Communication Papers of the of the 17th Conference on Computer Science and Intelligence Systems pp. 233–240
DOI: 10.15439/2022F20
ISSN 2300-5963 ACSIS, Vol. 32

Financial News Effect Analysis on Stock Price Prediction Using a Stacked LSTM Model

Alexandre Heiden
Santa Catarina State University
Graduate Program in Applied Computing
Joinville, SC – Brazil
Email: alexandreheiden@hotmail.com

Rafael Stubs Parpinelli
Santa Catarina State University
Graduate Program in Applied Computing
Joinville, SC – Brazil
Email: rafael.parpinelli@udesc.br

***Abstract*—In the age of information, it is understood that social media provides valuable reference for many contexts, including the financial market. Although having high volume, publications on social media are not necessarily reliable. In this context, this research aims to examine the influence of financial news coming from a more transparent source, the newspaper The New York Times. This source provides fact-checked news, but the volume of information is lower compared to social media. The strategy proposes a difficult challenge, the application of a Machine Learning model on a limited dataset. The LSTM-based stock price prediction model proposed has two features, news sentiment and historical data of the assets. Experiments indicate that the model performs better when the news' sentiments are considered and demonstrates potential to accurately predict stock prices up to around 35 days into the future, comparing the results obtained with the real prices on the period.**

I. Introduction

THE formidable technology present in the current financial market provides a great quantity of indicators for the investors to guide their decisions in a more positive way. This information requires resources to be processed and analyzed, resources that are dominated by big fintechs and usually lacks in the hands of the small investors. An investor without tools faces uncertainty towards their investment decisions, due to the high volatility of the stock market.

This volatility is explained due to the fact that the stock price depends directly on the decisions taken by the companies [1], decisions that are unpredictable to some extent, supported by the high competitiveness of the market. The market requires adaptability from the investors to build a portfolio able to balance the risk and return factors. To achieve this, investors have a framework of mathematical models, which can help them building their portfolio.

The primordial attempt to abstract the responsibility of the analysis and decision-making of the investor was the development of the Portfolio Optimization Problem (POP). The fundamentals to the problem revolves around optimally allocating an amount of capital based on the historical stock prices time series, in order to maximize the portfolio returns. Several models have already been implemented to solve this problem [2].

While POP provides enough tools to analyze the current state of the market, the future is completely undefined. The results gathered using an ordinary POP model are not able to truly determine the next movements of the market. To accomplish this, the model needs to analyze not only the historical prices of the assets, but also more market indicators.

Recently, Machine Learning (ML) models have been studied and developed to take an even more ambitious step in the area of investment portfolio optimization: the stock price prediction. The ML models can be trained using a wide array of factors, including but not limited to financial news related to companies and socio-economic factors. These models are generally used to predict the stock price movement during a period of time, in order to aid the investor's decision-making process. The predominant techniques are based on supervised learning models, such as bagging, boosting ensemble classifiers, and Artificial Neural Networks (ANN) for making stock price predictions.

In this research, along with the classic indicator, being the historical data of the assets prices, the model developed is paired with the sentiment analysis of financial news related to the companies, collected from the newspaper The New York Times. The model is based on the Long Short-Term Memory (LSTM) artificial recurrent neural network, which is properly tuned with an AutoML strategy in order to verify the effect of news sentiment on the results and to predict raw stock prices on a pre-defined time window.

The literature showcases a few studies confirming the relevance of news sentiments in the stock price prediction process, but the majority of the results are only applied on a theoretical scope. These assertions should only be truly accepted if the model results are also replicated in a real investment scenario, to create evidence of the news sentiments' relevance and also investigate the longevity of the accuracy of the predictions generated by the model.

For this reason, the predictions generated are applied in a real investment scenario, using a risk/return strategy for the creation of portfolios based on the predicted stock price values, in order to evaluate the ability of the model in creating portfolios with high accuracy for a relatively long period of time. The experiments indicate the possibility of maintaining high accuracy up until around 35 days in the future when the model considers the investors' general sentiments acquired via financial news.

The remainder of this paper is organized as follows. Section 2 briefly discusses the relevancy of the study and further details on the stock price prediction approaches; Section 3 displays related work on the area; Section 4 elaborates the architecture and characteristics of the proposed model; Section 5 discusses about the validation and results of the model; and Section 6 points out the conclusions, limitations of this study and future work.

II. Background

A. Sentiment Analysis

At least in a financial context, it is hard to evaluate the efficiency of a sentiment analysis model individually. It is difficult to evaluate this component by itself due to the fact that it cannot be treated as a detached component, because the result of the sentiment analysis model is directly consumed by the prediction model. Furthermore, sentiment analysis models are usually generic, with no evidence displaying a better performance for a specific scenario, in this case, on financial news. The only factor that can truly influence the choice of a sentiment analysis model is the database selected, as there are faster models (for databases with large volume, such as Twitter) and more robust models (for databases with small volume, such as newspapers).

B. Valence Aware Dictionary for Sentiment Reasoning (VADER)

In order to workaround the lack of information, a sentiment analysis model that works well with a small database should be applied when working with a newspaper as source of news. The VADER sentiment analysis model accomplishes this need and is explored in this work, because the model does not require a large database to achieve high accuracy [3].

The creators of VADER have demonstrated the model's potential by comparing it with several other strategies. The results were better than all of them, both for an analysis of comments on social networks and for analysis of newspaper news. Although VADER is not specially designed to work in an investment market scope, the results showed great generalization. The application of the model on this paper will verify VADER's efficiency on a purely financial context.

C. Stock Price Prediction

Prediction methods generally fall into three categories: fundamental analysis, technical analysis or Machine Learning.

Technical analysis denotes the study of past prices, using charts as the main tool. This analysis assumes that market reactions to news are instantaneous and therefore does not take them into account in its attempts at predictions. The objective of technical analysis is to identify patterns in historical series, in order to anticipate changes in the market [4].

Fundamental analysis looks at indicators that affect supply and demand in the market. The idea is to collect and process the information before it reflects its consequences in the market. This in-between represents an opportunity to dispose of stocks that are about to go down or buy stocks that are about to go up. This type of analysis uses data about companies to predict market movements, with news as the main source.

Machine Learning denoted a transition to more technological and robust prediction models. It is very common for ML-based strategies to overlap the fundamental analysis concept, as the models generally use a number of indicators and/or an amount of market data. ML techniques are also useful to analyze how the stocks behave when subject to different market scenarios, in order for the investors to be able to make decisions with more confidence.

D. Long Short-Term Memory (LSTM)

Unlike sentiment analysis, the predictive model has an approach that stands out. Recent studies indicate that the LSTM network, an architecture specially designed to process data in the form of historical series, has better performance than the other identified models [5], but there is a lack of evidence to prove that the results are good enough to generate efficient portfolios scenarios with real data.

What makes the LSTM artificial neural network stand out from the other models is the ability of identifying long-term dependencies, as long as the training data is properly segmented into sub-sequences with a well-defined beginning and end [6]. This requirement is fulfilled when analyzing any kind of historical series, as any sequential subset of the series can be effectively listed as training data. The model's performance is directly affected by the quantity and size of these subsets: a model with a small number of subsets or with a window too short could have it's memorization capacity inhibited and, on the other hand, a model with a large number of subsets or with a window too large could suffer an execution bottleneck during model training.

III. Related Work

Stock price prediction is an inherently complex task due mainly to the volatility of the investment market [7]. Since the investment market presents the possibility of large returns in a relatively short period of time, naturally it draws a lot of interest. Researchers from all around have presented several different approaches to tackle the problem.

Rana, Uddin, and Mhoque [8] proposed three prediction models: Linear Regression (LR), Support Vector Regression (SVR), and LSTM. The authors compared the three models and highlighted the superiority of the LSTM model. For the LSTM model, different activation functions and optimizers were paired, reaching the conclusion that the combination that generated the best accuracy was the activation by Hyperbolic Tangent with the Adam optimizer.

Du and Tanaka-Ishii [9] created their own sentiment analysis model and applied on news extracted from the Wall Street Journal (WSJ) and the Reuters & Bloomberg database (R&B), weighting each news item in relation to its respective stock. The authors used a Multi Layer Perceptron (MLP) for the prediction model and the results were computed by an optimization model based on the classic Markowitz model [10]. The authors evaluated their strategy observing 18 selected stocks from the American market. Their model obtained better results for the R&B database than other models on the literature.

The work of Xing, Hoang, and Vo [11] shows a prediction model based on an MLP network, which is applied on the currency trading scenario. The authors attempt to predict the appreciation or not of the US Dollar and the Euro. The analysis performed by the authors is extremely complete, and by comparing their results on several investment fronts, they confirm the predictive power of a model using high frequency news without any kind of technical analysis.

Maqsood *et al* [12] proposed a Convoluted Neural Network (CNN) model to work with the 4 major stocks from the US, Hong Kong, Turkey and Pakistan. The model uses historical series of stocks and a simple sentiment analysis strategy based on the proposal of the SentiWordNet lexical resource, mapping sentiments from Twitter publications in a dictionary of 4000 words. The authors concluded that not all events impact the financial market, but the sentiment analysis implemented is too simple for this statement to be generalized.

Patil, Wu, Potika, and Orang [13] combined graph theory with CNN analyzing spatiotemporal relationships between different stocks, modeling the financial market as a graph. The model used financial indicators and news as input to predict prices for 30 US stocks. The results indicate that the application of graph theory produces better results than ordinary statistical models for time series prediction.

Jin, Yang, and Liu [14] implemented an LSTM model for predicting Apple stock prices. The model applies a type of decomposition to the historical price series, in order to simplify the sequences and make them more predictable. A CNN model was developed for binary categorization (positive, negative) of posts from a forum to make the prediction model more robust, considering the content of posts. The model showed good results, but it is not possible to generalize any results due to the fact that the experiments were performed with only one stock.

An overview of the works can be seen in Table I. The focus of the research was precisely works that combine sentiment analysis with price prediction, so most of the materials present in the table explore both premises. Differences between works in terms of study scenario, algorithms used, and evaluation metrics are also highlighted in the table. There is considerable heterogeneity in the choice of sentiment analysis model, indicating that the application scenario strongly influences the choice of model. For the prediction model, CNN and LSTM stand out, having studies with relevant and recent contributions. At the end of the table, in bold, the model proposal for this work is presented.

IV. Proposed Model

The essence of the model presented in this work is highlighted in two steps: sentiment analysis and price prediction. Both stages have different requirements and will end up working together.

The first stage, sentiment analysis, has the purpose of processing the raw news of the companies studied to define the position of investors during the period studied. The values obtained are used to create the training and validation samples necessary for the model.

The second stage, stock price predictions, has the goal of developing a prediction model intelligent enough to generalize results to all studied stocks, which represent a subset of the universe of assets on a stock exchange.

A. Database Description

The experiments are carried out with the historical series of assets that make up the Top 50 of the S&P Index, with indicators for the first quarter of 2021. The stocks considered are presented in Table II.

All the data used in this research is publicly available, the historical series of assets being extracted from Yahoo Finances and the news from the New York Times newspaper's API. The period studied starts in January 1, 2016 and ends in December 31, 2020. In the 5 years considered, a maximum of 500 news per asset were extracted, with priority for the most relevant news from the section with the theme "finance". The number 500 was selected empirically, as the majority of the companies studied did not have more than 500 news on the period considered.

The separation between training and validation data is not done randomly, a strategy commonly applied in artificial neural networks. As the data is a historical series, the values are dependent on their predecessors, for example: the price of the day d depends directly on the price of the days $d-1$, $d-2$ and so on, therefore, the separation of the sets is carried out according to the closing date. The training/validation split is given at $80/20$, culminating in the years 2016-2019 being used for training and the year 2020 being used for validation.

TABLE I
MAIN APPROACHES IDENTIFIED IN LITERATURE

Reference	Scenario	Sentiment Analysis	Prediction Model	Evaluation Metrics
[8]	Spain	-	LR, SVR, LSTM	RMSE
[9]	USA	Original	MLP	Return
[11]	Exchange	BERT	MLP	Accuracy
[12]	USA, Hong Kong, Turkey, Pakistan	SentiWordNet	CNN	RMSE, MAE
[13]	USA	-	Graphs + CNN	RMSE, MAPE, MAE
[14]	USA	CNN+word2vec	LSTM	MAE, RMSE, MAPE
This work	**USA**	**VADER**	**LSTM**	**RMSE, MAPE**

B. Portfolio Selection

To analyze the results on a real investment scenario, the predictions generated are subjected to a portfolio selection strategy. This strategy is unrelated to the model, and is only used as a performance measure.

With the predictions value at hand, it is possible to calculate the expected returns for every asset, as well as the measured risk. By calculating the risk and return, portfolios can be selected in three different ways: (I) maximizing returns; (II) minimizing risk, and (III) maximizing return/risk ratio.

Maximizing returns is the most aggressive strategy, as the risk values are not considering at any point of the portfolio selection. In the other hand, minimizing risk is the most conservative strategy. The strategy used in this work is to maximize the return to risk ratio, which aims to pick the assets producing high returns while offering a somewhat low risk. In order to add security to the portfolio, it is good practice to dilute the investment in a number of assets. This number represents the portfolio's cardinality (k). There is not a set in stone to find the value for k, but there is evidence that k should always be at least 3 [15], which is the value for the cardinality in this research.

C. Model Operation

In the sentiment analysis stage, the news of each asset were individually submitted to the VADER model, resulting in a sentiment value calculated for every news. This value is in the $[-1, 1]$ interval, with $s = -1$ representing the most negative sentiment possible, 1 representing the most positive sentiment possible and 0 representing a completely neutral sentiment.

As 500 news are considered for every year, some days will necessarily have more than one news for an asset. In these cases, the average of the news' sentiments is calculated for those specific days. There is also the possibility of a day having no news for an asset. In these cases, the neutral value $s = 0$ is considered.

The product of this stage is the calculation of sentiment values for each closing day for each analyzed asset. These values are fundamental for the creation of training and validation samples on the next step. The samples have the sentiment values and historical stock prices for every asset.

In the price prediction stage, the LSTM model is implemented. An ordinary LSTM model consists of three layers: a regular input layer, an LSTM layer of size n, and a dense layer with a single node, responsible for the consolidation of the LSTM layer output into a palpable prediction value. The model proposed in this work has an additional LSTM layer. Stacking LSTM layers has an interesting trade-off: it allows the model to represent more complex patterns, but increases the computational cost of every epoch. The architecture was defined empirically.

TABLE II
TICKERS CONSIDERED

AAPL	ABBV	ABT	ACN	ADBE
AMZN	AVGO	BAC	BRK-B	CMCSA
COST	CRM	CSCO	CVX	DHR
DIS	FB	GOOG	GOOGL	HD
INTC	JNJ	JPM	KO	LLY
MA	MCD	MDT	MRK	MSFT
NEE	NFLX	NKE	NVDA	ORCL
PEP	PFE	PG	PM	PYPL
T	TMO	TSLA	TXN	UNH
V	VZ	WFC	WMT	XOM

D. Parameter Tuning

One of the most important and difficult to tackle problems when working with ML algorithms is parameter tuning. An ordinary ML algorithm has a set of adjustable parameters, a fact that is also common to the model in this work. In the design of any ANN, the number of layers must be adjusted, the size of each layer, which training algorithm will be used, as well as other parameters.

The fact is that in the face of an universe of adjustable parameters, an infinite number of configurations can be established. Training a ML model until it reaches convergence is a time consuming process, so training a large number of configurations is completely unfeasible. There are strategies to reduce the amount of configurations to be trained, such as empirical experiments and sub-divisions of the search space.

There is also the possibility of applying Auto-ML, algorithms that configure the parameters of a model automatically. These algorithms abstract much of the manual decision-making during the adjustment, but it is interesting that at least some empirical testing is done beforehand to define the intervals where the algorithm should search. In this work, the *Hyperband* algorithm from the *KerasTuner* library is applied to perform the parameter adjustment.

The empiric tests performed beforehand indicates that the model's architecture should have no more than four layers, two of them being hidden layers and the remaining two being the input and output layers [16]. The hyperparameters subjected to tuning are:

- Number of nodes in the LSTM layers: minimum 16 nodes, maximum 128 nodes, with a step of 16 nodes;
- Learning rate: values of 10^{-2}, 10^{-3}, and 10^{-4};
- Window size: minimum 30 days, maximum 120 days, with a step of 30 days.

This generates a set of $8 * 8 * 3 * 4 = 768$ combinations of configurations, a number rather large of models needed to be tuned. Hyperband's ability of computing models with early stopping and remarkable speedup prove to be almost a necessity on a scenario with this many possible configurations.

V. EXPERIMENTS & RESULTS

The general idea of the experiments on this paper are summarized to four points: (I) parameter tuning of the proposed model; (II) analysis of financial news influence on the prediction model; (III) generation of stock price predictions for a set time window, and (IV) application of the generated predictions on a real investment scenario.

The model was implemented using the Python language and all experiments were carried out in a controlled environment, in equipment with the following specifications: Intel i7 Core™ i7-4770 processor operating at 3.9 GHz, with 16 Gigabytes of RAM and running a GNU/Linux operational system with kernel 4.8.10.

A. Parameter Tuning

Before submitting the model to any experiments, the calculation of the news' sentiments was performed with the VADER model, using the implementation exposed by the creators in [3]. This stage is not necessarily a part of the LSTM model itself, but is necessary to create the input data.

With the sentiments calculated and paired with the historical series of the stocks, the LSTM model has all the data necessary for it to be fitted. The execution hyperparameters are refined using the strategy defined on Section IV-D. This is performed for both the model without sentiment values as a feature and with. Hyperband was executed with a factor of $f = 2$ and a total budget of $e = 10000$ epochs for a total of 50 times and the results were collected.

1) Results for the Model without Sentiments: The five best configurations generated for the model without sentiments are presented on Table III, from best (1) to worst (5), based on their Mean Absolute Percentage Error (MAPE) values, a performance metric which measures the model accuracy as a ratio. The model was fitted for 100 epochs using these configurations 50 times each, having the value for MAPE calculated. The mean of the results for the 50 trials of each configuration is shown on Fig. 1.

TABLE III
BEST CONFIGURATIONS GENERATED BY HYPERBAND FOR THE MODEL WITHOUT SENTIMENTS

Label	Layer #1	Layer #2	Learning Rate	Window Size
1	64	48	10^{-2}	60
2	48	48	10^{-2}	60
3	32	48	10^{-2}	90
4	48	32	10^{-2}	120
5	48	48	10^{-2}	30

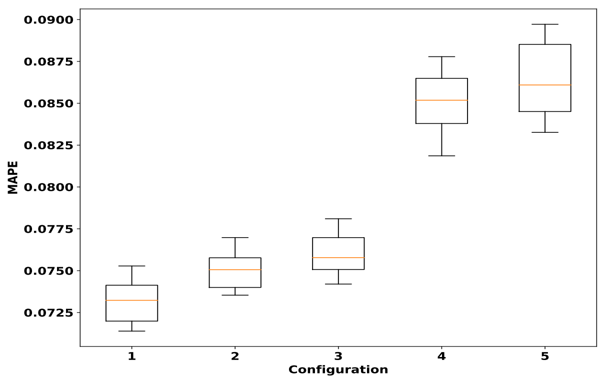

Fig. 1. MAPE for the 5 Best Configurations

TABLE IV
ONE-FACTOR ANOVA FOR CONFIGURATIONS 1 AND 2

Source	SS	df	MS	F	p-value
Treatment	0.0002	2	0.0001	83.2989	1.1102×10^{-16}
Error	0.0002	147	0		
Total	0.0004	149			

Configurations (1), (2), and (3) stand out from (4) and (5), specially configuration (1), which has the best results for MAPE. An one-factor ANOVA test is applied on the samples for configurations (1), (2), and (3) to verify if the difference between samples are statistically significant. Table IV presents the results, which suggests that at least one treatment is significantly different with a significance level of $\alpha = 0.05$. To identify the difference in between each treatment, a post-hoc test such as the Tukey HSD is indicated on this situation.

To run the Tukey test based on the $k = 3$ treatments, $df = 147$ degrees of freedom for the error and significance levels $\alpha = 0.01$ and $\alpha = 0.05$, the critical values $Q^{k=3,df=147}_{\alpha=0.01} = 4.1850$ and $Q^{k=3,df=147}_{\alpha=0.05} = 3.3487$ are obtained, respectively. To find the value for the Tukey HSD Q statistic, the Equations 1 and 2 are calculated.

$$Q_{i,j} = \frac{|\bar{x_i} - \bar{x_j}|}{s_{i,j}} \tag{1}$$

$$s_{i,j} = \frac{\sigma_\epsilon}{\sqrt{H_{i,j}}} \tag{2}$$

$H_{i,j}$ is the harmonic mean of the observations from configurations (i) and (j). σ_ϵ is the square root of the mean squared error calculated on the ANOVA test precursor of the Tukey test. The results, shown in Table V, assert the ANOVA test by confirming statistical difference for every pair of configurations. Therefore, configuration (1) is proven to be statistically better than configurations (2) and (3) and is used for the remaining experiments.

2) Results for the Model with Sentiments: Similarly, the experiment is performed for the model with sentiments. The five best configurations generated are presented on Table VI, from best (1) to worst (5). MAPE is calculated on the same way and is shown on Fig. 2.

It looks like the best configuration outperforms the remaining configurations by quite a margin, even the runner-up. Both the minimum value and the median are better for configuration (1). Selecting configurations (1) and (2), the one-factor ANOVA test is applied to verify if the difference

TABLE V
TUKEY HSD FOR CONFIGURATIONS (1) AND (2)

Pair	Tukey HSD Q	p-value	Inference
(1), (2)	11.7179	$p < 10^{-3}$	Significant
(1), (3)	17.9798	$p < 10^{-3}$	Significant
(2), (3)	6.2618	$p < 10^{-3}$	Significant

TABLE VI
BEST CONFIGURATIONS GENERATED BY HYPERBAND FOR THE MODEL WITH SENTIMENTS

Label	Layer #1	Layer #2	Learning Rate	Window Size
1	48	48	10^{-2}	60
2	48	64	10^{-2}	60
3	16	48	10^{-2}	90
4	48	16	10^{-2}	120
5	16	48	10^{-2}	30

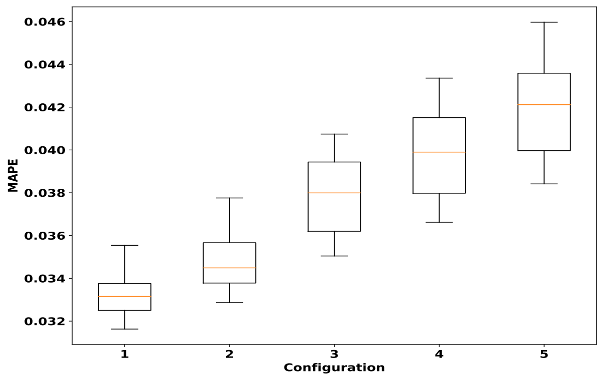

Fig. 2. MAPE for the 5 Best Configurations

between the samples are statistically significant. Table VII shows the test results, which suggests the treatments are indeed significantly different with a significance level $\alpha = 0.05$.

With the data acquired on the ANOVA test, a Tukey test was carried out to confirm the difference between the pair of configurations (1) and (2). For the test with $k = 2$ treatments, $df = 98$ degrees of freedom for the error and significance levels $\alpha = 0.01$ and $\alpha = 0.05$, the critical values $Q_{\alpha=0.01}^{k=2,df=98} = 3.7150$ and $Q_{\alpha=0.05}^{k=2,df=98} = 2.8065$ are obtained, respectively. With the values at hand, the confidence limits for the pair of configurations are set and the Tukey HSD Q are calculated, using Equations 1 and 2.

The final result is shown on Table VIII. The tests confirms the hypothesis from ANOVA, reassuring that configurations (1) and (2) are statistically different. Therefore, configuration (1) is proven to be statistically better than configuration (2) and is used for the remaining experiments.

B. Analyzing Financial News Influence

The first tests after establishing the tuning of the model aims to investigate the influence of applying news sentiments to historical series to make predictions. The model, with

TABLE VII
ONE-FACTOR ANOVA FOR CONFIGURATIONS 1 AND 2

Source	SS	df	MS	F	p-value
Treatment	0.0001	1	0.0001	34.8378	5.1488×10^{-8}
Error	0.0002	98	0		
Total	0.0002	99			

TABLE VIII
TUKEY HSD FOR CONFIGURATIONS (1) AND (2)

Pair	Tukey HSD Q	p-value	Inference
(1), (2)	8.3472	$p < 10^{-3}$	Significant

tuning referring to the best configurations without and with sentiments identified on Section V-A, was executed 50 times without the sentiment attribute and 50 times with the sentiment attribute, on the dataset containing the 5 years of series historical and news stories. Root Mean Squared Error (RMSE) was calculated at the end of each execution for every ticker, as well as the average of the normalized RMSE, represented by RMSE-N. The objective of normalizing the RMSE is to mitigate the discrepancies generated by the gross share price: a company with more expensive shares has a higher RMSE than a company with cheaper shares, even in scenarios where the proportional error is smaller. The average RMSE-N values calculated for each asset are shown in the Table IX.

The results show superiority of the model using sentiments as a feature, having a better performance for every ticker analyzed. The RMSE-N values are similar for every ticker, indicating the capabilities of generalization from the model. To confirm the results on Table IX, on Fig. 3 the boxplots referent to the RMSE-N values for the 50 tickers are presented. The mean of the RMSE-N is approximately three times higher for the model without sentiments as a feature, certifying the efficiency of the application of news' sentiments.

Furthermore, to validate the hyperparameters tuning, the average of training and validation losses curves for the 50 runs of the model with sentiments feature are displayed on Fig. 4. Results show both curves decaying to a certain point of stabil-

TABLE IX
MODELS COMPARISON (WITH AND WITHOUT SENTIMENT FEATURE)

	RMSE-N			RMSE-N	
Ticker	Without	With	Ticker	Without	With
AAPL	0.16217	0.03641	MA	0.17366	0.05188
ABBV	0.09216	0.04734	MCD	0.08461	0.04920
ABT	0.14920	0.05156	MDT	0.06753	0.05126
ACN	0.15730	0.04010	MRK	0.18300	0.06483
ADBE	0.23952	0.04764	MSFT	0.21505	0.04691
AMZN	0.17517	0.05637	NEE	0.17395	0.05358
AVGO	0.07489	0.03368	NFLX	0.20258	0.05361
BAC	0.18811	0.05210	NKE	0.08748	0.03864
BRK-B	0.19654	0.06785	NVDA	0.16485	0.04028
CMCSA	0.11541	0.04631	ORCL	0.10840	0.04493
COST	0.32205	0.04946	PEP	0.09034	0.05540
CRM	0.17471	0.04560	PFE	0.05463	0.05287
CSCO	0.13646	0.06998	PG	0.18623	0.05203
CVX	0.16035	0.05146	PM	0.33774	0.05437
DHR	0.17426	0.03413	PYPL	0.13417	0.03225
DIS	0.04160	0.03896	T	0.11346	0.05044
FB	0.10427	0.05738	TMO	0.19514	0.03491
GOOG	0.16468	0.04464	TSLA	0.09386	0.06502
GOOGL	0.17081	0.04911	TXN	0.21595	0.05035
HD	0.08458	0.04226	UNH	0.15182	0.04465
INTC	0.25737	0.06388	V	0.18331	0.06031
JNJ	0.11604	0.07057	VZ	0.12535	0.07817
JPM	0.19586	0.05880	WFC	0.13376	0.03780
KO	0.05499	0.06043	WMT	0.24476	0.05127
LLY	0.23789	0.11573	XOM	0.21062	0.04595

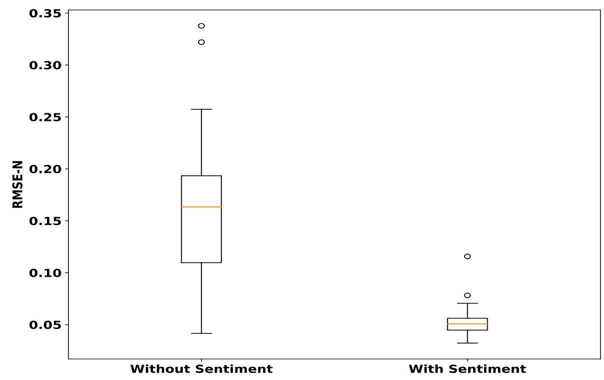

Fig. 3. RMSE-N for the 50 Stocks for the Model with and without Sentiment

ity, with a small gap between them. The first fact indicates that the model was trained enough to generalize well with the data input, and continuing the training would eventually culminate in over-fitting. The second fact is expected, since the model's loss for training should be lower than the loss for validation. The results indicate that the model is well trained.

C. Predicting Stock Prices into the Future

The main objective of the model is predicting stock prices for the 50 tickers analyzed. As explained before, the LSTM model works with a window size to represent the memorization capacity. This is crucial to understand why is it so difficult to predict prices into the future: the model has to operate with limited points of real data.

In this experiment, the model is used to predict the stock prices for the next 50 days for every ticker. The RMSE-N of the predictions generated are listed in Table X. As expected, the RMSE-N values are noticeably higher for the predictions, since the model works with data not used during the training process. Naturally, creating the predictions outside of the training scope is the biggest challenge for the model, and a decrease in accuracy is expected as the predictions distance themselves from the training scope, in other words, the further into the future, the worse the quality of predictions.

TABLE X
NORMALIZED RMSE COMPARISON FOR TRAINING AND PREDICTIONS

	RMSE-N			RMSE-N	
Ticker	Training	Predictions	Ticker	Training	Predictions
AAPL	0.03641	0.43611	MA	0.05188	0.30879
ABBV	0.04734	0.94908	MCD	0.04920	0.52883
ABT	0.05156	0.85740	MDT	0.05126	0.48973
ACN	0.04010	0.71989	MRK	0.06483	0.70439
ADBE	0.04764	0.35472	MSFT	0.04691	0.35621
AMZN	0.05637	1.15072	NEE	0.05358	0.39933
AVGO	0.03368	1.65071	NFLX	0.05361	0.97983
BAC	0.05210	0.44360	NKE	0.03864	1.40680
BRK-B	0.06785	0.5726	NVDA	0.04028	0.95906
CMCSA	0.04631	0.55155	ORCL	0.04493	0.38606
COST	0.04946	0.31010	PEP	0.05540	0.79192
CRM	0.04560	0.39452	PFE	0.05287	0.87950
CSCO	0.06998	0.32509	PG	0.05203	0.48418
CVX	0.05146	0.24973	PM	0.05437	0.41451
DHR	0.03413	0.78150	PYPL	0.03225	1.11321
DIS	0.03896	1.07256	T	0.05044	0.64139
FB	0.05738	1.00459	TMO	0.03491	0.56151
GOOG	0.04464	0.79797	TSLA	0.06502	1.65771
GOOGL	0.04911	0.78201	TXN	0.05035	0.91743
HD	0.04226	0.59678	UNH	0.04465	0.68341
INTC	0.06388	0.55860	V	0.06031	0.25727
JNJ	0.07057	0.87682	VZ	0.07817	1.11059
JPM	0.05880	0.56786	WFC	0.03780	0.24410
KO	0.06043	1.39544	WMT	0.05127	0.25203
LLY	0.11573	0.84523	XOM	0.04595	0.38433

D. Investigating the Predictions on a Real Investment Scenario

The final experiment has the goal of moving the prediction results from a theoretical standpoint to a practical analysis. Using the predictions generated, a portfolio is setup and compared to a baseline, being the S&P 500 index. The best way to generate this portfolio would be using the RMSE of the predictions, but on a realistic scenario the RMSE is impossible to calculate, due to the real values being unknown.

The solution is to pick the portfolio based on the calculated risk of the predictions generated, as explained on Section IV-B. The portfolio selected has a cardinality value of $k = 3$, in order to create a minimum degree of diversification, consisting of the three stocks with the smallest values of CVaR, being the tickers WFC, WMT, and V. Fig. 5 display the predicted and real performance of the portfolio, assuming an equal portion of investment in each stock being $1/3$ of the total capital. Alongside, the performance on the S&P Index is also exhibited as a baseline measure.

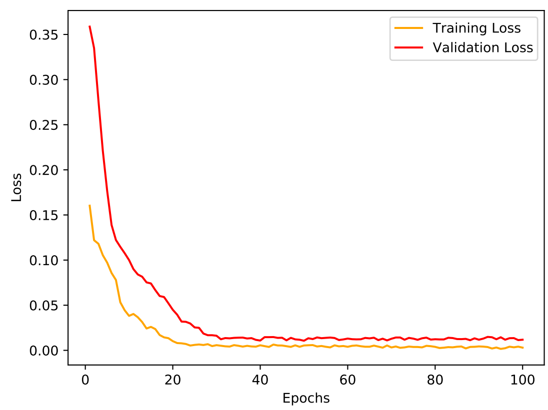

Fig. 4. Training and Validation Losses

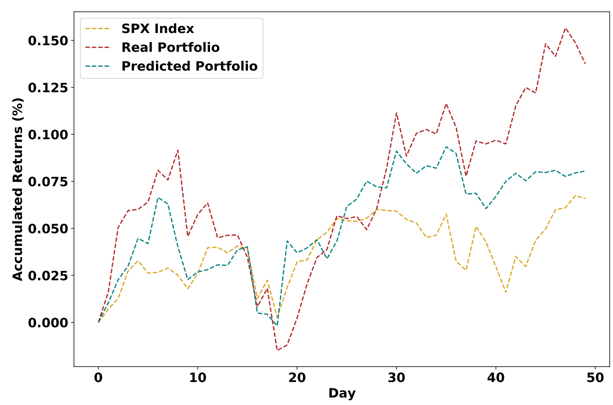

Fig. 5. Portfolio Performance & Comparison

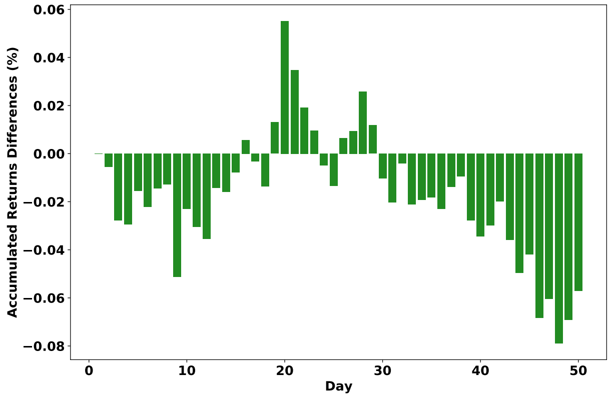

Fig. 6. Accumulated Returns Difference for Predicted and Real Portfolios

Many insights can be extracted from this experiment. The first positive fact is both the real and predicted portfolios have a better accumulated return value than the baseline compared. On the other hand, the final predicted value is a bit distant than the real value. It's noticeable that the model's predictions follow the real values really well up until the mark of 35 days, but the accuracy collapses after this point. It is understandable and expected that the accuracy would decrease as the days passes, and the breakpoint in this model seem to be around day 35. Fig. 6 displays the accumulated returns difference for predicted and real portfolios to show a less visual, more mathematical perspective for the analysis. The model objective is to keep the difference as close as possible to zero, and it is noticeable that the difference rapidly rises beyond day 35.

VI. Conclusion and Future Work

The prediction of asset prices in the financial market proves to be an ambitious and complicated task, due to the market being affected by many different factors. In this paper, one of these factors, financial news associated to the companies, was aggregated to the classic analysis of the historical series of stock prices. The model proposed presents two stages, beginning with the sentiment analysis of the news collected from the New York Times newspaper performed by VADER and ending on a stock price prediction model based on the LSTM architecture. The model also showcased the use of a robust parameter tuning strategy, being *AutoKeras' Hyperband*.

Pairing financial news sentiments with the assets historical prices gives more abstraction power to the prediction model, especially one with a dominant recurrence strategy such as LSTM. The model itself has no limitations for the application of even more features, although further investigation would be required to not only gather a new and improved dataset, but for the creation of the input samples as well.

The experiments included the tickers composing the Top 50 of the S&P 500 index, on a period of 5 years. The results indicate that the model is able to predict prices with good accuracy within a time window of around 35 days, given the scenario in which the model was applied and it's configuration. The results were validated by simulating the construction of a portfolio using the values for the three tickers with best return/risk ratio and comparing with the real values for the same portfolio. The simulation sustained the results, displaying a very similar performance for the predicted and real portfolios up until around day 35. The predicted portfolio also displayed better performance than the SPX index, which is relevant for the possibility of using the model on the real investment market.

Even after a display of good results, there is room for improvement. The five year period studied is enough for demonstration, but a realistic investigation could use a longer period to generate more data points for model training. With more training samples, the prediction could be attempted into a more distant future, while still maintaining good accuracy. Another possibility is focusing on short-time instead, a strategy used by day traders.

References

[1] E. F. Fama, "Efficient Capital Markets: II", *The journal of finance*, vol. 46, 1991, pp. 1575–1617.
[2] K. Liagkouras, K. Metaxiotis, "Efficient Portfolio Construction with the Use of Multiobjective Evolutionary Algorithms: Best Practices and Performance Metrics", *International Journal of Information Technology & Decision Making*, vol. 14, 2015, pp. 535–564, DOI: 10.1142/S0219622015300013.
[3] C. Hutto, E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text", *International AAAI Conference on Web and Social Media*, vol. 8, 2014.
[4] R. Schumaker, H. Chen. "Textual Analysis of Stock Market Prediction Using Financial News Articles", *AMCIS Proceedings*, 2006, p. 185.
[5] M. Roondiwala, H. Patel, S. Varma, "Predicting Stock Prices Using LSTM", *International Journal of Science and Research (IJSR)*, vol. 6, 2017, pp. 1754–1756.
[6] F. A. Gers, J. Schmidhuber, F. Cummins, "Learning to Forget: Continual Prediction with LSTM", *Neural Computation*, vol. 12, 2000, pp. 2451–2471, DOI: 10.1162/089976600300015015.
[7] K. Adam, A. Marcet, J. P. Nicolini, "Stock Market Volatility and Learning", *The Journal of Finance*, vol. 71, 2016, pp 33–82, DOI: 10.1111/jofi.12364.
[8] M. Rana, M. M. Uddin, M. M. Mhoque. "Effects of Activation Functions and Optimizers on Stock Price Prediction Using LSTM Recurrent Networks", *International Conference on Computer Science and Artificial Intelligence*, vol. 3, 2019, pp. 354–358, DOI: 10.1145/3374587.3374622.
[9] X. Du, K. Tanaka-Ishii, "Stock Embeddings Acquired From News Articles and Price History, and an Application to Portfolio Optimization", *Annual Meeting of the Association for Computational Linguistics*, vol. 58, 2020, pp. 3353–3363, DOI: 10.18653/v1/2020.acl-main.307.
[10] H. Markowitz, "Portfolio Selection", *The Journal of Finance*, vol. 7, 1952, pp. 77–91.
[11] F. Xing, D. H. Hoang, D.-V. Vo, "High-frequency News Sentiment and its Application to Forex Market Prediction", *Hawaii International Conference on System Sciences*, vol. 54, 2020.
[12] H. Maqsood *et al.* "A Local and Global Event Sentiment Based Efficient Stock Exchange Forecasting Using Deep Learning", *International Journal of Information Management*, vol. 50, 2020, pp. 432–451, DOI: 10.1016/j.ijinfomgt.2019.07.011.
[13] P. Patil, C. S. M. Wu, K. Potika, M. Orang, "Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models", *International Conference on Software Engineering and Information Management*, vol. 3, 2020, pp. 85–92, DOI: 10.1145/3378936.3378972.
[14] Z. Jin, Y. Yang, Y. Liu, "Stock Closing Price Prediction Based on Sentiment Analysis and LSTM", *Neural Computing and Applications*, vol. 32, 2020, pp. 9713–9729, DOI: 10.1007/s00521-019-04504-2.
[15] R. Cheng, J. Gao. "On Cardinality Constrained Mean-CVaR Portfolio Optimization", *27th Chinese Control and Decision Conference*, 2015, p. 1074–1079, DOI: 10.1109/CCDC.2015.7162076.
[16] A. Heiden, R. Parpinelli. "Applying LSTM for Stock Price Prediction with Sentiment Analysis", *15th Brazilian Congress of Computational Intelligence*, 2021, p. 1–8, DOI: 10.21528/CBIC2021-45.