieu d’aider la décision. De plus en comparant aux méthodes classiques, figure 6 et 7, nous remarquons que TDQN + LTSM avec contexte performe mieux que les modèles classiques mais le modèle linéaire reste assez correcte. De plus en comparant avec l’étape précédente, notre TDQN + LSTM avec contexte n’excelle pas forcément les résultats sans contexte.

Performance Indicator	B&H (Testing)	Performance Indicator	S&H (Testing)	Performance Indicator	MATF (Testing)	Performance Indicator	MAMR (Testing)
Profit & Loss (P&L)	7961	Profit & Loss (P&L)	-7981	Profit & Loss (P&L)	6856	Profit & Loss (P&L)	-3460
Annualized Return	28.80%	Annualized Return	-100.00%	Annualized Return	25.92%	Annualized Return	-19.08%
Annualized Volatility	26.55%	Annualized Volatility	44.17%	Annualized Volatility	24.82%	Annualized Volatility	28.28%
Sharpe Ratio	1.239	Sharpe Ratio	-1.591	Sharpe Ratio	1.178	Sharpe Ratio	-0.610
Sortino Ratio	1.559	Sortino Ratio	-2.204	Sortino Ratio	1.801	Sortino Ratio	-0.813
Maximum Drawdown	38.43%	Maximum Drawdown	82.28%	Maximum Drawdown	14.86%	Maximum Drawdown	51.09%
Maximum Drawdown Duration	62 days	Maximum Drawdown Duration	250 days	Maximum Drawdown Duration	20 days	Maximum Drawdown Duration	204 days
Profitability	100.00%	Profitability	0.00%	Profitability	42.31%	Profitability	56.67%
Ratio Average Profit/Loss	inf	Ratio Average Profit/Loss	0.000	Ratio Average Profit/Loss	3.181	Ratio Average Profit/Loss	0.492
Skewness	-0.476	Skewness	0.145	Skewness	0.404	Skewness	-0.291

Figure 6 - Evaluation de performance en phase de test des méthodes classiques pour Apple

Performance Indicator	B&H (Testing)	Performance Indicator	S&H (Testing)	Performance Indicator	MATF (Testing)	Performance Indicator	MAMR (Testing)
Profit & Loss (P&L)	2960	Profit & Loss (P&L)	-2980	Profit & Loss (P&L)	-7327	Profit & Loss (P&L)	854
Annualized Return	24.06%	Annualized Return	-7.38%	Annualized Return	-100.00%	Annualized Return	18.99%
Annualized Volatility	53.05%	Annualized Volatility	46.04%	Annualized Volatility	52.61%	Annualized Volatility	58.03%
Sharpe Ratio	0.507	Sharpe Ratio	-0.154	Sharpe Ratio	-0.989	Sharpe Ratio	0.358
Sortino Ratio	0.741	Sortino Ratio	-0.205	Sortino Ratio	-1.231	Sortino Ratio	0.538
Maximum Drawdown	52.76%	Maximum Drawdown	54.04%	Maximum Drawdown	79.85%	Maximum Drawdown	65.30%
Maximum Drawdown Duration	205 days	Maximum Drawdown Duration	144 days	Maximum Drawdown Duration	229 days	Maximum Drawdown Duration	159 days
Profitability	100.00%	Profitability	0.00%	Profitability	34.38%	Profitability	67.65%
Ratio Average Profit/Loss	inf	Ratio Average Profit/Loss	0.000	Ratio Average Profit/Loss	0.533	Ratio Average Profit/Loss	0.496
Skewness	0.542	Skewness	-0.024	Skewness	-0.309	Skewness	0.550

Figure 7 - Evaluation de performance en phase de test des méthodes classiques pour Tesla

Conclusion. Des réseaux divers ont été introduits dont les résultats sont disponibles dans le dépôt. Nous traitons l'essentiel dans ce rapport annexe. L'introduction du contexte pourrait améliorer les performances du bot sur les cours volatiles en particulier pour le GRU, mais cela n'est pas forcément le cas pour LSTM. Dans le modèle PPO, le contexte a pu montrer des améliorations sur les cours volatiles mais les résultats restent à confirmer avec plus d'application sur des cours volatiles et en effectuant une moyenne des résultats. Dans l'ensemble, à certaines exceptions près, les modèles IA performant mieux que les classiques, en particulier ceux basées sur TDQN, et l'ajout de réseaux récurrents aident à la décision. Cependant, dans certaines conditions, nous voyons que PPO peut également se tirer du lot et voir ses résultats améliorer. Pour la suite, il faudrait envisager un ajustement des paramètres sur l'environnement, le contexte et l'architecture du réseau pour améliorer les performances du *trading* bot.