

Faculté des Sciences et Ingénierie
Sorbonne Université

Projet Cryptofolio : Plate-forme d'optimisation de portfolio de cryptomonnaies

JULIEN DENES, MICHAËL TRAZZI
Sous la supervision de THIBAUT LUST

Master 1 Informatique, spécialité ANDROIDE
Année 2017 – 2018, Semestre 2



Table des matières

Introduction	2
Ressources	2
1 Présentation générale du produit	3
2 Revue de la littérature existante	4
2.1 Les cryptomonnaies comme actifs financiers	4
2.2 Algorithmes d'optimisation de portefeuilles	4
2.3 Application des algorithmes aux portefeuilles de cryptomonnaies	5
3 Évaluation et choix des algorithmes d'optimisation	7
3.1 Implémentation des algorithmes	7
3.1.1 Source initiale	7
3.1.2 Représentation des portefeuilles	7
3.1.3 Données de trading	8
3.1.4 Algorithmes de référence et CNN	8
3.2 Choix empirique des meilleurs algorithmes	8
3.2.1 Métriques	9
3.2.2 Sur du court terme	9
3.2.3 Sur du long terme en période « propice »	10
3.2.4 Sur du long terme en période « néfaste »	12
3.3 Étude des algorithmes retenus	13
3.3.1 Uniform Constant Redistribution Portfolio (UCRP)	14
3.3.2 Best Stock	14
3.3.3 Exponential Gradient (EG)	15
3.3.4 Universal Portfolio (UP)	15
3.3.5 Anticor	16
3.3.6 Le réseau neuronal convolutif	16
4 Développement logiciel	19
4.1 Organisation des tâches	19
4.1.1 Étapes de développement	19
4.1.2 Calendrier	19
4.2 Pages et fonctionnalités du produit	20
4.2.1 Landing page	20
4.2.2 Inscription et connexion	20
4.2.3 Dashboard et barre de navigation	21
4.2.4 Outil d'optimisation	21
4.2.5 Enregistrement des portefeuilles	23
4.3 Organisation du code	23
4.3.1 Framework Django	23
4.3.2 Code des algorithmes	24
4.4 Choix techniques	24
4.4.1 Connexion et stockage des données des utilisateurs	24
4.4.2 Récupération des données relatives aux cryptomonnaies	25
4.4.3 Fréquence et données d'entraînement du CNN	25
4.5 Limitations du produit final	26
4.5.1 Nombre de portefeuilles possibles	26
4.5.2 Automatisation et modularité de l'implémentation	26
Conclusion	28
Références	29

Introduction

Les monnaies numériques, dites cryptomonnaies, sont aujourd'hui en plein essor. L'année 2017 a vu apparaître des centaines de nouvelles monnaies sur le marché et le bitcoin, la plus connue d'entre elles, ne cesse d'accumuler les records. En 2016, sa valeur a bondi de plus de 120%, dépassant les 1 000 dollars. En 2017, elle pulvérise ses propres exploits et s'évalue à près de 20 000 dollars. La progression fulgurante de son prix attire ainsi toujours plus d'émulation autour d'elle, et ceux qui souhaitent profiter de cette croissance sont toujours plus nombreux. L'investissement dans les cryptomonnaies présente néanmoins un risque important, et est réservé aux investisseurs présentant une excellente tolérance au risque. La haute volatilité des cours des cryptomonnaies exige en effet des nerfs d'acier : la valeur du bitcoin a par exemple été divisée par 3 entre décembre 2017 et janvier 2018.¹ Paradoxalement, ces nouvelles formes d'actifs financiers sont toujours plus accessibles, grâce au développement de plate-formes d'échanges faciles d'utilisation et sécurisées. Un investisseur se retrouve donc confronté à une grande diversité de cryptomonnaies, de plus en plus accessibles à l'achat, et doit choisir de répartir son capital dans un sous-ensemble d'actifs. Il est amené à se constituer ce qu'on appelle un portefeuille de cryptomonnaies, ou portfolio.

Dans le cadre de ce projet, nous avons donc développé à cet effet une plate-forme d'optimisation de portfolio de cryptomonnaies basée sur différentes théories modernes de l'optimisation de portfolio d'actifs. Nous fournissons une aide à la décision et à la gestion de portefeuille pour des utilisateurs de tous types, aussi bien habitués de la spéculation que novices en la matière. Celle-ci permettra à tout utilisateur, avec ou sans connaissances, d'obtenir en échange de très peu d'informations sur son portefeuille en entrée (comme les monnaies qu'il souhaite y voir) une suggestion d'investissement dans chaque cryptomonnaie en sortie, basée sur des algorithmes de trading adaptés au marché des cryptomonnaies. Cette plate-forme prendra la forme d'un site internet, accessible à tous et permettant une forme de suivi des portefeuilles.

Ce rapport détaille et explique les étapes de développement de cette plateforme. Nous présenterons d'abord le principe général du produit (partie 1), puis nous détaillerons les travaux académiques sur lesquels nous nous sommes basés (2). Nous réaliserons ensuite une étude empirique puis théorique des algorithmes retenus pour la plate-forme (3). Nous détaillerons finalement les principaux aspects logiciels du produit, comme ses fonctionnalités, son calendrier de développement et ses limites (4).

Ressources

Le présent rapport, les figures qu'ils contient, les sources, ainsi que le code de ce projet, sont disponibles sur Github à l'adresse : <https://github.com/mtrazzi/cryptofolio/>.

La plate-forme d'optimisation de portefeuilles de cryptomonnaies, résultat finale de ce projet, est disponible à l'adresse : <https://michaeltrazzi.wixsite.com/cryptoptimisation/> (landing page) et <https://cryptooptimize.herokuapp.com/> (outil d'optimisation).

1. Source : <https://coinmarketcap.com/>

1 Présentation générale du produit

Le produit final de ce projet prend la forme d'une aide à la décision sur internet pour des individus qui souhaiteraient investir dans les cryptomonnaies. L'outil central de cette plateforme web consiste en une interface de suggestion de compositions possibles de portfolio. L'utilisateur a notamment entre les mains la possibilité de spécifier les paramètres de son portefeuille, comme les monnaies dans lesquelles il souhaite investir, sa quantité de capital initial, et l'algorithme d'optimisation qu'il souhaite utiliser.

La plate-forme propose également d'autres outils pour permettre aux investisseurs de mieux comprendre le marché, comme un affichage des capitalisations des monnaies, de leur volumes d'échanges et de leur valeur. Elle propose des outils de contrôle et de pérennisation de l'investissement, comme de la possibilité d'enregistrer et de suivre valeur d'un portfolio.

Il est cependant important de préciser que l'on ne souhaite pas créer une interface de gestion directe de portefeuille. Il n'est pas envisagé, dans le cadre de ce projet, de pouvoir permettre l'achat et la vente de monnaies depuis notre site, mais cette option reste envisageable dans une extension future de ce projet.

Les détails techniques des fonctionnalités seront évoqués plus en détail dans la partie 4.2.

2 Revue de la littérature existante

La première étape nécessaire de ce projet est d’investiguer la possibilité d’utiliser des algorithmes de trading sur le marché des cryptomonnaies à travers de précédents travaux de recherche. Cela nécessite de valider plusieurs pré-requis : d’abord que les cryptomonnaies peuvent effectivement être considérées comme des actifs financiers, sur lesquels il est donc possible de spéculer ; il est ensuite nécessaire de nous intéresser à l’existence d’algorithmes d’optimisation de portefeuilles, et d’en établir une liste. Enfin, reste en suspens la question de la combinaison des deux : l’applicabilité de ces algorithmes au marchés des cryptomonnaies.

2.1 Les cryptomonnaies comme actifs financiers

Elendner et al. [1] proposent, dans leur publication « **The Cross-Section of Cryptocurrencies as Financial Assets** », une étude approfondie de la possibilité de considérer les cryptomonnaies comme des actifs d’investissement alternatifs. Ils évaluent en particulier leurs propriétés, en les comparant à celles des actifs standards. Leurs premières conclusions sont celles que l’on pourrait attendre : bien qu’en moyenne légèrement positifs (entre 0 et 1%), la croissance quotidienne des principales monnaies est très volatile : le record de croissance en une journée de l’Ethereum est par exemple de 55%, mais son record de perte est tout aussi grand (−48%). Leur volatilité journalière moyenne (une mesure de variation) oscille entre 3% et 10%, tandis qu’elle varie généralement entre 0 et 3% pour des actifs classiques. Les cryptomonnaies présentent donc à première vue un intérêt du point de vue de leur rapide croissance, mais présentent un risque élevé. Les auteurs s’intéressent donc dans un second temps aux possibilités de diversifications de portefeuille qu’elles présentent. Leurs conclusions sont cette fois-ci moins instinctives : les cryptomonnaies sont deux à deux assez peu corrélées en terme d’évolution quotidienne de leur valeur, et sont bien moins corrélés entre eux que les actifs boursiers traditionnels entre eux. Cela est également valable pour leur volatilité. Elendner et al. enquêtent finalement sur la corrélation entre cryptomonnaies et d’autres actifs traditionnels, comme les monnaies nationales, l’or, ou les bons du Trésor américains. Leurs conclusions montrent une très forte indépendance. Leur conclusion est donc que les cryptomonnaies présentent un grand intérêt pour la diversification de portfolio, que ceux-ci en soient uniquement composés de monnaies virtuelles ou qu’ils soient mixés avec des actifs plus traditionnels.

2.2 Algorithmes d’optimisation de portefeuilles

L’article de Li et Hoi [2] « **Online Portfolio Selection : A Survey** » propose un panorama des principales techniques d’optimisation de portfolio de la littérature, et une analyse des principes mathématiques sous-jacents. Ils distinguent d’abord la « Mean Variance Theory » et la « Capital Growth Theory ». La première, basée sur la théorie de Markovitz [3], ne s’intéresse qu’à une seule période de temps : on sélectionne un portfolio fixé en cherchant le meilleur compromis entre profit et risque. La seconde théorie se focalise sur une sélection de portfolio séquentielle, c’est à dire découpant une période en sous-séquences et autorisant une modification du portfolio à la fin de chacune de ces périodes. Les techniques de cette catégorie sont regroupées sous le nom de « Online Portfolio Selection », sur lesquelles cet article se focalise. Les auteurs découpent ces techniques en 5 catégories : les références, les stratégies « Follow-the-Winner », les stratégies « Follow-the-Loser », les approches de pattern-matching, et les algorithmes de méta-apprentissage. Les algorithmes de références sont des algorithmes suivant des logiques très simples, et servent de repères. Les stratégies Follow-the-Winner désignent un ensemble d’approches basées sur l’hypothèse que des actifs performants à une période t le seront également à la période $t + 1$. Au contraire, les stratégies Follow-the-Loser s’appuie sur l’hypothèse que les stocks ayant eu de bons résultats à une période t performeront moins bien au temps $t + 1$ (et vice versa), suivant la théorie selon laquelle les prix des actifs tendent vers le prix moyen au cours du temps. Les approches de pattern-matching cherchent quant à elles à identifier des motifs reconnaissables pour anticiper les mouvements de valeur. Plus diverse, la catégorie des algorithmes de méta-apprentissage désigne des

stratégies qui combinent différentes approches. Cet article nous fournit une source très riche pour le choix de nos algorithmes, et nous choisissons de ne pas les détailler ici pour nous y consacrer en profondeur plus tard dans notre étude (voir partie 3.3).

Moody et Saffell [4] proposent, dans « **Learning to Trade via Direct Reinforcement** », un modèle d'algorithme de trading basé sur leurs précédentes publications, qui s'appuie sur une stratégie d'apprentissage renforcé récurrent (ou direct). L'algorithme qu'ils présentent s'affranchit de l'apprentissage d'une fonction évaluée, en se basant plutôt sur un feedback immédiat de performance lors de l'exécution d'une action. Ils cherchent ainsi à s'affranchir de deux biais majeurs des algorithmes d'apprentissage sur des données financières : la malédiction de la dimensionnalité de Bellman, due au fait que les précédents algorithmes s'appuient sur de la programmation dynamique, et la nécessité de s'appuyer sur des modèles prédictifs. Leur solution : une fonction de récompense immédiate. Ils utilisent pour celle-ci la différentiation d'utilité entre le portfolio au temps t et celle au temps $t + 1$. La fonction d'utilité utilisée est le ratio de Sharpe, défini comme le gain divisé par la déviation standard (on ne maximise ainsi pas le profit mais le profit ajusté par le risque). On calculera alors une utilité espérée par rapport aux situations passées, le gain et la déviation standard ne pouvant être calculées pour un temps futur. Les résultats empiriques montrent que cet algorithme présente de meilleures performances que d'autres basées sur des fonctions évaluées (Q-Learning) sur le marché des échanges de monnaies US Dollar/British Pound. Il réussit en effet à dégager en moyenne sur 25 ans un bénéfice de 15% par an. Les auteurs insistent cependant sur la stabilité et la périodicité de ce marché.

D'autres travaux similaires font suite à celui-ci dans l'application des stratégies d'apprentissage pour le trading algorithmique. **Ban, El Karoui et Lim [5]** étudient ainsi un modèle d'apprentissage par renforcement très similaire, en y ajoutant deux techniques supplémentaires : la régularisation et la validation croisée. Leurs travaux cherchent de cette manière à contraindre la variance des estimations de gain et de risque, réduisant ainsi l'erreur d'évaluation de performance. Leurs évaluations théoriques font preuve d'un avantage sur les algorithmes de référence.

Heaton, Polson et Witte [6] tentent quant à eux d'appliquer un algorithme de deep learning pour la prédiction d'évolution de valeur d'un portfolio. Leur argument principal est que les réseaux de neurones pourraient avoir une capacité à déceler des interactions dans les données invisibles par les algorithmes fortement basés sur des modèles. La prédiction d'évolution est cependant souvent décriée dans la littérature (par exemple [4] et [7]), et ce, semble-t-il, à juste titre. Les résultats de cet article semblent en effet montrer une assez mauvaise précision, à cause d'une accumulation d'infimes imprécisions des estimations. Leur algorithme resterait donc, pour les auteurs, applicable à des problèmes simples comme le passage d'ordres d'achat ou de vente, mais la volonté de prédire l'évolution des valeurs semble encore trop complexe. Elle est surtout souvent vue comme non nécessaire à l'optimisation de portefeuilles.

2.3 Application des algorithmes aux portfolios de cryptomonnaies

Dans « **Application of Machine Learning Algorithms for Bitcoin Automated Trading** », **Żbikowski [8]** étudie en premier la possibilité d'appliquer un algorithme de machine learning pour le trading de cryptomonnaies. Il ne s'intéresse cependant qu'au Bitcoin, et utilise 2220 points de données périodiques espacés de 15 minutes dans le temps. Il applique deux algorithmes : une machine à support de vecteurs (SVM) avec « Box Theory », et une SVM avec pondération par le volume. Ces deux algorithmes se montrent compétents dans la réalisation d'une stratégie de trading, le premier obtenant un retour sur investissement de 10.58%, le second de 33.58%. L'exposition au risque est également meilleure que la stratégie utilisée comme référence.

L'article de **Jiang et Liang [7]** « **Cryptocurrency Portfolio Management with Deep Reinforcement Learning** » s'intéresse à l'application de techniques de machine learning innovantes au problème de gestion de portefeuilles. L'algorithme d'apprentissage qu'ils proposent ne s'appuie sur aucune connaissances expertes, et ne tente pas de faire de la prédiction de prix. Son principe repose sur une stratégie d'apprentissage par renforcement adaptée au cas du portfolio. Celle-ci est appliquée à plusieurs structures de réseaux neuronaux, dont un réseau neuronal convolutif (RNN),

un réseau neuronal récurrent (RNN) plus classique, et un réseau Long Short Term Memory (LSTM), lesquelles doivent prédire le portefeuille à adopter à la période suivante. Les résultats théoriques sont testés en pratique en effectuant un trading des données passées, avec ré-optimisation toutes les 30 minutes. Les auteurs formulent notamment des hypothèses de liquidité totale des marchés (un ordre peut être passé et exécuté immédiatement) et de non-influence de l'agent sur le marché. Le Bitcoin est considéré être le « cash », ou cryptomonnaie de réserve. A la fin de chaque période, en achetant/vendant, la composition du portefeuille est modifiée. Les résultats obtenus tendent à montrer que cette approche surpasse toutes les méthodes traditionnelles. Le CNN fournit les meilleurs résultats, suivie de près par le RNN, puis par le LSTM. Les auteurs insistent sur quelques limites de leur modèle, notamment sur les hypothèses faites de liquidité totale et de non influence. Celles-ci peuvent en effet ne pas être vérifiées sur le marché de petites cryptomonnaies émergentes, et des tests en situation réelle sont nécessaires pour vérifier ces résultats. Cette approche novatrice nous semblant très prometteuse, nous avons décidé de proposer cet algorithme parmi ceux de notre plate-forme. Nous ne détaillons donc volontairement pas le modèle ici, mais celui-ci sera étudié en détail dans la partie 3.3.6.

3 Évaluation et choix des algorithmes d'optimisation

Comme expliqué plus tôt, l'aide à la décision de ce projet propose à l'utilisateur divers algorithmes pour optimiser son portfolio, et selon différents critères.

Nous allons alors comparer et analyser ci-dessous les principaux algorithmes décrits par LI et HOI [2] : Online Moving Average Reversion (OLMAR, [9]), Universal Portfolio (UP, [10]), Anticor [11], Passive Aggressive Mean Reversion (PAMR, [12]), Nonparametric Kernel Based Log Optimal Strategy (BK, [13]), M0 [14], Robust Median Reversion (RMR, [15]), Confidence Weighted Mean Reversion (CWMR, [16]), Exponential Gradient (EG, [17]) et Weighted Moving Average Mean Reversion (WMAMR, [18]), ainsi que des algorithmes de référence basiques développés dans [10] et [19] : Best Stock (Best), Buy and Hold (BAH), Uniform Buy and Hold (UBAH), et Best Constant Rebalanced Portfolios (BCRP). Nous choisissons volontairement de ne détailler pour le moment que le fonctionnement des algorithmes de références, et de ne détailler par la suite que les algorithmes que nous retiendrons à la suite de l'étude empirique. A ces algorithmes dit « classiques », nous ajoutons le réseau neuronal convolutif (CNN dans la suite) étudié par JIANG et LIANG [7] et évoqué dans la revue de la littérature.

3.1 Implémentation des algorithmes

Nous détaillons dans un premier temps les choix et modalités de l'implémentation des algorithmes pour la réalisation de ces tests. En particulier, la source du code de ces derniers, le choix de représentation des portefeuilles, des données utilisées, et une courte explication des algorithmes de référence basiques utilisés par la suite.

3.1.1 Source initiale

Le code des algorithmes de ce projet a été réalisé sous Python 3, et se base sur le code de JIANG et LIANG [7] disponible en ligne² sous licence libre GPL. Nous l'avons largement modifié pour permettre une plus grande flexibilité, en particulier pour la prédiction de la composition d'un unique portefeuille (sans itérations). Nous nous sommes donc largement appuyés sur les choix techniques des auteurs (par exemple le choix des hyperparamètres des réseaux de neurones), puisque ceux-ci sont extensivement justifiés dans leur article.

3.1.2 Représentation des portefeuilles

Chaque algorithme, quel que soit son modèle, retourne un vecteur de la taille du nombre de monnaies préalablement choisies, où chaque composant est un nombre compris entre 0 et 1, et tel que l'ensemble somme à 1. Ce vecteur indique donc la répartition de l'investissement à placer dans chacune des monnaies. Le premier composant est le Bitcoin, utilisé comme monnaie « de référence » ou « de réserve », du fait que ce soit la plupart du temps par elle que passent les achats des autres monnaies, qui ne peuvent être échangées avec des dollars. Même s'il paraît plus rationnel de prendre le dollar comme monnaie de réserve (puisque c'est le comportement que les utilisateurs de notre plate-forme adopteront certainement), il est plus logique pour nous d'étudier la valeur de nos portfolio en Bitcoin dans ce rapport. En effet, les prix d'une monnaie sont souvent donnés en Bitcoin puisqu'uniquement échangeable avec lui, et prendre cette monnaie comme référence simplifie les transformations. Le second composant du vecteur représente pour sa part le dollar. Il est en réalité représenté par une cryptomonnaie, l'USDT, disponible sur la blockchain du Bitcoin et dont la valeur est strictement indexée sur celle du dollar. Elle a justement été créée dans le but de fournir une alternative plus stable aux acheteurs de cryptomonnaies, tout en restant facilement échangeable contre ces dernières sur les plate-formes où il n'est pas possible d'investir avec des monnaies nationales (comme Poloniex).

2. <https://github.com/ZhengyaoJiang/PGPortfolio>

On appellera souvent ce vecteur « vecteur de poids » dans la suite, et le noterons mathématiquement ω . On notera également Ω_m l'ensemble des portfolios valides composés de m monnaies :

$$\Omega_m = \left\{ \omega \in \mathbb{R}_+^m \mid \sum_{i=1}^m \omega(i) = 1 \right\} \quad (1)$$

où $\omega(i)$ désigne le $i^{\text{ème}}$ terme de ω .

3.1.3 Données de trading

La plupart des algorithmes nécessitent également des données sur le prix ou l'évolution des prix des monnaies. Ces données sont fournies par l'interface de programmation (API) de Poloniex.com, l'une des plate-formes d'échange les plus populaires, et dont l'API est facile d'utilisation. Celle-ci offre la possibilité de récupérer des données sur les échanges de monnaies (volume, prix, valeur acheteur et vendeur) depuis sa création en 2014, et ce pour différentes fréquences (5, 15 ou 30 minutes, 2h, 4h ou 24h). Pour les tests qui suivent, nous choisirons pour tous les algorithmes d'effectuer une ré-optimisation du portfolio toute les 30 minutes. Cette durée semble en effet correspondre à la situation d'un utilisateur optimisant son portfolio très souvent, mais pas non plus à la vitesse d'un trading haute fréquence pour lequel notre produit n'est pas conçu. Cela a peu d'impact sur les algorithmes d'optimisation classiques, pour lesquels la fréquence n'entre jamais en paramètre, mais elle en a pour l'entraînement du réseau de neurones. Nous reviendrons sur ces implications lors de l'étude théorique des modèles.

3.1.4 Algorithmes de référence et CNN

Les algorithmes de référence sont en fait quatre algorithmes particuliers de type Buy and Hold, c'est à dire des portfolios où les poids sont fixés à la première itération puis jamais modifiés par la suite. Le premier, nommé BAH dans la suite, est le portfolio tel que 100% de sa valeur est placée dans le dollar. Il nous permet ainsi de comparer tous les algorithmes à la décision de ne pas investir dans les cryptomonnaies. Le second, Uniform Buy and Hold (UBAH), consiste à diviser équitablement les poids entre chaque monnaie ($1/m$ dans le cas de m monnaies). Best Stock (Best) place quand à lui 100% de sa valeur dans la monnaie qui aurait le mieux performé sur une période de temps précédant le début de sa mise en ligne, tandis que Best Constant Rebalanced Portfolio (BCRP) choisit comme portfolio initial celui qui aurait le mieux performé sur cette même période (et dont les poids peuvent être répartis entre différentes monnaies contrairement à Best). Ces algorithmes, simplistes, nous serviront par la suite de repères pour détecter si les stratégies plus ou moins complexes mises en place par les autres algorithmes apportent réellement une plus-value de stabilité ou de rentabilité.

A noter enfin que le CNN a été implémenté par JIANG et LIANG [7] en utilisant Tensorflow (bibliothèque d'apprentissage par réseaux de neurones développée par Google). Nous reviendrons en détail sur celui-ci dans la section 3.3.1.

3.2 Choix empirique des meilleurs algorithmes

Dans les tests suivants, on choisit de travailler avec un portfolio de 10 monnaies en plus de la monnaie de réserve (le Bitcoin), à savoir : le dollar (USDT), l'Ethereum (ETH), le Ripple (XRP), le Bitcoin Cash (BCH), le StarCoin (STR), le Monero (XMR), le Litecoin (LTC), le Biteshares (BTS), le Siacoin (SC) et l'Ethereum Classic (ETC), qui constituent dans cet ordre les 10 monnaies les plus échangées sur Poloniex au cours des 30 jours précédents à l'écriture de ce rapport (mai 2018). On initialise le portfolio de l'utilisateur en Bitcoin uniquement, c'est-à-dire que le premier élément du vecteur de poids est 1 et le reste 0.

On anticipe que l'optimisation de portfolio peut répondre à deux objectifs différents : maximiser ses gains sur un temps court, avec un besoin de rentabilité immédiate, ou au contraire

chercher un placement sur le long terme, avec peut-être un risque moindre et une plus grande patience vis-à-vis des fluctuations (positives ou négatives) de la valeur du portfolio.

Les tests présentés par la suite s'appuient largement sur le code de JIANG et LIANG [7], très adapté puisqu'écrit pour étudier dans un but semblable d'évaluation des différents algorithmes (dans leur cas, pour montrer la supériorité du CNN).

3.2.1 Métriques

On propose d'utiliser trois métriques pour évaluer les algorithmes : la valeur finale du portfolio, le ratio de Sharpe et le maximum drawdown. Grâce à l'étude de ces métriques, nous comparerons les avantages et inconvénients de chacune des méthodes d'optimisation pour justifier les algorithmes présents sur notre plate-forme.

La valeur finale (VF) a pour but de mesurer le profit généré par le portfolio. On le ramène bien entendu à la valeur initiale du portfolio, puisqu'on s'intéresse à la démultiplication (ou la division) de l'investissement initial de l'utilisateur. On l'obtient donc par le calcul $VF = p_T/p_0$ où p_t désigne le prix du portfolio à la $t^{\text{ième}}$ période, pour $0 \leq t \leq T$. On se contentera, pour nos tests, de fixer p_0 à 1 BTC.

La mesure de rentabilité n'est cependant pas suffisante, dans la mesure où elle ne prend pas en compte le risque encouru pour obtenir un tel résultat. Le ratio de Sharpe (RS) [20] cherche à pallier à ce manque et offre une mesure de rentabilité marginale par unité de risque, soit, en termes mathématiques, la moyenne de son rendement sans risque divisé par sa déviation :

$$SR = \frac{\mathbb{E}[\rho_t - \rho_F]}{\sqrt{\text{Var}[\rho_t - \rho_F]}} \quad (2)$$

où $\rho_t = \frac{p_t - p_{t-1}}{p_{t-1}}$ désigne le taux de rendement de l'algorithme pour la période t , et ρ_F le rendement du placement sans risque, ici celui de garder le portfolio fixe à 1 Bitcoin.

Enfin, le maximum drawdown (MDD) [21] est une mesure de la tendance du portfolio à prendre des risques. Là où le ratio de Sharpe traite indifféremment les variations à la hausse et à la baisse, indiquant ainsi une bonne valeur de la volatilité, le MDD sert à mesurer « le pire mouvement possible » dont est capable l'algorithme. Cela correspond mathématiquement à la plus grosse perte que le portfolio a réalisé entre un pic et un creux qui le suit dans le temps, soit à la formule :

$$MDD = \max \left\{ 0, \max_{\substack{t \\ k > t}} \left\{ \frac{p_t - p_k}{p_t} \right\} \right\} \quad (3)$$

3.2.2 Sur du court terme

La première situation imaginée est celle d'un utilisateur souhaitant obtenir des gains sur une très courte période, par exemple dans la crainte de faire de trop grosses pertes. Le but de cette partie est donc d'étudier le comportement des algorithmes sur une période de cette durée (e.g. sur une semaine). On a choisi la semaine du 25 juillet au 1^{er} août 2017, semaine qui correspond à une période assez « calme » et classique dans l'évolution du prix des cryptomonnaies, comme le montre la Figure 1. L'évolution des valeurs de portfolio conseillés par les différents algorithmes est affichée dans la Figure 2.

Les résultats de cette étude sur court terme, disponibles dans la Table 1, nous permettent d'apprécier au premier coup d'oeil la performance du réseau neuronal convolutif. En plus de dégager le plus grand profit (+13% de l'investissement initial, en seulement 7 jours), il réussit également à obtenir le ratio de Sharpe le plus faible. Il est suivi par l'algorithme Best, dont la décision a été de conserver l'ensemble de l'investissement en Bitcoin, conservant ainsi la mise initiale et sans

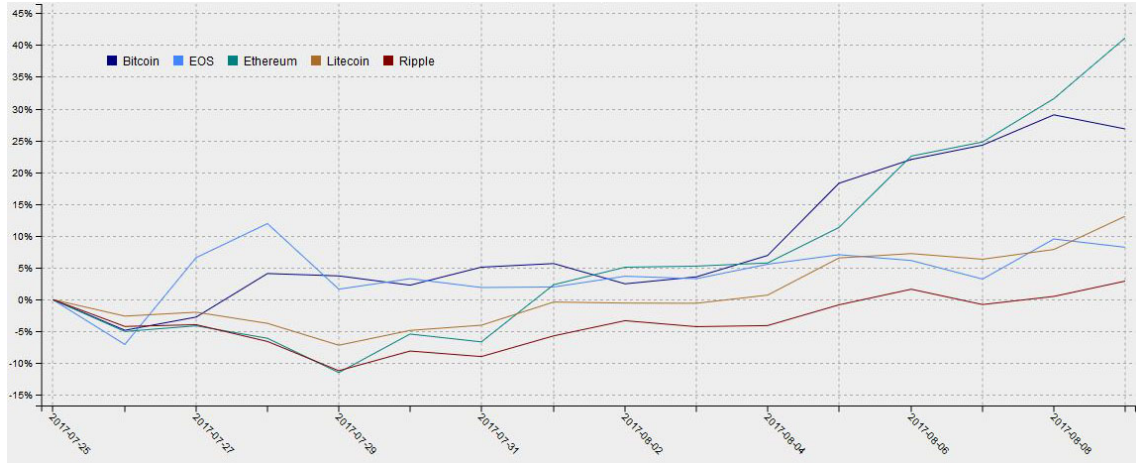


FIGURE 1 – Retour sur investissement (ROI) de 5 monnaies entre le 25/7/2017 et le 9/8/2017

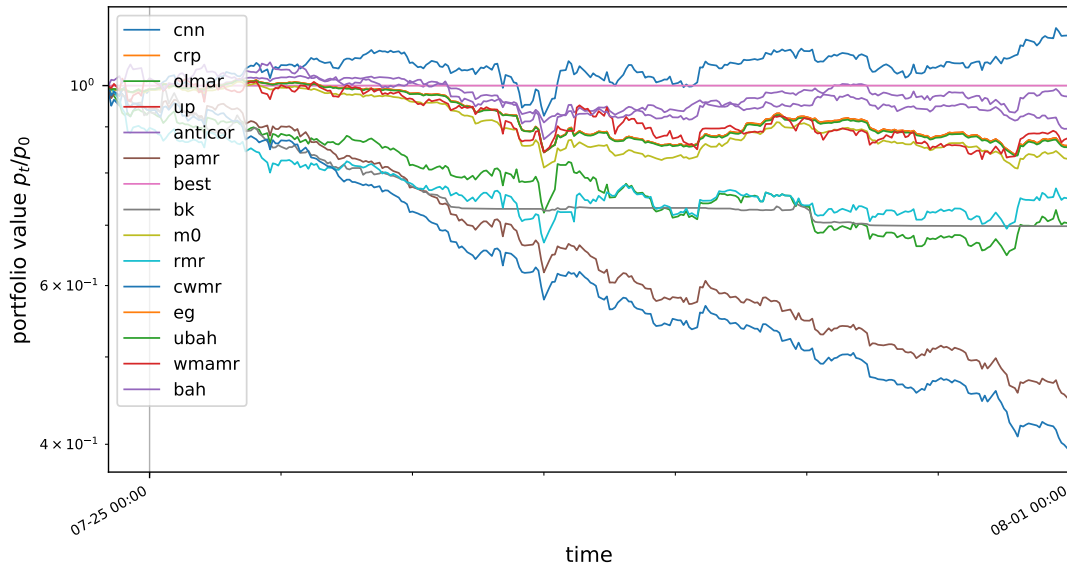


FIGURE 2 – Évolution des portefeuilles de chaque algorithme (25 juillet au 1^{er} août 2017)

risque (le ratio de Sharpe n'est cependant pas calculable dans ce cas là, cf. équation (2)). les autres algorithmes ne réussissent quant à eux pas à dégager de bénéfices sur cette période. On peut toutefois remarquer parmi eux la performance d'Anticor, qui obtient le plus petit maximum drawdown et la plus petite perte.

3.2.3 Sur du long terme en période « propice »

On cherche dans cette situation à observer le comportement des algorithmes en période de grande croissance du prix de la majorité des monnaies, afin de mesurer en quelques sortes quels algorithmes profitent le mieux d'une telle situation. On se focalise donc dans ce cas là sur leur capacité à maximiser le gain, le risque étant assez peu présent. Une telle situation décrit par exemple assez bien le climat propice de la fin de l'année 2017, comme le montre la Figure 3. On choisit donc d'effectuer cette étude sur la période du 1^{er} septembre au 1^{er} décembre 2017. L'évolution des valeurs des portefeuilles sont affichés dans la Figure 4.

Algorithme	Croiss. moy.	VF	MDD	RS
UBAH	0.999503	0.857542	0.175826	-0.090797
BAH	0.999659	0.895841	0.156347	-0.047898
UCRP	0.999489	0.854162	0.177425	-0.095134
Best	1.000000	1.000000	0.000000	N/A
CNN	1.000504	1.136845	0.154795	0.040674
Anticor	0.999930	0.972036	0.150485	-0.009877
OLMAR	0.998904	0.703084	0.342543	-0.088098
PAMR	0.997386	0.447489	0.551817	-0.233060
WMAMR	0.999633	0.876064	0.186466	-0.030030
CWMR	0.996959	0.393266	0.606756	-0.266809
RMR	0.999111	0.748939	0.327542	-0.072642
UP	0.999502	0.857191	0.175851	-0.091108
EG	0.999502	0.857368	0.175913	-0.091007
BK	0.998827	0.698287	0.298748	-0.170781
M0	0.999393	0.828398	0.198717	-0.095927

TABLE 1 – Performance des algorithmes sur une période courte

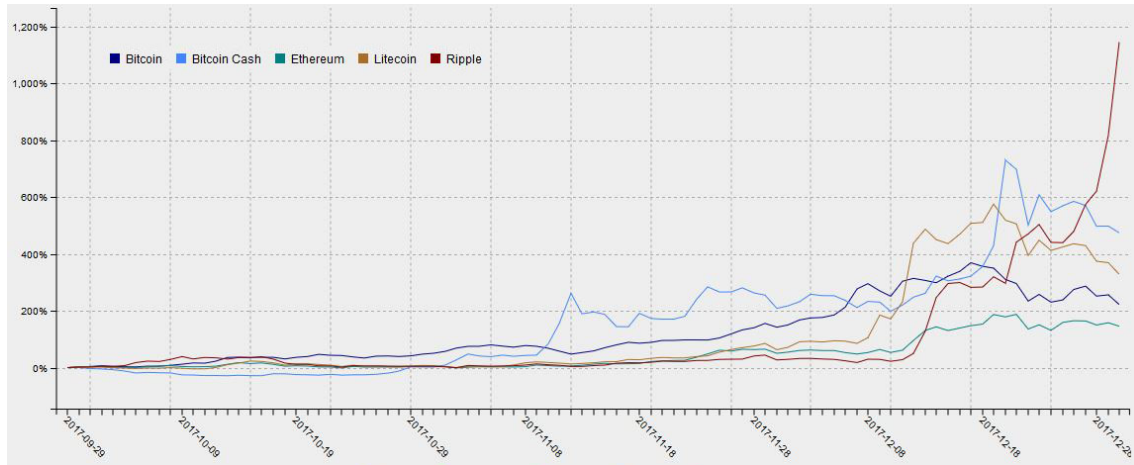


FIGURE 3 – Retour sur investissement (ROI) de 5 monnaies entre le 29/9/2017 et le 1/1/2018

Les résultats de la Table 2 nous montrent malheureusement là encore l’incapacité de la majorité des algorithmes à dégager un bénéfice, même en trois mois sur une période qui semble le plus propice pour cela. A nouveau, seuls Best et le CNN sont capables d’obtenir un multiplicateur de valeur supérieure à 1, et le CNN s’illustre par sa capacité à multiplier l’investissement initial par 5. Ils sont également les seuls à obtenir un ratio de Sharpe supérieur à 0. L’algorithme Uniform Constant Redistribution Portfolio minimise le maximum drawdown mais perd 34% de son investissement initial. Certains algorithmes s’illustrent par leur mauvais résultats, comme CWMR et PAMR qui divisent leur capital par plus de 1000, et OLMAR et RMR par près de 100. Ces résultats questionnent largement l’applicabilité des algorithmes de trading « classiques » à des marchés différents de ceux pour lesquels ils ont été pensés. Ceux-ci ont peut-être été sur-spécialisés, en intégrant toujours plus de connaissances expertes sur les marchés d’actifs classiques, et ne s’appliquent donc pas aussi bien au marché des cryptomonnaies. Une majorité des algorithmes qui performant le moins bien sont en effet datés des années 2010, tandis que ceux qui s’en sortent le mieux sont les plus anciens (par exemple UP 1991, EG 1998, UCRP 1956, etc.). Nous discuterons de la sous-jacence d’hypothèses sur le marché dans les modèles des algorithmes lors de leur étude approfondie (partie 3.3).

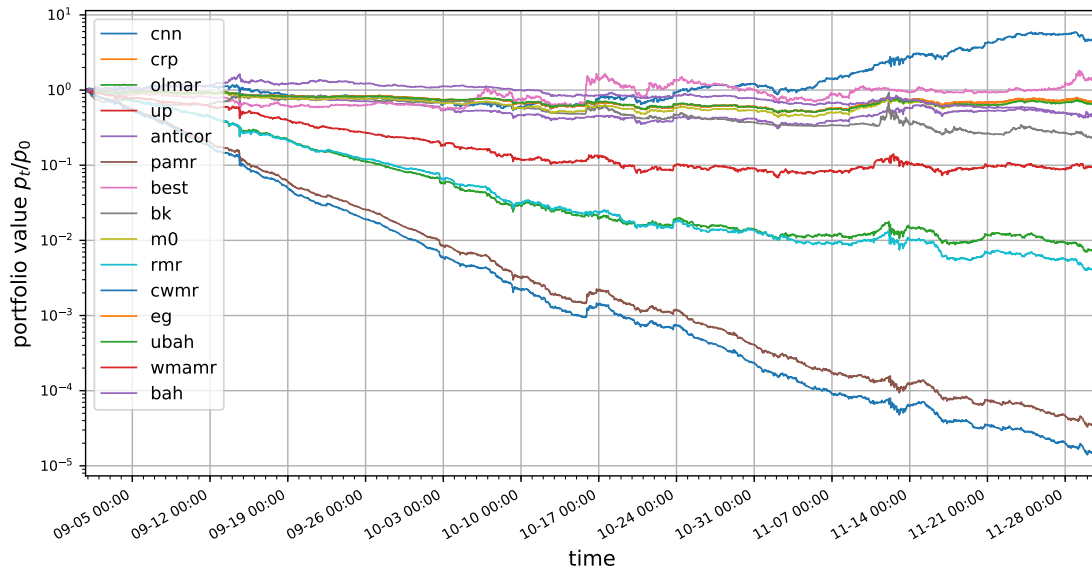


FIGURE 4 – Évolution des portefeuilles de chaque algorithme (1^{er} septembre au 1^{er} décembre 2017)

Algorithme	Croiss. moy.	VF	MDD	RS
UBAH	0.999940	0.713298	0.506966	-0.010091
BAH	0.999870	0.484452	0.737925	-0.015162
UCRP	0.999927	0.661413	0.503504	-0.010777
Best	1.000240	1.422701	0.585365	0.013452
CNN	1.000494	4.912568	0.542773	0.031007
Anticor	0.999873	0.466154	0.696386	-0.012878
OLMAR	0.998998	0.007047	0.993178	-0.059715
PAMR	0.997759	0.000034	0.999967	-0.138442
WMAMR	0.999575	0.093635	0.934437	-0.027190
CWMR	0.997566	0.000015	0.999986	-0.148947
RMR	0.998858	0.003859	0.996357	-0.068339
UP	0.999939	0.711239	0.506651	-0.010140
EG	0.999939	0.711675	0.506802	-0.010129
BK	0.999755	0.231534	0.772612	-0.017986
M0	0.999935	0.677607	0.577664	-0.009125

TABLE 2 – Performance des algorithmes sur une période propice

3.2.4 Sur du long terme en période « néfaste »

Les marchés des cryptomonnaies ne demeurant malheureusement pas toujours en pleine expansion, on s'intéresse cette fois à la capacité des algorithmes à limiter les risques, voir la perte, pour les utilisateurs qui utiliseraient notre produit dans une telle période. On étudie donc cette fois les décisions des algorithmes sur la période du 1^{er} janvier au 1^{er} avril 2018, qui correspond à une période de grande chute des prix des monnaies et une faible rentabilité des monnaies, comme illustré par la Figure 5. L'évolution de la valeur des portefeuilles de chaque algorithme est disponible dans la Figure 6.

Les résultats de l'étude de la Table 3 confortent ceux obtenus précédemment. Le CNN obtient de nouveau de très bons résultats, réussissant à dégager un bénéfice malgré la situation difficile, et doublant presque la valeur de l'investissement. Le portefeuille de Buy and Hold (BAH)(consistant



FIGURE 5 – Retour sur investissement (ROI) de 5 monnaies entre le 24/12/2017 et le 25/3/2018

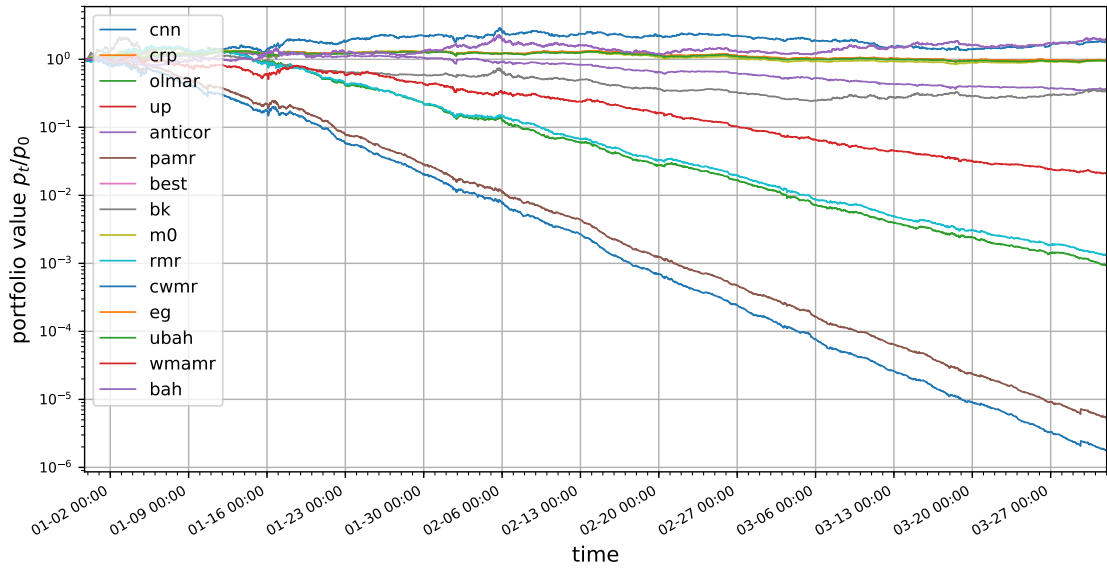


FIGURE 6 – Évolution des portfolios de chaque algorithme (1^e janvier au 1^e avril 2018)

à tout investir dans le dollar et donc à n'acheter aucune cryptomonnaie) arrive logiquement en seconde position. De même, Best, qui ne peut investir que dans une unique monnaie, semble avoir détecté également l'opportunité de n'acheter que du dollar. Ces deux portfolio disposent d'ailleurs du plus faible Sharpe ratio, ce qui témoigne de leur sécurité. UP et EG sont de nouveau presque à l'équilibre mais accusent d'une légère perte. Ils disposent cependant de très bonnes valeurs de maximum drawdown.

3.3 Étude des algorithmes retenus

Les études de la section précédente nous ont montré qu'un petit groupe d'algorithmes présente souvent les meilleurs résultats. Ce sont ces six algorithmes que l'on a choisis de conserver, d'implémenter dans notre produit et de proposer aux utilisateurs. Il s'agit de Uniform Constant Redistribution Portfolio (UCRP), Best Stock (Best), le réseau neuronal convolutif (CNN), Exponential Gradient (EG), Universal Portfolio (UP) et Anticor. Dans la suite, nous étudierons en détail les principes sur lesquels ils reposent.

Algorithmme	Croiss. moy.	VF	MDD	RS
UBAH	1.000008	0.986813	0.309470	0.001729
BAH	1.000220	1.989104	0.488889	0.019891
UCRP	1.000004	0.964661	0.310212	0.000799
Best	1.000220	1.989104	0.488889	0.019891
CNN	1.000222	1.824969	0.536745	0.017242
Anticor	0.999803	0.366518	0.731929	-0.023750
OLMAR	0.998482	0.000969	0.999380	-0.117113
PAMR	0.997289	0.000005	0.999995	-0.210151
WMAMR	0.999186	0.021587	0.982921	-0.067846
CWMR	0.997031	0.000002	0.999998	-0.224027
RMR	0.998558	0.001357	0.999178	-0.113020
UP	1.000008	0.985659	0.307534	0.001673
EG	1.000008	0.985941	0.308693	0.001690
BK	0.999814	0.346746	0.887654	-0.017139
M0	1.000009	0.952937	0.388243	0.001414

TABLE 3 – Performance des algorithmes sur une période néfaste

3.3.1 Uniform Constant Redistribution Portfolio (UCRP)

Longuement étudiée par COVER [10], cette stratégie illustre nous montrent que des solutions simples peuvent encore avoir du succès dans des situations où peu de connaissances expertes sont disponibles. Considéré aujourd’hui comme un algorithme de référence, seulement utilisé pour comparaison avec des algorithmes plus poussés, cet algorithme est un cas particulier d’algorithme suivant la stratégie « Constant Redistribution Portfolio ». Celle-ci repose sur l’idée simple de balancer à chaque période t le portfolio vers un portfolio fixe ω , soit en d’autres termes : $\forall t \in [0, T]$, $\omega_t^{\text{CRP}} = \omega$. La stratégie UCRP est celle pour laquelle le portfolio fixe est répartie équitablement entre chaque monnaie, c’est à dire :

$$\omega_t^{\text{UCRP}} = \left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m} \right) \quad \forall t \in [0, T], \text{ avec un portfolio de } m \text{ monnaies} \quad (4)$$

Elle est cependant à différencier de la stratégie Buy and Hold et de son pendant Uniform Buy and Hold, en cela que la stratégie BAH ne touche pas à la répartition des monnaies après avoir fixé le portefeuille au temps $t = 0$. Dans UCRP, si la valeur de l’une des monnaies change, par exemple augmente fortement, la valeur du portefeuille est recalculée puis la part allouée à cette monnaie sera réduite, pour conserver une distribution de valeur égale entre toutes les monnaies.

3.3.2 Best Stock

Développé également dans [10], l’algorithme Best est également un algorithme utilisé fréquemment comme référence. Dans l’étude empirique réalisé précédemment, il surpasse pourtant toutes les autres stratégies, à l’exception de celle du CNN. Best est un cas particulier d’algorithme Buy and Hold (BAH), stratégie qui fonctionne comme on l’a vu précédemment avec un portfolio fixé à $t = 0$, puis non modifié. Sa valeur cumulative finale est donc :

$$VF = \omega_0^{\text{BAH}} \cdot \left(\bigodot_{t=0}^T y_t \right) \quad (5)$$

avec \odot l’opérateur de produit terme à terme, \cdot le produit scalaire, et y_t le vecteur de prix des monnaies à la période t .

Partant de cette simple formule, l’idée de l’algorithme Best est de trouver le portefeuille fixe qui aurait maximisé le gain sur une période antérieure à la date de début du trading on-

line, idéalement finissant à la date où celui-ci commence. Le vecteur fixe de poids choisi par cet algorithme est donc :

$$\omega_0^{\text{Best}} = \arg \max_{\omega \in \Omega_m} \left(\omega \cdot \left(\bigodot_{t=0}^T y_t \right) \right) \quad (6)$$

3.3.3 Exponential Gradient (EG)

L'algorithme EG, développé par HELMBOLD et al. [17], appartient à la catégorie des approches « Follow-the-Winner », c'est à dire une approche dont l'hypothèse sous-jacente est qu'un actif ayant été performant à une période t devraient également l'être au temps $t + 1$. Contrairement aux algorithmes précédents, l'approche de EG est on-line, c'est à dire que la décision du prochain portfolio se fait en fonction des résultats du temps présent. Exponential Gradient est formulé comme le problème d'optimisation suivant :

$$\omega_{t+1}^{\text{EG}} = \arg \max_{\omega \in \Omega_m} (\eta \log (\omega \cdot y_t - R(\omega, \omega_t))) \quad (7)$$

où $\eta > 0$ est le coefficient d'apprentissage et $R(\omega, \omega_t)$ dénote le terme de régularisation. On peut globalement interpréter cette équation comme une recherche du portfolio avec la meilleure performance à la période précédente mais avec une volonté de garder le portfolio suivant proche du précédent. Le terme de régularisation employé est :

$$R(\omega, \omega_t) = \sum_{i=1}^m \omega(i) \log \frac{\omega(i)}{\omega_b(i)} \quad (8)$$

Les auteurs montrent par la suite que ce problème peut être simplifié en la formule :

$$\omega_{t+1}^{\text{EG}}(i) = \frac{1}{k} \omega_t(i) \exp \left(\eta \frac{y_t(i)}{\omega_t \cdot y_t} \right) \quad \forall i \in [1, \dots, m] \quad (9)$$

où k est le terme de normalisation qui assure que les poids du vecteurs somment à 1.

3.3.4 Universal Portfolio (UP)

Imaginé également dans [10], la stratégie UP est aussi de type « Follow-the-Winner ». Son idée est de calculer les performances de tous les portfolios valide au cours des périodes écoulées, puis d'en effectuer un mixage pondéré par ces performances de gain. Formellement, la procédure est la suivante :

$$\omega_{t+1}^{\text{UP}} = \int_{\Omega_m} \omega S_t(\omega) d\omega \Bigg/ \int_{\Omega} S_t(\omega) d\omega \quad (10)$$

où l'intégration se fait donc sur Ω_m , l'ensemble des portfolios valides, et $S_t(\omega)$ est défini par :

$$S_t(\omega) = \prod_{k=1}^t \omega \cdot y_k \quad (11)$$

L'apparente difficulté dans la formulation et la compréhension de cet algorithme se retrouve dans sa complexité temporelle, puisque cette dernière est de $O(n^m)$ avec n le nombre de périodes et m le nombre de monnaies. Il existe une large littérature qui s'intéresse aux possibilités de son amélioration, notamment pour le calcul des poids S_t . La procédure utilisée dans notre implémentation réalise un échantillonnage de portfolios pour approximer l'intégrale, réduisant ainsi la complexité à $O(npm^2)$ où p désigne le nombre de portefeuilles dans l'échantillon.

3.3.5 Anticor

Anticor (pour anti-corrélation), suit pour sa part une approche « Follow-the-Loser », basée sur l'intuition qu'un actif ayant eu de mauvais résultats à une période est plus susceptible d'en avoir des bons à la période suivante. Contrairement par exemple à UP, qui ne fait aucune hypothèse de distributions, la stratégie d'Anticor repose sur l'hypothèse que le marché suit le principe de mean reversion, selon lequel le prix d'un actif tendra à évoluer vers son prix moyen. Cette stratégie cherche à exploiter le marché en pariant sur la consistance de corrélations croisées retardées, et d'auto-corrélation négative récentes.

Elle s'intéresse donc à deux fenêtres temporelles pour les prix : $X_1 = \log(y_{t-2\alpha+1}^{t-\alpha})$ et $X_2 = \log(y_{t-\alpha+1}^t)$, deux matrices extraites de y_t pour une fenêtre de temps donnée α et passées au log. Elle utilise ensuite aux matrices de corrélation croisées entre X_1 et X_2 :

$$M_{cov}(i, j) = \frac{1}{\alpha - 1} (X_1(i) - \mu_1(i))^T (X_2(j) - \mu_2(j)) \quad (12)$$

et

$$M_{cor}(i, j) = \begin{cases} \frac{M_{cov}(i, j)}{\sigma_1(i)\sigma_2(j)} & \text{si } \sigma_1(i), \sigma_2(j) \neq 0 \\ 0 & \text{sinon} \end{cases} \quad (13)$$

où $\mu_k = (\mu_k(1), \dots, \mu_k(m))$ est le vecteur des moyennes des colonnes de X_k , et de même σ_k celui des écarts-types.

Finalement, la décision de redistribution des poids entre les monnaies i et j ne se fait que si l'on détecte une corrélation entre ces deux monnaies, et que la première monnaie se déplace « plus rapidement » que la seconde, i.e. ssi $M_{cor}(i, j) > 0$ et $\sigma_2(i) > \sigma_2(j)$. Dans ce cas, la quantité de monnaie à déplacer de i à j , disons $d_{i \rightarrow j}$ est proportionnelle à cette corrélation ainsi qu'à l'auto corrélation de chaque monnaie. On peut écrire ce processus sous forme algorithmique :

$$\forall i, j \in [1, \dots, m], d_{i \rightarrow j} = \begin{cases} M_{cor}(i, j) + A(i) + A(j) & \text{si } M_{cor}(i, j) > 0 \text{ et } \sigma_2(i) > \sigma_2(j) \\ 0 & \text{sinon} \end{cases} \quad (14)$$

avec

$$A(k) = \begin{cases} |M_{cor}(k, k)| & \text{si } M_{cor}(k, k) < 0 \\ 0 & \text{sinon} \end{cases} \quad (15)$$

La ré-allocation de poids est finalement :

$$\omega_{t+1}^{\text{Anticor}}(i) = \omega_t(i) + \sum_{i \neq j} (t_{j \rightarrow i} - t_{i \rightarrow j}) \quad (16)$$

avec $t_{j \rightarrow i}$ le transfert de i à j défini par

$$t_{j \rightarrow i} = \frac{\omega_t(i) \cdot d_{i \rightarrow j}}{\sum_j d_{i \rightarrow j}} \quad (17)$$

3.3.6 Le réseau neuronal convolutif

Comme expliqué plus tôt, cet algorithme a été imaginé et présenté dans un article par JIANG et LIANG [7]. Le principe de cet algorithme est de se débarrasser de toute modélisation ou hypothèses sur l'évolution du marché, contrairement par exemple aux approches Follow-the-Winner ou Follow-the-Loser. En procédant ainsi, nos expériences ont montré qu'on obtenait des résultats bien supérieurs, mais on sacrifie dans un même temps toute capacité à expliquer la logique derrière la décision prise par l'algorithme. La domination du CNN semble ainsi montrer que les logiques de fluctuation des marchés financiers (et de celui des cryptomonnaies en particulier) sont loin d'être comprises, et que les connaissances expertes qui ont été conçues et testées sur des

marchés boursiers classiques (voir les résultats de [2]) ne donnent pas des résultats aussi bons sur les marchés constitués de nouveaux types d'actifs comme les cryptomonnaies.

Le principe général de l'algorithme est de suivre un apprentissage par renforcement, avec utilisation d'une politique déterministe. L'environnement est défini comme le marché financier des cryptomonnaies, et l'état de l'agent (sa connaissance de l'environnement) comme les prix actuels et passés des monnaies, considérés traditionnellement par les analystes financiers [22]. L'état d'un agent à un temps t est donc simplement défini par le portefeuille qu'il hérite du temps précédent et d'un tenseur de prix $X_t : s_t = (X_t, \omega_{t-1})$. Il dispose également d'une fonction de reward, directement induite par la valeur finale du portfolio p_T :

$$p_T = p_0 \prod_{t=1}^{T+1} \mu_t y_t \cdot \omega_{t-1} \quad (18)$$

où p_0 désigne l'investissement initial, y_t le vecteur de prix des monnaies au temps t , ω_t le vecteur de poids du portfolio, et μ_t un facteur de réduction, par excellence $1 - c$ où c désigne le taux de commission. La fonction de reward n'est autre que la moyenne logarithmique de cette valeur :

$$R = \frac{1}{T} \sum_{t=1}^{T+1} \ln(\mu_t y_t \cdot \omega_{t-1}) \quad (19)$$

Enfin, la politique (qui n'est autre qu'une fonction de projection de l'espace d'état vers celui des actions, i.e. $\pi_\theta : S \mapsto A$), est obtenue grâce à un algorithme de montée de gradient, où la métrique de performance n'est autre que le reward associé au paramètres $\theta : R(s_1, \pi_\theta(s_1), \dots, s_T, \pi_\theta(s_T), s_{T+1})$. Il est intéressant de noter que cet agent va donc simplement chercher à maximiser son profit sans s'intéresser au risque encouru, mais réussit tout de même à minimiser ce dernier comme nous l'avons vu au cours des tests empiriques.

La structure adoptée pour l'implémentation de cet agent est celle d'un réseau neuronal convolutif (CNN) appliqué à la prédiction de portefeuille comme pourrait le faire un CNN classique appliqué à la reconnaissance d'image. Il prend en entrée une matrice de taille $3 \times m \times n$, où 3 correspond aux trois features « closing » (prix d'échange), « highest » (prix le plus haut) et « low » (prix le plus bas), m au nombre de monnaies choisies (hors monnaie de réserve) et n la fenêtre de temps i.e. un nombre de périodes précédentes. L'algorithme prend également en entrée le portfolio de la période précédente, pour permettre à l'algorithme de minimiser les coûts de transaction.

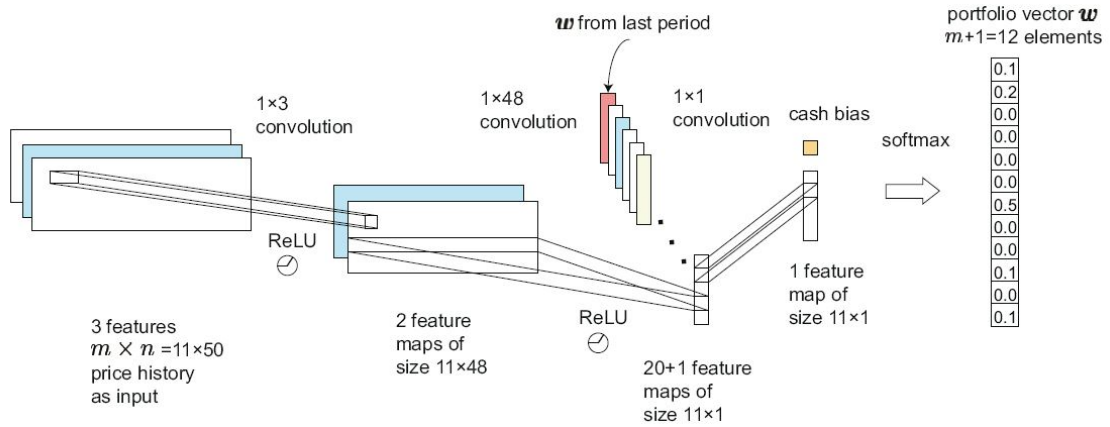


FIGURE 7 – Structure du réseau neuronal convolutif (CNN)

La structure du réseau est basée sur une dichotomie avec celle des CNN utilisés pour la détection d'image, où les 3 valeurs RGB sont remplacées par les 3 valeurs des features. L'astuce de la structure employée est que les m lignes correspondant aux m monnaies sont traitées sur des réseaux identiques mais indépendants (EIEE, Ensemble of Identical Independent Evaluators), et ne s'inter-connectent qu'à la fin en un vote (scoring) utilisant une fonction softmax pour garantir

la non-négativité des portfolio prédits et la sommation à 1. La Figure 7, tirée de l'article cité plus tôt, résume cette organisation : on observe une décomposition en 4 convolutions, où les lignes ne se mélangent jamais. Le portfolio de l'étape précédente entre en jeu à la troisième étape, et la monnaie de réserve uniquement à la quatrième, lors de l'agrégation des résultats des IIE. JIANG et LIANG [7] ont implémenté cette structure de convolutions sous Tensorflow, structure que nous avons repris dans notre projet.

Le CNN est sensible à plusieurs hyperparamètres, qu'il est nécessaire de fixer pour une utilisation future au sein du logiciel. On retiendra notamment $n = 50$, le nombre de périodes à considérer dans notre matrice X_t , , fenêtre assez large pour fournir des données fortement corrélées (très proches dans le temps) et peu corrélées (plus anciennes) à l'algorithme. Enfin, le dernier hyperparamètre est la fréquence des données passées en entraînement. Nous choisirons de fixer cette fréquence à 30 minutes, fréquence également utilisée par JIANG et LIANG [7], parce qu'elle semble correspondre à la fréquence maximale qu'un utilisateur puisse faire de notre outil sans avoir plutôt besoin d'un outil de trading haute fréquence, et elle permet également de garantir que l'ensemble des ordres d'achat ou de vente pourront être passés et exécutés avant une prochaine optimisation, notamment sur des marchés avec une plus faible liquidité.

4 Développement logiciel

Dans cette partie, nous aborderons les étapes et les choix effectués pour l'intégration des algorithmes précédemment choisis vers la création d'une interface d'aide à la décision en ligne. Nous décrirons succinctement l'organisation des tâches qui a été suivie, puis rentrerons dans le détail des fonctionnalités proposées et de leur disposition au sein de l'interface, avant de discuter des choix techniques qu'il a été nécessaire de réaliser.

4.1 Organisation des tâches

La réalisation de ce projet, où nous n'étions que deux à travailler, a nécessité organisation et répartition des tâches. Nous détaillons dans la suite quelles ont été les grandes étapes de travail, comment le travail a été réparti, et de quelle manière il a été organisé.

4.1.1 Étapes de développement

Pendant le premier mois nous avons travaillé étroitement avec Thibaut Lust, notre directeur de projet. La problématique essentielle était d'élaborer un cahier des charges pour les mois à venir. Il a été question d'établir un projet raisonnable à implémenter dans le temps imparti, qui apporterait de la valeur à l'utilisateur, et qui aurait un sens d'un point de vue algorithmique. Ce travail s'est avéré étroitement lié à celui de la revue de la littérature. En effet, pour comprendre ce qui était possible de faire, il nous fallait d'abord envisager l'étendue des possibilités et considérer les travaux déjà effectués.

Après ce travail de recherche bibliographique et cahier des charges, nous nous sommes répartis le travail de la façon suivante :

Julien Denes :

1. Étude théorique des algorithmes.
2. Compréhension du code implémenté par JIANG et LIANG [7], ré-écriture du code pour le simplifier, le rendre plus rapide, l'épurer.
3. Amélioration dans l'implémentation pour pouvoir accéder aux monnaies/algorithmes qui nous intéressaient et se connecter aux API d'échanges de cryptomonnaies.
4. Choix des algorithmes, analyse de performance sur des données que nous avons choisi pour comparer les performances des algorithmes retenus.

Michaël Trazzi :

1. Étude des besoins de l'utilisateur et de la faisabilité des fonctionnalités. Code html/css destiné à faire une première maquette avec un environnement de développement (bootstrap). Création d'une « landing page » pour comprendre les besoins client.
2. Prise en main de l'environnement de développement Django 1.8 (pour Python 3.5). Correspondance entre le front-end (html/css) et des « views »/« templates » en Django.
3. Interface entre la plateforme Django et les algorithmes implémentés en back-end.
4. Amélioration des fonctionnalités proposées, gestion de la base de données, procédures d'authentifications, d'enregistrement/suppression de portfolio, de graphiques.
5. Mise en production de la plateforme sur un serveur distant en utilisant Heroku.

4.1.2 Calendrier

Le calendrier suivant résume les grandes étapes suivies lors de la réalisation de ce projet, et leur agencement temporel.

— Lundi 29 janvier : début du semestre et début du projet

- Mercredi 28 février : cahier des charges réalisé
- Mercredi 7 mars : revue de la littérature terminée
- Mercredi 15 mars : cahier de bord terminé, liste des algorithmes établie
- Mercredi 21 mars : récupération des données des monnaies établie
- Mercredi 18 avril : analyse et test des performances des algorithmes terminés
- Mercredi 25 avril : début de l'implémentation de la plateforme web
- Mercredi 16 mai : implémentation de la plateforme web terminée, mise en production
- **Vendredi 25 mai : rapport à rendre**
- **Mardi 5 : soutenance**

4.2 Pages et fonctionnalités du produit

Les sections suivantes détaillent les différentes fonctionnalités du produit final, et présentent des captures des pages sur lesquelles elles sont implémentées. Ce site est implémenté en Python Django, permettant une intégration plus aisée des algorithmes écrits en Python à des pages web.

4.2.1 Landing page

La Figure 8 présente la landing page de notre site, et disponible à l'adresse <https://michaeltrazzi.wixsite.com/cryptooptimisation>, hébergée par la plate-forme Wix. Celle-ci remplit un rôle assez standard, qui est donc d'informer sur l'existence de l'outil, expliquer son principe et inciter à l'utilisation. C'est elle que l'on cherchera à mettre en avant, par exemple avec un bon référencement. Elle dispose d'un bouton central qui invite les utilisateurs à continuer vers l'application centrale, hébergée par Heroku.

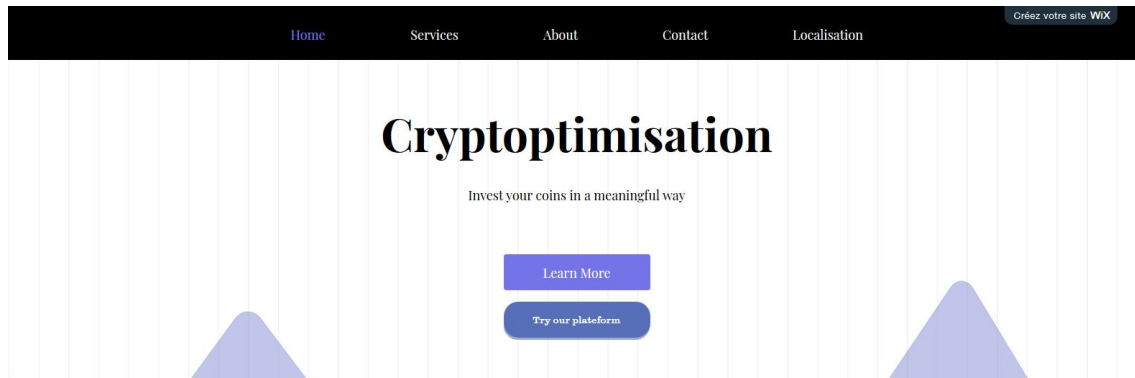


FIGURE 8 – Extrait de la landing page

4.2.2 Inscription et connexion

En cliquant sur ce bouton, l'utilisateur est redirigé vers la page de connexion de l'application (<https://cryptooptimize.herokuapp.com/>). Comme montré dans la Figure 9, celle-ci invite soit à la connexion, soit à l'inscription. L'étape d'inscription nécessite que l'utilisateur enregistre un identifiant et un mot de passe, reliés à une adresse mail valide. Un email automatique d'activation est donc envoyé sur celle-ci avec d'effectuer une vérification.

Le bon fonctionnement de la plate-forme nécessite en effet que l'utilisateur puisse s'identifier de manière unique, puisqu'elle dispose d'une possibilité d'enregistrer des portefeuilles. De plus, puisque notre plate-forme pourrait à plus long terme aller plus loin dans les possibilités de gestion de portefeuilles, il est nécessaire d'offrir un espace sécurisé pour le stockage de données relatives au capital des utilisateurs.

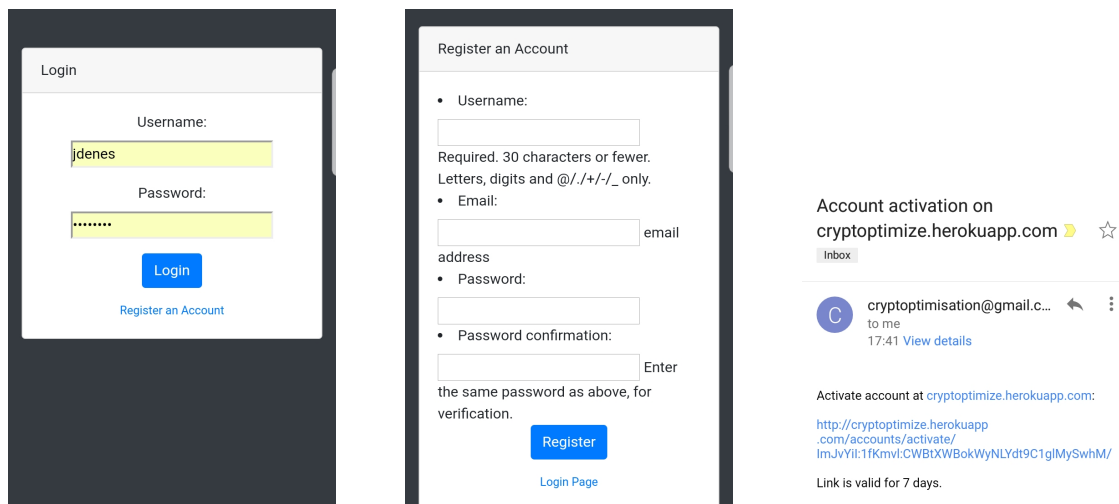


FIGURE 9 – Pages de login, d'inscription et email de vérification de l'adresse

4.2.3 Dashboard et barre de navigation

Une fois connecté, l'utilisateur accède à son tableau de bord. Comme visible sur la Figure 10, celui-ci dispose de deux fonctionnalités. En haut, la possibilité de suivre la composition et l'évolution de la valeur d'un portefeuille. Celle-ci n'est malheureusement pas opérationnelle à l'heure de l'écriture de ce rapport (25 mai 2018). En dessous, un tableau permet à l'utilisateur de suivre la capitalisation et l'évolution des valeurs des principales cryptomonnaies. A gauche de cette page, un menu de navigation permet à l'utilisateur d'atteindre les fonctionnalités suivantes.

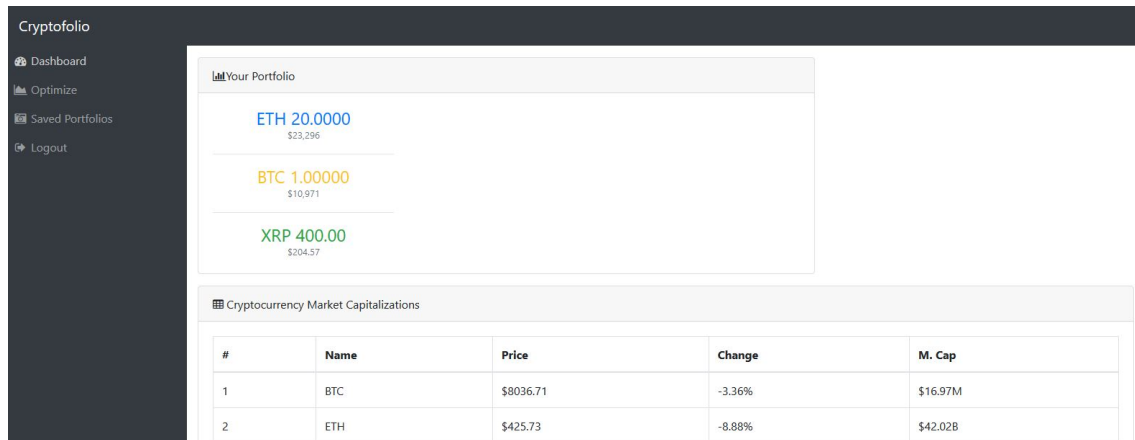


FIGURE 10 – Tableau de bord et menu de navigation

4.2.4 Outil d'optimisation

La fonctionnalité centrale est bien entendu la page d'optimisation de portefeuille. Comme on le voit sur la Figure 11, on peut y entrer le nom du portefeuille, l'investissement de départ, et choisir son algorithme d'optimisation. Cette dernière option nécessitera bien entendu d'ajouter par la suite une documentation pour permettre aux utilisateurs de comprendre les différences entre eux. La Figure 12 nous montre également la forme du résultat. Elle présente notamment la possibilité d'enregistrer le portefeuille généré.

Cryptofolio

Dashboard

Optimize

Saved Portfolios

Logout

Portfolio

Portfolio Name

Enter name

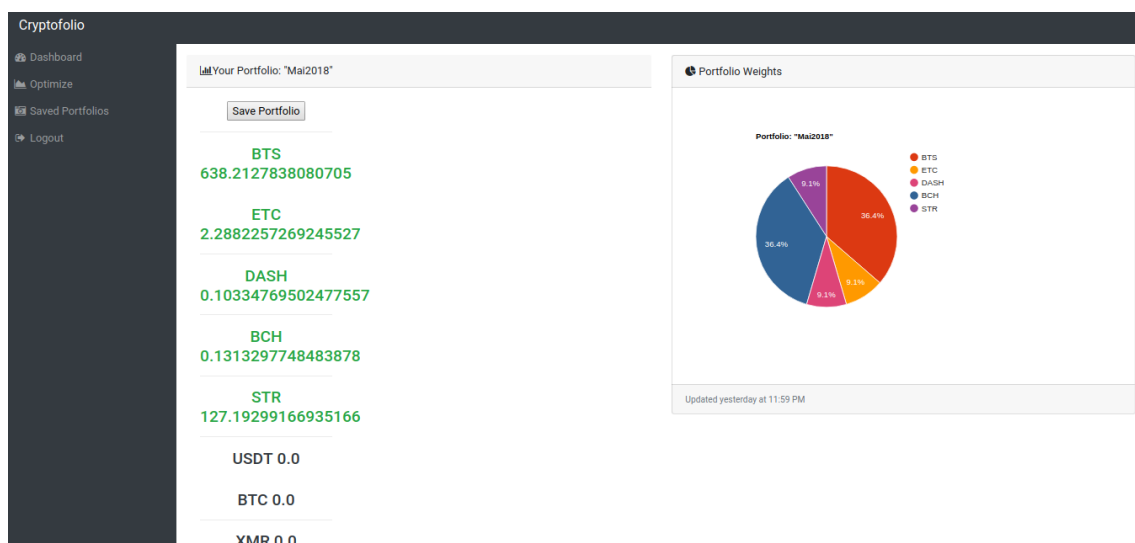
Initial Investment (BTC)

Enter a Value in BTC

Réseau Convolutif (CNN)

Optimize

FIGURE 11 – Interface d’optimisation



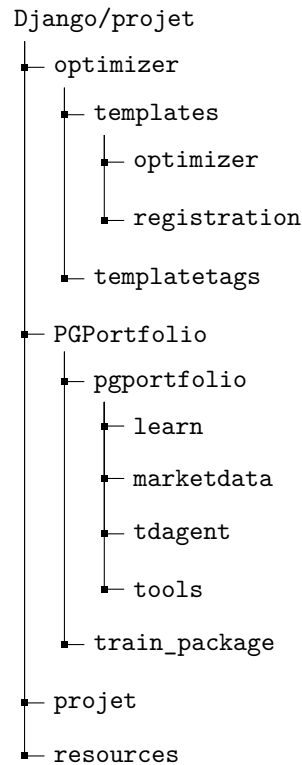


FIGURE 14 – Arborescence des dossiers de code

4.2.5 Enregistrement des portefeuilles

Les portfolios ainsi enregistrés sont disponibles sur la page dédiée, comme montré dans la Figure 13. Elle permettra à plus long terme à l'utilisateur de suivre l'évolution de la valeur de ces derniers, et éventuellement de les ré-optimiser plus tard.

4.3 Organisation du code

Depuis la racine du repository, le code de ce projet se trouve dans le dossier `Django/projet`. Les fichiers peuvent être globalement découpés en deux catégories : les fichiers dédiés à l'implémentation des algorithmes, disponibles dans `Django/projet/PGPortfolio`, et ceux servant au framework `Django` (par défaut, le reste). Nous détaillons dans la suite la structure de ces deux ensembles. La structure générale du dossier `Django/projet` est décrite dans la Figure 14.

Bien que disponible en ligne, la plate-forme peut également être lancée en local. Il suffit de lancer la commande `python manage.py runserver` une fois avoir changé le dossier courant (il faut se placer au même niveau que `manage.py`, donc dans `Django/projet`), sous réserve d'avoir au préalable installé les packages python requis donnés dans `Django/projet/requirements.txt`.

4.3.1 Framework Django

A la racine, on trouve les fichiers nécessaires à la gestion de base de données : enregistrement/suppression de portfolios et piecharts dans les csv `portfolios.csv` et `charts.csv`, données utilisateurs dans `db.sqlite3`. On y trouve également un détail des packages requis (`requirements.txt` et `conda-requirement.txt`), les fichiers nécessaires au déploiement automatique du serveur sur la plateforme Heroku (`Pipfile`, `Procfile`, `runtime.txt`) ainsi que le fichier Python nécessaire au bon fonctionnement d'un serveur Django : `manage.py`.

Dans `Django/projet/optimizer/templates` sont présentes les différentes pages HTML que l'on peut trouver sur le site. Le sous-dossier `Django/projet/optimizer/templates/optimizer` contient les pages que voit un utilisateur connecté :

- `dashboard.html` (tableau de bord)
- `optimize.html` (interface d'optimisation)
- `portfolios.html` (ensemble des portfolios sauvés par l'utilisateur)
- `result.html` (résultat de l'optimisation)

Dans `Django/projet/optimizer/templates/registration`, on trouve l'ensemble des fichiers servant à implémenter le masque HTML des fonctionnalités d'inscription (notamment envoi automatique de mails), connexion et gestion des identifiants.

A la racine de `Django/projet/optimizer` et `Django/projet/projet` on trouvera des fichiers `urls.py` et `views.py` qui ont pour rôle de faire la transition entre le serveur Django et les templates HTML. Dans `Django/projet/projet/settings.py` sont définis l'ensemble des constantes nécessaires au bon fonctionnement d'un serveur Django. Pour que la plateforme envoie effectivement des mails avec votre serveur mail, il faudra en particulier changer la variable `EMAIL_HOST_PASSWORD` de `EMAIL_HOST_USER`.

4.3.2 Code des algorithmes

Cette implémentation est concentrée dans le dossier `Django/projet/PGPortfolio`. A sa racine, le fichier central `main.py` fait appel aux autres composants du code de ce dossier pour prédire le portfolio et a vocation à être appelé dans `Django/projet/optimizer/views.py`.

Dans `PGPortfolio/pgportfolio/learn`, on retrouve les fichiers servant à implémenter le réseau neuronal convolutif et ses comportements (`nnagent.py` et `network.py`). Le répertoire `PGPortfolio/pgportfolio/tdagent` contient quant à lui les fichiers d'implémentation des algorithmes de trading « classique ». A sa racine, `tdagent.py` contient la classe générale de ces algorithmes, et les fichiers contenus dans `algorithms`, qui correspondent aux différents algorithmes, héritent de celle-ci.

Dans `PGPortfolio/pgportfolio/marketdata`, le fichier `poloniex.py` regroupe les outils de communication avec l'API de Poloniex. `PGPortfolio/pgportfolio/tools` contient pour sa part des fichiers utilitaires améliorant la gestion des données notamment pour la lecture des paramètres des algorithmes (`configprocess.py`) et le traitement des données de l'API (`data.py`).

Enfin, le dossier `PGPortfolio/train_package` contient un certain nombre de sous-dossiers, où chacun contient les fichiers correspondants à un CNN entraîné (identifié par un chiffre). Ces fichiers suivent des formats propres à Tensorflow et stockent une représentation des poids du réseau neuronal. Le fichier `net_config.json` stocke quant à lui les paramètres de ce réseau.

4.4 Choix techniques

Plusieurs choix techniques ont dû être réalisés pour l'implémentation de la plate-forme d'optimisation. Nous détaillons ci-dessous les principales, et les motivations qui nous ont poussés à les adopter.

4.4.1 Connexion et stockage des données des utilisateurs

L'un des objectifs de notre produit est de pouvoir fournir aux utilisateurs un suivi de leurs portfolios, en plus de la possibilité de les optimiser. Il est donc nécessaire que ceux-ci disposent d'une possibilité de s'identifier, et donc de posséder espace personnel avec connexion sécurisée. Nous avons donc fait le choix d'imposer la création d'un login et d'un mot de passe pour la première utilisation du site, et une nécessité de se connecter pour y accéder par la suite. Les données relatives

à ces identifiants sont stockées dans une base de donnée intégrée à Django qui prend en charge le chiffrement des données sensibles comme les mot de passe.

En outre, nous proposons aux utilisateurs qui s'inscrivent sur la plateforme la possibilité de se créer un compte. Ils remplissent alors un formulaire qu'ils soumettent à notre site. Nous leur envoyons automatiquement par mail un lien d'activation pour qu'ils confirment leur compte. Leur compte validé, ils peuvent se connecter de façon sécurisée à notre plateforme.

4.4.2 Récupération des données relatives aux cryptomonnaies

L'optimisation des portefeuilles, tout comme l'affichage des cours des monnaies, ou le suivi de l'évolution de la valeur des portefeuilles, nécessitent l'accès à des données en temps réel des prix de ces monnaies. Nous avons pour cela fait le choix d'utiliser l'interface de programmation (API) de la plate-forme d'échange Poloniex, l'une des plus large plate-forme à l'heure actuelle. Celle-ci est en effet facile d'utilisation grâce à des requêtes simples, et propose un temps de réponse très court quelle que soit la taille des données demandées. C'est également pour cette raison que nous avons choisi, pour l'ensemble des instruments de notre produit, de ne pas utiliser de base de données pour stocker les informations nécessaires mais d'interroger à chaque fois l'API. Nous avons en effet constaté, après avoir travaillé avec le code source de l'article de JIANG et LIANG [7] qui utilisait pour sa part une base de données, que deux inconvénients se présentaient. Tout d'abord celui de l'espace, puisque la taille de la base de données croît rapidement avec l'ajout de points de données, jusqu'à peser plus d'un demi Go pour les données d'une dizaine de mois. La seconde, qui lui est liée, est celle du temps : si une économie de temps pouvaient être réalisée grâce à la constitution de la base puis aux requêtes SQL lorsque les données devaient être lues de nombreuses fois, elle devient inutile lorsqu'on ne souhaite y accéder qu'une fois périodiquement, comme dans le cas de l'optimisation du portefeuille.

4.4.3 Fréquence et données d'entraînement du CNN

L'une des décisions associée à l'utilisation du CNN est celle de la fréquence de ré-entraînement de celui-ci, problématique fortement liée à celle de la quantité de données passées en input pour cet entraînement.

Pour le choix de la quantité de données, nous basons notre raisonnement sur le théorème d'approximation universelle [23], selon laquelle un réseau neuronal à propagation avant composé d'un nombre fini de neurones (perceptrons) et dotés d'une fonction d'activation continue, croissante, non-constante et bornée peut approximer n'importe quelle fonction continue sur un sous-ensemble de \mathbb{R}^n . Les hypothèses sont bien vérifiées dans notre CNN, puisque celui-ci utilise une fonction Rectified Linear Unit (RELU, définie par $g(x) = \max(0, x)$) dans les convolutions du EIIE, puis une fonction softmax pour les combiner. Il est de plus évident qu'il existe une infinité de fonctions $f: y_t \mapsto \omega_t$ continues et définies de \mathbb{R}^m vers $\Omega_m \subset \mathbb{R}^m$, dont on chercherait à maximiser l'argmax de l'équation (18). A partir de ce théorème, on peut donc écarter un questionnement sur le fait de donner « trop » de données en entraînement à l'algorithme, et notamment des données « trop anciennes ». Le théorème d'approximation universelle et cette équation nous indiquent en effet que le CNN cherchera la fonction qui maximise la valeur amortie, et donc celui qui se comporte mieux en moyenne, et donc dans toutes les situations possibles. S'il est légitime de penser que les mouvements de marché actuels sont plus ressemblant aux mouvements du passé proche de d'un passé plus ancien, ce raisonnement nous pousse à penser qu'il est moins risqué de fournir de nombreux motifs en entraînement, fussent-ils obsolètes. On choisit donc de fournir à l'algorithme les matrices de valeur des monnaies X_t sur deux ans précédant la date actuelle.

A partir de cette décision, il nous semble légitime de penser que l'ajout de quelques dizaines de périodes à la matrice X_t , voir même de quelques centaines, ne changera pas fondamentalement le résultat (la matrice X_t contiendra en effet $2 \times 365 \times 24 \times 2 = 35040$ périodes, une dizaine semblent marginales). Il est cependant nécessaire d'être conscient que cette hypothèse pourrait s'avérer fallacieuse, puisque l'on ne peut connaître l'importance relative donnée à chaque période

par le CNN en fonction de son ancienneté. A partir de celle-ci cependant, nous avons décidé que le ré-entraînement s'effectuerait tous les 7 jours (soit un changement de 336 périodes).

4.5 Limitations du produit final

Par manque de temps ou face à des limitations techniques, certaines caractéristiques initialement prévues dans le projet ont dû être revues à la baisse ou supprimées. Nous les détaillons dans la suite ainsi que les choix techniques que nous avons réalisés pour les contourner.

4.5.1 Nombre de portefeuilles possibles

L'un des problèmes majeurs liés à l'utilisation du réseau neuronal convolutif est lié à sa nécessité d'un entraînement en fonction de chaque situation. Nous avons discuté plus tôt des modalités liées à la fréquence des points de données (on a retenu 30 minutes entre chaque), celles liées à l'actualisation du réseau avec l'arrivée de nouvelles données (et donc le choix de la fréquence à laquelle le ré-entraîner), mais il reste à discuter la principale difficulté liée au choix des monnaies. En effet, si M est le nombre de monnaies disponibles, alors le nombre de portefeuilles possibles est $2^M - 1$. Si l'on se restreint à la plate-forme Poloniex seule, qui propose un peu plus de 80 monnaies, cela représente plus de 10^{24} portefeuilles que l'utilisateur peut demander à notre produit d'optimiser, et donc tout autant de réseaux entraînés à avoir sous la main ! Face à cette difficulté, un premier choix a été de limiter le nombre de monnaies disponibles au choix sur notre plate-forme à 12, à savoir les 11 monnaies principales en terme de capitalisation de marché en plus du Bitcoin (et du dollar) : Ethereum, Ripple, Bitcoin Cash, EOS, Litecoin, Cardano, Stellar, IOTA, TRON, NEO, Monero et Dash. En plus de présenter une facilité pour le CNN, cette solution permet d'assurer que la liquidité des monnaies sélectionnées sera toujours suffisante pour que les ordres de l'utilisateur (achat ou vente) puissent être exécutés.

Mais cette solution nous rapporte tout de même à plus de 4000 configurations en considérant que Bitcoin et dollar doivent nécessairement y apparaître, et reste impossible à envisager puisque l'entraînement d'un CNN nécessite dans le meilleur cas 2 à 3h, et nécessite entre 2 et 6 Mo d'espace de stockage. La solution que nous avons décidé d'adopter dans le rendu actuel s'appuie donc sur un constat que nous avons fait sur le portefeuille ω^{CNN} conseillé par le CNN : celui-ci est bien souvent composé d'une unique monnaie, parfois de 2 à 3, mais rarement de plus. En s'appuyant sur ces caractéristiques, on procède de la sorte : quelles que soient les monnaies sélectionnées par l'utilisateur, la plate-forme interroge le CNN sur l'affectation optimale avec les 14 monnaies. Si les 1 à 3 monnaies conseillées sont présentes dans le portefeuille souhaité, on affiche ω^{CNN} tel quel. Si certaines ne le sont pas, on supprime la ou les monnaies absentes puis on normalise sur celles présentes. Si aucune n'est présente, on indique à l'utilisateur que l'algorithme ne peut trouver de solution avec sa sélection et lui en conseille une autre basée sur ω^{CNN} .

A la date limite de rendu de ce projet (25 mai 2018), il n'est pas possible pour l'utilisateur de choisir la composition de portfolio qu'il souhaite. L'optimisation se fait donc pour un portfolio composé des 14 monnaies sans possibilité de restriction. Ceci n'est dû qu'à un manque de temps et nous tenterons de l'ajouter sous peu.

4.5.2 Automatisation et modularité de l'implémentation

Comme nous l'avons vu dans la partie 3.3.6, il est nécessaire de ré-entraîner périodiquement notre CNN sous Tensorflow, une fois par semaine (la période que nous avons choisie). Nous souhaitons dans un premier temps automatiser ce processus, de manière à ce que l'entraînement se déclenche sur le serveur à heure fixée. Cependant, cette fonctionnalité n'a pas pu être implémentée par manque de temps (au moment de l'écriture de ce rapport). Elle nécessiterait en effet la rédaction de quelques scripts pour le lancement de l'entraînement du réseau et la suppression des plus anciens, sans interrompre la continuité du service. Pour l'heure, nous nous contentons d'entraîner

le CNN en local puis de transférer les fichiers du réseau Tensorflow en ligne. Nous avons en effet volontairement choisi de nous focaliser sur les fonctionnalités axées utilisateurs, comme l'amélioration du design ou la possibilité d'enregistrer des portfolios lorsque le temps nous manquait.

Le problème duquel découle ce problème est le manque de modularité de notre code. En effet, si celui-ci fonctionne parfaitement pour l'optimisation à un temps donné, il est assez peu adaptable à d'autres cadres. On peut par exemple penser aux tests sur des données passées, pour lesquels nous avons dû passer par du code de JIANG et LIANG [7].

Conclusion

L'implémentation actuelle du produit répond à la problématique initiale : permettre à des utilisateurs sans connaissance sur les cryptomonnaies ou l'optimisation de portefeuille de disposer d'un outil de conseil pour la prise de décision sur l'investissement dans les cryptomonnaies. Le développement de cette plate-forme a également été l'occasion de s'intéresser à l'applicabilité des algorithmes de trading à des marchés d'un nouveau type, celui des cryptomonnaies, et de constater à quel point les stratégies d'optimisation de portefeuilles classiques sont peu performantes sur ces derniers, alors que les taux de croissances affichés sont très largement supérieurs aux actifs boursiers sur lesquels ils sont traditionnellement appliqués. Cette étude montre donc qu'il est nécessaire de repenser les modèles et les hypothèses sur le marché sur lesquelles ces stratégies reposent, mais également que l'étude des marchés des cryptomonnaies est encore un domaine trop peu exploré par les chercheurs en finance alors qu'il présente des caractéristiques tout à fait uniques. Le succès du réseau neuronal convolutif semble montrer qu'il existe bien des patterns reconnaissables pour anticiper les mouvements de valeurs, cependant les connaissances expertes actuelles ne peuvent les déceler.

Ce projet appelle bien entendu à de nombreuses extensions et améliorations. Il serait par exemple nécessaire d'améliorer le temps de calcul des algorithmes de recommandation de portefeuilles, celui-ci atteignant souvent plus d'une dizaine de seconde, temps d'attente supérieur à ce qu'un utilisateur pourrait attendre aujourd'hui d'un site internet. Plusieurs autres fonctionnalités pourraient être intégrées à la plateforme, comme par exemple des suivis d'évolution de valeurs et d'échanges plus précis, une plus grande customisation des valeurs à suivre, ou encore la possibilité de suivi sur du long terme de l'évolution d'un portefeuille. L'extension par excellence serait celle de la possibilité de passer des transactions directement depuis la plate-forme : l'utilisateur obtiendrait le portefeuille conseillé par l'algorithme de son choix puis demanderait à la plate-forme d'effectuer le changement de portefeuille sur la plate-forme d'échange sur laquelle celui-ci est constitué. Une autre avancée pourrait être de proposer les données d'un plus grand nombre de plate-formes d'échange afin d'augmenter le nombre de monnaies disponibles. Le développement ultime pourrait être de permettre à l'utilisateur de donner le contrôle quasi-total de son portefeuille à notre outil, et que celui-ci gère sa ré-optimisation à fréquence élevée pour permettre une maximisation des profits sans contraindre l'utilisateur à revenir en permanence sur la plate-forme.

Références

- [1] Hermann ELENDRER et al. « The Cross-Section of Crypto-Currencies as Financial Assets ». In : *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1*. Sous la dir. de David LEE KUO CHUEN et Robert H. DENG. Academic Press, 2018. Chap. 7, p. 145–170.
- [2] Bin LI et Steven C.H. HOI. « Online Portfolio Selection : A Survey ». In : *ACM Computing Surveys (CSUR)* 46.3 (2014).
- [3] Harry MARKOWITZ. « Portfolio Selection ». In : *The Journal of Finance* 7.1 (1952), p. 77–91.
- [4] J. MOODY et M. SAFFELL. « Learning to Trade via Direct Reinforcement ». In : *IEEE Transactions on Neural Networks* 12.4 (2001), p. 875–889.
- [5] Gah-Yi BAN, Nouredine EL KAROUI et Andrew E.B. LIM. « Machine Learning and Portfolio Optimization ». In : *Management Science* (2016).
- [6] J.B. HEATON, Nick POLSON et Jan WITTE. « Deep Learning for Finance : Deep Portfolios ». In : *Applied Stochastic Models in Business and Industry* 33.1 (2017), p. 3–12.
- [7] Zhengyao JIANG et Jinjun LIANG. « Cryptocurrency Portfolio Management with Deep Reinforcement Learning ». In : *Intelligent Systems Conference*. 2017.
- [8] Kamil ŻBIKOWSKI. « Application of Machine Learning Algorithms for Bitcoin Automated Trading ». In : *Machine Intelligence and Big Data in Industry*. Sous la dir. de Dominik RYŻKO et al. Springer International Publishing, 2016, p. 161–168.
- [9] Bin LI et al. « Moving Average Reversion Strategy for On-line Portfolio Selection ». In : *Artificial Intelligence* 222 (2015), p. 104–123.
- [10] Thomas M. COVER. « Universal Portfolios ». In : *Mathematical Finance* 1.1 (1991), p. 1–29.
- [11] Allan BORODIN, Ran EL-YANIV et Vincent GOGAN. « Can we Learn to Beat the Best Stock ». In : *Journal of Artificial Intelligence Research* 21 (2004), p. 579–594.
- [12] Bin LI et al. « PAMR : Passive Aggressive Mean Reversion Strategy for Portfolio Selection ». In : *Machine Learning* 87.2 (2012), p. 221–258.
- [13] László GYÖRFI, Gábor LUGOSI et Frederic UDINA. « Nonparametric Kernel-based Sequential Investment Strategies ». In : *Mathematical Finance* 16.2 (2006), p. 337–357.
- [14] Allan BORODIN, Ran EL-YANIV et Vincent GOGAN. « On the Competitive Theory and Practice of Portfolio Selection ». In : *Latin American Symposium on Theoretical Informatics*. 2000, p. 173–196.
- [15] Dingjiang HUANG et al. « Robust Median Reversion Strategy for Online Portfolio Selection ». In : *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. 2013, p. 2006–2012.
- [16] Bin LI et al. « Confidence Weighted Mean Reversion Strategy for Online Portfolio Selection ». In : *ACM Transactions on Knowledge Discovery from Data* 7.1 (2013), p. 4.
- [17] David P. HELMBOLD et al. « On-line Portfolio Selection Using Multiplicative Updates ». In : *Mathematical Finance* 8.4 (1998), p. 325–347.
- [18] Li GAO et Weiguo ZHANG. « Weighted Moving Average Passive Aggressive Algorithm for Online Portfolio Selection ». In : *5th International Conference on Intelligent Human-Machine Systems and Cybernetics*. T. 1. 2013, p. 327–330.
- [19] David H. GLUSS THOMAS M. COVER. « Empirical Bayes Stock Market Portfolios ». In : *Advances in Applied Mathematics* 7.2 (1986), p. 170–181.
- [20] William F. SHARPE. « The Sharpe Ration ». In : *The Journal of Portfolio Management* 21.1 (1994), p. 49–58.
- [21] Malik MAGDON-ISMAIL et Amir F. ATIYA. « Maximum Drawdown ». In : *Risk Magazine* 17.10 (2004), p. 99–102.
- [22] Andrew W. LO, Harry MAMAYSKY et Jiang WANG. « Foundations of Technical Analysis : Computational Algorithms, Statistical Inference, and Empirical Implementation ». In : *The Journal of Finance* 55.4 (2000), p. 1705–1770.
- [23] Kurt HORNIK. « Approximation Capabilities of Multilayer Feedforward Networks ». In : *Neural Networks* 4.2 (1991), p. 251–257.