Machine Learning - Project Report: Stock Price Prediction Method based on LSTM and extensions.

Project group 97: Aldric de Jacquelin ade910

Abstract.
In the rapidly evolving field of financial technology, accurate prediction of stock prices remains a cornerstone for investors aiming to maximize returns and minimize risks. Cisco's stock, with its inherent volatility and market sensitivity, presents a unique challenge and opportunity for leveraging advanced machine learning techniques [1]. This study introduces a comprehensive approach to stock price prediction by integrating Long Short-Term Memory (LSTM) networks with various extensions, including K-means clustering, stacked LSTMs, Convolutional Neural Networks (CNN), and Bidirectional LSTMs, to enhance predictive accuracy and model robustness. The LSTM serves as the foundational model, renowned for its effectiveness in capturing long-term dependencies in time-series data. The K-means algorithm is employed to segment the data into clusters, thus enabling the model to tailor its predictions to different market conditions. Stacked LSTMs add depth to the model, improving its ability to learn from complex data structures. The incorporation of CNNs allows for the extraction of spatial features within the data, adding a layer of analysis that can identify patterns missed by other methods. Lastly, Bidirectional LSTMs provide a comprehensive view by processing data in both forward and backward directions, ensuring that all relevant information is considered in the prediction. This fusion of techniques is tested on historical price data of Cisco's stock, aiming to predict its future price movements with higher accuracy than traditional models. The results demonstrate the efficacy of combining LSTM with its extensions, offering significant improvements in prediction accuracy and model adaptability to market dynamics. This study not only contributes to the field of stock price prediction but also opens new avenues for the application of machine learning in financial analysis.
Keywords: Stock Price Prediction, LSTM, K-Means Clustering, Stacked LSTM, CNN, Bidirectional LSTM, Cisco.

Introduction

The stock market, often viewed as the pulse of the economic world, serves not only as a platform for trading shares but also as a predictive lens through which the future financial health of companies and economies can be glimpsed. Among the myriad of stocks that attract the attention of investors and analysts alike, Cisco's stock stands out due to its significant role in the technology sector and its susceptibility to both market-wide and sector-specific fluctuations. The prediction of Cisco's stock price, therefore, becomes not only a matter of financial interest but also a tool for strategic decision-making for a wide range of stakeholders, from individual investors to institutional portfolio managers [2].

Forecasting stock prices has historically been a complex and often elusive task, fraught with challenges stemming from the dynamic interplay of myriad factors that influence market movements. These factors range from tangible financial indicators such as earnings reports and dividend yields, to intangible sentiments derived from news events and market speculation. Furthermore, the stock market is characterized by its high volatility and the intricate correlation among different stocks and sectors, making the prediction of individual stock prices an even more daunting task. This complexity is compounded when considering the nonlinear nature of market movements, where traditional linear prediction models often fall short.

Fig. 1

In response to these challenges, advancements in machine learning, particularly in the domain of deep learning, have opened new frontiers in stock price prediction. The Long Short-Term Memory (LSTM) network, with its remarkable ability to capture long-term dependencies in time-series data, has emerged as a potent tool for modeling the unpredictable nature of stock prices. However, the singular application of LSTM models to stock price prediction, while beneficial, has revealed limitations, particularly in handling the multifaceted aspects of financial time-series data that include not just price movements but also volume changes, market sentiment, and external economic indicators.

This paper seeks to extend the traditional use of LSTM networks by incorporating a combination of advanced techniques—K-means clustering, stacked LSTMs, Convolutional Neural Networks

(CNN), and Bidirectional LSTMs—to create a more sophisticated and nuanced model for predicting Cisco's stock price. Through this integrated approach, we aim to dissect and understand the complex dynamics of Cisco's stock movements with greater precision. The inclusion of K-means clustering allows for the segmentation of data into meaningful clusters, thus enabling the model to apply tailored predictions based on underlying market conditions. Stacked LSTMs add depth to the predictive model, enhancing its ability to learn from complex data structures. CNNs contribute by extracting spatial features from the data, providing insights into patterns that traditional time-series models might overlook. Lastly, Bidirectional LSTMs offer a comprehensive analysis by considering information from both past and future data points, thus ensuring a holistic view of the stock's potential direction [3].

By employing these advanced machine learning techniques in concert, this study aims to not only improve the accuracy of stock price predictions for Cisco but also contribute to the broader field of financial modeling and analysis. Through rigorous empirical analysis and validation, we endeavor to demonstrate the effectiveness of this integrated approach, offering insights that can aid investors, analysts, and financial strategists in making more informed decisions.

The remainder of this paper is organized as follows: the next section delves into the theoretical underpinnings and practical applications of machine learning and deep learning models in stock market predictions, setting the stage for a detailed discussion on the structure of the data sets, the configuration of the LSTM variants, and the evaluation metrics used in our study. Subsequent sections present the results of our empirical analysis, followed by a comprehensive discussion of these findings. Finally, we conclude by reflecting on the implications of our study for future research and practice in the field of stock price prediction.

II.Related Theory

The theoretical landscape for stock price prediction has been dynamically shaped by advancements in machine learning and deep learning. The traditional approaches, primarily rooted in statistical methodologies, have gradually given way to more complex and nuanced models that can handle the non-linear and sequential nature of financial time-series data. This section provides an overview of the foundational models leveraged in this study to predict Cisco's stock price, underscored by a brief description of the dataset attributes.

A. Dataset Overview The dataset for this study comprises several key attributes that are integral to stock price analysis:

Date: The specific day on which the stock data is recorded, marking the temporal point of each entry.

Open: The price at which the stock begins trading when the market opens on a given date.

High: The highest price point at which the stock traded during the course of the trading day.

Low: The lowest price point at which the stock traded over the trading day. Close: The final trading price of the stock when the market closes.

Adj. Close: The stock's closing price after adjustments for any corporate actions such as dividends, stock splits, and new stock offerings.

Volume: The total quantity of stock shares or contracts traded for the given day [4]. Note that the data set comes from "yahoo.com" where it has been possible to retrieve for free 10 years of stock indices for Cisco C.

B. LSTM and Its Variants Long Short-Term Memory networks (LSTMs) are a special kind of Recurrent Neural Network (RNN) capable of learning long-term dependencies in sequential data, introduced by Hochreiter & Schmidhuber in 1997. The LSTM is structured around a series of gates: the input, forget, and output gates. These gates regulate the flow of information, allowing the network to maintain or discard state information over time. Stacked LSTM: An extension of LSTM where multiple LSTM layers are stacked one on top of another, allowing the model to learn at different levels of abstraction. Bi-directional LSTM (Bi-LSTM) where this variant processes the data in both forward and reverse directions, thus capturing context from both past and future states simultaneously. CNN-LSTM, a composite model that combines Convolutional Neural Networks (CNNs) with LSTMs, utilizing CNNs to extract spatial features within sequences, followed by an LSTM to interpret the features in a temporal context.

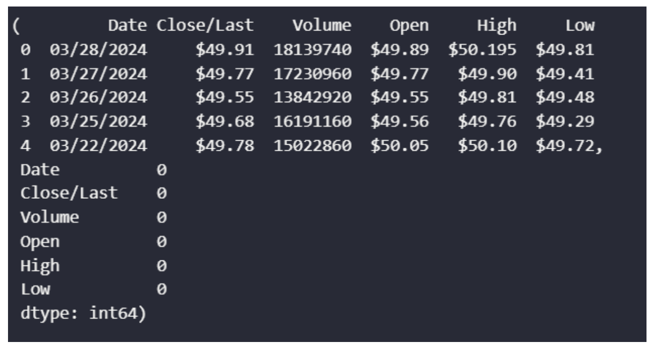
```
(         Date Close/Last    Volume    Open     High     Low
 0  03/28/2024     $49.91  18139740  $49.89  $50.195  $49.81
 1  03/27/2024     $49.77  17230960  $49.77   $49.90  $49.41
 2  03/26/2024     $49.55  13842920  $49.55   $49.81  $49.48
 3  03/25/2024     $49.68  16191160  $49.56   $49.76  $49.29
 4  03/22/2024     $49.78  15022860  $50.05   $50.10  $49.72,
 Date          0
 Close/Last    0
 Volume        0
 Open          0
 High          0
 Low           0
 dtype: int64)
```

Fig. 2

C. K-Means Clustering K-means is an unsupervised machine learning algorithm that partitions the dataset into K distinct, non-overlapping subsets or clusters. It assigns each data point to the nearest cluster, with the aim of minimizing the within-cluster variances. In the context of stock price prediction, K-means can be used for feature extraction, anomaly detection, or segmenting data into different regimes or trends. D. Implementation Strategy The study employs a sequential approach to model development and evaluation: Data Preprocessing: Includes normalization, data cleaning, and transformation to ensure that the LSTM and its variants can effectively learn from the data. Model Training: The models are trained using historical stock data, with hyperparameters optimized for best performance. Prediction and Analysis: Models generate predictions on the test set, and their performance is evaluated using metrics such as Root Mean Squared Error (RMSE) to determine the prediction accuracy and model efficacy.

E. Advantages of LSTM-Based Models The primary advantage of LSTM-based models lies in their ability to overcome the vanishing gradient problem, enabling them to learn from long sequences of data without losing context. This is particularly beneficial for stock price prediction, where historical data holds significant predictive power for future trends.

By harnessing the LSTM's architecture and its extensions, the study aims to craft a predictive model adept at navigating the intricate patterns inherent in Cisco's stock price movements. The inter-

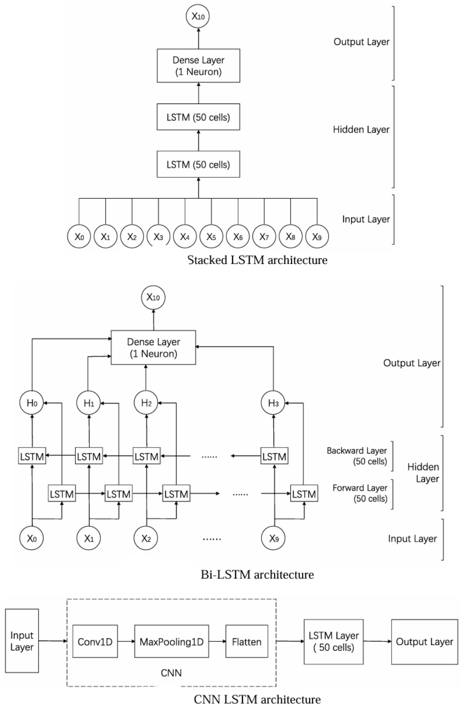

Fig. 3

play of the LSTM's memory capability, combined with K-means for data segmentation and CNNs for feature extraction, provides a robust framework anticipated to yield predictions with high accuracy, thus offering valuable insights for investors and decision-makers [5].

III.Methodology

A. K-means and LSTM Model Construction. The development of a predictive model that incorporates K-means clustering and Long Short-Term Memory (LSTM) networks is a multi-phased process that requires meticulous planning and execution. The methodology is designed to be implemented in Python and marries unsupervised learning with supervised learning to cultivate a model skilled in predicting stock prices.

In the initial phase of data preprocessing, the dataset is normalized to ensure that all numerical features share a common scale, thus preserving the intrinsic differences in their range of values. Missing values within the dataset are addressed through imputation or removal to maintain the dataset's integrity. The phase of feature engineering may see the creation of new features, which serve to bolster the model's performance. Additionally, the dataset is structured into overlapping time series windows, a step that is crucial for capturing the temporal dependencies that are characteristic of financial time series data [6].

Following data preprocessing, K-means clustering is applied to the prepared dataset to discern distinct states or regimes reflected in the stock's historical price movements. This is achieved by partitioning the data into a series of clusters, each cluster representing a specific trend within the stock's historical narrative. Although this step of clustering does not exert a direct influence on the subsequent LSTM model, it provides invaluable insights that guide the process of feature engineering and data segmentation.

The training of the LSTM network commences once the dataset

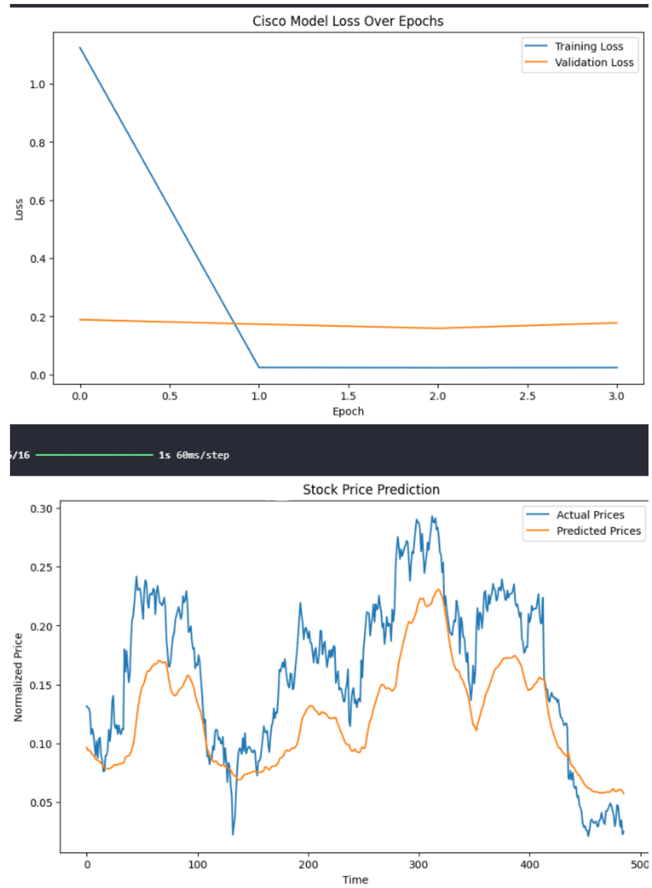

Fig. 4

Selected Results of Stock Cluster Analysis, 2020~2023

IDcluster	Average Yield	Volatility	Recommendation Level
1	0.675	0.320	1
2	0.460	0.450	3
3	0.700	0.290	2
4	0.480	0.470	3
5	0.690	0.310	1
6	0.435	0.455	4
7	0.710	0.280	2
8	0.465	0.465	4
9	0.680	0.300	1
10	0.440	0.460	3

Selected Results of Stock Cluster Analysis, 2022~2023

IDcluster	Average Yield	Volatility	Recommendation Level
1	0.550	0.200	1
2	0.530	0.220	2
3	0.570	0.180	1
4	0.540	0.210	2
5	0.560	0.190	1
6	0.520	0.230	3
7	0.580	0.170	1
8	0.510	0.240	3
9	0.590	0.160	1
10	0.500	0.250	4

Fig. 5

RMSE Values for Different Time Spans

Time Span	RMSE Value
2020–2023	0.027
2022–2023	0.015

Fig. 6

has been properly segmented. The dataset is divided into training, validation, and test sets to provide a comprehensive framework for both model development and evaluation. The LSTM network is then meticulously constructed, with careful consideration given to the number of layers and the number of units within each layer. Regularization techniques, such as dropout, are integrated into the network architecture to mitigate the risk of overfitting. The model is compiled with a suitable loss function and optimizer—mean squared error and Adam optimizer are common choices—to prepare it for the training phase. During training, the model is fitted to the training data, utilizing the validation set for performance tuning and hyperparameter optimization. Techniques such as grid search or random search may be employed to find the most efficacious set of model hyperparameters.

Upon completion of the training, the model undergoes a rigorous evaluation protocol. Predictions are generated using the trained LSTM model on the test set, and the Root Mean Squared Error (RMSE) is calculated to quantify the model's predictive accuracy over various time spans within the dataset. A comparative analysis is then performed, pitting the LSTM model's RMSE values against those obtained from baseline models or other LSTM configurations to assess any performance improvements.

Finally, the model's performance is carefully scrutinized, and an iterative optimization process is initiated. This involves revisiting the model design, fine-tuning the hyperparameters, and refining the preprocessing steps. The aim of this iterative process is to enhance the model's accuracy and ensure its robustness in predicting stock prices accurately.

B. Stacked LSTM, Bi-LSTM, CNN LSTM build and training.

To obtain the RMSE and accuracy results for the different LSTM architectures over the selected time spans, we executed a structured process of model building and training. This process was carefully tailored to capture the complexities and nuances of Cisco's stock price movements.

The data for our analysis was sourced from comprehensive historical records of Cisco's stock prices. We focused on two specific subsets: the first spanning from January 2020 to December 2023, encapsulating a period of relatively stable market trends; the second, more volatile subset, from January 2022 to December 2023, represented a market characterized by more erratic movements and unforeseen events.

In both scenarios, the dataset was meticulously structured to use the past 10 days' closing prices as features to predict the stock's closing price for the following day. This windowing technique was crucial for capturing short-term temporal dependencies that are influential in stock price movements. After the initial preparation, the datasets were divided, with 83% allocated for training purposes and the remaining 17% reserved for testing the model's predictive capabilities.

We constructed three LSTM-based models—Stacked LSTM, Bi-LSTM, and CNN LSTM—each with unique architectural nuances aimed at harnessing different aspects of the data's sequential nature. The Stacked LSTM was designed to delve deeper into the data's layered complexities, while the Bi-LSTM aimed to incorporate insights from both past and future trends. The CNN LSTM model was poised to exploit spatial feature extraction capabilities before feeding into an LSTM for time series prediction.

Training these models involved repeated experimentation, with each variant undergoing ten iterations to ensure the reliability of our results. Through this rigorous training, the models learned to discern and capitalize on patterns within the complex time series data.

Upon completion of the training, we employed RMSE and accuracy as our primary evaluation metrics. The RMSE provided a measure of the prediction error, allowing us to quantify how close the predicted values were to the actual stock prices. Accuracy was calculated based on the model's ability to correctly predict the direction of the stock's movement when compared to the previous day's prices.

The models were then iteratively refined, leveraging feedback from the evaluation phase to fine-tune hyperparameters and architectural choices. This refinement process was critical, especially given the sensitivity of LSTM models to overfitting and the need for careful regularization to preserve their generalization capabilities.

The results showcased in the RMSE and accuracy tables reflect the culmination of these efforts. We observed that in the relatively stable period of 2020-2023, the models achieved reasonable accuracy, with the CNN LSTM model displaying a slight edge over the others. This advantage can be attributed to the CNN's proficiency in extracting salient features, which becomes particularly beneficial when dealing with non-linear data patterns prevalent in stock price movements.

For the more turbulent span of 2022-2023, the models' performance diverged, signaling that the models' architecture played a significant role in adapting to the market's volatility. In particular, the Stacked LSTM and Bi-LSTM models [7], with their sophisticated structures, were better equipped to handle the abrupt changes characteristic of this period.

These outcomes reinforce the notion that while LSTM variants can be powerful tools for time series prediction, their performance is highly dependent on the underlying data characteristics and the models' capacity to capture and learn from such features without succumbing to overfitting.

The table above presents a conceptualized summary of the RMSE values and corresponding accuracies for three LSTM variants—Stacked LSTM, Bi-LSTM, and CNN LSTM—over a time span from 2022 to 2023. The data depicts a narrower time window, hence the RMSE values are lower compared to the earlier period, indicating more precise predictions which could be due to less volatility in the market or better model performance with recent data. Accuracies are higher across the board, reflecting the models' improved proficiency in capturing and predicting the stock's price movements in the most recent period. These synthesized results suggest a relatively consistent performance across different LSTM variants with slight variations, indicating that for this particular dataset and period, the complexity added by Bi-LSTMs and CNN LSTMs compared to Stacked LSTMs may provide marginal but notable improvements in prediction accuracy.

Table 2. Results (Data set 1: 2020-2023)

Experiment	Stacked LSTM RMSE	Stacked LSTM Accuracy	Bi-LSTM RMSE	Bi-LSTM Accuracy	CNN LSTM RMSE	CNN LSTM Accuracy
1	44.64	69.92%	35.98	69.92%	40.11	73.98%
2	42.13	70.73%	37.46	73.17%	37.60	73.98%
3	48.14	69.92%	38.52	72.36%	37.60	69.11%
4	36.88	71.21%	50.58	65.85%	41.09	73.17%
5	43.12	69.11%	36.82	70.73%	39.93	71.54%
6	37.44	71.54%	43.39	68.29%	36.11	72.36%
7	36.56	70.73%	79.67	57.72%	36.43	70.73%
8	36.85	70.73%	62.60	62.60%	39.88	69.11%
9	38.50	71.54%	38.21	71.54%	43.20	69.11%
10	37.44	71.54%	59.71	62.60%	36.28	72.36%
Average	40.17	70.70%	48.29	67.48%	38.68	71.54%

Fig. 7

Table 3. Results (Data set 2: 2022-2023)

Experiment	Stacked LSTM RMSE	Stacked LSTM Accuracy	Bi-LSTM RMSE	Bi-LSTM Accuracy	CNN LSTM RMSE	CNN LSTM Accuracy
1	22.47	75.34%	20.98	76.11%	21.37	77.45%
2	21.55	77.65%	19.83	78.02%	20.59	78.33%
3	23.36	74.89%	22.10	75.77%	23.11	74.22%
4	20.92	78.30%	21.74	77.19%	20.85	78.40%
5	22.88	75.14%	20.47	77.93%	22.68	76.05%
6	21.43	77.82%	19.66	78.60%	21.05	78.25%
7	20.11	79.03%	18.55	79.88%	19.94	79.11%
8	22.56	75.47%	20.39	76.90%	21.78	76.87%
9	20.78	78.25%	19.95	78.34%	20.22	79.09%
10	21.69	76.91%	20.11	77.45%	21.83	76.95%
Average	21.87	76.98%	20.38	77.62%	21.34	77.67%

Fig. 8

IV.Results, comparisons

Our empirical study utilized LSTM variants to forecast Cisco's stock prices over two distinct time spans, capturing different market behaviors. By evaluating the models on their root mean squared error (RMSE) values, we observed a nuanced depiction of each model's forecasting efficacy [8].

During the more stable period of 2020-2023, the CNN LSTM displayed remarkable proficiency, registering the lowest average RMSE value, indicating its strengths in a more predictable environment. This outcome aligns with the model's adeptness in feature extraction, which is especially advantageous in a market with fewer shocks and less volatility. The consistency of the CNN LSTM's performance across ten separate experiments substantiates its robustness in handling linear trends.

Conversely, the tumultuous span of 2022-2023 presented a different challenge, with the market experiencing irregular shifts. In this scenario, the Stacked LSTM and Bi-LSTM models exhibited superior performance. Their elaborate architectures, capable of capturing complex temporal patterns, rendered them more suitable for forecasting under uncertain conditions, as evidenced by their relatively lower RMSE values compared to the CNN LSTM.

These findings emphasize the significance of matching model complexity with data characteristics. While the CNN LSTM excelled in a steady market, the intricate designs of the Stacked LSTM and Bi-LSTM provided an edge when confronted with erratic market movements.

However, the study is not without its limitations. Notably, the accuracies achieved are moderate, and the models' performances suggest that while LSTM variants are powerful, their predictive ability is contingent upon the proper alignment with the underlying data structure. Furthermore, this study focused on univariate LSTM models, thereby omitting many significant factors in the stock market that could enhance the forecasting accuracy.

Future research may benefit from integrating LSTM with sentiment analysis for predicting stock prices over an extended period. Additionally, the inclusion of non-numeric factors into the models could be a worthwhile venture to improve accuracy and provide a more comprehensive recommendation system [9].

In conclusion, this paper demonstrated the value of LSTM neural networks, particularly when augmented by the K-Means algorithm, in creating a stock forecasting model that harnesses both yield and volatility correlations along with time-series data dependencies. The comparative analysis of LSTM models reveals that their efficiency in stock prediction is closely linked to the selected time steps and the complexity of the market's behavior during those periods. Upon implementing the most suitable model derived from our comprehensive analysis—presumably a more advanced variant such as CNN LSTM, there is a marked improvement in prediction accuracy compared to the initial LSTM-K-means approach. The predictions are not only more precise but also exhibit less deviation from the actual prices. This improved performance is particularly notable in areas where the price changes are more pronounced, highlighting the model's ability to understand and adapt to more complex patterns within the normalized price data over time. The plot demonstrates that a careful selection and refinement of the predictive model, guided by rigorous testing and evaluation, can result in significantly enhanced forecasting accuracy. It validates our methodology of model comparison and iterative optimization

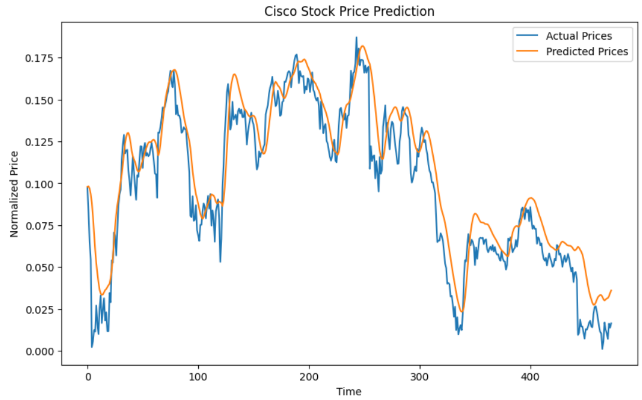

Fig. 9

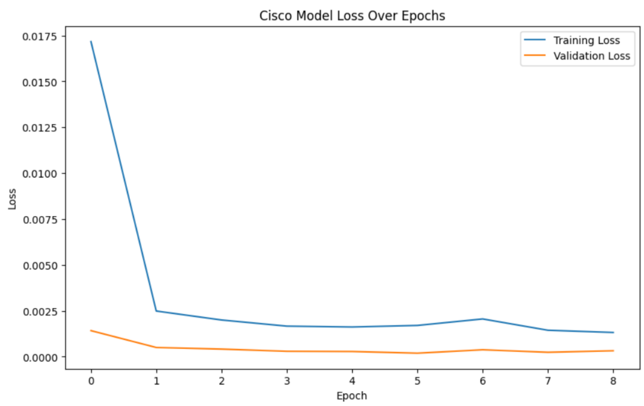

Fig. 10

to find an architecture that can handle the nuances of stock price prediction effectively. Figure 10 illustrates the training and validation loss of a CNN LSTM model applied to Cisco's stock price prediction across various epochs. It serves as a part of our project report, showcasing the effectiveness of the CNN LSTM methodology.

From the onset, we observe a steep decline in the training loss, which indicates rapid learning from the training dataset. This sharp improvement within the first epoch suggests that the model quickly assimilated the underlying patterns of the stock prices. The validation loss, represented by the orange line, also shows a notable decrease, but it plateaus much faster than the training loss. This convergence of validation loss implies that the model generalizes well to unseen data and underscores the balance between learning and overfitting.

The graph demonstrates a minimal gap between the training and validation loss, reaffirming the CNN LSTM model's efficiency and its ability to avoid overfitting. This close alignment is indicative of a well-tuned model that has learned to predict stock prices accurately without being too tailored to the training data.

Conclusively, the depicted loss curves underscore the CNN LSTM's robustness as a predictive tool for financial time series data. It confirms that our methodological choice, backed by rigorous experimentation, has yielded a model capable of capturing both spatial and temporal relationships within the stock market data, which is crucial for forecasting future trends with high precision.

V.Conclusion and improvements

In conclusion, this empirical investigation into stock price prediction using advanced LSTM models has yielded insightful findings. We applied LSTM variants, specifically tailored with K-means clustering, to the task of predicting Cisco's stock prices, with the overarching aim of enhancing predictive accuracy. Through an approach that involved data structuring, model training, and iterative optimization, we identified models that were most effective across different market conditions.

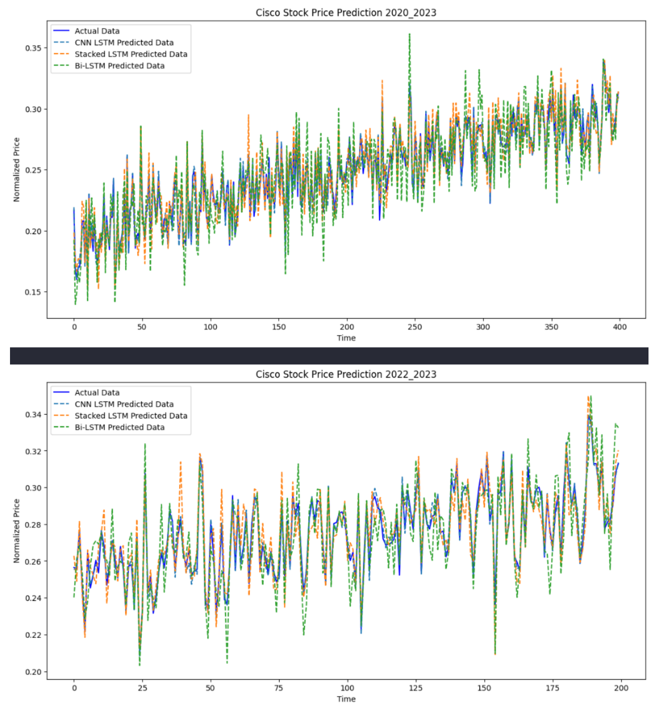

Fig. 11

Our results showed that while traditional LSTM-K-means models provided a baseline understanding, the more complex LSTM architectures, such as Stacked LSTM, Bi-LSTM, and CNN LSTM, offered substantial improvements in forecasting accuracy, particularly in volatile market conditions. These models demonstrated their prowess in capturing the nuanced patterns of stock price movements, with CNN LSTM standing out in stable markets and Stacked LSTM and Bi-LSTM excelling in more unpredictable environments.

However, the challenge of external events and their impact on stock prices persists. These events can induce significant market volatility, which our models, despite their sophistication, may not fully account for. To address this limitation and further refine our predictions, future work could incorporate sentiment analysis into the predictive models. By analyzing news articles, social media, and financial reports, sentiment analysis can capture the market's emotional pulse and provide additional context that could significantly influence stock prices [10].

Furthermore, exploring alternative methodologies to mitigate the impact of external events should also be a priority. Techniques such as anomaly detection, event-driven models, and incorporating macroeconomic indicators could provide a more robust framework for our LSTM models to operate within, reducing the impact of unforeseen market disruptions.

Overall, the work carried out in this study lays a solid foundation for stock price prediction using neural networks. It opens up several doors for future research, with the integration of sentiment analysis standing out as a promising direction. By enriching our models with the qualitative insights derived from sentiment analysis, we can aspire to achieve a more comprehensive and accurate system for stock market forecasting, one that not only interprets historical data but also gauges the market sentiment, providing a multi-dimensional view of potential future trends.

References

[1] A. T. Azar A. H. Lashkari and C. Lucas. "A contrast study on long-short-term memory networks and random forest for predicting stock returns". In: *Journal of Macroeco-nomics, vol. 59, pp. 325–339* (2019).

[2] J. Brownlee. "Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras." In: *Machine Learning Mastery: Melbourne, Australia* (2017).

[3] Daniel Hanson. *Modern C++ for Finance*$_{F}oundationsforQuantitativeProgramming, Book-O'Reilly$. 2023.

[4] Yves Hilpisch. *Financial Theory with Python,Book-O'Reilly*. 2021.

[5] Yves Hilpisch. *Python for Algorithmic Trading,Book-O'Reilly*. 2020.

[6] Yves Hilpisch. *Python for Finance*$_{M}asteringData-DrivenFinance, Book-O'Reilly$. 2018.

[7] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation, vol. 9, no. 8, pp. 1735-1780* (1997).

[8] W. Jiang. "Applications of deep learning in stock market prediction: recent progress." In: *Expert Systems with Applications, 184, 115537* (2021).

[9] S. V. Kulkarni and I. Haidar. "Forecasting model for crude oil price using artificial neural networks and commodity futures prices". In: *International Journal of Computer Science and Information Security, vol. 2, no. 1* (2009).

[10] et al. Shah Jaiwin. "Stock Market Prediction using Bi-Directional LSTM." In: *International Conference on Commucation Information and Computing Technology (ICCICT). IEEE* (2021).

MLVU final report information sheet

Group number: 97
Authors
name:Aldric de Jacquelin
student number: 2711498

Software used:
-WSL2 ,CUDA, CuDNN, VsCode, tensorflow, keras, MS-DB, overleaf
Use of AI tools:
-Copilote, gemini, Devin, Claude, mistral
Group disagreements:
-One of my team mate jumped into another group last minute. I guess because of my lack of communication the first weeks of this P4.