École Centrale Lyon

Projet de recherche
Rapport

Stock direction prediction analysis

Élèves :
Mohammed AKHARMOUCH
Badreddine SAADIOUI

Enseignant :
Christian De peretti

Table des matières

Résumé

Ce projet, inscrit dans le prestigieux programme de recherche de l'École Centrale de Lyon, vise à repousser les frontières de la prédiction boursière en exploitant des techniques avancées d'apprentissage automatique, notamment les modèles Long Short-Term Memory (LSTM) et Random Forest. En analysant minutieusement les performances de ces modèles sur les données de diverses entreprises, nous avons découvert que les LSTM surpassent les stratégies traditionnelles et même les approches modernes dans la plupart des scénarios, en particulier pour les entreprises caractérisées par une forte volatilité. Les résultats fascinants de cette étude ouvrent la voie à des perspectives futures ambitieuses : enrichir les modèles avec des données supplémentaires, développer des approches hybrides innovantes, et optimiser les hyperparamètres pour atteindre une précision de prédiction encore inégalée. Ce projet est une avancée majeure vers la transformation des pratiques de prédiction financière.

Summary

This project, part of the prestigious research program at École Centrale de Lyon, aims to push the boundaries of stock market prediction by leveraging advanced machine learning techniques, specifically Long Short-Term Memory (LSTM) and Random Forest models. Through meticulous analysis of these models' performance on data from various companies, we found that LSTM models consistently outperform traditional strategies and even modern approaches in most scenarios, particularly for companies with high volatility. The fascinating results of this study pave the way for ambitious future endeavors : enriching the models with additional data, developing innovative hybrid approaches, and optimizing hyperparameters to achieve unparalleled prediction accuracy. This project represents a significant leap towards transforming financial prediction practices.

Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à M. Christian de Peretti et à M. Philippe Michel pour leur accompagnement précieux, leur expertise et leurs conseils tout au long de ce projet. Leur soutien continu et leur capacité à inspirer et à motiver ont été d'une aide inestimable pour la réalisation de ce travail.

On souhaite également remercier l'École Centrale de Lyon pour nous avoir offert cette opportunité exceptionnelle de nous engager dans un projet aussi stimulant et enrichissant. Cette expérience nous a été une étape clé dans notre parcours académique et professionnel, et nous sommes très reconnaissant pour les ressources et le soutien fournis par l'école.

Un remerciement spécial s'adresse également à madame Amal BEN HAMIDA qui a partagé avec nous ses connaissances et expertise, enrichissant ainsi notre apprentissage et notre compréhension tout au long du projet.

Enfin, on voudrais exprimer notre gratitude à tous ceux qui ont contribué directement ou indirectement à la réussite de ce projet. Chaque conseil, chaque mot d'encouragement, et chaque critique constructive ont été des pierres ajoutées à l'édifice de notre développement professionnel et personnel.

1 Introduction

1.1 Contexte

La prédiction des rendements boursiers constitue un enjeu majeur pour les investisseurs, les analystes financiers et les gestionnaires de portefeuille. Cette capacité à anticiper les fluctuations du marché boursier offre des opportunités significatives d'optimisation des rendements et de minimisation des risques associés aux investissements. Dans un environnement financier en constante évolution, où les décisions doivent être prises rapidement et de manière informée, les avancées technologiques, notamment en apprentissage automatique et en traitement des données, jouent un rôle crucial.

L'impact potentiel de ces modèles est énorme. Par exemple, une gestion de portefeuille basée sur des prévisions précises pourrait améliorer significativement les rendements des investissements. Si une entreprise de gestion de fonds utilise ces modèles pour optimiser son portefeuille, elle pourrait potentiellement augmenter son rendement annuel de plusieurs points de pourcentage, ce qui pourrait se traduire par des millions de dollars de profits supplémentaires pour les investisseurs.

L'importance de ces avancées est mise en lumière par des cas pratiques comme celui de Nvidia, une société pionnière dans le domaine des processeurs graphiques et de l'intelligence artificielle. Par exemple, l'application d'un modèle prédictif précis aux actions de Nvidia aurait permis à un investisseur d'acheter des actions à 11.95 USD le 13 octobre 2022 et de les vendre à 125.20 USD le 12 juin 2024, générant ainsi un rendement de 946%, soit un profit de 113,250 USD pour un investissement initial de 10,000 USD. Ce cas illustre parfaitement le potentiel des techniques de prévision boursière pour transformer les décisions d'investissement et maximiser les profits.

Les avancées récentes en intelligence artificielle (IA) ont révolutionné les méthodes de prédiction boursière. Les réseaux de neurones récurrents (RNN) et leurs variantes comme les réseaux de neurones à mémoire à long terme (LSTM) sont utilisés pour traiter des données séquentielles et capturer les dépendances temporelles dans les données de marché. Les LSTM, en particulier, sont appréciés pour leur capacité à gérer des séries temporelles longues et complexes, ce qui est essentiel pour la prédiction des prix des actions.

En parallèle, les techniques de traitement du langage naturel (NLP) sont également employées pour analyser les sentiments des articles de presse et des réseaux sociaux, offrant des insights supplémentaires sur les mouvements potentiels du marché. Cette approche permet de capturer non seulement les tendances historiques des données financières, mais aussi les facteurs externes qui peuvent influencer les décisions des investisseurs.

Historiquement, les méthodes de prédiction boursière s'appuyaient principalement sur l'analyse fondamentale et l'analyse technique. L'analyse fondamentale se concentre sur l'évaluation de la valeur intrinsèque d'une action en examinant les états financiers, les rapports de gestion et d'autres indicateurs économiques. L'analyse technique, quant à elle, utilise des modèles statistiques et graphiques basés sur les prix historiques et les volumes de transactions pour prévoir les tendances futures.

Les techniques modernes de prévision boursière intègrent une variété d'approches mathématiques, statistiques et informatiques pour analyser des ensembles de données complexes et volumineux. Par exemple, les modèles de séries temporelles comme ARIMA (AutoRegressive Integrated Moving Average) et les modèles de volatilité stochastique (GARCH) sont utilisés pour capturer les comportements dynamiques des séries de prix des actions.

Les réseaux de neurones profonds (DNN), les forêts aléatoires, et les machines à vecteurs de support (SVM) sont également couramment utilisés pour la classification et la régression des données boursières. Ces modèles sont capables d'identifier des patterns non linéaires et des interactions complexes entre les variables, améliorant ainsi la précision des prédictions.

Les données doivent être normalisées pour réduire les biais et les variations extrêmes qui pourraient affecter la performance des modèles. Par exemple, l'utilisation de techniques de mise à l'échelle comme MinMaxScaler peut transformer les valeurs des données pour qu'elles se situent dans une plage uniforme, améliorant ainsi la convergence et la précision des modèles d'apprentissage automatique. On va détailler plus ça dans l'état d'art et présenter les avantages et les inconvénients de tous ces modèles et ces méthodes en se basant sur plusieurs articles et revues.

1.2 Objectifs du Projet

Cette recherche s'inscrit dans le cadre du programme de recherche de l'École Centrale de Lyon, spécifiquement conçu pour les étudiants internationaux accueillis au S8. Sous la supervision de Monsieur Christian de Peretti, nous nous concentrons sur un projet alliant finance et intelligence artificielle avec les objectifs suivants :

1. **Évaluer l'état de l'art** : Examiner les techniques et modèles actuels utilisés pour la prédiction des rendements boursiers. Cela inclut une analyse approfondie des méthodes existantes, des outils et des approches employées par les chercheurs et les professionnels du secteur financier.
2. **Explorer et tester de nouvelles approches** : Développer et évaluer des modèles prédictifs innovants, intégrant des algorithmes d'apprentissage automatique avancés. Nous visons à améliorer la précision des prévisions et à optimiser les rendements financiers en utilisant des données historiques et actuelles et des transformations sur les données ainsi qu'une analyse spécifiquement sectorielle sur des PME francaise.

1.3 Structure du Rapport

Ce présent rapport sera structuré de manière à présenter de façon claire et logique les différentes étapes et résultats du projet. Voici les principales sections :

1. **Introduction** : Présentation du contexte, des objectifs et du plan du rapport.
2. **État de l'Art** : Analyse des recherches et des modèles existants dans le domaine de la prédiction des rendements boursiers. Cette section inclura une revue de la littérature ainsi qu'une discussion des défis courants rencontrés dans la prédiction boursière.

3. **Méthodologie** : Description des données utilisées, des modèles appliqués et des techniques de prétraitement des données. Détails sur l'implémentation des modèles, y compris les paramètres et les configurations spécifiques.
4. **Résultats** : Présentation des résultats obtenus à partir des modèles testés, avec des analyses comparatives et des interprétations et évaluations des performances des modèles.
5. **Conclusion et perspectives futures** : Résumé des principales conclusions du projet et perspectives pour des recherches futures.

Cette recherche ambitionne de nous introduire aux frontières de la prédiction financière en utilisant les techniques les plus avancées d'apprentissage automatique, tout en fournissant des insights pratiques pour améliorer la prise de décision financière et spécialement tester et évaluer ces modèles dans des secteurs spécifiques pour des PME et comparer la performance avec des entreprises du CAC40.

État d'art

2 État de l'Art

2.1 Prédiction de Stock : Un Aperçu

L'analyse des prix des actions a évolué à travers l'utilisation de diverses méthodes prédictives, classées en trois grandes approches : techniques statistiques, techniques d'apprentissage automatique. Les techniques statistiques, notamment les modèles ARIMA, sont traditionnellement privilégiées pour les séries temporelles linéaires, tandis que les modèles de réseaux de neurones artificiels (ANN) ont tendance à mieux performer avec des données non linéaires. La littérature révèle des résultats mixtes concernant la supériorité de l'une ou l'autre approche, suggérant une adaptation en fonction des spécificités des données traitées.

Les avancées en IA ont considérablement influencé les méthodes de prédiction boursière. L'IA, en particulier les techniques d'apprentissage automatique, offre des outils puissants pour détecter des patterns complexes dans les données historiques et optimiser les prédictions. Ce processus comprend l'observation des données, la planification des solutions possibles, l'optimisation pour trouver la meilleure solution, et l'action basée sur cette solution optimale. Les études comparatives entre ARIMA et ANN montrent des avantages respectifs selon la nature des séries temporelles analysées.

2.2 Modèles Traditionnels

L'étude des modèles traditionnels pour la prédiction des prix des actions repose principalement sur l'utilisation des modèles traditionnels tels que ARMA et ARIMA remonte au milieu du siècle dernier, où ils ont été largement appliqués pour résoudre les problèmes de prédiction des séries temporelles. Dès les années 1950, ces modèles ont été adoptés pour leur capacité à modéliser efficacement les données dans un cadre temporel limité, malgré leur performance réduite sur les séries à long terme. Plus tard, en 1991, Trench W.F.[1] a amélioré le modèle ARMA avec des coefficients de pondération explicites, augmentant ainsi sa précision sur des plages de temps restreintes.

Et aussi modèles tels que les modèles ARMA (Moyenne Mobile Autorégressive) et ARIMA (Moyenne Mobile Intégrée Autorégressive) sont couramment utilisés pour analyser et prédire les des prix des actions. Ces méthodes ont prouvé leur efficacité dans la modélisation des séries temporelles financières, avec des applications notables signalées dans les travaux de chercheurs comme Abul Basher et Sadorsky en 2016 [2]. Les modèles GARCH (Hétéroscédasticité Conditionnelle Autorégressive Généralisée), qui examinent la volatilité des marchés, sont également fréquemment employés pour leur capacité à analyser les variations de variance au fil du temps.Cependant, l'utilisation du modèle GARCH traditionnel peut se heurter à plusieurs obstacles, notamment en raison des propriétés non stationnaires des données, de la persistance élevée de la variance conditionnelle, et des comportements asymétriques et non linéaires des marchés. En parallèle, l'intégration

des techniques d'apprentissage automatique dans les études récentes offre des alternatives innovantes pour surmonter les limites des approches statistiques traditionnelles, en fournissant des outils plus adaptatifs et robustes pour l'analyse financière.

2.3 Modèles d'Apprentissage Automatique

L'évolution des modèles d'apprentissage automatique pour la prédiction des marchés financiers s'est accélérée au cours de la dernière décennie, avec l'adoption croissante de techniques avancées telles que les support vecteur machines (SVM) et les Random Forest (RF). Des études telles que celles de X.Z.Li et J.M.Kong (2014) [3], et ont aussi démontré l'efficacité des SVM dans la prédiction des prix des actions, tandis que J.M.Kong en 2018 a validé l'application réussie des RF dans ce domaine. Ces méthodes offrent une meilleure généralisation par rapport aux techniques statistiques traditionnelles, bien que leurs performances dépendent fortement des variables d'entrée choisies.

L'apprentissage profond, avec des architectures comme les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN) et les réseaux LSTM (Long Short-Term Memory), a pris une place prépondérante dans la recherche contemporaine. Par exemple, Hoseinzade et Haratizadeh [7] (2019) ont utilisé des CNN pour prédire la direction quotidienne des indices boursiers, et Long (2019) [7] a employé des modèles d'apprentissage profond pour anticiper les mouvements de prix sur des intervalles très courts. Ces techniques, en exploitant des volumes massifs de données historiques, ont démontré une capacité supérieure à modéliser les dynamiques complexes et non linéaires des marchés financiers.

2.3.1 Support Vector Machines (SVM)

Les Support Vector Machines (SVM) sont des modèles robustes principalement développés pour les problèmes de classification, mais ils sont également appliqués aux problèmes de régression à travers la régression vectorielle de support (SVR), utilisable dans la prévision des prix des actions. Cette méthode est efficace pour traiter des données non linéaires, ce qui est fréquent dans les données financières où les relations linéaires simples ne suffisent pas à modéliser les dynamiques complexes du marché. La combinaison de l'analyse du spectre singulier (SSA) et des SVM par Xiao et al. [4] a proposé une approche novatrice pour l'analyse et la prévision des prix des actions, soulignant la capacité des SVM à capturer des tendances complexes dans les données de marché.

En outre, Ismail et Mohd AL [5] . ont introduit une méthode qui intégre les SVM avec d'autres techniques telles que les réseaux de neurones artificiels, et les forêts aléatoires pour renforcer l'exactitude des prédictions de direction des prix des actions. Ce genre d'approche hybride illustre comment les SVM peuvent être utilisés en combinaison avec d'autres méthodes pour améliorer la fiabilité et la précision des prévisions, ce qui est crucial dans les applications financières où les décisions doivent être prises rapidement et sur la base de prévisions précises.

2.3.2 K-Nearest Neighbors (KNN)

Le K-Nearest Neighbors (KNN) est une méthode simple mais puissante, utilisée pour prédire les prix futurs des actions en fonction des valeurs historiques les plus proches. Cette technique repose sur la proximité et la similarité des caractéristiques pour prédire les valeurs futures, ce qui en fait un outil précieux dans les contextes où les données comportementales ou les tendances passées sont fortement indicatives des performances

futures. Les améliorations apportées à l'algorithme KNN standard, telles que celles développées par Bingjie Loua , et Yixuan Dong [6]., ont permis d'augmenter la précision des prédictions en ajustant les paramètres de l'algorithme pour mieux capturer les complexités des séries temporelles financières.

2.3.3 Régression Linéaire

La régression linéaire est un modèle statistique de base souvent utilisé pour établir une relation entre une variable dépendante (le prix des actions) et une ou plusieurs variables indépendantes (par exemple, des facteurs économiques, des indicateurs de marché, etc.). En raison de sa simplicité et de sa transparence, la régression linéaire est particulièrement appréciée dans les milieux financiers pour la modélisation des relations linéaires où les entrées et les sorties sont supposées être proportionnelles. Cependant, cette méthode peut être limitée par son incapacité à traiter les relations non linéaires qui sont souvent présentes dans les données financières, ce qui peut nécessiter l'utilisation de techniques plus complexes comme les SVM ou les réseaux de neurones pour les modèles prédictifs plus précis.

2.3.4 Évaluation Comparative et Sélection des Modèles

Method	Comparisons	Dataset	Targets	Input features	Metrics	Results
ANN	SPSS statistics tool	Bombay Stock Exchange Limited	Future direction of the stock price movements	Opening price, high price, low price, and closing price	AAE, MAE, RMSE	ANNs provide higher accuracy
ANN	ANN_SCG, ANN_LM, ANN_BR	Reliance Private Limite from Thomson Reuter Eikon	Stock prices and movements	Tick Data, and 15-min Data	MAPE, MSE	ANN_SCG obtained best performance
KNN	Baseline KNN, regression prediction	Historical data of stock Neimengyiji	History	High, low, open, and close	Standard error	Improved KNN yielded the best result
EEMD–MKNN–TSPI	EEMD–MKNN, MKNN–TSPI	NAS, DJI, S&P 500, Russell 2000; and stock data from 04 regions	Opening and closing price	Opening and closing prices	MAPE, MASE, NMSE	EEMD–MKNN–TSPI model outperforms the EEMD–MKNN and MKNN–TSPI models
SSA–SVM	ANFIS, SVM, EEMD–ANFIS, EEMD–SVM, and SSA–ANFIS	Shanghai Stock Exchange Composite Index	Daily closing price	Closing price	MSE, MAPE, DS, R^2	SSA–SVM model exhibiting the best prediction performance
SVM	LR, ANN, RF	Kuala Lumpur Composite Index, Kuala Lumpur Stock Exchange Industrial, Kuala Lumpur Stock Exchange Technology	Next day movement	Stock returns, technical indicators, connected components, Holes	Average of the prediction performances	Support vector machine with persistent homology generates the best outcome
Random Forest	LR, LDA, NB, KNN, K*, C4.5, CART, ANN, SVM	Indonesia Stock Exchange	Prediction of the LQ45 index	15 variables (volum, value, …)	Accuracy, recall, precision	RF had the best performance
Random Forest	XGBoost, Bagging Classifier, AdaBoost, Extra Trees Classifier, Voting Classifier	NYSE, NASDAQ, NSE	Direction of stock price movement	40 technical indicators and the OHLCV variables	Accuracy, precision, f1-score, specificity, and AUC	Extra Trees classifier outperformed the other models

FIGURE 1 – Performances des Modèles ML dans la Prédiction des Stocks [10]

Reference No.	Method	Comparisons	Dataset	Targets	Input features	Metrics	Results
50	CNN	LR, CNN-Rand, CNN-Corr, LR With FS	BIST 100 Index	Hourly stock price direction	25 technical indicators with different time lags	Macro-Averaged F-Measure	CNN-Corr classifier yielded the best performance
51	CNN + frequent patterns	ARIMA, Wavelet + ARIMA, HMM, LSTM, SFM	S&P 500 and 07 individual stocks	Trend of stock price	Closed value	Accuracy, recall, precision, f1-score	Proposed method outperformed the others with a 4%–7% accuracy improvement
55	LSTM	Random Forest	S&P 500	Directional movements of stock price	Adjusted closing prices and opening prices	Various metric (mean, std error, sharpe ratio, ...)	LSTM outperforms random forests
56	LSTM	LASSO-LSTM, PCA-LSTM, LASSO-GRU, PCA-GRU	Shanghai Composite Index	Stock price trend	Open, high, low, trading volume, and other technical indicators	RMSE, MAE	LSTM and GRU with LASSO yielded better accuracy than models with PCA
62	BiLSTM	WAE-BLSTM, W-BLSTM, W-LSTM, BLSTM, LSTM	S&P500	Next day closing price	Open, high, low, close (OHLC), 08 technical indicators	MAE, RMSE, R^2	WAE-BLSTM model outperformed the other models. MAE (0.0211), RMSE (0.0272), and R^3 (0.8934)
64	AE-BiLSTM-ECA	CNN, LSTM, BiLSTM, CNN-LSTM, AE-LSTM, CNN-BiLSTM, AE-BiLSTM, BiLSTM-ECA,	Shanghai Stock Composite Index (SSCI) and CSI 300	Closing price	Seven characteristics such as closing, high, open, low, previous day's closing price, up or down amount and and	MSE, RMSE, MAE, MAPE	AE-BiLSTM-ECA obtain the best accuracy. CSI 300 stock data: MSE: 3158.452 RMSE: 56.200 MAE: 36.681

FIGURE 2 – Analyse détaillée des performances des modèles LSTM, BiLSTM et leurs variantes en comparaison avec RF [10]

Dans l'analyse des modèles de prévision des prix des actions, les approches de machine learning et de deep learning telles que Random Forest et LSTM se distinguent par leurs performances supérieures. Les méthodes démontrent que les deux modèles sont parmi les plus efficaces, avec LSTM excellant dans la gestion des dépendances temporelles complexes et Random Forest performant robustement avec de grands ensembles de données. En raison de leur efficacité avérée, notre étude continuera de se concentrer sur ces deux modèles, en les intégrant avec d'autres aspects et variables, pour améliorer davantage la précision et la fiabilité des prédictions dans divers secteurs économiques.

2.4 Erreurs courantes dans la prédiction des stocks

Dans cette section, nous allons examiner les erreurs les plus courantes observées dans les projets de prédiction de stock. Ces erreurs, bien que fréquemment rencontrées, peuvent être évitées en adoptant des approches appropriées. L'objectif est de fournir une vue d'ensemble des défis rencontrés et des solutions possibles pour améliorer la performance des systèmes de prédiction.

2.4.1 Jeux de données insuffisants

L'une des erreurs les plus fréquentes dans la modélisation est l'utilisation de jeux de données insuffisants. Les stratégies de trading sont des systèmes complexes nécessitant un cycle de prédiction, d'évaluation, de rétroaction et de recalibration. Pour évaluer correctement la performance d'un modèle, il est crucial d'utiliser un ensemble de données totalement inédit. Malheureusement, de nombreux systèmes de trading sont développés en utilisant uniquement un ensemble de données d'entraînement et un ensemble de données de vérification, ce qui est insuffisant. Un ensemble de validation totalement indépendant est nécessaire pour vérifier si le système généralise vraiment bien aux nouvelles données.

Sans cela, le système risque d'être sur-ajusté aux données spécifiques de l'entraînement, rendant les résultats inapplicables dans un contexte réel.

2.4.2 Échelle inappropriée

Une autre erreur courante est l'utilisation d'une échelle inappropriée pour les valeurs cibles prédictives. Représenter les valeurs cibles avec leurs valeurs réelles peut sembler donner une vue précise des valeurs cibles, mais cela peut introduire des erreurs importantes dans la prédiction. Par exemple, une prédiction non mise à l'échelle peut donner l'impression d'une faible erreur alors qu'en réalité, l'erreur est significative et rend le système de prédiction inutile pour le trading. Pour éviter cela, il est recommandé de pré-traiter et de mettre à l'échelle les données, en divisant les valeurs cibles par une valeur maximale historique appropriée, afin de garantir que les données cibles se situent dans une plage appropriée on utilisera alors une méthode Min Max Scaling.

2.4.3 Suivi des séries temporelles

Une erreur fréquente dans l'analyse des séries temporelles est de produire des résultats trop optimistes qui semblent trop bons pour être vrais. Ces résultats proviennent souvent de systèmes qui ne font que prédire le prix du jour précédent, satisfaisant ainsi la fonction de minimisation de l'erreur sans fournir de véritables prédictions de mouvement des prix. Pour éviter cette erreur, il est crucial de concevoir des paires d'entrée-sortie qui ne se limitent pas aux prix eux-mêmes, mais incluent des informations permettant de prédire les mouvements des prix.

2.4.4 Mesures de performance inappropriées

Le problème ici réside dans la confiance excessive accordée aux mesures de performance classiques pour valider le succès d'un système de trading. Des mesures telles que les courbes ROC, les graphiques RMS et autres mesures de performance typiques peuvent masquer des problèmes dans la conception du système. Pour éviter cette erreur, il est recommandé de mettre en place un simulateur de trading et d'utiliser le prédicteur conçu pour simuler des transactions basées sur ses prédictions. En effectuant des transactions réelles basées sur les prédictions, les erreurs deviennent rapidement apparentes, ce qui permet d'évaluer plus précisément la performance du système et à la fin on calculera le ratio de sharp pour évaluer notre modèle.

2.4.5 Prédiction de la direction vs. prédiction de la valeur

Un autre problème majeur est que la majorité des articles et des projets se concentrent sur la prédiction de la valeur des actions. Cependant, dans le contexte du trading, prédire la valeur exacte d'une action n'est pas aussi utile que de prédire la direction ou le rendement. Les décisions de trading dépendent de la capacité à prédire si le prix va augmenter ou diminuer. Par conséquent, il est plus pertinent de développer des modèles qui prédisent la direction des mouvements des prix plutôt que leur valeur exacte. Un modèle qui prédit correctement la direction peut générer des profits, même si les prédictions de valeur ne sont pas parfaitement précises.

Cette section a examiné plusieurs erreurs courantes dans la littérature sur la prédiction des marchés. Les erreurs identifiées comprennent l'utilisation de jeux de données insuffisants, une échelle inappropriée, un suivi des séries temporelles incorrect, des mesures de performance inappropriées, et la focalisation sur la prédiction de la valeur au lieu de la direction. En adoptant des approches alternatives et en évitant ces erreurs, On essaiera plusieurs modèles pour améliorer la fiabilité et l'applicabilité de nos prédictions et ainsi générer du bénéfice.

Méthodologie

3 Méthodologie

3.1 Présentation des Données

Nous avons utilisé les données historiques fournies par Yahoo Finance. Ces données incluent plusieurs variables cruciales pour la modélisation prédictive et sont disponibles à une fréquence horaire. L'utilisation de données de haute fréquence permet de capturer les fluctuations intra-journalières du marché, offrant ainsi une granularité et une richesse d'informations essentielles pour des prédictions précises. Dans cette section, nous allons décrire en détail les données utilisées, justifier notre choix de source de données et expliquer comment ces données sont préparées pour être utilisées dans notre modèle de prédiction.

Les données historiques récupérées de Yahoo Finance incluent les variables suivantes :

- **Opening Price** (Prix d'ouverture) : Le prix auquel une action s'échange pour la première fois au début de la séance de marché.

- **Closing Price** (Prix de clôture) : Le prix final auquel une action s'échange à la fin de la séance de marché.

- **Highest Price** (Prix le plus élevé) : Le prix maximum atteint par une action au cours de la séance de marché.

- **Lowest Price** (Prix le plus bas) : Le prix minimum atteint par une action au cours de la séance de marché.

- **Volume** (Volume des transactions) : Le nombre total d'actions échangées pendant une séance de marché.

Datetime	Open	High	Low	Close	Adj Close	Volume
2022-06-17 09:30:00-04:00	102.800003	106.180000	102.510002	105.230003	105.230003	24579527
2022-06-17 10:30:00-04:00	105.260002	105.653603	104.019997	105.620003	105.620003	9370452
2022-06-17 11:30:00-04:00	105.629997	105.860001	104.820000	105.699997	105.699997	6084924
2022-06-17 12:30:00-04:00	105.725098	106.980003	105.459999	106.429901	106.429901	8309250

FIGURE 3 – Format données yfinance

Ces variables sont essentielles pour analyser le comportement des actions et pour construire des modèles prédictifs robustes. En particulier, les prix d'ouverture et de clôture fournissent des informations sur le sentiment général du marché et la dynamique quotidienne des prix. Les prix les plus élevés et les plus bas offrent des indications sur la volatilité intra-journalière, tandis que le volume des transactions peut signaler l'intérêt et l'activité des investisseurs.

3.1.1 Justification du Choix de Yahoo Finance

- **Fiabilité et Accessibilité des Données :** Yahoo Finance est une source reconnue et fiable de données financières. Elle offre un accès gratuit et facile à une vaste gamme de données historiques pour un grand nombre d'actifs financiers.

- **Fréquence et Granularité des Données :**La disponibilité des données à une fréquence horaire sur Yahoo Finance est un avantage majeur. Cette granularité permet de capturer les mouvements subtils et rapides du marché qui peuvent ne pas être visibles dans les données quotidiennes.

- **Facilité d'Intégration :** Yahoo Finance offre API facile à appeler avec un script python en une seule ligne de code.

3.1.2 Préparation des Données :

Avant d'utiliser les données pour l'entraînement de notre modèle de prédiction, nous avons effectué plusieurs étapes de préparation des données. Ces étapes incluent la collecte, le nettoyage, la transformation et la normalisation des données.

- **Collecte des Données :** Les données ont été collectées en utilisant l'API de Yahoo Finance.

- **Nettoyage des Données :** Le nettoyage des données est une étape cruciale pour éliminer les anomalies et les valeurs manquantes. Les données manquantes peuvent être imputées ou supprimées selon leur importance et l'impact potentiel sur les modèles. Par exemple, les valeurs manquantes pour les prix peuvent être imputées en utilisant des méthodes telles que l'interpolation linéaire.

3.1.3 Transformation des Données :

Pour améliorer la pertinence des données pour la modélisation, plusieurs transformations ont été appliquées. Ces transformations incluent l'ajout d'indicateurs techniques clés :

— **Moyennes Mobiles Exponentielles (EMA)** : Les EMAs sont des moyennes pondérées qui donnent plus de poids aux prix récents. Elles sont calculées en utilisant la formule suivante :

$$EMA_t = \alpha \times Price_t + (1 - \alpha) \times EMA_{t-1}$$

où $Price_t$ est le prix à l'instant t, EMA_{t-1} est la valeur EMA à l'instant précédent, et α est le coefficient de lissage (habituellement $2/(N+1)$ pour une période N).

Par exemple, pour une EMA sur 20 périodes :

$$EMA_{20,t} = \alpha \times Price_t + (1 - \alpha) \times EMA_{20,t-1}$$

— **Indice de Force Relative (RSI)** : Le RSI est un oscillateur qui mesure la vitesse et le changement des mouvements de prix. Il est calculé comme suit :

$$RSI = 100 - \frac{100}{1 + RS}$$

où RS est le rapport moyen des gains sur les pertes sur une période spécifiée (généralement 14 jours).

— **Rendement** : Le rendement représente la variation du prix d'une action sur une période donnée. Il est calculé comme la différence entre le prix de clôture ajusté (*Adjusted Close*) et le prix d'ouverture (*Open*).

$$Rendement = AdjClose_t - Open_t$$

où $AdjClose_t$ est le prix de clôture ajusté à l'instant t et $Open_t$ est le prix d'ouverture à l'instant t.

— **Direction** : La variable Direction indique la direction du rendement de l'action, ce qui est crucial pour la partie où on va développer un bot stratégique de trading. Elle est définie comme :

$$Direction_t = \begin{cases} 1 & \text{si } Rendement_t > 0 \\ 0 & \text{si } Rendement_t \leq 0 \end{cases}$$

où $Rendement_t$ est le rendement à l'instant t. Cette variable permet à l'algorithme de prendre des décisions de trading basées sur la direction anticipée des mouvements de prix.

Ces indicateurs fournissent des perspectives supplémentaires sur le comportement des prix et sont essentiels pour la construction de modèles prédictifs précis et robustes.

3.1.4 Normalisation des Données

La normalisation des données est une étape cruciale pour garantir la performance optimale des modèles d'apprentissage automatique, en particulier des réseaux de neurones récurrents tels que les LSTM (Long Short-Term Memory). Les valeurs des actions peuvent varier considérablement et ne sont pas limitées à une plage fixe. Cela peut entraîner des problèmes de stabilité numérique et de convergence lors de l'entraînement des modèles. La normalisation permet de résoudre ces problèmes en mettant toutes les variables sur une échelle commune.

Les deux méthodes couramment utilisées pour normaliser les données de stock sont :

— **Min-Max Scaling** : Cette méthode transforme les données pour qu'elles soient dans un intervalle spécifique, généralement entre 0 et 1. La formule est :

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

où x est la valeur originale, $x_{\min}$ et $x_{\max}$ sont respectivement les valeurs minimale et maximale des données. Cette méthode est utile lorsque les données n'ont pas de valeurs extrêmes ou outliers significatifs, car ces valeurs peuvent fausser la transformation.

— **Standardization (Z-score Normalization)** : Cette méthode transforme les données pour qu'elles aient une moyenne de 0 et un écart-type de 1. La formule est :

$$z = \frac{x - \mu}{\sigma}$$

où x est la valeur originale, μ est la moyenne des données, et σ est l'écart-type. La standardization est particulièrement utile lorsque les données contiennent des valeurs extrêmes ou outliers, car elle rend chaque caractéristique comparable en termes de variations standard.

Ainsi, nous avons choisi d'appliquer la Standardization (Z-score Normalization) pour normaliser nos données de stock. Cette méthode assure une meilleure robustesse face aux valeurs aberrantes et facilite la comparaison entre les différentes caractéristiques des données.

3.2 Long Short-Term Memory (LSTM)

Dans le cadre de notre projet, il est crucial de comprendre les architectures de réseaux de neurones avancées utilisées pour modéliser les données temporelles. Parmi ces architectures, les Réseaux de Neurones Récurrents (RNN) et leurs variantes, les Long Short-Term Memory (LSTM), jouent un rôle essentiel. Cette section détaille les concepts des RNN, les problèmes de gradient associés, les solutions proposées, et se concentre en profondeur sur les LSTM et leurs variantes.

3.2.1 Réseaux de Neurones Récurrents (RNN)

Les Réseaux de Neurones Récurrents (RNN) sont une classe de réseaux de neurones spécialement conçus pour traiter les données séquentielles. Contrairement aux réseaux de neurones traditionnels qui supposent que toutes les entrées et les sorties sont indépendantes les unes des autres, les RNN prennent en compte les dépendances temporelles en utilisant des connexions récurrentes. Cela signifie que les RNN possèdent une "mémoire" interne qui leur permet de capturer les informations sur les états précédents et de les utiliser pour les états futurs.

Les RNN sont particulièrement adaptés aux tâches telles que la prédiction des séries temporelles, le traitement du langage naturel, et l'analyse des séquences vidéo. En fait, il existe plusieurs types de réseaux de neurones récurrents (RNN) :

— **One-to-Many (Un-à-Plusieurs)** : Description informatique d'une image. Un réseau de neurones convolutifs (CNN) est utilisé pour classer les images, puis un RNN est employé pour interpréter les images et générer du contexte.

— **Many-to-One (Plusieurs-à-Un)** : Analyse de sentiment d'un texte (évaluer la positivité ou la négativité du texte).

— **Many-to-Many (Plusieurs-à-Plusieurs)** : Traduction automatique d'une langue dont le vocabulaire change en fonction du genre du sujet. Aussi, le sous-titrage d'un film.

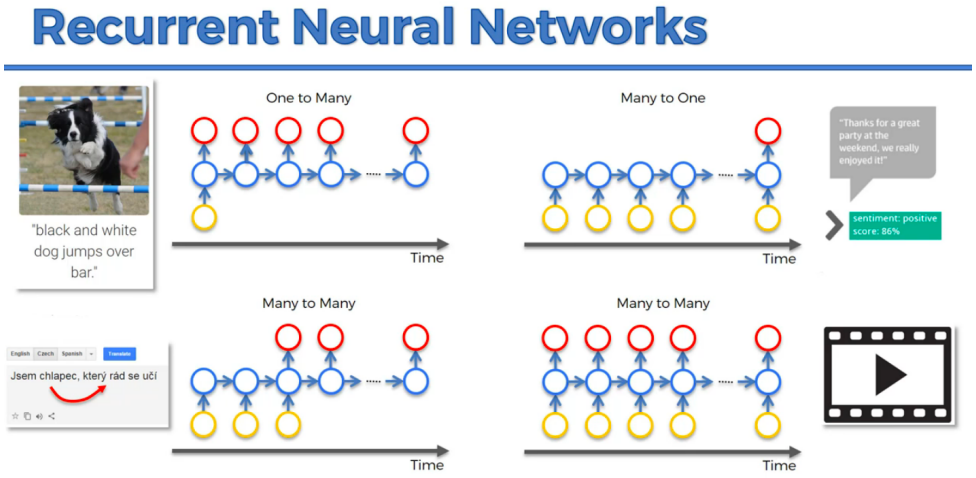

FIGURE 4 – Différents types de réseaux de neurones récurrents (RNN) [8]

Dans le contexte de la prédiction des prix des actions, les RNN peuvent capturer les tendances et les motifs dans les données historiques pour faire des prédictions informées sur les mouvements futurs du marché.

3.2.2 Problème de Gradient des RNN (Expanding ou Vanishing)

L'un des principaux défis des RNN est le problème de gradient, qui peut se manifester sous forme de gradient vanishing (décroissant) ou gradient exploding (explosif). Le gradient est une mesure de la sensibilité de la fonction de perte par rapport aux poids du réseau, utilisé pour mettre à jour les poids pendant l'entraînement via la rétropropagation.

— **Gradient Vanishing** : Lorsque le gradient devient extrêmement petit, les poids des couches précédentes du réseau sont mis à jour très lentement, ce qui empêche le réseau d'apprendre les dépendances à long terme. Cela signifie que les informations importantes des premiers états temporels sont perdues.

— **Gradient Exploding** : À l'inverse, lorsque le gradient devient très grand, il peut entraîner des mises à jour instables et des oscillations des poids, rendant l'entraînement du réseau difficile et imprévisible.

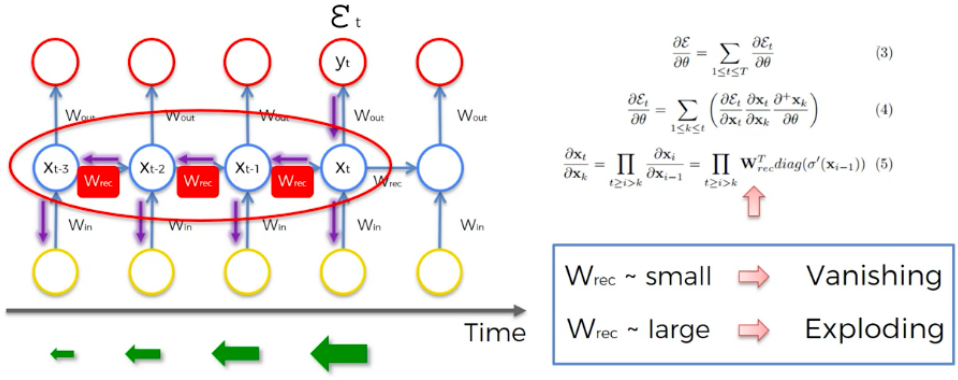

FIGURE 5 – Problème de gradient RNN Expanding et Vanishing [8]

Lorsque le poids du gradient d'un RNN, W_{rec}, est inférieur à 1, nous obtenons un gradient qui disparaît. Lorsque W_{rec} est supérieur à 1, nous obtenons un gradient qui explose. Par conséquent, nous pouvons fixer $W_{\text{rec}} = 1$.

3.2.3 Solutions au Problème du Gradient Expanding

Pour atténuer le problème du gradient exploding, plusieurs techniques peuvent être appliquées :

1. **Truncated Back-propagation** : Cette méthode arrête la rétropropagation après un certain nombre de pas temporels, réduisant ainsi le risque de gradients explosifs. Cependant, cela peut également limiter la capacité du réseau à apprendre des dépendances à long terme.
2. **Penalties** : Les pénalités appliquées aux gradients peuvent les réduire artificiellement, empêchant ainsi leur explosion.
3. **Gradient Clipping** : Cette technique impose une limite maximale aux valeurs du gradient, empêchant leur augmentation excessive. Le gradient est "clippé" à une valeur seuil pour maintenir la stabilité des mises à jour des poids.

3.2.4 Solutions au Problème du Gradient Vanishing

Pour résoudre le problème du gradient vanishing, les approches suivantes sont souvent utilisées :

1. **Weight Initialization** : Une initialisation judicieuse des poids peut minimiser le problème de gradient vanishing. Des techniques comme l'initialisation de Xavier ou He peuvent aider à maintenir des gradients dans des plages appropriées.
2. **Echo State Network** : Conçu pour résoudre le problème du gradient vanishing, ce type de réseau de neurones récurrents utilise une couche cachée connectée de manière éparse, avec des poids fixés et assignés de manière aléatoire.
3. **Long Short-Term Memory Networks (LSTM)** : Les LSTM sont conçus spécifiquement pour traiter les problèmes de dépendance à long terme en utilisant une architecture modifiée avec des cellules de mémoire et des portes de régulation.

3.2.5 Long Short-Term Memory Networks (LSTM)

Les LSTM sont une variante des RNN conçue pour résoudre les problèmes de gradient vanishing et exploding. Ils introduisent des mécanismes de mémoire et de régulation pour permettre au réseau de retenir des informations sur de longues séquences temporelles. Une cellule LSTM est composée de plusieurs composants clés.

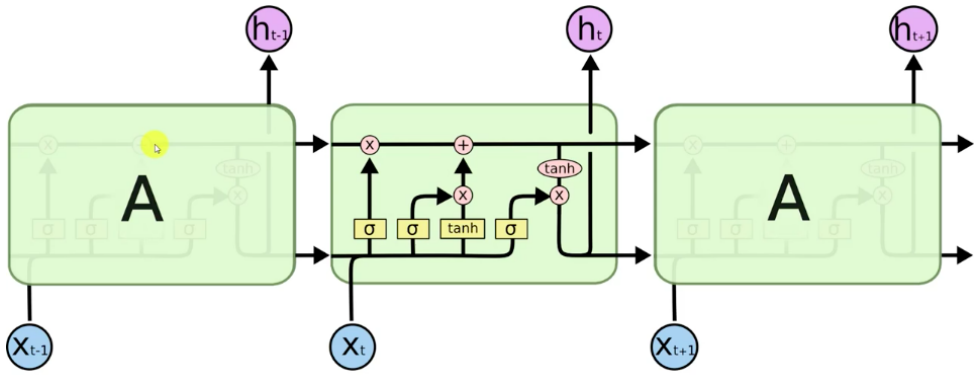

FIGURE 6 – Structure LSTM [8]

— Les cercles représentent des couches (vecteurs).
— 'A' représente les couches des cellules de mémoire.
— 'h' représente les couches de sortie (états cachés).
— 'X' représente les couches d'entrée.
— Les lignes représentent les valeurs transférées.
— Les lignes concaténées représentent des pipelines fonctionnant en parallèle.
— Les fourches sont des moments où les données sont copiées.
— L'opération point par point (X) représente des valves (de gauche à droite : valve d'oubli, valve de mémoire, valve de sortie).
— Les valves peuvent être ouvertes, fermées ou partiellement ouvertes selon une fonction d'activation.
— L'opération point par point (+) représente un raccord en T, permettant le passage si la valve correspondante est activée.
— L'opération point par point (Tanh) représente une fonction tangente qui produit des valeurs comprises entre -1 et 1.
— L'opération de la couche Sigma représente une fonction d'activation sigmoïde (valeurs de 0 à 1).

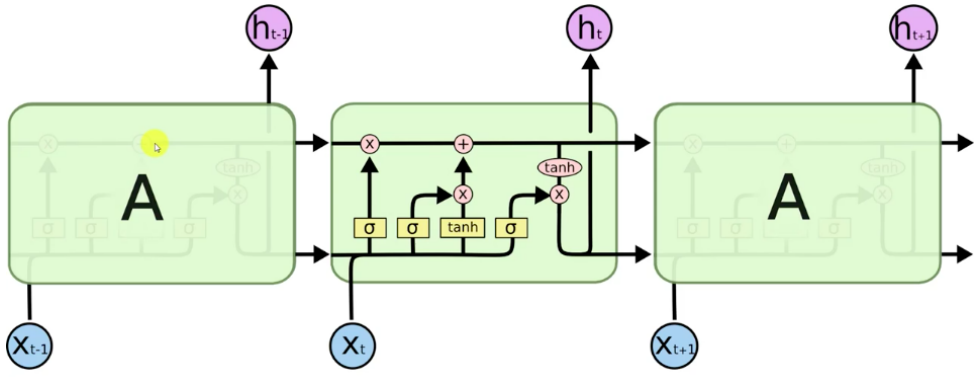

FIGURE 7 – Structure LSTM [8]

3.2.6 Étape 1 : Décision de la Porte d'Oubli (Forget Gate)

La valeur d'entrée X_t et la valeur de l'état caché précédent h_{t-1} déterminent si la porte d'oubli doit être ouverte ou fermée, en utilisant une fonction sigmoïde.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

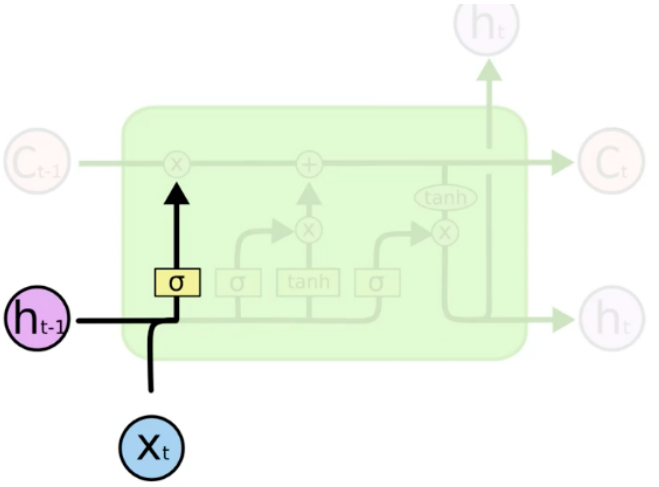

FIGURE 8 – LSTM Step 1 [8]

3.2.7 Étape 2 : Décision de la Porte d'Entrée (Input Gate)

La nouvelle valeur X_t et la valeur de l'état caché précédent h_{t-1} déterminent si la porte d'entrée doit être ouverte ou fermée, ainsi que l'étendue des valeurs à laisser passer (tanh de -1 à 1).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

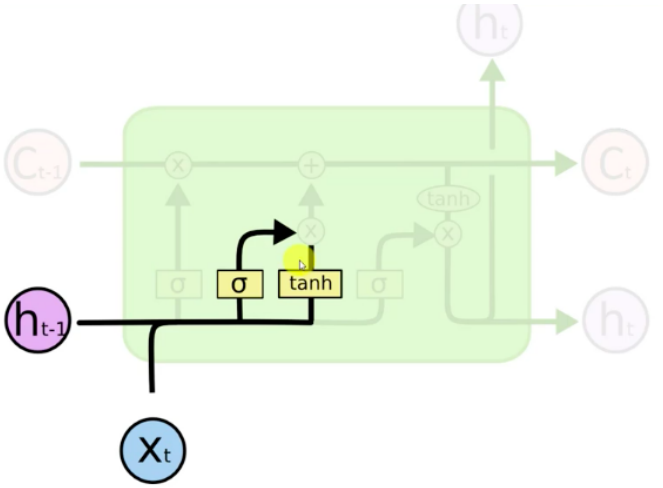

FIGURE 9 – LSTM Step 2 [8]

3.2.8 Étape 3 : Mise à Jour de la Mémoire de la Cellule (Cell State Update)

Décider de l'étendue de la mise à jour de la cellule de mémoire C_t à partir de la cellule de mémoire précédente C_{t-1}. Les portes d'oubli et de mémoire sont utilisées pour décider cela.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

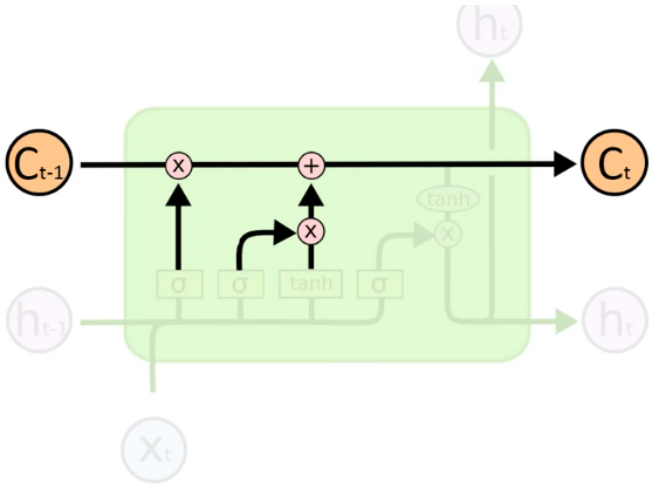

FIGURE 10 – LSTM Step 3 [8]

3.2.9 Étape 4 : Décision de la Porte de Sortie (Output Gate)

La nouvelle valeur X_t et la valeur de l'état caché précédent h_{t-1} déterminent quelle partie du pipeline de mémoire sera utilisée comme sortie h_t.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$

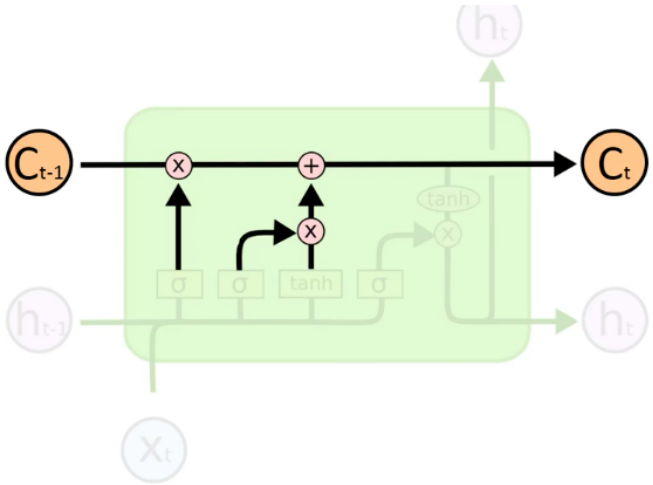

FIGURE 11 – LSTM Step 4 [8]

3.3 Random Forest

Dans notre projet, en plus d'utiliser les réseaux de neurones LSTM, nous avons également implémenté l'algorithme Random Forest pour comparer les performances et tester la robustesse des prédictions dans divers secteurs.

3.3.1 Qu'est-ce que Random Forest ?

Le Random Forest est un algorithme d'apprentissage automatique couramment utilisé, inventé par Leo Breiman et Adele Cutler, qui combine les résultats de multiples arbres de décision pour obtenir une prédiction finale. Sa simplicité d'utilisation et sa flexibilité expliquent son adoption large, car il peut traiter à la fois des problèmes de classification et de régression.

3.3.2 Arbres de Décision

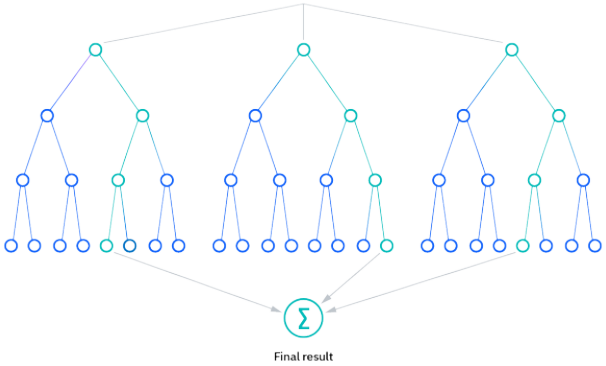

FIGURE 12 – Random forest : Arbes de décision [9]

L'algorithme Random Forest est constitué de multiples arbres de décision. Un arbre de décision commence par une question simple et se ramifie en fonction des réponses à des questions successives, formant ainsi des nœuds de décision qui divisent les données. Chaque question aide à arriver à une décision finale représentée par un nœud feuille. Les arbres de décision sont entraînés à l'aide de l'algorithme CART (Classification and Regression Tree) et utilisent des métriques telles que l'impureté de Gini, le gain d'information ou l'erreur quadratique moyenne (MSE) pour évaluer la qualité des divisions.

Cependant, les arbres de décision sont souvent sujets à des problèmes de biais et de surapprentissage. Le modèle Random Forest atténue ces problèmes en formant un ensemble d'arbres de décision non corrélés, produisant ainsi des résultats plus précis.

3.3.3 Méthodes d'Ensemble

Les méthodes d'ensemble, comme le Random Forest, combinent plusieurs classificateurs pour obtenir une prédiction agrégée. Les deux méthodes d'ensemble les plus connues sont le bagging (bootstrap aggregation) et le boosting. Dans le bagging, un échantillon aléatoire de données est sélectionné avec remplacement pour former des ensembles de données de formation multiples. Chaque modèle est alors entraîné indépendamment, et leurs prédictions sont moyennées (pour la régression) ou agrégées par un vote majoritaire (pour la classification).

3.3.4 Algorithme Random Forest

L'algorithme Random Forest est une extension de la méthode de bagging, utilisant à la fois le bagging et la randomisation des caractéristiques pour créer une forêt d'arbres de décision non corrélés. La randomisation des caractéristiques (ou feature bagging) génère un sous-ensemble aléatoire de caractéristiques, garantissant ainsi une faible corrélation entre les arbres de décision.

Lors de la construction de la forêt, chaque arbre est formé à partir d'un échantillon bootstrap de l'ensemble de données d'entraînement, et un tiers de cet échantillon est mis de côté comme données de test (out-of-bag, oob). Ensuite, une nouvelle instance de randomisation est introduite via le feature bagging. Pour les tâches de régression, les prédictions individuelles des arbres de décision sont moyennées, tandis que pour les tâches de classification, un vote majoritaire est utilisé pour déterminer la classe prédite. Les échantillons oob sont utilisés pour la validation croisée, finalisant ainsi la prédiction.

3.3.5 Avantages et Défis du Random Forest

Avantages

- **Réduction du risque de surapprentissage** : Les arbres de décision ont tendance à s'ajuster étroitement aux échantillons de formation, mais l'ensemble des arbres dans une forêt aléatoire, en moyennant les prédictions de multiples arbres non corrélés, réduit la variance globale et l'erreur de prédiction.
- **Flexibilité** : Le Random Forest peut gérer à la fois des tâches de régression et de classification avec une grande précision. La randomisation des caractéristiques permet également d'estimer les valeurs manquantes, en maintenant l'exactitude même avec des données partielles.
- **Facilité de détermination de l'importance des caractéristiques** : Il est simple d'évaluer l'importance des variables dans un modèle Random Forest, généralement mesurée par l'importance Gini ou la réduction moyenne de l'impureté.

Défis

- **Processus long** : Le traitement de grands ensembles de données par des forêts aléatoires peut être lent, car chaque arbre doit traiter les données de manière individuelle.
- **Besoin de ressources importantes** : Le stockage et le traitement de grandes quantités de données nécessitent des ressources importantes.
- **Complexité accrue** : Comparé à un arbre de décision unique, une forêt d'arbres est plus complexe et peut être plus difficile à interpréter.

3.4 Implémentation du Random Forest

Le diagramme ci-dessous illustre le flux de travail de l'implémentation de notre modèle de prédiction des rendements des actions et du bot de trading :

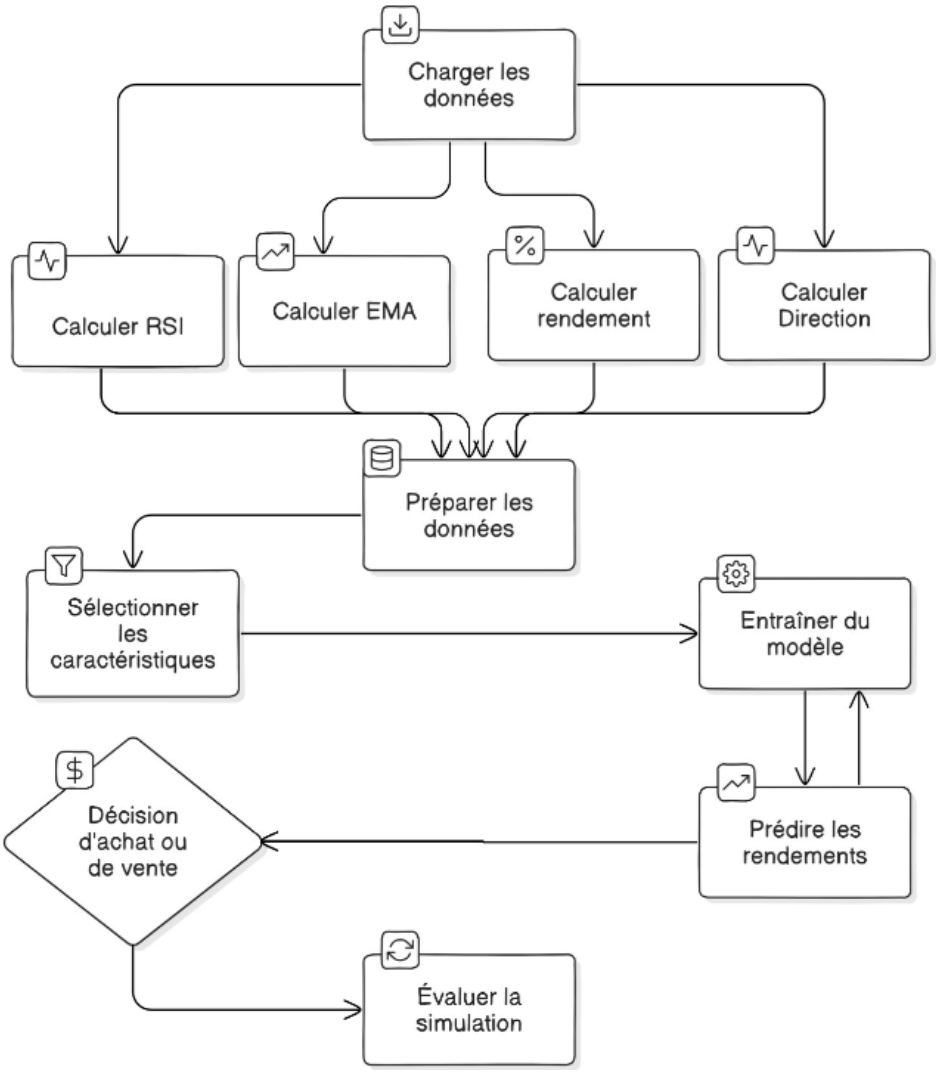

FIGURE 13 – Processus global de l'implémentation du modèle et du bot de trading

1. **Charger les Données** : Les données historiques des actions sont téléchargées de yfinance.
2. **Calcul des Indicateurs Techniques** : Les indicateurs tels que le RSI et les moyennes mobiles exponentielles (EMA) sont calculés.
3. **Préparation des Données** : Les données sont nettoyées et transformées.
4. **Sélection des Caractéristiques** : Les caractéristiques pertinentes sont sélectionnées pour le modèle.
5. **Entraînement du Modèle** : Le modèle Random Forest est entraîné.
6. **Prédiction des Rendements** : Le modèle prédit les rendements futurs.
7. **Décision d'Achat ou de Vente** : Les décisions de trading sont prises en fonction des prédictions.
5. **Ré-entraînement du Modèle** : Le modèle Random Forest est ré-entraîné après chaque décision pour prédire la prochaine heure.

8. **Évaluer la Simulation** : La performance de la stratégie de trading est évaluée via le backtesting.

Pour le Random Forest on a utilisé deux approches premières c'est l'utilisation du rendements des actions. Voici les étapes détaillées de l'implémentation :

3.4.1 Sélection des Caractéristiques et Préparation des Données

Nous avons sélectionné les caractéristiques pertinentes pour notre modèle : `Open`, `High`, `Low`, `Adj Close`, `Volume`, `RSI`, `EMA20`, `EMA50`, `EMA200`, et `Volatility`. Les données ont été divisées en ensembles d'entraînement et de validation pour évaluer les performances du modèle.

3.4.2 Entraînement du Modèle

Nous avons utilisé l'algorithme `RandomForestRegressor` de `sklearn` avec les hyper-paramètres suivants :

- Nombre d'arbres : 100
- Profondeur maximale : 10
- Critère de division : Mean Squared Error (MSE)
- Nombre minimum d'échantillons pour diviser un nœud interne : 10
- Nombre minimum d'échantillons dans un nœud feuille : 5

Le modèle a été entraîné sur un sous-ensemble des données d'entraînement pour éviter le surapprentissage.

```
model = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42,
min_samples_split=10, min_samples_leaf=5)
model.fit(X_train_split, y_train_split)
```

3.4.3 Développement du Bot de Trading pour évaluation de performance

Nous avons développé un bot de trading pour exécuter des transactions basées sur les prédictions du modèle Random Forest. Voici les étapes détaillées :

Logique du Bot de Trading : Le bot utilise les prédictions de rendement pour décider d'acheter ou de vendre des actions. La stratégie d'achat consiste à investir lorsque le rendement prédit est positif et la volatilité est faible. Inversement, la stratégie de vente est déclenchée lorsque le rendement prédit est négatif.

Backtesting et Évaluation : On commence la simulation du bot avec un capital initial de 10 000 $ et ajuste les positions en fonction des prédictions du modèle. On mis à jour le capital à chaque itération qui représente une heure en fonction des transactions effectuées et on ré-entraine le modèle à nouveau pour reprédire la prochaine heure. On a aussi calculé le ratio de Sharpe pour évaluer la performance ajustée au risque de la stratégie.

```
capital_initial = 10000

for i in range(len(validation_data)):
    current_data = validation_data.iloc[i:i+1]
    X_current = current_data[features]
    predicted_return = model.predict(X_current)[0]
    current_volatility = current_data['Volatility'].values[0]

    if predicted_return > 0 and capital > 0:
        montant_investir = capital * (predicted_return / 100) / (current_volatility + 1)
        nb_actions_acheter = (montant_investir / current_data['Adj Close'].values[0]) * 0.2
        capital -= nb_actions_acheter * current_data['Adj Close'].values[0]
        position += nb_actions_acheter
    elif predicted_return <= 0 and position > 0:
        montant_vendre = position * (predicted_return / 100) * (current_volatility + 1)
        nb_actions_vendre = (montant_vendre / current_data['Adj Close'].values[0]) * 0.2
        capital += nb_actions_vendre * current_data['Adj Close'].values[0]
        position -= nb_actions_vendre

    total_value = capital + (position * current_data['Adj Close'].values[0])
    historique_capital.append(total_value)

    X_train_split = pd.concat([X_train_split, X_current])
    y_train_split = pd.concat([y_train_split, pd.Series([predicted_return], index=[current_data.index[0]])])
    model.fit(X_train_split, y_train_split)

dernier_adj_close = validation_data['Adj Close'].iloc[-1]
if position > 0:
    capital += position * dernier_adj_close
historique_capital.append(capital)
```

Listing 1 – Code du Bot de Trading pour Random Forest

3.4.4 Utilisation de la Direction des Prix

En plus de l'approche basée sur les rendements, nous avons également implémenté une méthode utilisant la direction des prix, c'est-à-dire la prédiction de la tendance à la hausse ou à la baisse des prix.

Pour cela, nous avons transformé la variable cible en une variable binaire indiquant si le prix de clôture à l'heure suivante est supérieur au prix de clôture actuel (1 pour une hausse, 0 pour une baisse). Nous avons utilisé l'algorithme RandomForestClassifier de sklearn pour entraîner le modèle avec les mêmes caractéristiques que pour le modèle basé sur les rendements.

Le bot de trading utilise les prédictions de direction pour prendre des décisions d'achat et de vente. La logique du bot est similaire à celle utilisée pour les rendements, mais les décisions sont basées sur la direction prédite des prix.

Dans la section des résultats et discussions, nous verrons que la prédiction de la direction des prix s'est avérée être aussi performante, voire plus performante pour certaines entreprises, par rapport à l'approche basée sur les rendements. Cela démontre la flexibilité et l'efficacité du modèle Random Forest pour la prédiction des mouvements des prix des actions.

3.5 Implémentation du Bot de Référence

Afin de comparer les performances de notre modèle Random Forest et du modèle LSTM, nous avons implémenté un bot de référence utilisant une stratégie simple de "buy and hold".

3.6 Stratégie de Buy and Hold

La stratégie de "buy and hold" consiste à acheter une certaine quantité d'actions avec un capital initial à un moment donné et à les conserver jusqu'à la fin de la période d'observation. Voici les étapes détaillées de l'implémentation de cette stratégie :

1. **Téléchargement des Données** : Nous avons téléchargé les données historiques des actions sur la période de la simulation par exemple on prend le début 1er juin 2024 et la fin 10 juin 2024.

2. **Capital Initial et Calcul du Nombre d'Actions** : Le bot commence avec un capital initial de 10 000 $. Le prix d'ouverture de la première journée de trading est utilisé pour calculer le nombre d'actions que le bot peut acheter avec ce capital.

3. **Calcul du Capital Final** : À la fin de la période, le capital final est calculé en multipliant le nombre d'actions détenues par le prix d'ouverture de la dernière journée de trading. Le profit est ensuite calculé comme la différence entre le capital final et le capital initial.

Voici le code implémentant cette stratégie :

```
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np

ticker = '^FCHI'
start_date = '2024-06-01'
end_date = '2024-06-10'
data = yf.download(ticker, start=start_date, end=end_date)

initial_capital = 10000
capital = initial_capital

initial_price = data['Open'].iloc[0]
final_price = data['Open'].iloc[-1]

num_shares = capital // initial_price

final_capital = num_shares * final_price

profit = final_capital - initial_capital
profit_percentage = (profit / initial_capital) * 100
```

Listing 2 – Code du Bot de Référence pour la Stratégie de Buy and Hold

Les résultats obtenus à partir de la stratégie de "buy and hold" nous fourniront une référence pour évaluer les performances des autres modèles Random Forest et le LSTM. En comparant les profits réalisés et les ratios de Sharpe obtenus par ces différentes stratégies, nous serons en mesure de déterminer l'efficacité et la robustesse de nos modèles de prédiction.

Dans la section des résultats et discussions, nous analyserons les performances des différentes stratégies de trading et discuterons des avantages et des inconvénients de chacune d'entre elles.

3.7 Implémentation Pratique du Modèle LSTM

On a développé un modèle LSTM (Long Short-Term Memory) en utilisant les bibliothèques TensorFlow et Keras. Voici les étapes détaillées de notre implémentation :

3.7.1 Préparation des Données

Même approche que celle du Random Forest, téléchargement des données de Yahoo Finance puis ajout des indicateurs techniques : RSI, EMA, volatilité, rendements et directions. Les données sont ensuite normalisées par le z-score pour correspondre à l'échelle requise par le modèle LSTM.

Le code suivant montre la normalisation :

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_validation_scaled = scaler.transform(validation_data[features])
X_train_scaled = X_train_scaled.reshape((X_train_scaled.shape[0], 1,
    X_train_scaled.shape[1]))
X_validation_scaled = X_validation_scaled.reshape((X_validation_scaled.
    shape[0], 1, X_validation_scaled.shape[1]))
```

3.7.2 Construction du Modèle LSTM

On a consruit le modèle en utilisant Keras. ON a testé avec modèle comprend deux couches LSTM avec 50 unités chacune, suivies d'une couche dense avec une activation sigmoïde pour la classification binaire.

3.7.3 Choix des Couches et Paramètres

Couches LSTM : L'utilisation de deux couches LSTM permet au modèle de mieux comprendre et d'apprendre les représentations hiérarchiques des données séquentielles. La première couche LSTM, avec l'option `return_sequences=True`, retourne les séquences complètes de sorties pour chaque élément de la séquence d'entrée. Cela permet à la seconde couche LSTM de traiter ces séquences complètes et d'extraire des caractéristiques plus abstraites et de haut niveau.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 1, 50)             12200

 lstm_5 (LSTM)               (None, 50)                20200

 dense_2 (Dense)             (None, 1)                 51

=================================================================
Total params: 32451 (126.76 KB)
Trainable params: 32451 (126.76 KB)
Non-trainable params: 0 (0.00 Byte)
```

FIGURE 14 – Architecture de notr modèle LSTM

Première Couche LSTM : La première couche LSTM capture les motifs à court terme et les informations locales présentes dans les données de séries temporelles.

Seconde Couche LSTM : La seconde couche LSTM, qui reçoit les séquences complètes de la première couche, apprend les dépendances temporelles à plus long terme, permettant ainsi de capturer les tendances et les relations à long terme dans les données.

Ce choix d'architecture permet de modéliser à la fois les dynamiques à court et à long terme des données financières, améliorant ainsi la capacité prédictive du modèle.

Couche Dense : La couche dense finale avec une fonction d'activation sigmoïde est appropriée pour une tâche de classification binaire, permettant de prédire si le prix de l'action va augmenter ou diminuer.

Fonction de perte et Optimiseur : La fonction de perte utilisée est la binary crossentropy, qui est bien adaptée pour les tâches de classification binaire. L'optimiseur Adam a été choisi pour sa capacité à s'adapter dynamiquement au taux d'apprentissage pendant l'entraînement, ce qui en fait un choix populaire pour de nombreuses applications de deep learning.

Ces choix ont été faits pour leur efficacité à capturer les relations temporelles complexes dans les données financières et leur performance dans des tâches de classification.

```
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(X_train_scaled.
    shape[1], X_train_scaled.shape[2])))
model.add(LSTM(50))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
    accuracy'])
model.fit(X_train_scaled, y_train, epochs=10, batch_size=64, verbose=1)
```

3.7.4 Développement du Bot de Trading

Un bot de trading a été développé pour exécuter des transactions basées sur les prédictions du modèle LSTM même que dans le random forest. Le bot utilise les prédictions de direction pour décider d'acheter ou de vendre des actions. La logique de prise de décision est basée sur les prédictions de la direction des prix pour l'heure suivante.

Le bot commence avec un capital initial de 10 000 $ et ajuste les positions en fonction des prédictions. Le modèle LSTM est réentraîné après chaque prédiction pour inclure les nouvelles données.

```
capital_initiale = 10000
for i in range(len(validation_data)):
    current_data = validation_data.iloc[i:i+1]
    if current_data.empty:
        break
    X_current = current_data[features]
    X_current_scaled = scaler.transform(X_current).reshape((1, 1, len(features)))
    prediction = (model.predict(X_current_scaled) > 0.5).astype(int)[0][0]
    if prediction == 1 and capital > 0:
        position = capital / current_data['Adj Close'].values[0]
        capital = 0
    elif prediction == 0 et position > 0:
        capital = position * current_data['Adj Close'].values[0]
        position = 0
    historique_capital.append(capital + (position * current_data['Adj Close'].values[0]))
    X_train_scaled = np.append(X_train_scaled, X_current_scaled, axis=0)
    y_train = np.append(y_train, [current_data['Direction'].values[0]])
    model.fit(X_train_scaled, y_train, epochs=1, batch_size=64, verbose=0)
final_adj_close = validation_data['Adj Close'].iloc[-1]
if position > 0:
    capital = position * final_adj_close
historique_capital.append(capital)
print(f'Capital final : ${capital:.2f}')
print(f'Retour : {(capital - capital_initiale) / capital_initiale * 100:.2f}%')
```

Listing 3 – Code du Bot de Trading pour LSTM

Résultats et Discussion

4 Résultats et Discussion

Dans cette partie, nous présentons les résultats de nos tests effectués sur trois modèles : le bot de référence, le modèle LSTM, et le modèle Random Forest. Nous avons évalué ces modèles sur plusieurs entreprises, en utilisant deux métriques principales : le rendement et le ratio de Sharpe. Le ratio de Sharpe, calculé comme $\frac{R_p - R_f}{\sigma_p}$, mesure la performance ajustée au risque d'un investissement, où R_p est le rendement du portefeuille, R_f le taux sans risque, et σ_p l'écart type du rendement du portefeuille.

4.1 Résultats Financiers pour des PME

Entreprise	**Modèle / Stratégie**	**Rendement**	**Ratio de Sharpe**
Transat A.T. TRZ.TO	Bot de référence	-2.78%	-6.52
	LSTM	0.21%	2.81
	Random Forest	-1.07%	-18.62
Lululemon LULU	Bot de référence	-2.75%	5.43
	LSTM	3.50%	1.24
	Random Forest	1.22%	-10.16
Lumibird SA LBIRD.PA	Bot de référence	-5.51%	-13.78
	LSTM	1.96%	1.52
	Random Forest	0.39%	29.24
Arkema S.A AKE.PA	Bot de référence	-5.51%	-11.67
	LSTM	1.22%	-2.34
	Random Forest	-0.34%	-21.18
Rémy Cointreau RCO.PA	Bot de référence	-4.20%	-5.08
	LSTM	-1.89%	-2.69
	Random Forest	-1.53%	-26.24
CAC 40 FCHI	Bot de référence	-19.89%	0.29
	LSTM	0.68%	0.21
	Random Forest	0.03%	-0.28

TABLE 1 – Rendements des modèles financiers par entreprise pour des PME

L'analyse des PME montre que le modèle LSTM surpasse les autres approches avec des rendements et des ratios de Sharpe positifs pour des entreprises telles que Lululemon (3.50%, 1.24) et Lumibird SA (1.96%, 1.52). Le bot de référence affiche systématiquement des rendements négatifs, illustrant ses limites. Le modèle Random Forest présente des résultats contrastés, allant de -18.62 (TRZ.TO) à 29.24 (LBIRD.PA).

4.2 Résultats Financiers par Secteur

4.2.1 Secteur de l'Énergie

Entreprise	Modèle / Stratégie	Rendement	Ratio de Sharpe
Exxon Mobil Corporation	Bot de référence	-3.28%	-5.24
XOM	LSTM	-1.80%	-1.18
	Random Forest	-1.92%	-18.59
EONGY	Bot de référence	-2.02%	-6.83
EONGY	LSTM	0.73%	7.68
	Random Forest	0.59%	36.53
Engie SA	Bot de référence	-4.22%	-10.63
ENGIY	LSTM	1.23%	0.26
	Random Forest	0.55%	4.06
Petróleo Brasileiro S.A.	Bot de référence	-2.69%	-11.84
PBR	LSTM	-1.64%	2.76
	Random Forest	-2.33%	28.65

TABLE 2 – Rendements des modèles financiers dans le secteur de l'énergie

Dans le secteur de l'énergie, le modèle LSTM a généré des rendements positifs pour EONGY (0.73%) et Engie SA (1.23%), avec des ratios de Sharpe de 7.68 et 0.26 respectivement. Le modèle Random Forest présente un ratio de Sharpe exceptionnel de 36.53 pour EONGY.

4.2.2 Secteur de l'Industrie

Entreprise	Modèle / Stratégie	Rendement	Ratio de Sharpe
Holcim AG	Bot de référence	0.98%	18.26
HCMLY	LSTM	1.71%	5.37
	Random Forest	0.53%	-17.39
BASF SE	Bot de référence	-3.64%	-32.61
BASFY	LSTM	-0.39%	4.58
	Random Forest	-0.98%	-24.04
Dow Jones Industrial	Bot de référence	-4.21%	-2.15
DOW	LSTM	-1.27%	-1.24
	Random Forest	1.04%	-48.77
Vinci SA	Bot de référence	-1.75%	-11.19
DG.PA	LSTM	0.95%	-5.64
	Random Forest	0.15%	-27.37

TABLE 3 – Rendements des modèles financiers par entreprise dans le secteur industriel

Dans le secteur industriel, le LSTM a affiché des rendements positifs pour Holcim AG (1.71%) et Vinci SA (0.95%), avec des ratios de Sharpe de 5.37 et -5.64 respectivement. Random Forest a montré une performance notable pour Dow Jones Industrial (1.04%).

4.2.3 Secteur des Services et de la Distribution

Entreprise	Modèle / Stratégie	Rendement	Ratio de Sharpe
Tesco PLC	Bot de référence	0.18%	-17.55
TSCO.L	LSTM	0.97%	0.26
	Random Forest	0.11%	21.94
The Procter & Gamble	Bot de référence	-2.07%	4.71
PG	LSTM	-0.18%	-1.13
	Random Forest	-0.13%	-29.06
Danone S.A	Bot de référence	0.36%	-0.34
DANOY	LSTM	0.89%	-2.65
	Random Forest	-0.31%	35.70
The Estée Lauder	Bot de référence	-2.08%	-9.84
EL	LSTM	0.07%	-2.76
	Random Forest	-1.83%	-35.56

TABLE 4 – Rendements des modèles dans le secteur des services et de la distribution

Dans le secteur des services et de la distribution, le LSTM a généré des rendements positifs pour Tesco PLC (0.97%) et Danone S.A (0.89%), avec des ratios de Sharpe de 0.26 et -2.65 respectivement. Le Random Forest a montré un ratio de Sharpe exceptionnel de 35.70 pour Danone S.A.

4.3 Comparaison Globale des Modèles

En comparant les performances des différents modèles, nous observons plusieurs tendances clés :

— **Supériorité des LSTM** : Le modèle LSTM surpasse systématiquement les autres modèles avec des rendements positifs et des ratios de Sharpe favorables. Par exemple, Lululemon (3.50%, 1.24) et Engie SA (1.23%, 0.26) démontrent la capacité du LSTM à gérer les séquences temporelles complexes.

— **Performance du Random Forest** : Le Random Forest présente des résultats variés avec des rendements positifs et des ratios de Sharpe élevés dans certains cas spécifiques, comme EONGY (36.53) et Danone S.A (35.70).

— **Limites du Bot de Référence** : Le bot de référence "buy and hold" affiche systématiquement des rendements négatifs et des ratios de Sharpe défavorables, mettant en évidence ses limitations dans la gestion de la volatilité du marché.

4.4 Analyse Comparative entre PME et Grandes Entreprises

En comparant les PME et les grandes entreprises, nous observons les tendances suivantes :

— **Rendements des PME** : Les PME, telles que Lululemon (3.50%, 1.24) et Lumibird SA (1.96%, 1.52), montrent des rendements élevés mais avec une volatilité accrue. Les modèles LSTM réussissent à capturer les dynamiques complexes de ces entreprises.

— **Stabilité des Grandes Entreprises** : Les grandes entreprises, telles que celles du CAC 40 ou dans le secteur de l'énergie, présentent des rendements plus stables. Par exemple, le CAC 40 avec un ratio de Sharpe de 0.29 et Engie SA avec 0.26 démontrent une plus grande résilience aux fluctuations du marché.

Conclusion et Perspectives Futures

5 Conclusion et Perspectives Futures

Cette étude a mis en lumière l'efficacité des modèles d'apprentissage automatique, notamment les LSTM, pour la prédiction des rendements boursiers. Les résultats montrent que les modèles LSTM offrent une amélioration notable des performances par rapport aux approches traditionnelles, particulièrement dans des contextes de données complexes et volatiles.

Le modèle Random Forest, bien qu'efficace dans certains contextes spécifiques, montre des performances moins stables globalement. Ces résultats soulignent l'importance de choisir le bon modèle en fonction des caractéristiques des données et des objectifs spécifiques de prédiction.

Voici quelques perspectives pour améliorer nos modèles :

5.1 Enrichissement des Données

Intégrer davantage de données exogènes telles que des indicateurs macroéconomiques, des actualités financières et des sentiments des réseaux sociaux. Ces données supplémentaires vont enrichir les modèles et permettre des prédictions encore plus précises, surtout si le comportement des données est imprévisible.

L'analyse sentimentale, par exemple, peut être intégrée en utilisant des techniques de traitement du langage naturel (NLP). En scrappant des avis et des news en temps réel et en analysant les sentiments exprimés dans les articles, les tweets, et les rapports financiers, nous pouvons créer de nouvelles variables qui enrichissent nos modèles. Ce processus est complexe car il nécessite une collecte de données en temps réel, un nettoyage et une structuration des données textuelles, ainsi que le développement d'algorithmes pour différencier les nouvelles pertinentes des nouvelles non pertinentes ou fausses.

On doit donc construire des pipelines robustes pour le scraping de données en temps réel. Ces pipelines vont traiter de grandes quantités de données textuelles, en utilisant des techniques de NLP pour extraire les sentiments et les informations clés. Par exemple, des modèles de classification de texte tels que GPT-3 peuvent être utilisés pour analyser le ton et le contenu des nouvelles, fournissant ainsi des indications précieuses sur l'humeur du marché.

La quantité de contenu généré par l'IA sur Internet connaît une croissance rapide.[13] D'après certaines estimations, jusqu'à 90 % du contenu en ligne pourrait être généré par des intelligences artificielles d'ici 2025. et donc l'idée c'est de développer des systèmes basés sur des réseaux antagonistes génératifs (GAN) [14] pour différencier le contenu généré de manière automatique du contenu non généré. Les GAN sont des modèles d'apprentissage profond capables de générer du contenu réaliste, et dans l'entrainement on développe aussi en parallèle un modèle capable de détecter si le contenu est génératif ce qui va être pertinent pour nous dans ce cas.

5.2 Modèles Hybrides

Développer des modèles hybrides qui combinent les forces des LSTM avec d'autres techniques de machine learning et deep learning. Par exemple, intégrer des réseaux de

neurones convolutifs (CNN) pour capturer les motifs visuels dans les données de séries temporelles. Un modèle hybride pourrait utiliser les LSTM pour capturer les dépendances temporelles et les CNN pour détecter les motifs saisonniers ou les tendances périodiques dans les séries temporelles.

5.3 Optimisation des Hyperparamètres

Utiliser des méthodes avancées d'optimisation pour affiner les hyperparamètres des modèles. Une optimisation rigoureuse peut améliorer significativement les performances des modèles prédictifs. Jusqu'à aujourd'hui, il n'existe pas une fonction qui choisit automatiquement la meilleure structure du LSTM et les couches, avec différents paramètres. Une perspective future pourrait inclure le développement ou l'utilisation de frameworks d'optimisation d'hyperparamètres automatisés pour tester systématiquement différentes configurations de modèles et identifier celles qui offrent les meilleures performances.

En conclusion, cette étude a démontré le potentiel des modèles d'apprentissage automatique pour améliorer la prédiction des rendements boursiers. Les futures recherches devraient se concentrer sur l'enrichissement des données, l'optimisation des modèles et l'application pratique dans des environnements de marché réels, tout en tenant compte des implications éthiques et réglementaires. Avec ces perspectives, nous pouvons continuer à avancer dans le développement de modèles prédictifs robustes et efficaces, capables de naviguer dans les complexités des marchés financiers modernes.

Références

[1] William F. Trench, "Explicit weighting coefficients for predicting ARMA time series from the finite past," *Journal of Computational and Applied Mathematics*, vol. 34, pp. 251–262, 1991, North-Holland. https://www.sciencedirect.com/science/article/pii/037704279190047N.

[2] Basher, S.A. and Sadorsky, P. (2016) 'Hedging emerging market stock prices with oil, gold, VIX, and bonds : A comparison between DCC, ADCC and GO-GARCH', Energy Economics, 54, pp. 235–247. https://www.sciencedirect.com/science/article/pii/S0140988315003485.

[3] Li, X.Z., and Kong, J.M., "Application of GA–SVM method with parameter optimization for landslide development prediction," *Natural Hazards and Earth System Sciences*, vol. 14, no. 3, pp. 525–533, 2014. https://nhess.copernicus.org/articles/14/525/2014/.

[4] WEN Fenghuaa, XIAO Jihongb, HE Zhifanga, GONG Xua., "Stock Price Prediction Based on SSA and SVM,". Procedia Computer Science 31 (2014) 625 – 631 https://www.sciencedirect.com/science/article/pii/S1877050914004864.

[5] Ismail MS, Noorani MSM, Ismail M, et Mohd AL., "Predicting next day direction of stock price movement using machine learning methods with persistent homology : evidence from Kuala Lumpur Stock Exchange," vol. 93, 106422, 2020. https://www.sciencedirect.com/science/article/pii/S1568494620303628.

[6] Bingjie Loua, Bo Shaoaa, Chenchen Nia, Yixuan Donga, Shuning Yuea, Ming Zhu, "Quantitative Timing Strategy Model Based on Improved KNN," in : Procedia Computer Science 202 (2022) 61–66. https://www.sciencedirect.com/science/article/pii/S1877050922005439.

[7] Hoseinzade, E. and Haratizadeh "CNNpred : CNN-based stock market prediction using a diverse set of variables", Expert Systems with Applications, 129, pp. 273–285. https://www.sciencedirect.com/science/article/pii/S0957417419301915.

[8] Andrea Perlato, "Recurrent Neural Network in Theory," 2021. Available at : https://www.andreaperlato.com/aipost/recurrent-neural-network-in-theory/.

[9] IBM, "Random Forest," 2024. Available online : https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

[10] Arjun R. and Suprabha K. R., "A bibliometric review of stock market prediction : Perspective of emerging markets," Applied Computer Systems, vol. 25, no. 2, pp. 77, December 2020. https://doi.org/10.2478/acss-2020-0010.

[11] Google Colab, "Stock Price Prediction with LSTM and Random Forest," Available at : https://colab.research.google.com/drive/1FZJKlAHKSGUlh5b4YbLaKi6Rhhx9RJ2T?usp=sharing.

[12] Badreddine Saadioui, "Stock prediction with LSTM and Random Forest," GitHub repository, 2024. Available at : https://github.com/badreddinesaadioui/Stock-prediction-with-LSTM-and-Random-Forest.

[13] PwC, "The Future of Content in the Generative AI Age," Available at : https://www.pwc.com/us/en/tech-effect/ai-analytics/future-of-content-in-the-generative-ai-age.html .

[14] Badreddine Saadioui, "GAN for Wireless Signal Spoofing," GitHub repository, 2024. Available at : https://github.com/badreddinesaadioui/GAN-for-Wireless-signal-spoofing/blob/main/Article_Scientifique.pdf.