SANTA CATARINA STATE UNIVERSITY – UDESC
CENTER OF TECHNOLOGICAL SCIENCES – CCT
GRADUATE PROGRAM IN APPLIED COMPUTING – PPGCA

ALEXANDRE HEIDEN

STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY WITH SENTIMENT ANALYSIS

JOINVILLE
2022

ALEXANDRE HEIDEN

STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY WITH SENTIMENT ANALYSIS

Master thesis submitted to the Computer Science Department at the College of Technological Science of Santa Catarina State University in fulfilment of the partial requirement for the Master's degree in Applied Computing.

Advisor: Rafael Stubs Parpinelli

JOINVILLE
2022

Heiden, Alexandre
Stock Price Prediction Using Long Short-term Memory With Sentiment Analysis / Alexandre Heiden. - Joinville, 2022.
68 p. : il. ; 30 cm.

Advisor: Rafael Stubs Parpinelli.
.
Thesis (Master) - Santa Catarina State University, Center of Technological Sciences, Graduate Program in Applied Computing, Joinville, 2022.

1. Stock price prediction. 2. Machine Learning. 3. Long Short-Term Memory. 4. Sentiment analysis. 5. Newspaper articles. I. Parpinelli, Rafael Stubs . II. Santa Catarina State University, Center of Technological Sciences, Graduate Program in Applied Computing. III. Title.

ALEXANDRE HEIDEN

STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY WITH SENTIMENT ANALYSIS

Master thesis submitted to the Computer Science Department at the College of Technological Science of Santa Catarina State University in fulfilment of the partial requirement for the Master's degree in Applied Computing.

Advisor: Rafael Stubs Parpinelli

EXAMINATION BOARD:

Prof. Dr. Rafael Stubs Parpinelli
CCT/UDESC

Members:

Prof. Dr. Danton Diego Ferreira
UFLA/MG

Prof. Dr. Fabiano Baldo
CCT/UDESC

Joinville, July 22nd, 2022

ABSTRACT

Financial news has proven to be a valuable source of information for assessing stock market volatility. Most of the attention has been given to social media platforms, while news from vehicles such as newspapers are not as widely explored. Newspapers provide, albeit in a smaller volume, more reliable information than social media platforms. In this context, the present research aims to examine the influence of financial news on the stock price prediction problem, using the VADER sentiment analysis model to process news and feed sentiments as one of the features in a stock price prediction model based in LSTM, along with historical asset data. The hyperparameters of the model in question were studied and refined using the KerasTuner library. Lastly, the model was used to generate predictions outside its training scope, in order to verify the longevity of the prediction accuracy. The experiments indicate that the model presents better results when news sentiments are considered, presenting lower values for the MAPE error metric with statistical significance. The model demonstrates the potential to accurately predict the prices of 50 stocks in the *Top 50* of the S&P 500 index up to about 35 days into the future, asserted by comparing the expected returns between the predictions generated and the real values of the portfolio in a real investment environment.

Keywords: Stock price prediction. Machine Learning. Long Short-Term Memory. Sentiment analysis. Newspaper articles.

RESUMO

As notícias financeiras provaram ser uma fonte valiosa de informação para a avaliação da volatilidade do mercado de ações. A maior parte da atenção tem sido dada às plataformas de mídia social, enquanto as notícias de veículos como jornais não são tão amplamente exploradas. Os jornais fornecem, embora em um volume menor, informações mais confiáveis do que as plataformas de mídia social. Nesse contexto, a presente pesquisa visa examinar a influência das notícias financeiras no problema de previsão de preços de ações, usando o modelo de análise de sentimentos VADER para processar as notícias e alimentar os sentimentos como um dos atributos em um modelo de previsão de cotação de ações baseado em LSTM, junto com os dados históricos dos ativos. Os hiperparâmetros do modelo em questão foram estudados e refinados utilizando a biblioteca KerasTuner. Finalmente, o modelo foi utilizado para gerar previsões fora do seu escopo de treinamento, visando averiguar a longevidade da acurácia das previsões. Os experimentos indicam que o modelo apresenta melhores resultados quando os sentimentos das notícias são considerados, apresentando valores menores para a métrica de erro MAPE com significância estatística. O modelo demonstra potencial para prever com precisão os preços das 50 ações no *Top 50* do índice S&P 500 em até cerca de 35 dias no futuro, conforme averiguado ao comparar os retornos esperados entre as previsões geradas e os valores reais do portfólio em ambiente real de investimentos.

Palavras-chave: Previsão de cotações. *Machine Learning*. *Long Short-Term Memory*. Análise de sentimentos. Notícias de jornal.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANBIMA	Brazilian Association of Financial and Capital Market Entities
ANN	Artificial Neural Networks
ANOVA	Analysis of Variance
ATA	Automated Text Analysis
CNN	Convolutional Neural Network
CVaR	Continuous Value-at-Risk
EMH	Efficient Market Hypothesis
ETF	Exchange Traded Funds
GRU	Gated Recurrent Unit
LR	Linear Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NYSE	New York Stock Exchange
POP	Portfolio Optimisation Problem
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SOFNN	Self-Organizing Fuzzy Neural Network
SVM	Support Vector Machine
SVR	Support Vector Regression
VADER	Valence Aware Dictionary for sEntiment Reasoning

CONTENTS

1 INTRODUCTION

The financial market is an environment for trading assets, which are divided into monetary (currency and exchange), public and private bonds, commodities and shares. The stock market is the most popular among retail investors, a term coined to represent individual small investors, due to its high earning potential.

Shares are securities corresponding to a stock in the ownership of a company, so the shareholder of a given company has a right to assets and profit sharing (HANAOKA, 2014). When an investor acquires ownership of a set of shares from one or more companies, they become the owner of an investment portfolio.

Although the stock market is profitable, it also proves to be highly volatile, that is, fickle to the point that a stock can undergo sudden movements at any time. This behaviour is due to the fact that the stock price depends directly on the decisions taken by the companies (FAMA, 1991), decisions that are unpredictable to a certain extent, corroborated by the high competitiveness of the market. Therefore, the investor's objective is to build a portfolio that manages to balance the risk and return factors (CÂMARA et al., 2014). For this, the investor has a framework of mathematical models, which help him to build his portfolio.

1.1 PROBLEM

In order to help the investor's analysis and decision making, the Portfolio Optimisation Problem (POP) was created. The classic approach to the problem consists of optimally allocating investor capital based on the historical series of assets. This construction is relatively famous, and several models have already been implemented to solve this problem (LIAGKOURAS; METAXIOTIS, 2015).

Recently, models based on Machine Learning (ML) strategies have been widely studied to take an even more ambitious step in the area of investment portfolio optimisation: price predicting. Building a neural network and inserting historical asset data as a feature presents the potential for predicting stock prices with high accuracy (ROONDIWALA; PATEL; VARMA, 2017).

However, analysing only the historical series of prices is somewhat superficial, as there are multiple other factors that influence the price of an asset (BLICHFELDT; ESKEROD, 2008). Recent studies indicate that one of the main factors responsible for the variation of stock prices on the stock exchange are the news related to the respective owner companies (XU; COHEN, 2018), (SONG; LIU; YANG, 2017), (XING; HOANG; VO, 2020). Adding more elements to the analysis is of vital importance for the construction of a balanced portfolio, and the analysis of news sentiments shows positive results even during the COVID-19 pandemic (VALLE-CRUZ et al., 2021), a fact that destabilised the economy on a global scale and made the equity investment environment absolutely unstable.

1.1.1 Problem Characterisation

The scope of this work seeks to build the prediction of stock prices for two investment fronts: short and long term. The concept of short and long term can be interpreted differently from investor to investor, so it will be explored in a flexible way. Conceiving a satisfactory prediction allows the investor to have full control of the decision making regarding his portfolio, whether to buy, sell or hold shares.

Over the years, researchers have investigated the most varied methods to try to predict stock prices, such as linear programming models, statistical models and models based on ML strategies. This work proposes the use of ANN for the elaboration of the prediction model, more specifically the LSTM network.

1.1.2 Limitations

A price prediction model will never be able to produce results with 100% accuracy. There are many factors that influence the price of a stock and, in addition, the investment market environment is subject to completely unexpected factors, such as the COVID-19 pandemic. Even if there was a mathematical function that defines the investment market, this function would be so complex and would reside in a space with such high dimensionality that it would be impossible to approximate it.

1.2 MOTIVATION

The Brazilian Association of Financial and Capital Market Entities (ANBIMA) publishes an annual bulletin with information on the financial volume of private and retail. This report indicates that the investment market aimed at individuals grew, in the year of 2020, 13.4% in relation to the year of 2019, characterising the biggest annual variation since the beginning of the monitoring in 2014.

Therefore, there is a significant growth in the interest of retail investors in the stock market. Although there are financial technology companies, called Fintechs, that present portfolio management methods for these small investors, these services are usually quite expensive. Therefore, small investors are increasingly looking for their own investment strategies (CHENG et al., 2021), reducing dependence on a Fintech or a bank in their market applications.

However, the small investor has a significantly smaller amount of tools than a large financial institution. Leaving all decision-making in the hands of the investor can become oppressive for him, due to the massive amount of indicators in the market and factors that contribute to the volatility of asset prices.

The solution to the problem is to abstract the difficult decision making using computational intelligence, through ML strategies. Machines have much greater processing power than an investor alone, being able to absorb a large amount of information such as news from companies in the financial sector and quotations of their respective assets. The minimisation of human

interaction, in addition to improving decision-making, also leads to less conflict of interests and, on certain occasions, better market efficiency (BEKETOV; LEHMANN; WITTKE, 2018).

1.3 RESEARCH PROBLEM

The literature showcases 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. There is evidence of the results of stock price prediction models being improved by sentiment analysis using colloquial information sources, mainly social media (XU; COHEN, 2018) and investment forums (NGUYEN; SHIRAI; VELCIN, 2015) and also using formal information sources such as financial journals (DU; TANAKA-ISHII, 2020).

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.

1.4 OBJECTIVE

1.4.1 Main Objective

This dissertation has the general objective of improving the prediction of stock market movements with the help of financial news in a real investment environment, through the development of a stock price prediction strategy using the LSTM model, an ANN designed especially to work with forecasting of historical series in general.

1.4.2 Specific Objectives

Based on the main objective presented, the following specific objectives are proposed:

- Conduct a bibliographic review about POP, focusing on stock price prediction using *Machine Learning*;
- Develop a stock price prediction model based on the LSTM neural network with two indicators: historical series of prices and sentiment of news related to the investment market;
- Application of a framework for interpreting news and calculating sentiments, used in the prediction model;
- Perform a set of experiments, aiming to refine the adjustment of model parameters;
- Analyse the performance of the model in making predictions, investigating the maximum predictability time with acceptable error;

- Investigate the longevity of the accuracy of the predictions generated by the model, applying the results in a real investment scenario.

1.5 METHODOLOGY

The research is of primary nature because it presents the fulfilment of experimentation in order to clarify the research questions, in relation to the use of news as an indicator in the predicting process and the period in which the predictions have satisfactory accuracy in a real application scenario.

Regarding the objectives, the research has an explanatory character, as the analysis of results also seeks to explain the reason for the observed behaviour.

The work also fits as experimental research. The experiments are guided in a real investment scenario, and will aim to verify the importance of the news sentiment factor in relation to the accuracy of the predictions made, with the measurement of statistical measures to confirm the results. Furthermore, the final part of the experimentation consists of investigating the longevity of the accuracy of the predictions, also in a real investment scenario. By subjecting the predictions generated to a portfolio selection strategy, it's possible to compare the results with the performance of the real portfolio in the market and verify the consistency of the results.

1.5.1 Methodological Procedure

In order to identify the most successful implementations in the literature, a bibliographic review is carried out on the POP with emphasis on the theme of stock price prediction. In this phase, the possibility of implementing a more complete method is evaluated, with the analysis of one more indicator in addition to the historical series of prices: the sentiment acquired through the processing of newspaper news.

In an attempt to make predictions with the smallest possible error, a two-step model is developed based on the Valence Aware Dictionary for sEntiment Reasoning (VADER) framework for sentiment analysis and on the LSTM artificial recurrent neural network for price prediction. The application of the LSTM model ensures the resolution of the problem of long dependencies presented in prediction problems on historical series in general.

According to the literature review carried out, there is no application scenario common to the works found in the literature, so the analysis is performed using data from renowned institutions: the news will be collected from the New York Times newspaper and the assets analysed are related to companies that represent the Top 50 of the S&P index of the New York Stock Exchange (NYSE).

In order to evaluate the performance of the developed model, a set of experiments is carried out. The first experiments aim to perform the adjustment of the model's hyperparameters, using the framework KerasTuner. Once tuned, the model is subjected to tests to evaluate performance based on two metrics: Root Mean Squared Error (RMSE) and Mean

Absolute Percentage Error (MAPE). Lastly, the model is used to generate the prediction of stock prices in a scenario outside the training scope, predictions that will also be submitted to a portfolio selection strategy to verify their reliability and performance.

1.6 LIST OF PUBLICATIONS

- HEIDEN, Alexandre; PARPINELLI, Rafael. Applying LSTM for stock price prediction with sentiment analysis. In: **Anais do 15 Congresso Brasileiro de Inteligência Computacional**. Joinville, SC: SBIC, 2021. p. 1–8.
- HEIDEN. Alexandre; PARPINELLI, Rafael. Financial News Effect Analysis on Stock Price Prediction Using a Stacked LSTM Model. Accepted to **FedCSIS**, 2022.

1.7 DOCUMENT'S ORGANISATION

The work is organised in the following way: Chapter 2 presents the theoretical background necessary to discuss the work. Chapter 3 presents related works. Chapter 4 details the proposed model. Chapter 5 sets out the methodology and results of experimentation. Lastly, in Chapter 6, the final considerations of the work are presented and possible future works are pointed out.

2 THEORETICAL FOUNDATION

The present chapter has the objective of explaining the basic concepts used throughout the research. The introduction to some financial notions and to the models used are presented in order to fully understand the intricacies of the study.

2.1 STOCKS

A stock on the exchange is routinely known to represent a company or corporation. A stock is divided into shares, and the total number of shares is declared when the company enters the stock exchange. The holder of shares, even if it is a single share, is considered a partial owner of the company and is called a shareholder.

2.1.1 Shares

As shares represent a fraction of the company's ownership, different types of shares can be declared, each with appropriate specifications. In Brazil, there are two types of shares: preferential (PN) and common (ON). Preferential shares guarantee preference to holders in the division of profits (dividends) of the company. The common shares, in turn, grant voting rights to holders at the company's meetings, and a shareholder holding 50.01% or more of the common shares is considered a majority shareholder and controls all decision-making in the company. Table 1 shows the shares composition of the company Itaúsa S.A.[1], represented by ticker ITSA3, as example.

Table 1 – Composition of Share Capital of Itaúsa S.A.

Type	Total Shares	Shares in Circulation
Common (ON)	3.034.329.659	1.110.926.704 (36,61%)
Preferential (PN)	5.797.026.018	4.736.060.282 (81,70%)

Source: BM&F BOVESPA

2.1.2 Stock Prices

Every asset traded in the financial market has a value, in the case of stocks the shares have a liquid price. The price of a share has an extremely volatile behaviour, an intrinsic fact of the investment market. There are several socio-economic indicators that influence the price of a stock (BLICHFELDT; ESKEROD, 2008), the most notable being:

- Market trends: the law of supply and demand intrinsically influences stock prices. When a stock is on the rise, investor demand for the shares increases, generating an upward trend

[1] Source: <http://bvmf.bmfbovespa.com.br/cias-listadas/empresas-listadas/ResumoEmpresaPrincipal.aspx?codigoCvm=19348>. Acessed in June, 29th, 2022.

in the price. On the other hand, when a stock is down, the supply of papers increases, generating a downward trend in the price;

- Market manipulation: there are several mechanisms in order to manipulate stock prices. The dissemination of rumours on social media can have catastrophic consequences for the price of a stock. Another method of manipulation is the voluminous purchase of a certain share, causing the false impression that the company is on the rise, artificially inflating its demand and, consequently, its price;
- Political issues: countries with unstable political scenarios, such as Brazil, suffer drastic variations in stock corresponding to public companies in the market;
- Financial news: it is of vital importance to follow the news of companies in the market, as they directly influence their stocks. To some extent, it is possible to identify situations to enter or exit investment positions.

Market trends and manipulations are somewhat imperceptible in the short term, as there is no knowledge before these scenarios actually occur. These indicators require a human eye to be identified. Political issues and financial news can be absorbed by consuming newspaper news, whether by physical or electronic means.

In addition to its direct influence on market prices, another important factor in the use of newspaper news as an indicator is their availability to the general public. During the age of information, news are easily accessed in the most varied vehicles. These vehicles are full of information that certainly influence different areas of knowledge (HARIHARAN, 2012).

2.2 PREDICTING STOCK PRICES

An important theory that must be understood when working with stock price prediction is the Efficient Market Hypothesis (EMH) (FAMA, 1964). EMH indicates that all available information immediately reflects the current state of the market, suggesting that no predictions for future changes can be made. This hypothesis was proved false by its own creator (FAMA, 1991), but even overthrown, the hypothesis served as a catalyst for numerous researches in the area. Currently, there are two main types of approaches to predicting market behaviour: technical analysis and fundamental analysis.

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 (SCHUMAKER; CHEN, 2006).

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 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.

The key difference between technical and fundamental analysis is the incorporation of news into prediction models. News carries precious information about the market and represents a great impact on the prediction of stock prices (CHAN; CHUI; KWOK, 2001). The problem is that automating the interpretation of news proves to be a very complex task (TABARI et al., 2018), and this task characterises a large area of study, known as Sentiment Analysis.

2.3 ARTIFICIAL NEURAL NETWORKS

ANNs have been widely used to solve problems in the most varied areas of knowledge. Such computational model was designed with the aim of imitating the behaviour of the human brain, which is certainly the entity with the greatest existing computational power, due to its ability to process and organise information. ANNs provide a mechanism for handling problems oriented towards categorisation and pattern recognition of types and time series (WALCZAK, 2019). Models can be developed without knowledge about the characteristics of the distribution of the studied data or about the interaction between the variables of the problem, due to its non-parametric nature (WALCZAK, 2019).

An ANN resembles a human brain mainly in three ways (HAYKIN, 2007):

- Neuron: the main structure in the environment of an ANN is called an artificial neuron (popularly known as just a neuron). An ordinary ANN is composed of numerous neurons, which are the elements responsible for processing information;
- Learning: knowledge propagates through artificial neurons according to a learning process;
- Inter-connectivity: an ANN mimics the communication process between neurons (synapse). The connection between two neurons has a value called (synaptic) weight, which serves to store the absorbed knowledge.

2.3.1 Artificial Neuron

Routinely called a neuron, it is the most important component of an ANN. An artificial neuron is essentially composed of four elements (BRAGA; FERREIRA; LUDERMIR, 2007):

- Set of synapses: these are the connections between each neuron. Each connection has its own synaptic weight;
- Integrator: operator that sums the ANN input signals, weighted by the respective neuron synapses;

- Activation function: responsible for restricting the amplitude of the neuron's output value. Activation function types and their operation are explained in Section 2.3.5;
- Bias: value used to increase or decrease the input value of the activation function.

Using mathematical terms, a neuron is expressed by:

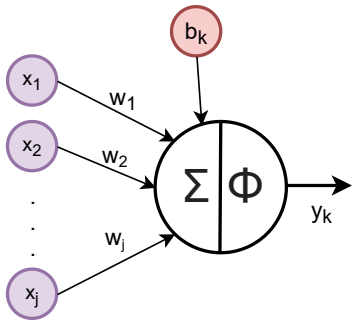
$$u_k = \sum_{j=1}^{m} w_k j x_j \tag{1}$$

$$y_k = \phi(u_k + b_k) \tag{2}$$

Equation (1) is the integrator's result for neuron k, represented by u_k, where $w_1, w_2, \ldots w_m$ are the synaptic weights of the neuron and x_j is the input signal of synapse j. Equation (2) is the output signal of neuron k, represented by y_k, where u_k is the integrator's result, b_k the bias value and ϕ is the activation function applied.

The structure of an artificial neuron is roughly sketched in Figure 1. An ANN is composed by the combination of a set of neurons, forming what is called an architecture.

Figure 1 – Structure of an Ordinary Artificial Neuron



Source: Author.

2.3.2 Architecture

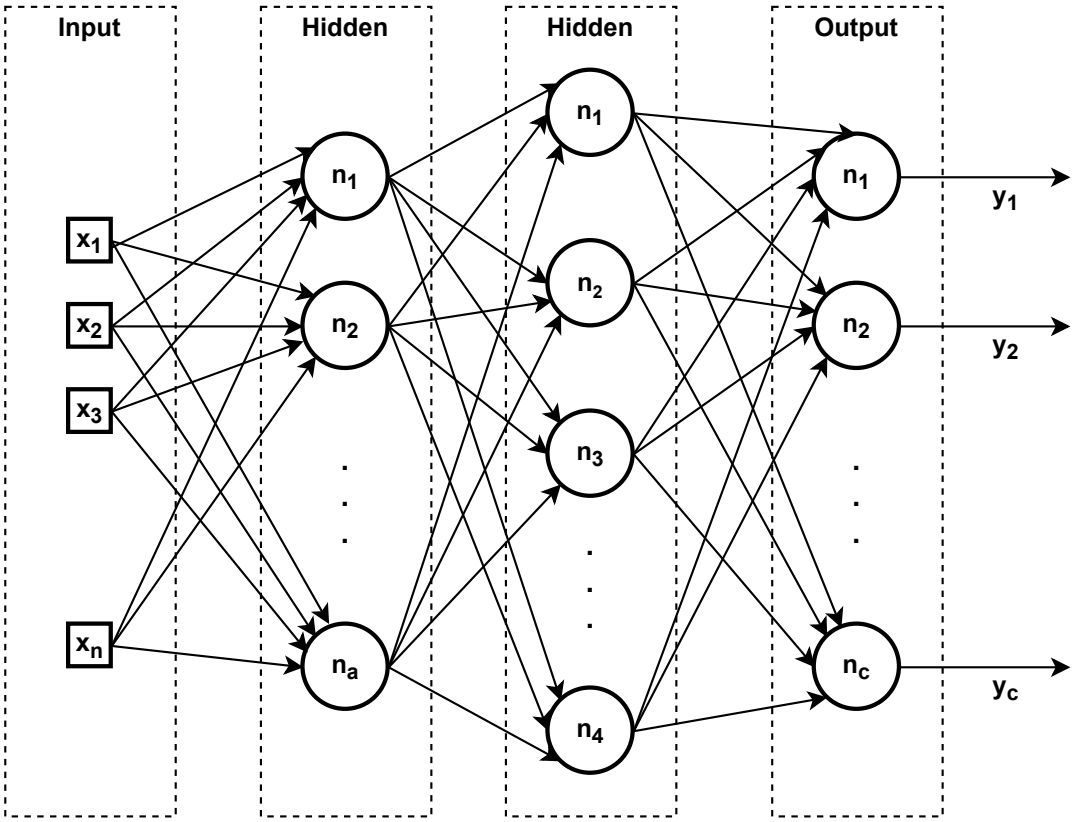
The way neurons are arranged and how synaptic connections between them are formed is called architecture. Every ANN architecture has a certain topology, which specifies the number of neurons and how the network layers will be configured. The architecture of an ANN is usually formed by an input layer, an output layer and none or more intermediate layers (also known as hidden layers).

The input layer is responsible for the absorption of input data, as well as the association of input weights. The intermediate layers are responsible for most of the network processing, aiming to extract the characteristics of the system. Finally, the output layer is responsible for transforming the data processed by the other layers into tangible information.

Commonly, the architecture of a neural network is classified into:

- Feedforward: in this type of architecture, information moves only in one direction, always forward: from the input layer it passes to the intermediate layers (if any) and ends the flow in the output layer, as outlined in Figure 2. Connections do not form a cycle and never move backwards; or
- Recurrent: In this type of architecture, connections can form loops and information can move backwards. This allows the network to maintain a kind of internal memory, so that more complex problems can be tackled, but with a higher computational cost. Figure 3 outlines the idea of recurrence in an ANN.

Figure 2 – Feedforward ANN Example



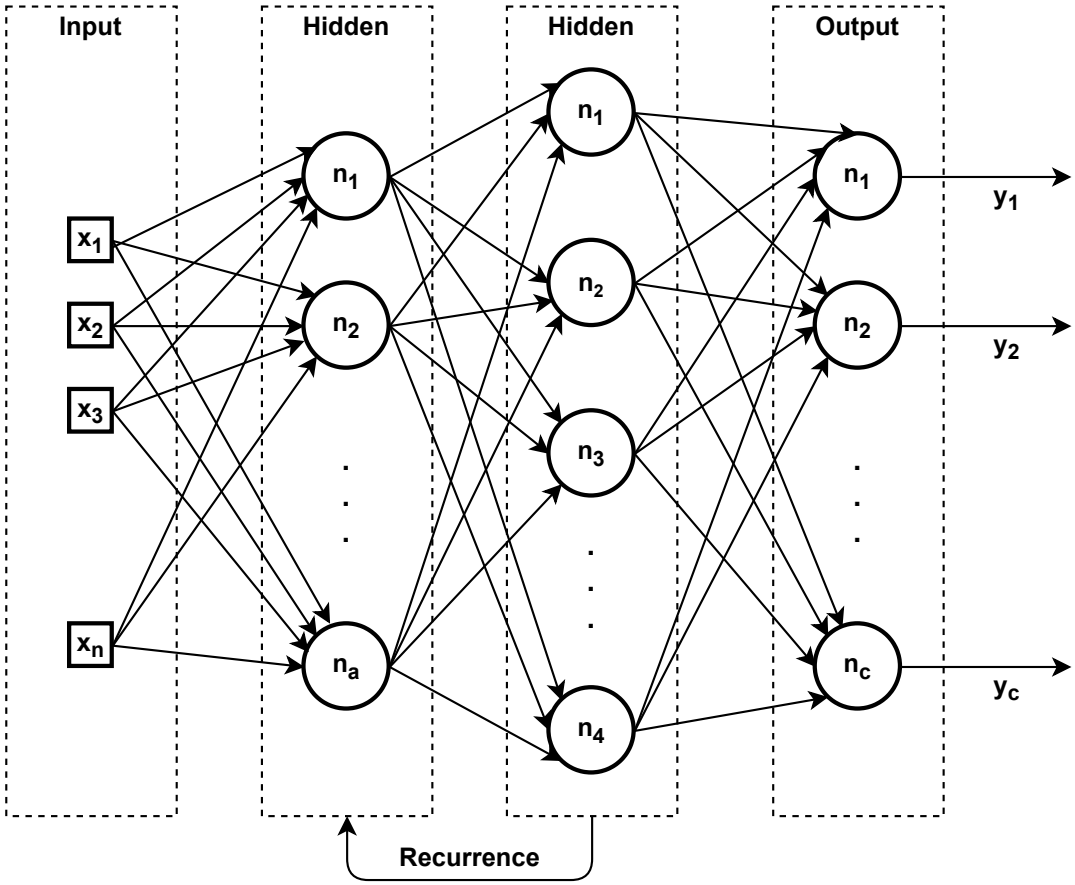
Source: Author.

2.3.3 Learning Algorithm

For an ANN to be able to properly solve a problem, the learning process is of fundamental importance. An ANN model has its performance adjusted according to its learning, which, in an ideal scenario, improves with each iteration of the process.

In the context of neural networks, the term learning boils down to the procedure of adjusting the internal parameters of the network: synaptic weights and bias values, via stimuli from the studied environment. During a learning iteration, the neural network must adapt to the stimuli, appropriately adjusting its parameters.

Figure 3 – Recurrence on an ANN



Source: Author.

The operator responsible for the learning process is called a learning algorithm. Learning algorithms can be divided into two main categories: supervised learning and unsupervised learning (WALCZAK, 2019). There are also two other categories, semi-supervised learning and reinforcement learning, which characterise problems less frequently. Both supervised and unsupervised learning algorithms need a collection of training samples, which are the input data. The big difference is in the labelling or not of the data.

In supervised learning, the network receives its desired responses along with the input data. This response represents the optimal network output for that particular input. The network parameters are adjusted according to the error signal, which is a value as a function of the current network response and the desired responses for each training instance. At each iteration of the process, we seek to reduce the error generated by the network, making its responses as close as possible to the labels received.

In unsupervised learning, nothing is known about the desired responses from the input data. As the network does not receive any kind of expected result, the network itself must be able to abstract relationships between the data in the set. The unsupervised learning process aims to group and organise data based on similarities found.

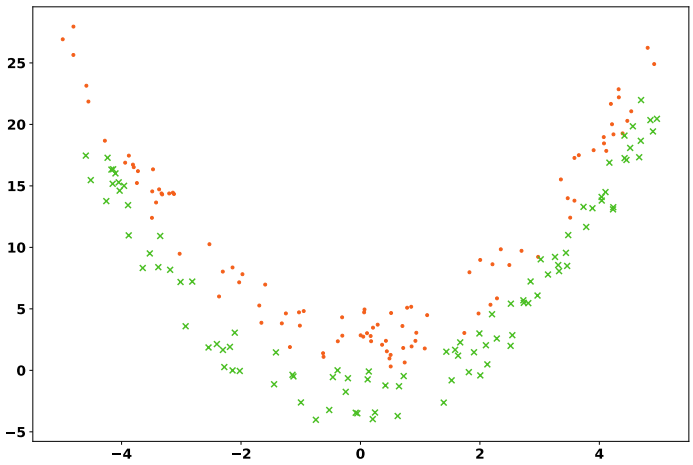
The learning algorithm used in the context of this research is

2.3.4 Problem Types

ANNs are used to solve problems of the most varied classes. It stands out, mainly, problems of classification, regression and clustering.

Classification is the name given to problems when you want to categorise a given set of data into different groups (classes). A traditional example of a classification problem is the approximation of a function to distinguish two different groups, as illustrated in Figure 4.

Figure 4 – Classification Problem with Two Groups



Source: Author.

In the illustrated scenario, there are two groups: green and orange. The objective of the classification problem is to find such a function that can make the best possible separation between the groups based on the known data, in order to be able to simulate the behaviour on unknown data, for example: identify to which group the point $p = (-4, -1)$, which would be classified as green in this example.

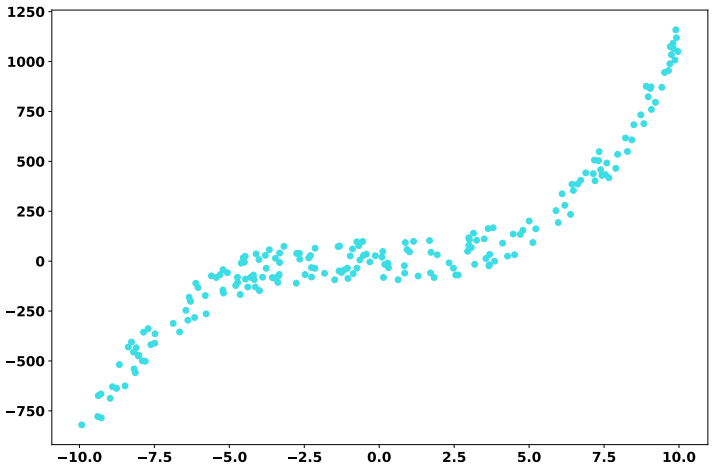
Regression is the name given to problems in which you want to estimate a continuous value for a function based on the input data. A classic example of a regression problem is trying to approximate a mathematical function that can represent the data set, as shown in Figure 5.

In a clustering problem, the objective is to separate the dataset into clusters, so that elements in the same cluster share similar properties. In Figure 6, an example of clustering is presented using the Euclidean distance between the points as a separation criterion.

2.3.5 Activation Function

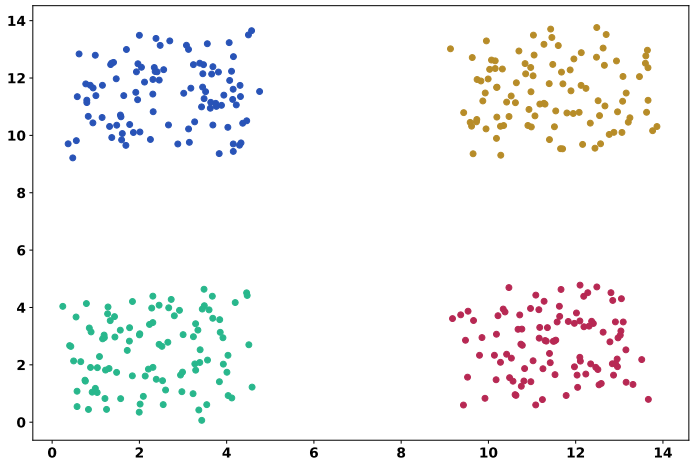
The activation function is responsible for the non-linear transformation performed along the input signal in a neuron. It is vitally important for an ANN to be able to abstract complex elements in a problem. Without an activation function, the synaptic weight and the bias represent only a linear transformation that, although simple to solve, is extremely limited in the face

Figure 5 – Simple Regression Problem



Source: Author.

Figure 6 – Clustering Example with Four Distinct Groups



Source: Author.

of complex problems. An ANN without an activation function is nothing more than a linear regression model (HAYKIN, 2007). There are several types of activation function, some of the most popular are:

- Binary Step: The binary step activation function limits the neuron's output to 0 or 1. This neuron is known as the McCulloch-Pitts model, due to its creators (MCCULLOCH; PITTS, 1943). Applying the binary step function converts the neuron output to the value 1 if its value is positive and 0 if its value is negative, according to Equation (3).

$$\phi(u_k) = \begin{cases} 1, & \text{if } u_k \geq 0 \\ 0, & \text{else} \end{cases} \tag{3}$$

- Piece-wise Step: This activation function is known to approximately simulate a non-linear amplifier. The piece-wise step function resembles the binary step function in its operation. Equation (4) shows the scenarios of the step activation function by parts, which basically boils down to keeping its original value if it is within the range $[-0.5, 0.5]$ or is converted to 0 or 1, if the value is below or above the range, respectively.

$$\phi(u_k) = \begin{cases} 1, & \text{if } u_k \geq 0.5 \\ u_k, & \text{if } 0.5 > u_k > -0.5 \\ 0, & \text{if } u_k \leq -0.5 \end{cases} \tag{4}$$

- Rectified Linear: Rectified linear activation function is widely used in models with high number of layers. Equation (5) denotes the function, which returns its own value if it is positive or 0 otherwise.

$$\phi(u_k) = \begin{cases} u_k, & \text{if } u_k \geq 0 \\ 0, & \text{else} \end{cases} \tag{5}$$

- Sigmoid: Perhaps the most used activation function, the sigmoid is a strictly increasing function that takes on values in the range $[0, 1]$. The most common example of a sigmoid is the logistic function, expressed by Equation (6).

$$\phi(u_k) = \frac{1}{1 + e^{a * u_k}} \tag{6}$$

  where a is the sigmoid slope coefficient. This parameter must be set, usually it is set to 1 or $1/4$. It is interesting to note that a value for a close to 0 approximates the sigmoid to the behaviour of a linear function and a value tending to infinity approximates the sigmoid to the behaviour of the threshold activation function.

- Hyperbolic Tangent: The hyperbolic tangent activation function has the same shape as a sigmoid, but assumes values in the range $[-1, 1]$. Allowing negative values can have analytical benefits in certain scenarios (HAYKIN, 2007). Equation (7) represents the hyperbolic tangent.

$$\phi(u_k) = tanh(u_k) = \frac{e^{u_k} - e^{-u_k}}{e^{u_k} + e^{-u_k}} \tag{7}$$

- Softmax: The softmax activation function takes a vector z of size k and normalises it to a probability distribution proportional to the exponential of the values, which results in values in the range $(0,1)$. Equation (8) represents the softmax.

$$\phi(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{8}$$

2.3.6 Long Short-Term Memory (LSTM)

The biggest difficulty of applying a common artificial neural network architecture in a price prediction scenario is the inability to deal with long-term dependencies, a factor of extreme importance for any problem that consists of a time series. LSTM is a special type of Recurrent Neural Network (RNN) explicitly designed to solve the long-term dependency problem by introducing a new memory cell concept to replace traditional artificial neurons in hidden layers (HOCHREITER; SCHMIDHUBER, 1997).

The cell state in an LSTM is designed to be able to remember information over a long period of time (GERS; SCHMIDHUBER; CUMMINS, 2000). This information is carefully regulated by three structures called gates, which will control what information will be discarded from the cell, what information will be fed into the cell, and what will be output from the cell. With this complex memory cell structure, LSTMs are able to dynamically assimilate data structure over a period of time with high predictability.

Due to the ability to process sequential data in a remarkable way, LSTMs prove to be an efficient mechanism in many different fields, including but not limited to computer science, statistics, linguistics, and medicine. All of these areas have complex tasks that revolve around predicting, classifying and analysing sequential data, tasks that LSTM is known to perform well (SMAGULOVA; JAMES, 2019).

2.3.6.1 Recurrence in Artificial Neural Networks

An ordinary RNN has a kind of internal memory, expressed by the state immediately prior to the current one. Arbitrarily defining a network with only one hidden layer, for the sake of simplicity, Equation (9) defines memory h in state t. Also for reasons of simplicity, it is opted for the matrix notation of the elements.

$$h_t = \phi(W[x_t, h_{t-1}] + b) \tag{9}$$

where W is the synaptic weight matrix connecting the input vector to the hidden layer, x is the input vector, h_{t-1} is the memory in the state before the state t and b the vector of bias. The output y of the layer in the state t is given by Equation (10).

$$y = V h_t \tag{10}$$

where V is the synaptic weight matrix connecting the hidden layer to the output layer.

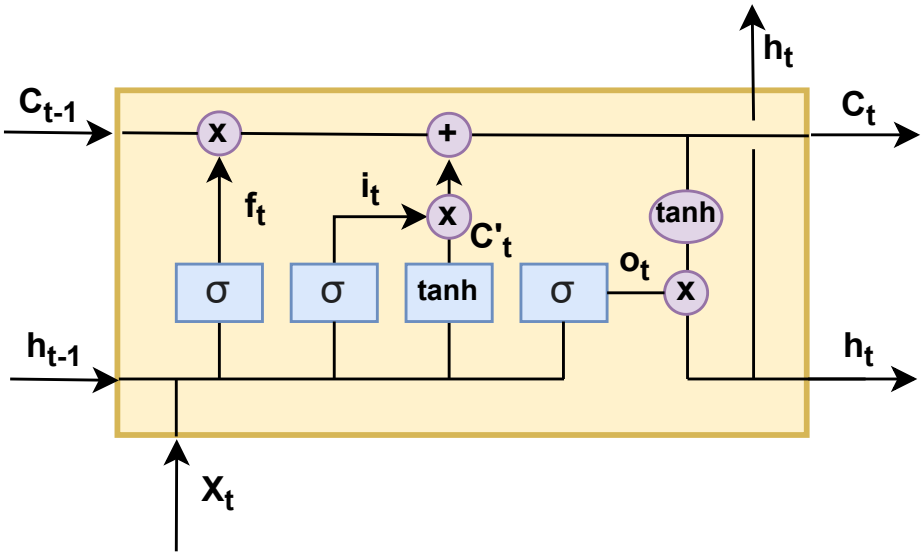
The problem with this formulation is that the network can only pass on information to the successor state. This fact limits short-term memory, but it is plausible that a given scenario needs to look for information in older contexts, something that the traditional recurrence construction does not allow (BENGIO; SIMARD; FRASCONI, 1994). The recurrence implementation in LSTMs, in turn, are designed to solve the dependency problem in the long run.

2.3.6.2 Recurrence in LSTMs

While an RNN overwrites its memory with each change of state, LSTMs use a more intelligent mechanism to manage changes in memory contents. This control allows information to be kept on the network for long periods of time. The operation of an LSTM cell, illustrated in Figure 7, consists of four steps:

- Decision of what to delete from memory (or forget);
- Decision what to add to memory (or remember);
- Actually update memory contents;
- Decide what the output value will be.

Figure 7 – LSTM Cell



Source: Adapted from (HOCHREITER; SCHMIDHUBER, 1997)

The cell state, denoted by C_t, is responsible for long-term memory. Within each LSTM cell, there are three elements responsible for controlling this state. The elements are called ports: forgetting port, entry port, and exit port. Details of the mathematical implementation can be found in (HOCHREITER; SCHMIDHUBER, 1997), but in this Section the general idea is presented.

The first step is to decide what information to eliminate from the cell state. The forgetting gate takes the memory of the previous state h_{t-1} and the input vector x_t and produces a vector of weights f_t in the domain $[0,1]$, controlled by a sigmoid according to Equation (11).

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{11}$$

The second step is to decide what new information will be added to the cell state. This step is divided into two steps. The input port operates analogously to the forgetting port, receiving the same values of h_{t-1} and x_t, producing a vector of weights i_t in the domain $[0,1]$, controlled by a sigmoid according to Equation (12). At the same time, a vector of candidates C'_t is listed to be added to the state, subject to the cell activation function, which is originally the hyperbolic tangent, according to Equation (13).

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{12}$$

$$C'_t = \tanh(W_C[x_t, h_{t-1}] + b_C) \tag{13}$$

The third step is to actually update the cell state. The previous state is multiplied by the result of f_t to forget what should be forgotten and sum what should be remembered, represented by $C'_t * i_t$. The Equation (14) presents the state update.

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{14}$$

The last step is to decide what the cell's output will be. The output is partially based on the current state C_t calculated in the previous step. The output port executes similarly to the others, receiving as input h_{t-1} and x_t, producing a vector of weights o_t in the domain $[0,1]$, controlled by a sigmoid according to Equation (15). The result is then multiplied by the current state subject to the hyperbolic tangent, according to Equation (16), finally representing the output of the LSTM cell.

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{15}$$

$$h_t = o_t * \tanh(C_t) \tag{16}$$

2.4 PARAMETER TUNING

One of the most important 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 a 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.

Empirical experiments are interesting to delimit upper and lower boundaries in the search space of parameters. For example, if in a set of tests it was found that the algorithm does not converge with 4 hidden layers nor with 5 hidden layers, it can be assumed that it will not converge with less than 4, given that the other parameters are set to the same values.

Subdivisions in the search space serve to decrease the amount of trained configurations. Dealing with the size of a layer, for example, instead of testing the values in the range of $[100, 200]$ nodes for the size of the layer, it is possible to test in intervals of 10 in 10, for example, to decrease significantly the time devoted to training the configurations. With the result of this experiment, one can choose to refine the surroundings of the found configuration.

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.

2.5 SENTIMENT ANALYSIS FOR STOCK PRICE PREDICTIONS

The process of determining the positivity, negativity or neutrality of a textual body is called sentiment analysis (PAWAR; JAWALE; KYATANAVAR, 2016). Normally, for us human beings, it is relatively simple to identify the sentiment of a text via textual interpretation. It is quite obvious that a sentence such as “It’s a beautiful sunny day today!” represents something positive and a sentence such as “Today’s game was cancelled due to rain” represents something negative. However, for a machine, it is extremely difficult to be able to categorise texts in this way.

Sentiment analysis is applied in various areas such as opinion mining, product review analysis, social media analysis and news analysis (YANG; MO; ZHU, 2014). In the scope of this work, sentiment analysis is used to help the price prediction model.

There are two main ways to automate the sentiment analysis process: methods based on the use of a dictionary and methods based on the application of ML. Using sentiment analysis, it

is possible to perceive intentions of companies and investors in real time, something extremely important for decision-making on the stock exchange.

Sentiment analysis models that apply ML rely on techniques to learn from automatically received textual bodies. No manual rules need to be developed. The neural network training process is entirely based on feature extraction. Models that apply the use of dictionaries, on the other hand, have to manually create said dictionaries to compute the overall sentiment of the input based on individual words.

Measuring sentiment aims to see whether financial news have an impact on market movements and to identify patterns of sentiment associated with these movements. In a volatile environment like the stock market, news sentiment is one of many indicators that can be useful in predicting the stock prices.

As explained in Section 1.4, newspaper news will be explored in this work. Newspaper news do not have a high volume of circulation. Although there is a considerable amount of newspapers, there are few that have high credibility. Furthermore, a newspaper is composed of numerous sections, limiting the number of news items in the financial section. Therefore, the sentiment analysis model applied must be able to work with a limited database.

2.5.1 Valence Aware Dictionary for sEntiment Reasoning (VADER)

The VADER model is based on a finely refined dictionary of lexical features for calculating emotional intensity in textual bodies, denoted sentiment scores. This score is obtained by adding the intensity of each word in the text, based on its entry in the dictionary. The total value is then normalised to the range $[-1, 1]$.

The sentiment analysis performed by the VADER model differs from other generic models in two aspects, which influenced the choice of the model for the scope of this work:

- Intensity of Lexicon Sentiment: in addition to determining the polarity of words, the VADER model is able to indicate the intensity of the sentiment expressed by the text. This fact is of vital importance for the interpretation of news in the financial scenario, where any change in intensity can denote a sudden variation in the investment market. Example: the sentence “I like orange juice” has a less intense feeling than “I love orange juice”, although both are positive;

- Lexicon Context Awareness: Words can have different meanings according to the context in which they are inserted. Analysing words individually may produce incorrect results in some cases. The VADER model manages to abstract these situations by analysing the context around the word. Example: in the sentence “The car repair will be very expensive”, the word “repair” represents something bad for the subject, because its context is “expensive”. However, in the sentence “The car’s repair was flawless”, “repair” represents something positive for the subject, as it was “flawless”.

The construction of a sentiment lexicon is a long, error prone, process. To develop VADER's dictionary, the authors collected a total of over 90000 unique ratings created by a group of human raters, as explained in detail by the creators in (HUTTO; GILBERT, 2014). Using this dictionary, the sentiment score of sentences are calculated by evaluating every word based on VADER's dictionary and summing their respective scores. This operation effectively returns a sentiment score in the interval of $[-4,4]$, which is normalised to a value in the interval $[-1,1]$.

3 RELATED WORK

The prediction of stock prices represents an extremely complex task since the beginning of the study in the area of financial computing, due to the high volatility of the investment market (ADAM; MARCET; NICOLINI, 2016). The area shows a lot of interest from researchers and presents several different approaches to tackle the problem.

3.1 LITERATURE REVIEW

Mittal and Goel (2012) used Twitter as a database. The sentiment analysis model implemented only categorised the news into 4 groups, representing the user's mood when publishing their tweet: calm, happy, alert or generous. The authors implemented 4 prediction models: linear regression, logistic regression, Support Vector Machine (SVM) and Self-Organizing Fuzzy Neural Network (SOFNN). According to the MAPE metric, the best predictor was SOFNN. This network has the need to use a time window value, the authors arbitrarily chose 3 days. With the predictor result, the authors implemented a greedy strategy to select the best portfolio, based solely on predicted closing values: there was no risk calculation.

Nguyen, Shirai and Velcin (2015) evaluated news from the Yahoo Finances forum on 18 stocks of the american market. The authors' sentiment analysis categorises each news item into one of 5 classes (Strong Buy, Buy, Hold, Sell, and Strong Sell) to feed the forecast model, using an SVM. This work presents binary prediction and only compares the accuracy between the models. The biggest problem on their approach is the fact that there is no way to state the size of the prediction window, that is, how many days should be taken into account in training to consolidate a prediction. The authors used a window of size 2, therefore, to predict the movement of the day d, the news and closing values of the days $d-2$ and $d-1$ were used.

Creamer (2015) used Automated Text Analysis (ATA) to classify news extracted from the *TR News Archive*. As a result, the author created the representation of the investor's view in the Black-Litterman portfolio optimisation model (BLACK; LITTERMAN, 1990). The proposed model, with 27 stocks from the European market, indicated better performance in terms of return compared to the market index, which is generally a good indicator. However, the model assumes that, for the generated investor view, the error co-variance is low and the confidence is high. This assumption is somewhat arbitrary, as the view generated cannot be generalised to all investors: some are more cautious, others more impulsive.

Ding et al. (2015) developed a deep learning model for event-based predicting in the investment market. Events were extracted from news and represented as vectors. A Convolutional Neural Network (CNN) was used to model the influences of events on short and long-term market movements. The results demonstrate that CNN networks capture long-term influences better than ordinary recurrent neural networks in the considered scenario.

Roondiwala, Patel and Varma (2017) implemented a LSTM model for predicting stock prices on the Indian stock market. By feeding the model only the historical data of the assets, the

authors were able to accurately predict the closing prices, based on the calculated RMSE values. The authors leave questions open for not using sentiment analysis in their model nor exploring the predictions generated with a portfolio optimisation algorithm.

Song, Liu and Yang (2017) implemented a relatively simple sentiment analysis, but it covers the need imposed by the prediction model - they only calculate two values: shock and news trend extracted from *Thomson Reuters News Analytics*. The Learning-to-rank model is composed of a neural network with gradient descent guided learning that has a cross-entropy loss function. This article does not predict prices, but a ranking of 512 assets in the American market and makes simulations using 4 different investment strategies, without using an optimiser. The strategy that generated the highest return was to buy the best-ranked assets and hold them for the entire time period studied.

Xu and Cohen (2018) developed a sentiment analysis model for posts collected from Twitter strongly based on the principle of Gated Recurrent Unit (GRU), which featured a layer in their original model for predicting stock prices, called StockNet. The application scenario was purely theoretical, presenting 88 benchmark stocks. The authors also needed to implement a strategy to filter the tweets, implying that many comments are disregarded. This work does not predict the gross closing value for the assets, but rather predicts the movement of the asset (binary prediction: the asset will appreciate or not) for the next few days. The authors did not use the result to build portfolios, they only compared the model's accuracy with other models in the literature.

Rana, Uddin and Hoque (2019) 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 optimisers were paired, reaching the conclusion that the combination that generated the best accuracy was the activation by Hyperbolic Tangent with the Adam optimiser.

Faizan (2019) used stocks, *Bitcoin* and even the price of gold in his experiments. The news were extracted from the NY Times and the sentiment analysis was performed using IBM's Watson API, but the author does not provide details on the method used by the API. The author used the Reference Point Method (RPM) to solve a bi-objective model (maximisation of return and minimisation of risk) of portfolio optimisation, considering the information coming from sentiment analysis in the form of constraints in the model, a unique approach in relation to the other studies studied. The results for the expected return were satisfactory, but the model is limited in terms of diversification of the generated portfolio.

Du and Tanaka-Ishii (2020) developed their own sentiment analysis model to investigate news extracted from the Wall Street Journal (WSJ) and the Reuters & Bloomberg database (R&B), categorising the news according to a metric to calculate the weight of each news item in relation to its respective stock. They used Multi Layer Perceptron (MLP) to predict prices and fed an optimisation model based on the classic Markowitz model (MARKOWITZ, 1952). Observing 18 selected stocks from the American market, the authors compared their strategy

with others presented in the literature. His model obtained better results for the R&B base, but worse results compared using the WSJ dataset when compared with a Word2Vec model that only considers news for the prediction of prices.

The work of Xing, Hoang and Vo (2020) is quite unique in relation to the others studied. The prediction model, a MLP network, is applied to predict whether the price of the US Dollar and the Euro will appreciate or not. The analysis performed by the authors is extremely complete, comparing the results of the model on several investment fronts with accuracy metrics. The results confirm the predictive power of a model using high frequency news without any kind of technical analysis.

Beraldi (2020) used the 7 Exchange Traded Funds (ETF) represented by IBOVESPA's B3 segment indices as assets and Twitter as a news base. The exercise is purely theoretical, as the Brazilian ETF market is not robust. Sentiment analysis is performed using the VADER technique on a social media context, which is quite interesting because it quantifies news sentiments in a way that facilitates the creation of a time series. The author did not specify which scores were used on VADER. The author implemented the Black-Litterman model for portfolio optimisation and fed it with sentiments representing the investor's view, similar to Creamer's approach (CREAMER, 2015), so this work does not use sentiments to predict prices. The author's model has better performance than the *benchmark* IBOVESPA, but the allocation of portfolios is poorly balanced, due to the characteristic of the Black-Litterman model. The author only measures the return of the portfolio and does not pay attention to risk, a fact that masks their poorly diversified portfolios, known to present high risk, especially in scenarios with high volatility such as during the COVID-19 pandemic.

Maqsood et al. (2020) proposed a 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 generalised.

Patil et al. (2020) combined graph theory with CNN analysing spatiotemporal relationships between different stocks, modelling 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 (2020) 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 categorisation (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 generalise any results due to the fact that the experiments were performed with only

one stock.

Jing, Wu and Wang (2021) created a prediction model with sentiment analysis and applied it on the Shanghai stock market. The model used CNN to calculate investors sentiment values on information extracted from a major stock forum and LSTM to predict stock prices of six companies. The experiment results confirms the performance increase of the model when sentiments are considered, when compared to selected baselines.

Gupta et al. (2022) used LSTM to predict prices of six stocks in the Indian market. The authors used a simple, but effective sentiment analysis tool called TextBlob to process information extracted from Twitter. Although the model presents results better than other compared models, it is hard to generalise the analysis because the authors considered a set with only six tickers on their experiments, which is rather limited given the magnitude of a stock market.

The research shows that financial sentiment has been explored in academic scenarios. To evaluate and understand this sentiment, a wide array of techniques have been used, generally depending on the differences presented by the scope of the project. Normally, financial news are used in attempt to predict market movement by analysing special patterns generated by the information acquired, which tend to display empirical evidences of correlation between investor sentiment and financial market movements. All of this is great on paper, but to a practical extent, the things seem a bit grey. There's little to no evidence of the results being consistent on a real investment scenario. The majority of the papers compare their results with results from another papers, but never seem to perform a tryout or simulation using the predictions generated.

An overview of the works can be seen in Table 2. 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, we present the model proposal for this dissertation.

Table 2 – Main Approaches Identified in the Literature

Reference	Scenario	Sentiment Analysis	Sentiment Source	Prediction Model	Evaluation Metrics	Portfolio Selection
(MITTAL; GOEL, 2012)	USA	Original	Social Media	SOFNN	MAPE	-
(NGUYEN; SHIRAI; VELCIN, 2015)	USA	Joint Sentiment/Topic	Investment Forum	SVM	Accuracy	-
(CREAMER, 2015)	Europe	ATA	Magazine	-	Sharpe Index	Black-Litterman
(DING et al., 2015)	USA	-	-	CNN	Accuracy	-
(ROONDIWALA; PATEL; VARMA, 2017)	India	-	-	LSTM	RMSE	-
(SONG; LIU; YANG, 2017)	USA	Indicators	Magazine	Learning-to-rank	Sharpe Index	Learning-to-rank
(XU; COHEN, 2018)	Theoretic	Original	Social Media	Original	MCC	-
(RANA; UDDIN; HOQUE, 2019)	Spain	-	-	LR, SVR, LSTM	RMSE	-
(FAIZAN, 2019)	USA/BTC	Watson	Newspaper	-	Risk/Return	Risk/Return (Covariance)
(DU; TANAKA-ISHII, 2020)	USA	Original	Newspaper	MLP	Expected Return	Expected Return
(XING; HOANG; VO, 2020)	Exchange	BERT	Newspaper	MLP	Accuracy	-
(BERALDI, 2020)	Theoretic	VADER	Social Media	-	Sharpe Index	Black-Litterman
(MAQSOOD et al., 2020)	USA, Asia	SentiWordNet	Social Media	CNN	RMSE, MAE	-
(PATIL et al., 2020)	USA	-	-	Graphs + CNN	RMSE, MAPE, MAE	-
(JIN; YANG; LIU, 2020)	USA	CNN+word2vec	Investment Forum	LSTM	MAE, RMSE, MAPE	-
(JING; WU; WANG, 2021)	China	CNN	Investment Forum	LSTM	MAPE	-
(GUPTA et al., 2022)	India	TextBlob	Social Media	LSTM	Accuracy	-
This dissertation	**USA**	**VADER**	**Newspaper**	**LSTM**	**RMSE, MAPE**	**Risk/Return (CVaR)**

Source: Author.

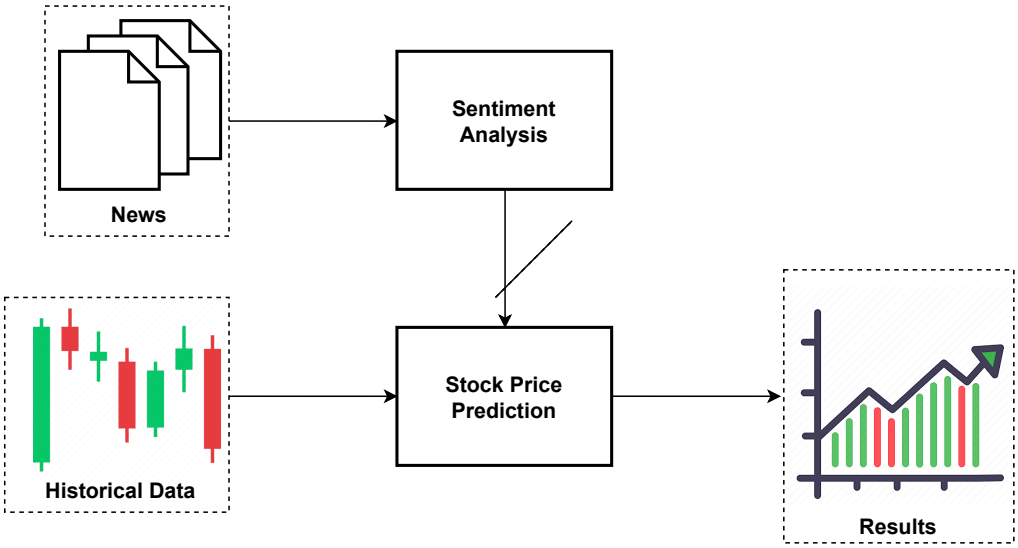
4 PROPOSED MODEL

The approach presented in this work separates the model into two stages: sentiment analysis and stock price predicting, as shown in Figure 8. The two steps have different requirements and work together.

The sentiment analysis stage aims to process the news of the companies studied and define the position of investors during the studied period, based on the sentiment obtained. Sentiment analysis tasks generally require some kind of data pre-processing to obtain good results, but this is not the case with the model applied in this work, as discussed in Section 4.1.1.

The price prediction stage, which is based on the result of the previous stage and on the historical series of prices of the assets in the studied period, aims to develop a model intelligent enough to generalise the prediction for all the assets studied, which will represent a portion of the universe of assets on a stock exchange. The price prediction task requires the development of a robust model, which has the need for data validation and parameter adjustment.

Figure 8 – Model Steps



Source: Author.

4.1 SENTIMENT ANALYSIS

The sentiment analysis process commonly requires some pre-processing of the data before actually performing the sentiment calculation. This process is not generic, as different databases require different refinements. Some of the most used techniques are:

- Removal of capital letters;
- Punctuation removal;

- Removal of stopwords[1];
- Lemmatization;
- Stemming.

In this work, the VADER model is used to calculate sentiments. The choice of the VADER framework is based on its applicability for sentiment analysis in databases with a limited number of samples, a characteristic that is apparent when working with newspaper news. The VADER model has five feelings intensity heuristics (HUTTO; GILBERT, 2014):

- Score: can increase intensity without changing feeling;
- Capital letters: can also amplify the feeling by affecting the mood of the text;
- Degree modifiers: decrease or increase the intensity according to the context used;
- Use of the word “but”: may indicate a change in polarity;
- Analysis of trigrams: allows identifying polarity changes caused by the negation of some element.

Due to these heuristics, it is noted that some pre-processing strategies will degrade the sentimentalisation results of the VADER model, rather than improving it. Therefore, no pre-processing routine will be applied in this work, as explained in the following section.

4.1.1 Sentiment Calculation

In the VADER model, the sentiment calculation returns four scores: negative, neutral, positive, and compound. Negative, neutral, and positive scores are proportions of the text that fall into each category, so the sum of the three scores must equal 1. These proportions represent the categorisation of each word in the text into negative, neutral or positive classes, without taking into account the heuristics of the VADER model. The compound score, in turn, is calculated by adding the valence scores[2] of each word in the text, adjusted according to the heuristics. The final value is normalised to the range $[-1, 1]$, with -1 being the negative extreme and $+1$ the positive extreme. Compound score is the sentimentality most used by most researchers, including the authors of the VADER model (HUTTO; GILBERT, 2014). In view of this, in this work its opted for the use of compound punctuation.

In order to justify the non-application of any pre-processing to the data, cases are presented where the techniques end up worsening the sentimentalisation results. For each of

[1] Stopword is the term denoted to commonly used words that are ignored by search engines, such as "the", "a", "is", "an", among others. It is worth mentioning that there is no universal list of stopwords, much less rules to define them.

[2] The valence score is not only based on the polarity (positive or negative) of each individual word, but also calculates the word’s intensity by analysing the applied context.

the 5 techniques listed in Section 4.1, a counter-example is presented, highlighting common textual construction cases where the application of the pre-processing strategy would degrade the sentiment calculation:

1) Removal of capital letters: Decreases the intensity of positive feeling in the “I LOVE sunny days” example. It is noticed that the simple fact of removing capital letters reduces the intensity of the positive feeling.

```
I LOVE sunny days
{'neg': 0.0, 'neu': 0.205, 'pos': 0.795, 'compound': 0.8286}

i love sunny days
{'neg': 0.0, 'neu': 0.222, 'pos': 0.778, 'compound': 0.7906}
```

2) Punctuation removal: lessens the intensity of negative feeling in the “I hate working on Mondays!” example. Removing the exclamation point at the end of the sentence reduces the expressiveness of the feeling, and therefore, the negativity in this case.

```
I hate working on mondays!
{'neg': 0.499, 'neu': 0.501, 'pos': 0.0, 'compound': -0.6114}

I hate working on mondays
{'neg': 0.481, 'neu': 0.519, 'pos': 0.0, 'compound': -0.5719}
```

3) Removal of stopwords: Drastic difference in sentiment in the example “Pizza is good, but salad is so bad”. In this example, the word “but” characterises a change in the feeling of the subject in the sentence, the removal of which causes a large part of the neutrality of the sentence to be converted into positivity, something that is incorrect given the original context.

```
Pizza is good, but salad is so bad
{'neg': 0.44, 'neu': 0.423, 'pos': 0.137, 'compound': -0.742}

Pizza good, salad bad
{'neg': 0.417, 'neu': 0.238, 'pos': 0.345, 'compound': -0.1531}
```

4) Lemmatization: drastic difference in feeling in the example “He has the bad habit of swimming after eating”. The big problem with lemmatization is that it produces sentences that do not make sense orthographically, in this example the word “have”, arising from “has” which is an inflection of the verb “have”, which has “have” as lemma. The word “have” gives no indication about the temporality or the subject of the sentence, making the sentence feel from relatively negative to almost neutral.

```
He has the bad habit of swimming after eating
{'neg': 0.304, 'neu': 0.696, 'pos': 0.0, 'compound': -0.5423}

He have the bad habit of swimming after eating
{'neg': 0.255, 'neu': 0.438, 'pos': 0.307, 'compound': -0.0772}
```

5) Stemming: this process of generating word roots also ends up mischaracterizing the meaning of sentences, but much more often. As the VADER model dictionary was created manually, the stemming process will cause the generated radicals not to be mapped to any dictionary entry, completely nullifying the parsing.

4.2 STOCK PRICE PREDICTING

The artificial neural network LSTM is capable of identifying long-term dependencies, as long as the training data is properly segmented into sub-sequences with well-defined beginning and end (GERS; SCHMIDHUBER; CUMMINS, 2000). This is guaranteed when analysing historical series of stock prices, as any sequential subset of the series can be effectively cast as training data. It is up to the researcher to define the size of these subsets, considering that a window too small can inhibit the memory capacity of the model and, on the other hand, a window too large will cause an execution bottleneck during model training.

The ability to keep information from the past is what guarantees the potential application of LSTM to the problem of predicting asset prices, since the previous price is a crucial element to predict the price in the future.

4.3 PREDICTION MODEL

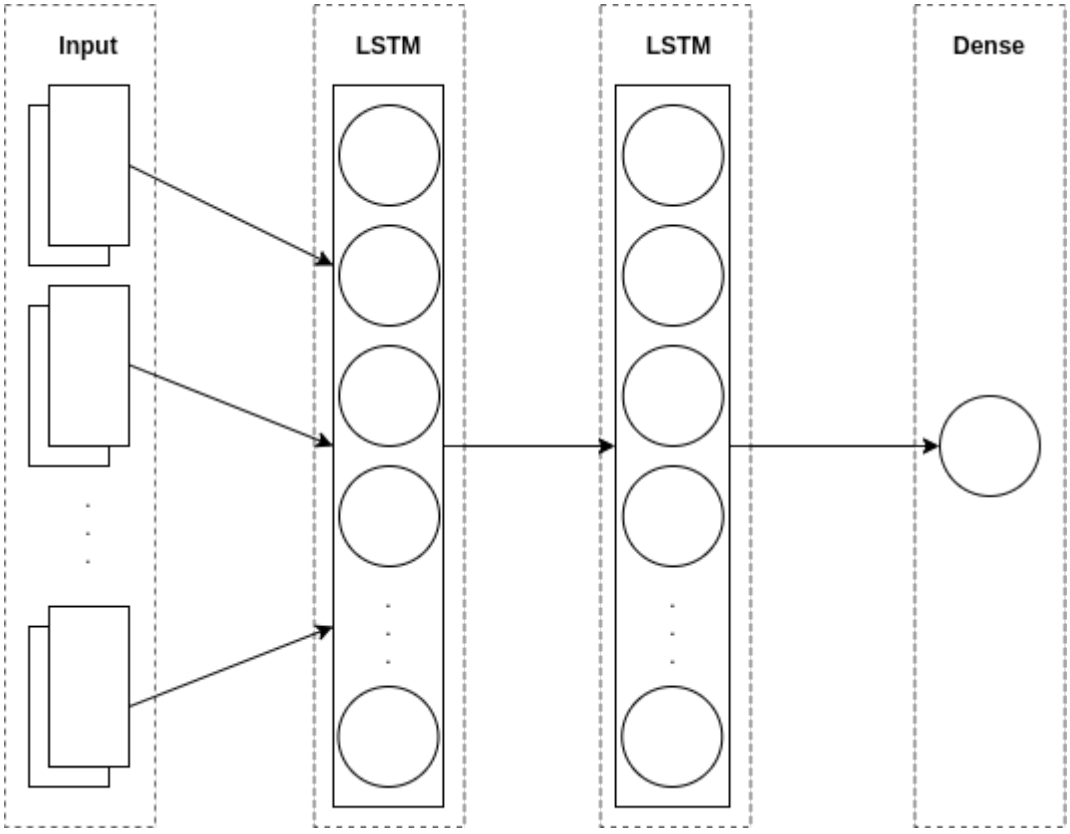
In the sentiment analysis stage, the set of news of each asset was individually submitted to the VADER model, resulting in the sentimentality of each news item. A maximum of 500 financial articles from the newspaper The New York Times for each asset per year is considered, which means that some days have more than one news for the asset. The limit of 500 news was empirically imposed, considering that a small minority of the companies analysed exceeded this value. For these cases, the sentiment of the day in question is given by the average of the sentiments of each news item. In addition, some days do not have any news for the asset. For these cases, sentiment is given as neutral, $s = 0$. At the end of the stage, a sentiment value will have been assigned to each closing day for each asset analysed. This value will be used as one of the characteristics of the prediction model.

In the price prediction stage, the historical series of the assets and their respective sentiment values are used as input in the prediction model. An ordinary LSTM model contains three layers: an input layer, an LSTM layer of size n, and a dense layer with a node deeply

connected to the output of the final LSTM layer, responsible for consolidating the input received from the LSTM layer into a final prediction value.

The model proposed in this work has an additional LSTM layer, as outlined in Figure 9. The stacking of two or more LSTM layers provides greater abstraction capability to the model (SCHMIDHUBER, 1992), (PASCANU et al., 2013), allowing the representation of more complex patterns. This strategy, on the other hand, increases the computational cost of the model. The architecture was defined empirically, experimenting models with different numbers of LSTM layers and evaluating the time complexity and model accuracy trade-off (HEIDEN; PARPINELLI, 2021).

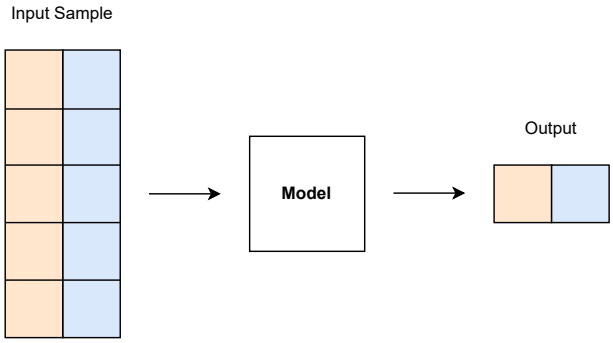
Figure 9 – Prediction Model Architecture



Source: Author.

Input data is formatted according to one parameter, the window size. This parameter indicates how many days will be considered "dependent" for the price prediction. For example, with a window size of $t = 30$, the 30 days prior to day d will be considered in the prediction. The larger the window, the greater the model's view of past information, but also the greater the model's computational cost. Each sample is a n-dimensional array of size t, n being the number of features in the prediction model and t the window size considered. The output of the model, controlled by the dense layer, is also a n-dimensional array, but this time has size 1, representing the prediction of the features values for a single day. An example of input and output data is shown on Figure 10.

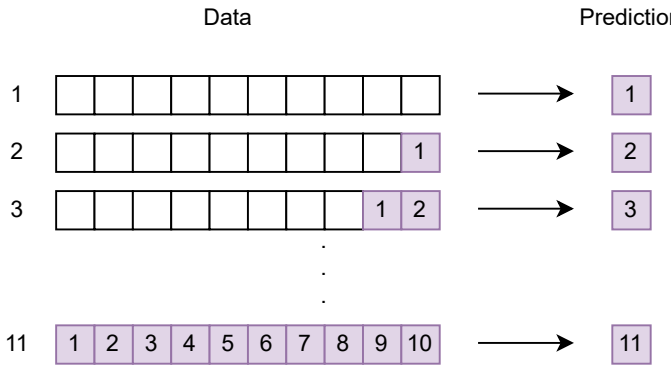
Figure 10 – Example of Input and Output for Two Features with Window Size of 5 Days



Source: Author.

In an example where the model works with a window size of 10, each prediction will be based on data from the last 10 days of the series, which is not a problem for the first prediction. As the predictions go further and further into the future, starting with the second prediction, the data used will be composed of actual data and predicted data. The further away the prediction, the less accurate it becomes, due to the use of more data acquired from the model's predictions. In the example of Figure 11, it is noted that from the prediction 11 onward, all data used for the prediction are the result of the model's predictions.

Figure 11 – Example of Predictions with Window Size of 10 Days



Source: Author

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.

4.4 PARAMETER TUNING USING HYPERBAND

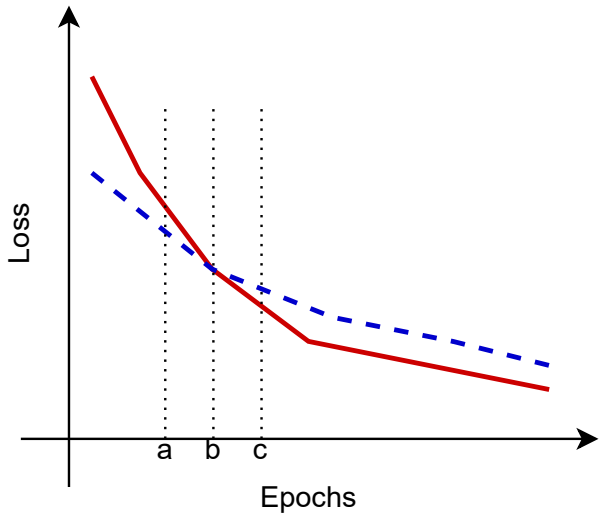
The main idea of Hyperband is to reduce the evaluation time from a model's configuration. There are two main ways of achieving this speedup:

- Early stopping: fit the model using a limited amount of epochs or with a set criteria to stop the training earlier than the usual;
- Train on a subset of data: reduce the amount of samples presented to the training algorithm for every epoch.

The application of Hyperband in this dissertation will use the early stopping strategy. It is important to understand that Hyperband's early stopping can have downsides. The deficiency of early stopping is shown on the subsequent example.

Figure 12 displays the loss over epochs for two different configurations of a ML model. Depending on where the early stopping occurs, different scenarios are obtained. It is unclear to determine which of the two configurations performs better when the fitting stops at b epochs. Furthermore, if the training stops before the b epochs mark, for example at a epochs, it will be decided that the configuration drawn in blue outperforms the configuration drawn in red, which is not technically correct because red converges to a better point as more epochs passes, for example at c epochs.

Figure 12 – Deficiency of Early Stopping



Source: Author

Hyperband builds a strategy to mitigate this problem. The algorithm requires three main components for it's execution: an array c of the configurations being tested, a factor f of which the configurations will be dropped over time and a total number of epochs e. The number of epochs is the most important resource, which will be allocated equally for each iteration of the algorithm. At the beginning, the number of configurations is high and therefore the model will be trained with a low number of epochs. As the iterations passes, the number of configurations remaining decreases, and the model trains with a higher number of epochs.

A good didactic example is shown on Table 3. Hyperband is selected to run over a set of 32 configurations, with a factor $f = 2$ and total epochs $e = 1000$. After each iteration

of the algorithm, the bottom half of the configurations will be dropped, this is the factor's role. This essentially means that the number of iterations is $\log_f length(c) - 1$, which in this case is $\log_2 32 - 1 = 5$. The number of iterations is used to split the total number of epochs, resulting in 200 total epochs by iteration. This number is split equally to train the model for every configuration. Naturally, this means that the best configuration will have outperformed most of it's peers on many different levels of early stopping.

The configurations dropped on early stages does not have the opportunity to have their respective models to be evaluated for a higher amount of epochs. Although Hyperband's strategy mitigates the problem, it is still recommended to run the algorithm multiple times to reduce the chances of early stopping causing a good configuration to be improperly excluded.

Table 3 – Example of Hyperband Resource Allocation

Iteration Number	1	2	3	4	5
Configurations Remaining	32	16	8	4	2
Resource Allocated	200	200	200	200	200
Epochs per Model	$\sim$6	$\sim$12	25	50	100

Source: Author

4.4.1 Parameter Tuning Strategy

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 (HEIDEN; PARPINELLI, 2021). 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.

4.5 EVALUATION METRICS

During the conception and experimenting phases of the model, a number of evaluation metrics are used: RMSE, MAPE and Conditional Value-at-Risk (CVaR). MAPE is applied for

the model's training, RMSE is used to evaluate the results accuracy and CVaR is complementary to the portfolio selection experiment, being the metric used for risk calculation.

4.5.1 Root Mean Squared Error (RMSE)

RMSE is defined as the square root of the average squared error e, given by Equation 17:

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=0}^{n} e_i^2} \tag{17}$$

The majority of ML algorithms use Mean Squared Error (MSE) instead, as it's faster to compute and easier to manipulate than RMSE. The reason RMSE is used to evaluate the results of the model on this research is the non-scalability to the original error of MSE. Squaring the error without computing it's square root later produces a value not scaled to the original scale, resulting in skewed values for the predictions' accuracy.

4.5.2 Mean Absolute Percentage Error (MAPE)

MAPE is a derivative of Mean Absolute Error (MAE), given by Equation 18:

$$\frac{100}{n} * \sum_{i=0}^{n} |\frac{y_i - x_i}{x_i}| \tag{18}$$

where y_i is the predicted value for timestamp i and x_i is the real value for the timestamp i.

MAPE is generally used as a loss function in model evaluation, due to it's low sensitivity to highest errors, the outliers. It's important to understand that the metric can't be used if the series of expected values contain zero values, because it would generate a division by zero. This isn't a problem on the scope of this research, because the stock prices never equal zero.

4.5.3 Continuous Value-at-Risk

CVaR is not a metric used on the ML context. CVaR is a portfolio risk measure and, in this research, is only used to help selecting the portfolios after the entire model is trained and the predictions are generated.

As defined by (ROCKAFELLAR; URYASEV, 2000), let $f(w,r)$ be a loss function associated with array $w \subset W \in \mathbb{R}^k$, representing the investment proportions of a portfolio and a random vector $r \in \mathbb{R}^T$, which represents the return values of the stocks in the portfolio. For each w, a loss function $f(w,r)$ has a probability distribution in $\mathbb{R}$ inducted by r, denoted $p(r)$. Therefore, the probability that $f(w,r)$ does not exceed a certain value ζ is given by:

$$\Psi(w,\zeta) = \int_{f(w,r)\leq\zeta} p(r)\,\mathrm{d}r \tag{19}$$

Fixing a certain value w, we have that, as a function of ζ, $\Psi(w,\zeta)$ is the cumulative distribution function of the losses associated with w. The function $\Psi(w,\zeta)$ is not decreasing with respect to ζ and is also assumed to be continuous with respect to ζ, for the sake of simplicity. This continuity results from the loss function $\Psi(w,\zeta)$ and the probability density $p(r)$.

With a significance level $\alpha \in (0,1)$, the values α-VaR and α-CVaR for a random variable representing the loss related to w, are denoted by $\zeta_\alpha(w)$ and $\phi_\alpha(w)$ and respectively defined by:

$$\zeta_\alpha(w) = \min\{\zeta \in \mathbb{R} : \Psi(w,\zeta) \geq \alpha\} \tag{20}$$

$$\phi_\alpha(w) = (1-\alpha)^{-1} \int_{f(w,r)\geq\zeta_\alpha(w)} f(w,r)p(r)\,\mathrm{d}r \tag{21}$$

For better understanding, consider a simple example. For an $\alpha = 0.95$, let $\mathrm{CVaR}(95) = -0.269$. The interpretation of the value of CVaR indicates that in the 5% worst case scenarios, the average loss of the portfolio is 2.69% of the invested capital. Note that the 5% is derived from the significance level value, being $1-\alpha$.

4.6 DATABASE DESCRIPTION

To analyse the efficiency of the proposed model in a real investment scenario, the experiments are carried out with the historical series of the assets that make up the Top 50 of the S&P 500 index, which is designed to include companies with the highest market capitalisation, with first quarter (Q1) indicators of 2021. Market capitalisation indicates the market value of a company, reflected by the share price and the number of shares available. The assets considered in the experiments are shown in Table 4.

The historical series of the assets were extracted from the platform Yahoo Finances and the news were extracted from the API of the newspaper The New York Times, more specifically the Article Search API. This service is publicly available as long as it is not used for commercial purposes. The resource provides many schemes, “Article” being the schema used on this research.

The period considered was from January 1st, 2016 to December 31st, 2020. For each asset in each year, the 500 most relevant articles from the news desks “finance” and “business” were considered. The order of relevance is given by the printing page, and the smaller the page number, the greater the relevance. The search terms used were the company symbol on the stock exchange and its respective name.

Some companies have less exposure than others, and the amount of news in some years does not reach the total of 500. Table 5 illustrates the occasions in which a company did not have 500 news in the year on the finance section in the newspaper The New York Times. Each table cell represents the amount of news for the asset for the respective year. Assets AAPL, AMZN, BAC, DIS, FB, GOOG, GOOGL, JNJ, MCD, MSFT, NFLX, T, V and WMT have 500 news every year and are therefore not listed.

Table 4 – Assets Considered

Ticker	Represented Asset
AAPL	Apple
ABBV	Abbvie
ABT	Abbott Laboratories
ACN	Accenture
ADBE	Adobe
AMZN	Amazon
AVGO	Broadcom
BAC	Bank of America
BRK-B	Berkshire Hathaway
CMCSA	Comcast
COST	Costco
CRM	Salesforce
CSCO	Cisco
CVX	Chevron
DHR	Danaher
DIS	Disney
FB	Facebook
GOOG	Alphabet
GOOGL	Alphabet
HD	Home Depot
INTC	Intel
JPM	JP Morgan & Chase
JNJ	Johnson & Johnson
KO	Coca-Cola
LLY	Eli Lilly
MA	Mastercard
NEE	Nextera Energy
NFLX	Netflix
MCD	McDonalds
MDT	Medtronic
MRK	Merck & Co
MSFT	Microsoft
NKE	Nike
NVDA	Nvidia
ORCL	Oracle
PEP	PepsiCo
PFE	Pfizer
PG	Procter & Gamble
PM	Phillip Morris
PYPL	PayPal
T	AT&T
TMO	Thermo Fisher Scientific
TSLA	Tesla
TXN	Texas Instruments
UNH	UnitedHealth
V	Visa
VZ	Verizon
WFC	Wells Fargo
WMT	Walmart
XOM	Exxon Mobil

Source: Author

Table 5 – Assets with Less Exposure

Ticker	2016	2017	2018	2019	2020
ABBV	113	101	110	99	83
ABT	125	113	140	110	166
ACN	92	80	114	126	185
ADBE	170	146	181	156	152
AVGO	73	134	1113	134	93
BRK-B	171	217	219	200	222
CMCSA	250	270	320	192	190
COST	156	180	165	167	392
CRM	160	124	202	160	175
CSCO	157	134	153	142	135
CVX	180	167	194	196	186
DHR	101	86	65	117	97
HD	232	277	285	242	313
INTC	286	345	379	258	207
JPM	500	392	427	385	397
KO	332	300	267	262	221
LLY	120	112	131	131	176
MA	137	131	161	155	159
MDT	97	86	73	110	111
MRK	124	78	116	45	72
NEE	56	67	33	49	81
NKE	342	343	424	464	377
NVDA	160	116	134	125	124
ORCL	301	251	231	263	331
PEP	141	150	163	141	138
PFE	249	167	177	154	500
PG	143	181	153	151	151
PM	168	150	172	158	135
PYPL	232	168	190	180	171
TMO	32	45	87	62	25
TSLA	388	451	500	459	417
TXN	172	165	169	163	191
UNH	135	146	115	113	124
VZ	372	313	295	256	228
WFC	500	394	333	277	333
XOM	279	338	231	181	174

Source: Author

5 EXPERIMENTS AND ANALYSIS

Different tests are performed to analyse the general performance of the model, aiming to establish an adequate adjustment of the model's hyperparameters, verify the importance of using news sentiments as an indicator in the model and verify the longevity of the accuracy of the predictions made by the model.

The model was implemented using the Python programming language and all experiments were performed in a controlled environment, on 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 operating system with *kernel* 4.8.10.

5.1 NEWS' SENTIMENT CALCULATION

The first stage of experimentation is the news' sentiment calculation. For each of the news from the 5 years, from 2016 to 2020, their respective sentiment is calculated with the VADER model, implemented exactly as exposed by the creators in (HUTTO; GILBERT, 2014).

At the end of the process, a sentiment value is established for each closing date in the period from 2016 to 2020 for each of the assets. For news published on dates when there is no stock market circulation, their sentiment values are carried forward to the nearest closing date. For example: the sentiment of a news item published on Saturday, when there is no trading session, will influence the decisions made on the next trading day, in this case Monday. For closing dates in which there was no news published for the asset, neutral sentiment is assigned $s = 0$.

5.2 TUNING THE MODEL'S HYPERPARAMETERS

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 Sections 4.4 and 4.4.1. 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.

5.2.1 Results for the Model without Sentiments

The five best configurations generated for the model without sentiments are presented on Table 6, from best (1) to worst (5). The model was fitted for 100 epochs using these configurations 50 times each, having the value for MAPE calculated. The mean of the MAPE values for the training split for the 50 trials of each configuration is shown on Figure 13.

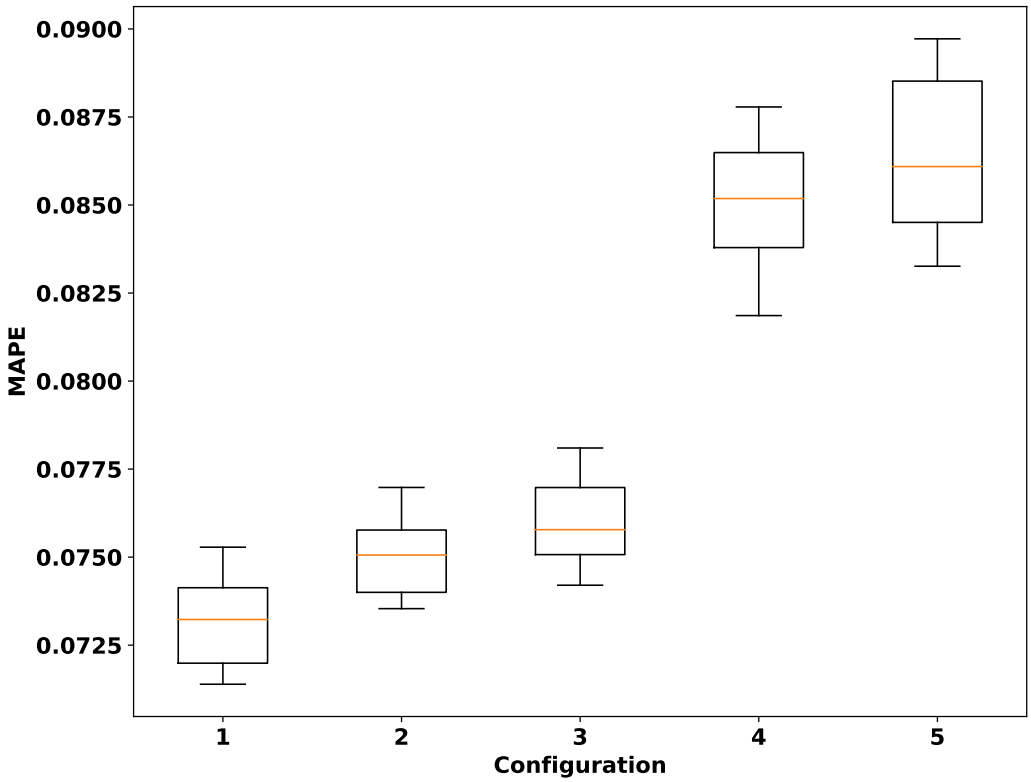
Configurations (1), (2) and (3) stand out from (4) and (5), specially configuration (1), which has the best results for MAPE. An one-factor ANOVA test is applied on the samples for configurations (1), (2) and (3) to verify if the difference between samples are statistically

Table 6 – 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

Source: Author

Figure 13 – MAPE for the 5 Best Configurations for the Model Without Sentiments



Source: Author

significant. Table 7 presents the rest results, which suggests that at least one treatment is significantly different with a significance level of $\alpha = 0.05$. To identify the difference between each treatment, a post-hoc test such as the Tukey HSD is indicated on this situation.

Table 7 – One-factor ANOVA for Configurations 1, 2 and 3

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			

Source: Author

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 $Q_{\alpha=0.01}^{k=3,df=147} = 4.1850$

and $Q_{\alpha=0.05}^{k=3,df=147} = 3.3487$ are obtained, respectively. To find the value for the Tukey HSD Q statistic, the Equations 22 and 23 are calculated.

$$Q_{i,j} = \frac{|\bar{x_i} - \bar{x_j}|}{s_{i,j}} \tag{22}$$

$$s_{i,j} = \frac{\sigma_\varepsilon}{\sqrt{H_{i,j}}} \tag{23}$$

$H_{i,j}$ 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 8, 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.

Table 8 – 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

Source: Author

5.2.2 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 9, from best (1) to worst (5). The MAPE for the training split is calculated on the same way and is shown on Figure 14.

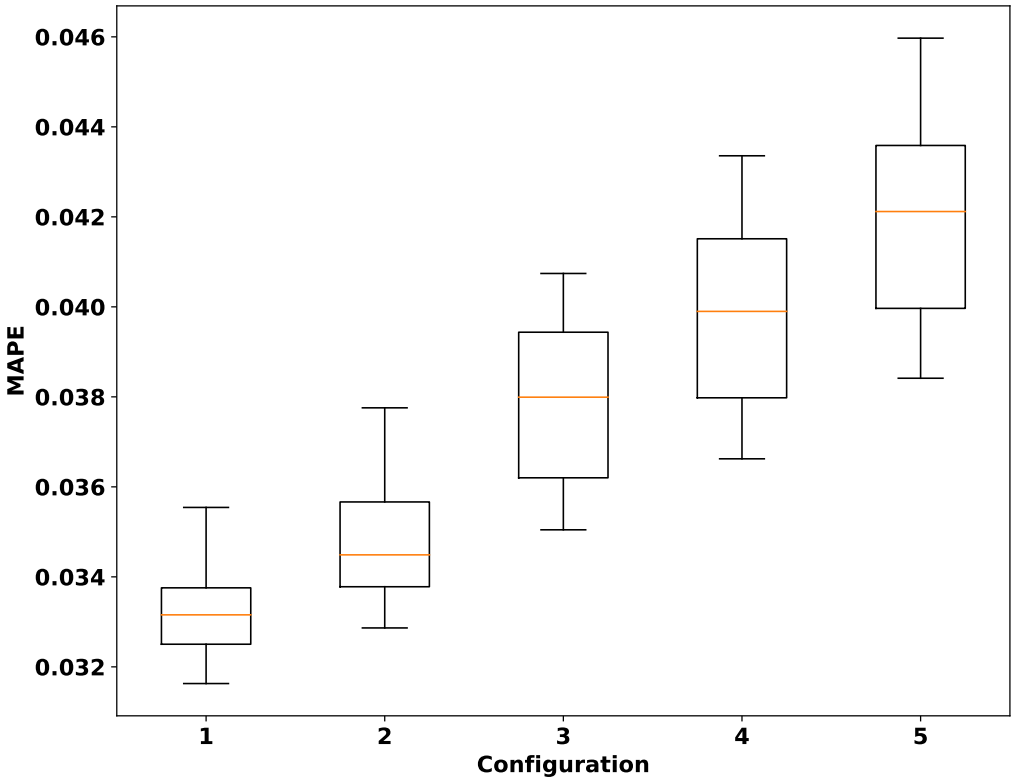
Table 9 – Best Configurations Generated by Hyperband for the Model with Sentiments

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

Source: Author

It looks like the best configuration outperforms the remaining configurations by quite a margin. Both the minimum value and the median are better for configuration (1), compared to any other configuration. Selecting configurations (1) and (2), the one-factor ANOVA test is applied to verify if the difference between the samples are statistically significant. Table 10

Figure 14 – MAPE for the 5 Best Configurations for the Model With Sentiments



Source: Author

Table 10 – 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			

Source: Author

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 these values, the confidence limits for the pair of configurations are set and the Tukey HSD Q are calculated, using Equations 22 and 23.

The final result is shown on Table 11. 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.

Table 11 – Tukey HSD for Configurations (1) and (2)

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

Source: Author

5.3 ANALYSIS OF SENTIMENT INDICATOR

The first tests after establishing the parameters of the model aim to investigate the influence of applying news sentiments to historical series to make predictions. The model, with parameters referring to the configuration (1) on Table 6, was executed 50 times without the sentiment attribute and the model with parameters referring to the configuration (1) on Table 9 was also executed 50 times, but this time with the sentiment attribute, on the dataset containing the 5 years of series historical and news stories. The RMSE was calculated at the end of each run for each set of assets, as well as the average of the RMSE for the stock prices normalised using min-max feature scaling, represented by RMSE-N. The objective of normalising the stock prices 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 and RMSE-N values calculated for each asset are shown in Table 12.

The results show superiority of the model using sentiment as a feature, having a better performance for all analysed stocks. The normalised RMSE values are similar for all stocks, indicating the model has a good generalisation and works well for any of the analysed stocks.

To confirm the results of Table 12, in Figure 15, the *boxplots* referring to RMSE-N values for the 50 assets are presented, using the model without and with the sentiment feature. It is noted that the average of the RMSE-N is approximately three times higher for the model without sentiments, certifying the effectiveness of using sentiment as a feature in the model.

Figure 15 – RMSE-N for the 50 Stocks for the Model With and Without Sentiment Feature

Source: Author

Table 12 – Comparison Between Models With and Without Sentiment Feature

	Without		With	
Ticker	RMSE	RMSE-N	RMSE	RMSE-N
AAPL	14.96988	0.16217	2.95058	0.03641
ABBV	3.84142	0.09216	1.97353	0.04734
ABT	7.30677	0.14920	2.52518	0.05156
ACN	18.22854	0.15730	4.64776	0.04010
ADBE	59.37529	0.23952	11.81055	0.04764
AMZN	333.79323	0.17517	107.42701	0.05637
AVGO	20.08617	0.07489	9.03264	0.03368
BAC	2.89823	0.18811	0.83580	0.05210
BRK-B	13.48688	0.19654	4.65646	0.06785
CMCSA	2.31630	0.11541	0.92951	0.04631
COST	35.84749	0.32205	5.50563	0.04946
CRM	27.59200	0.17471	7.20316	0.04560
CSCO	2.14452	0.13646	1.13374	0.06998
CVX	8.96068	0.16035	3.39546	0.05146
DHR	21.40698	0.17426	4.19363	0.03413
DIS	3.97419	0.04160	3.72190	0.03896
FB	16.72610	0.10427	9.20586	0.05738
GOOG	126.48249	0.16468	34.28603	0.04464
GOOGL	131.67108	0.17081	37.86126	0.04911
HD	12.02495	0.08458	6.00837	0.04226
INTC	5.88535	0.25737	1.50322	0.06388
JNJ	4.47591	0.11604	2.72195	0.07057
JPM	10.87005	0.19586	3.35437	0.05880
KO	1.15727	0.05499	1.27213	0.06043
LLY	12.46575	0.23789	6.06463	0.11573
MA	28.36808	0.17366	8.47512	0.05188
MCD	8.05574	0.08461	4.68446	0.04920
MDT	3.13823	0.06753	2.35347	0.05126
MRK	3.71097	0.18300	1.46756	0.06483
MSFT	21.34635	0.21505	4.32860	0.04691
NEE	5.98785	0.17395	1.84443	0.05358
NFLX	53.80244	0.20258	14.23824	0.05361
NKE	7.00174	0.08748	3.09277	0.03864
NVDA	15.99253	0.16485	3.90849	0.04028
ORCL	2.63314	0.10840	1.09154	0.04493
PEP	3.79264	0.09034	2.30927	0.05540
PFE	0.86237	0.05463	0.83454	0.05287
PG	8.55929	0.18623	2.39174	0.05203
PM	9.99516	0.33774	1.63356	0.05437
PYPL	21.10724	0.13417	5.07368	0.03225
T	1.36993	0.11346	0.64725	0.05044
TMO	51.02157	0.19514	9.12856	0.03491
TSLA	87.45676	0.09386	38.98505	0.06502
TXN	15.22252	0.21595	3.54928	0.05035
UNH	24.94108	0.15182	7.33508	0.04465
V	13.68782	0.18331	4.50377	0.06031
VZ	1.52807	0.12535	0.95294	0.07817
WFC	3.62921	0.13376	1.21531	0.03780
WMT	11.84669	0.24476	2.48174	0.05127
XOM	6.53585	0.21062	1.79656	0.04595

Source: Author.

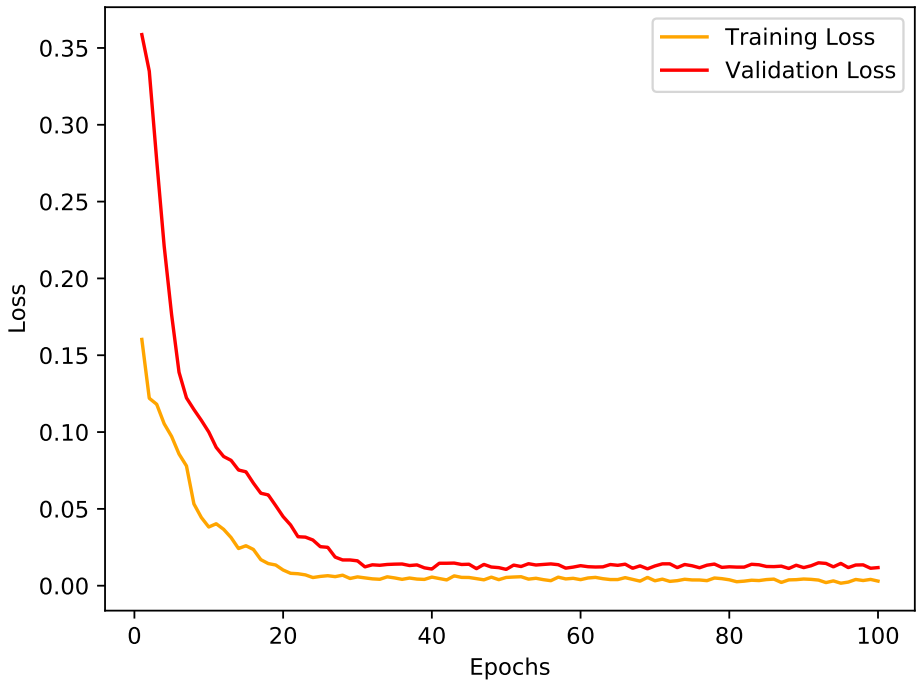
5.4 MODEL VALIDATION

With the model architecture and parameters defined, the model validation step is performed. The running procedure and creating the model input samples are explained in Section 4.3. To validate the selection of the hyperparameters, the loss curves in the training and validation of the model are investigated.

Loss is defined as the penalty over a bad prediction and is important to guide the model's training. When the prediction generated for a sample is the exact value expected, the loss for that specific sample is zero; otherwise, the loss is greater than zero and can be measured. Training a model has the objective of generating predictions with low loss across all the training samples.

The training and validation and loss curves, although they seem simple, have valuable information about the performance of the model. Figure 16 shows the average training and validation losses, measured using MAPE, for 50 runs of the final model, being the model with sentiments as a feature tuned with the best configuration resulted from Hyperband.

Figure 16 – Training and Validation Losses



Source: Author

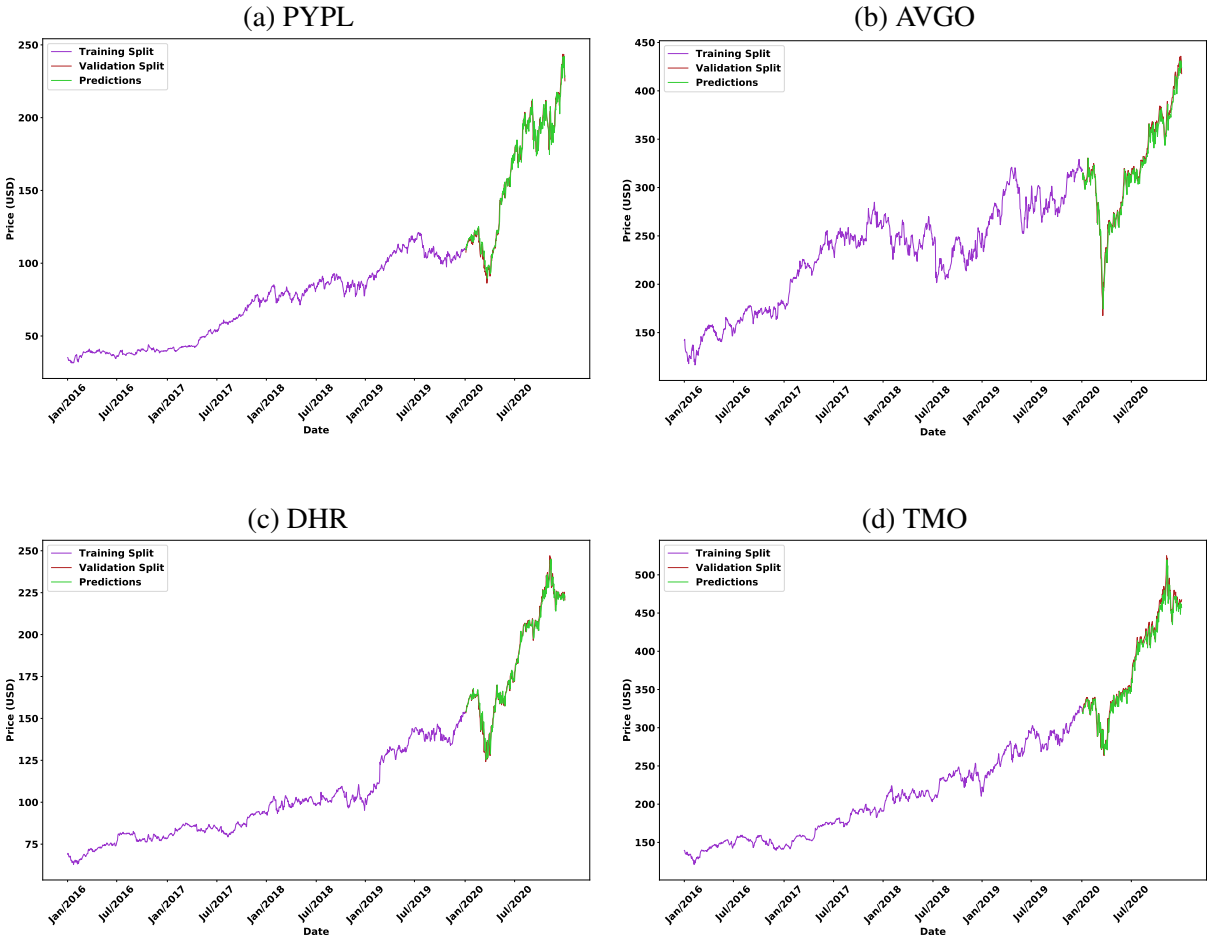
The results demonstrate that the two curves decay to a point of stability and have a small gap between them. The first fact indicates that the model has been sufficiently trained to generalise satisfactorily with the received data, and continuing training will eventually lead to overfitting. The second fact is expected, given that the model losses are smaller for the training data than for the validation data. This only characterises a problem if the gap is large. The results indicate that the model is well-fitted, but adding more data to the model would require further analysis. It is extremely important not to complicate the model architecture more than

necessary, so that it properly generalises the predictions on data not seen during training, which characterises the final stage of testing the model.

5.5 PREDICTING STOCK PRICES INTO THE FUTURE

The model's biggest challenge is to predict future stock prices, a scenario in which the model needs to operate with limited data. The question to be answered is how far into the future it is possible to extract predictions for stock prices with good accuracy. The following analysis is based on a subset of the analysed 50 assets, the 4 assets with the lowest normalised RMSE and the 4 assets with the highest normalized RMSE exposed in Table 12, which represents the best and worst performing assets, respectively. The idea is to illustrate the worst and best scenarios and abstract the intermediate scenarios, so that the analysis does not become too extensive. Figures 17 and 18 show the validation of the 4 assets with the best and worst performance, respectively.

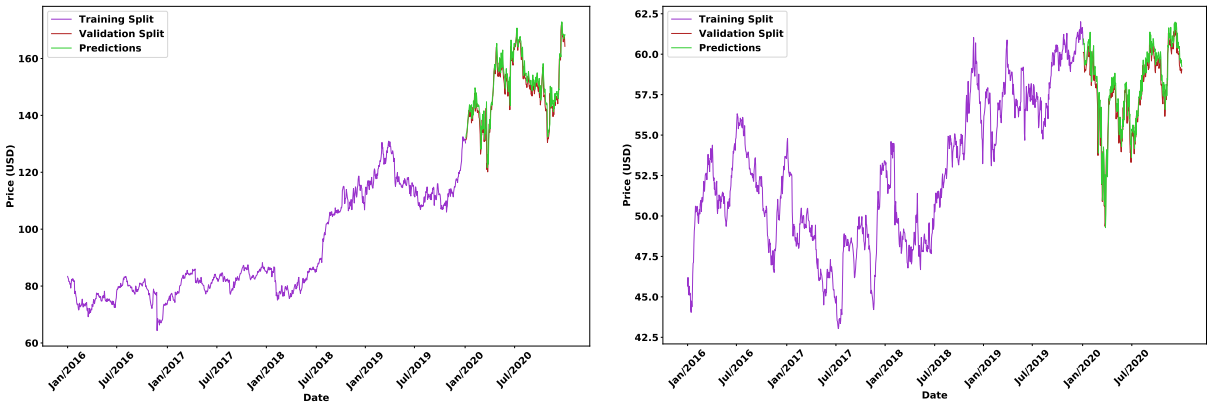
Figure 17 – Validation for Tickers with Best Performance



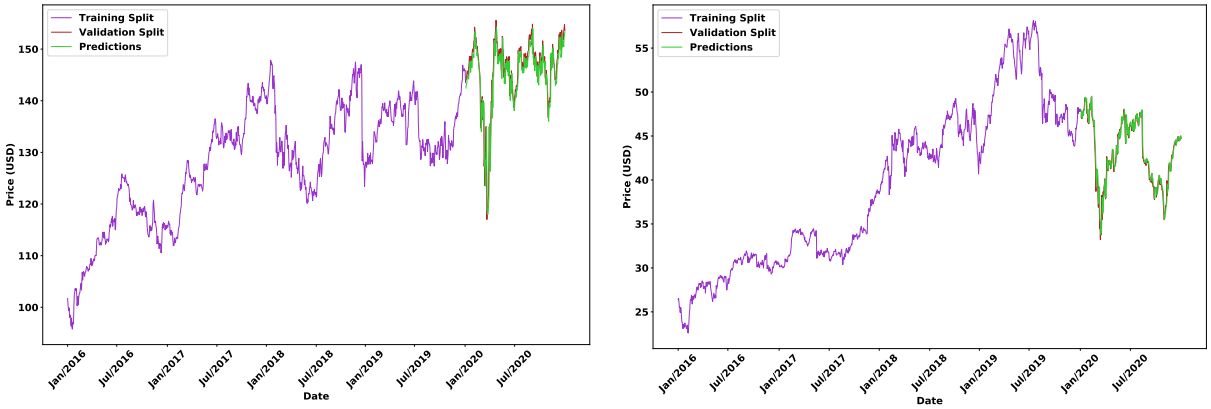
Source: Author.

Figure 18 – Validation for Tickers with Worst Performance

(a) LLY (b) VZ



(c) JNJ (d) CSCO



Source: Author.

Realistically, it is expected for the model to present better performance for some tickers than others. However, according to the illustrations in Figure 18, even the assets with the worst performances still have a very satisfactory validation curve, not far from the real values of the stock price in the period. It should also be noted that the model is able to distinguish the most different scenarios, be it uphill (Figure 17 (a)), downhill (Figure 18 (d)) or instability (Figure 18 (b)).

To verify the model's potential to predict prices outside the testing period, which ends on the last trading day of 2020, the first 50 prices for the year of 2021 are generated and compared with the actual closing value, values that are completely outside the scope of model training. In the same runs that generated the validation curves, the predictions were made. The average values of the RMSE and the normalised RMSE for the predictions of each asset are shown in Table 13.

Table 13 – Comparison Between Mean Values for RMSE and Normalised RMSE for Training and 50-day Future Predictions of Each Ticker

	Training		Predictions	
Símbolo	RMSE	RMSE-N	RMSE	RMSE-N
AAPL	2.95058	0.03641	11.55766	0.43611
ABBV	1.97353	0.04734	9.57022	0.94908
ABT	2.52518	0.05156	16.87315	0.85740
ACN	4.64776	0.04010	17.49767	0.71989
ADBE	11.81055	0.04764	28.53415	0.35472
AMZN	107.42701	0.05637	492.56845	1.15072
AVGO	9.03264	0.03368	110.85447	1.65071
BAC	0.83580	0.05210	3.70779	0.44360
BRK-B	4.65646	0.06785	20.97643	0.57265
CMCSA	0.92951	0.04631	5.23600	0.55155
COST	5.50563	0.04946	20.99689	0.31010
CRM	7.20316	0.04560	17.06722	0.39452
CSCO	1.13374	0.06998	1.81800	0.32509
CVX	3.39546	0.05146	6.82367	0.24973
DHR	4.19363	0.03413	26.13238	0.78150
DIS	3.72190	0.03896	41.70134	1.07256
FB	9.20586	0.05738	36.57719	1.00459
GOOG	34.28603	0.04464	319.24492	0.79797
GOOGL	37.86126	0.04911	309.47533	0.78201
HD	6.00837	0.04226	20.04585	0.59678
INTC	1.50322	0.06388	8.50004	0.55860
JNJ	2.72195	0.07057	14.16611	0.87682
JPM	3.35437	0.05880	17.39670	0.56786
KO	1.27213	0.06043	6.28473	1.39544
LLY	6.06463	0.11573	40.42931	0.84523
MA	8.47512	0.05188	21.22256	0.30879
MCD	4.68446	0.04920	8.17022	0.52883
MDT	2.35347	0.05126	4.12996	0.48973
MRK	1.46756	0.06483	8.40024	0.70439
MSFT	4.32860	0.04691	11.58663	0.35621
NEE	1.84443	0.05358	6.21476	0.39933
NFLX	14.23824	0.05361	91.13491	0.97983
NKE	3.09277	0.03864	22.42305	1.40680
NVDA	3.90849	0.04028	35.81343	0.95906
ORCL	1.09154	0.04493	4.70305	0.38606
PEP	2.30927	0.05540	12.05342	0.79192
PFE	0.83454	0.05287	3.34325	0.87950
PG	2.39174	0.05203	8.20877	0.48418
PM	1.63356	0.05437	3.64861	0.41451
PYPL	5.07368	0.03225	87.61025	1.11321
T	0.64725	0.05044	1.30048	0.64139
TMO	9.12856	0.03491	44.27727	0.56151
TSLA	38.98505	0.06502	530.61781	1.65771
TXN	3.54928	0.05035	17.70914	0.91743
UNH	7.33508	0.04465	27.51210	0.68341
V	4.50377	0.06031	8.51731	0.25727
VZ	0.95294	0.07817	4.90258	1.11059
WFC	1.21531	0.03780	2.51342	0.24410
WMT	2.48174	0.05127	5.34038	0.25203
XOM	1.79656	0.04595	7.89744	0.38433

Source: Author.

The results of Table 13 are fundamental to answer the main research question, about the longevity of predictions. Note that, when working with data that the model did not use during training, the error increases greatly. Making predictions outside the scope of tests is the greatest difficulty of the model, so it is expected that the accuracy will be lower and that it will collapse as the predictions move away from the scope of tests, that is, the further in the future, the worse the quality of predictions.

In order to identify the longevity of the generated predictions, a simple portfolio selection strategy is applied. This 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 and return of the predictions generated. By calculating the risk and return, portfolios can be selected in three different ways: maximising returns, minimising risk and maximising return/risk ratio. Maximising returns is the most aggressive strategy, as the risk values are not considering at any point of the portfolio selection. In the other hand, minimising risk is the most conservative strategy. In order to add security to the portfolio, it's good practice to dilute the investment in a number of assets. This number represents the portfolio's cardinality (k). There's not a set in stone to find the value for k, but there's evidence that k should always be at least 3 (CHENG; GAO, 2015).

In this analysis, the portfolios are generated minimising the risk and with cardinalities 3, 5 and 7, to create a degree of diversification. The returns were calculated assuming an equal portion of investment in each stock being $1/k$ of the total capital. Figures 19, 20 and 21 display the predicted and real performance of the three portfolios. Alongside, the performance on the S&P Index is also exhibited as a baseline measure. The real portfolios represent the same configuration of assets selected on the predicted portfolios, but with the real stock prices for the time period analysed.

Many insights can be extracted from this experiment. The first positive fact is indicated by the baseline. In every scenario, the real portfolio generated better accumulated return than the baseline, behaviour also displayed by the predicted portfolio. 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's understandable and expected that the accuracy would decrease as the days passes, and the breakpoint in this model seem to be around day 35.

Figures 22, 23 and 24 presents the difference of the expected returns, given by the predicted value minus the real value. It's interesting to focus on the days with a positive difference in returns. In these cases, the predicted portfolio sinalises a significantly higher return than the real portfolio, which is a negative fact because this masks a negative portion of the investment. This seems to be balanced on the big picture, with the majority of closing days reflecting a lower

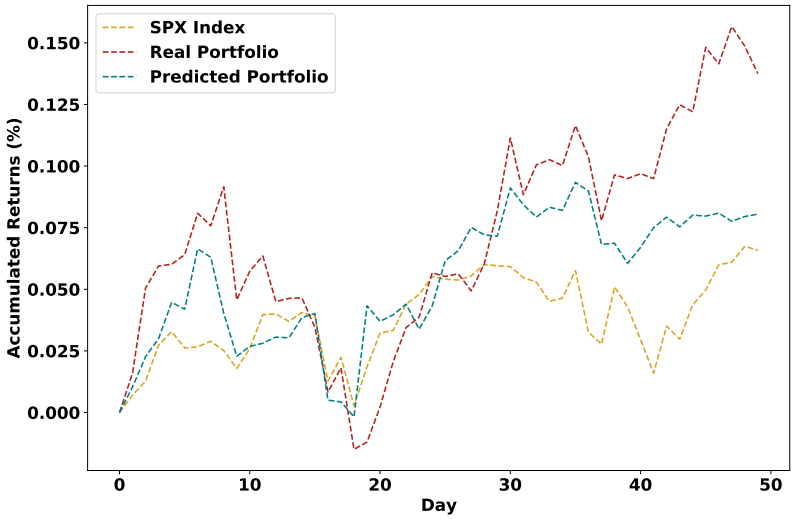


Figure 19 – Portfolio Performance & Comparison ($k = 3$)

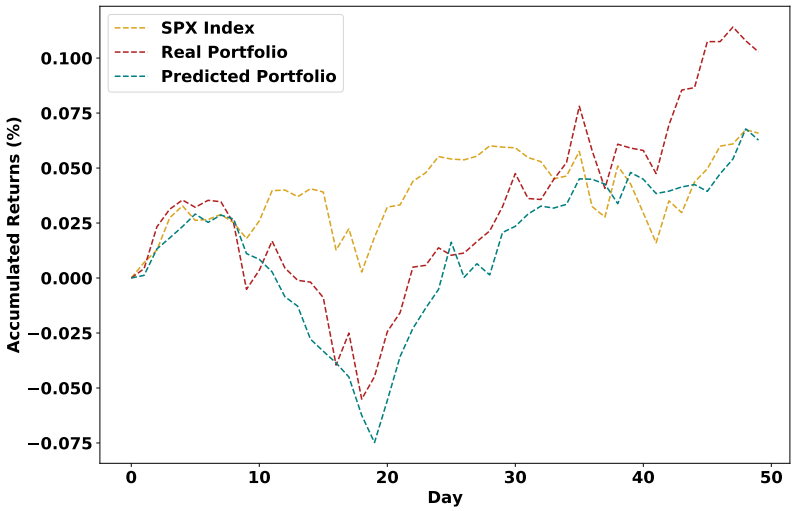


Figure 20 – Portfolio Performance & Comparison ($k = 5$)

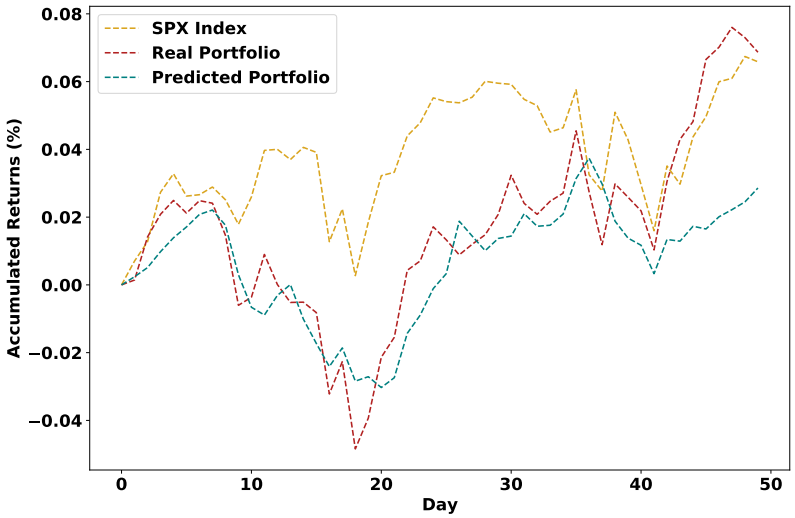


Figure 21 – Portfolio Performance & Comparison ($k = 7$)

return than the real value. It is important to understand that the worst scenarios are displayed when the difference is positive, because in these days the predictions indicate a return greater than the real portfolio, which leads the investor to believe the returns are better than they actually are.

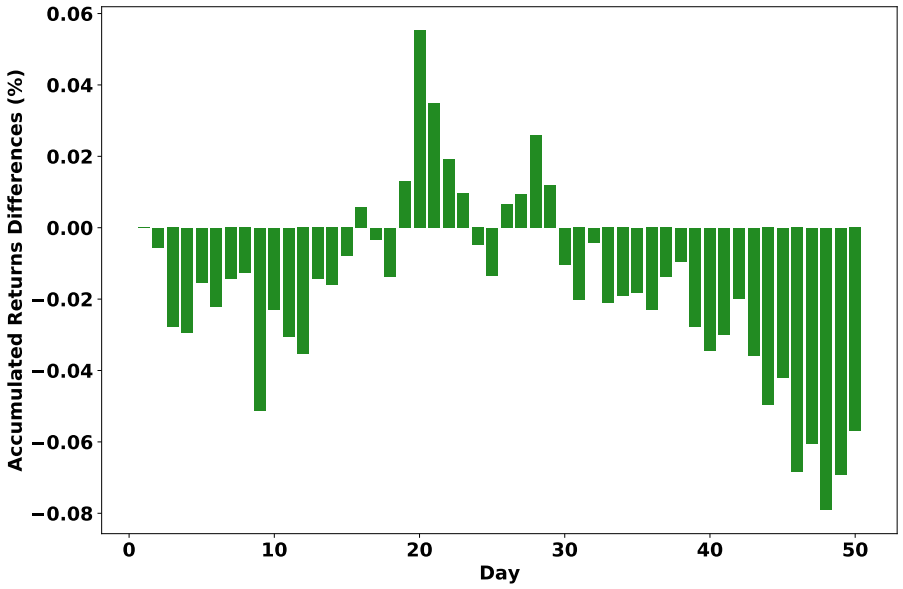


Figure 22 – Accumulated Returns Difference for Predicted and Real Portfolios ($k = 3$)

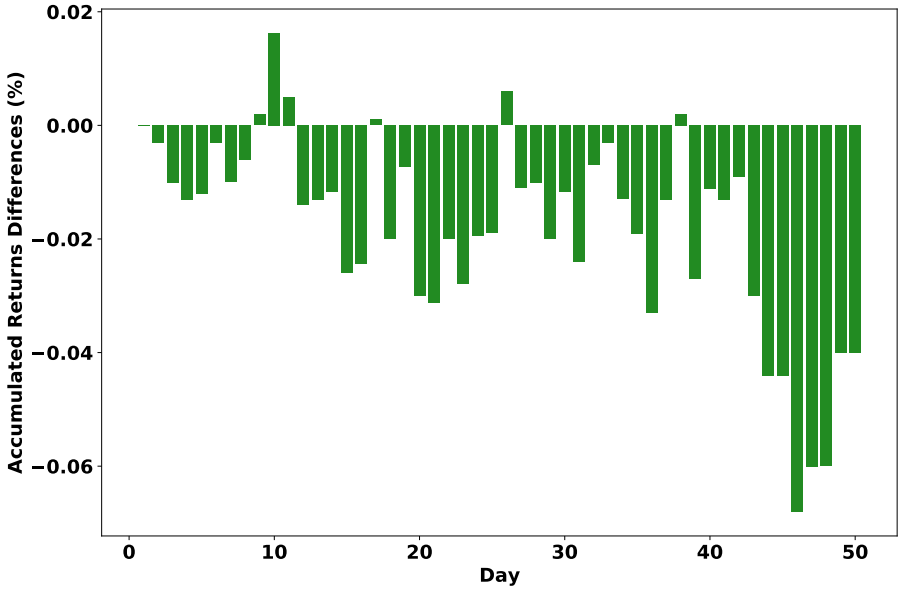


Figure 23 – Accumulated Returns Difference for Predicted and Real Portfolios ($k = 5$)

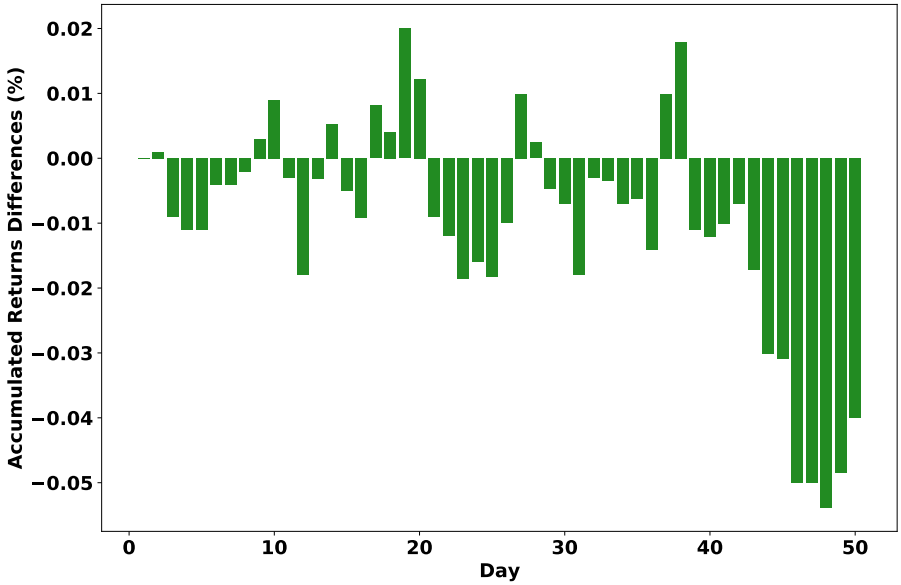


Figure 24 – Accumulated Returns Difference for Predicted and Real Portfolios ($k = 7$)

5.6 CHAPTER CONSIDERATIONS

This chapter presents the results from an extensive set of experiments, integrating every aspect of the implemented model evidence the positive and negative facets of the approach.

One of this research's objectives is to evaluate the importance of adding more indicators, in specific financial news sentiments, to the model and the impacts caused on it's accuracy. By tuning the model with and without considering the news sentiments feature using Hyperband, it was possible to properly compare their performance.

The results statistically confirm that adding financial news sentiments to the model as one of the features composing the samples increase the predictions' accuracy when compared to the model without this feature.

The model with sentiments and using the most optimal configurations of parameters found during the experiments was used to create predictions out of the training scope and with this predictions, a series of portfolios was generated and compared with the real performance of the same portfolios in a stock market investment scenario.

The results display that considering the scenario of application, the model is able to accurately predict prices up to 35 days into the future based on the analysis of difference of accumulated returns for the predicted portfolios and for their real values on that time period.

6 FINAL CONSIDERATIONS AND FUTURE WORKS

Stock price prediction is a very ambitious and complicated task, mainly because of the many factors that affect the stock exchange. With the great growth of the investment market today, the idea of being able to predict asset prices on the stock exchange is extremely coveted by any type of investor, and a tool with such power would help investors in the most difficult decisions in the market of investments.

This research presents a strategy to add one of the great indicators that influence stock prices in the investment market: the news associated with the companies that comprise the market. The sentiment analysis of the news collected from the New York Times was proposed using the VADER framework and the results were used as one of the features, together with historical stock prices data, of a stock price prediction model based on the LSTM architecture.

A set of experiments was carried out to verify the effectiveness of the proposed model, using a real investment scenario encompassing the assets belonging to the Top 50 of the NYSE S&P index, with indicators for the first quarter of 2021. The sentiment values of the news were calculated before the experimentation, so that they could be paired to the values of the historical prices of the assets.

The first stage of the tests was conducted to perform the adjustment of the model's hyperparameters. The algorithm Hyperband was implemented to perform the number of nodes on each LSTM layer, the model's learning rate and the window size for samples generation. From the test results, a statistical analysis was performed to choose the best configuration among all the listed ones.

With the hyperparameters established, the second stage of the tests starts. In order to confirm the relevance of the "sentiment" indicator, the model was run only with the historical series as input data attributes and, separately, with the historical series paired with sentiment values as input data attributes. The experiments indicate that the model using sentiment values excels, having a smaller error than the model without sentiment values for all 50 stocks analysed. A paper reporting the results obtained was published in the XV Brazilian Congress of Computational Intelligence (HEIDEN; PARPINELLI, 2021).

The third stage of the experiments analyses the validation of the model, using the parameters obtained in the first stage and building the input samples with the sentiment values, as a result of the analysis of the second stage. Analysing the model's loss curves, the results indicate that the model was sufficiently trained with the parameters used, for the inserted dataset. It is noteworthy that, if the input data is modified, the analysis must be redone.

Lastly, wrapping up the experiments, the model is used to predict the prices referring to the first 50 closing days of the trading session in a scenario outside the scope of tests. With the prediction values in hand, a portfolio selection strategy was implemented to simulate the application of the results in a real investment scenario, comparing the values generated with the real values, as well as the SPX index as baseline. The results indicate that the model is able to

generate a portfolio with consistent predictions up until around the 35th day into the future, mark where the accuracy of the predictions starts to collapse quickly.

The truth is that some movements in the financial market are completely unpredictable even from one day to the next. In prediction experiments, it is noticed that the error is progressively increasing and at some point the accuracy will collapse. The challenge is to make that point as far into the future as possible.

Even with display of good results, the research has areas for improvement. The five year period studied is enough for demonstration, but a more 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. Also, comparing the results generated by this LSTM model with another state-of-the-art technology, such as CNN, can provide useful insights regarding general performance, for example execution time, which was not in the scope of this research.

The performance increase generated by adding another feature to the model, being sentiments from financial news, fuel the possibility of the analysis of more indicators. A very interesting indicator used to guide buy and sell decisions in the investment market is dividends. It is very common for an investor to not make these decisions just based on a ticker's stock price, but also on the dividends paid by the company represented by the ticker. Studying a way to embed the dividends value as a feature on the model could improve the results even more. Other interesting indicators are trading volume and volatility index, although they are harder to measure.

Another possibility is focusing on short-time instead trying to push the predictions into the future, a strategy used by day-traders. Our model was designed to predict with accuracy as much in the future as possible, but this is not exactly a concern for day-traders. Day-trading is very popular amongst retail investors and is purely based on short-term market moves. Modifying the model to be applied in this context would certainly create a new interesting branch of analysis.

REFERENCES

ADAM, Klaus; MARCET, Albert; NICOLINI, Juan Pablo. Stock market volatility and learning. **The Journal of Finance**, Wiley Online Library, v. 71, n. 1, p. 33–82, 2016.

BEKETOV, Mikhail; LEHMANN, Kevin; WITTKE, Manuel. Robo advisors: quantitative methods inside the robots. **Journal of Asset Management**, Springer, v. 19, n. 6, p. 363–370, 2018.

BENGIO, Yoshua; SIMARD, Patrice; FRASCONI, Paolo. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BERALDI, Marcelo Vicente. **Robôs de investimento a partir de dados de redes sociais**. Dissertação (Mestrado) — Fundação Getúlio Vargas - Escola de Economia de São Paulo, 2020.

BLACK, Fischer; LITTERMAN, Robert. Asset allocation: combining investor views with market equilibrium. **Goldman Sachs Fixed Income Research**, Goldman Sachs & Co., v. 115, 1990.

BLICHFELDT, Bodil Stilling; ESKEROD, Pernille. Project portfolio management–there's more to it than what management enacts. **International Journal of Project Management**, Elsevier, v. 26, n. 4, p. 357–365, 2008.

BRAGA, Antônio de Pádua; FERREIRA, André Carlos Ponce de Leon; LUDERMIR, Teresa Bernarda. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC Editora Rio de Janeiro, Brazil:, 2007.

CÂMARA, Jackson Balthazar de Arruda et al. Diversificação entre classes de investimentos como estratégia para minimizar riscos e aumentar a rentabilidade em aplicações financeiras. **Congresso UFSC de Controladoria e Finanças & Iniciação Científica em Contabilidade**, 2014.

CHAN, Yue-cheong; CHUI, Andy CW; KWOK, Chuck CY. The impact of salient political and economic news on the trading activity. **Pacific-Basin Finance Journal**, Elsevier, v. 9, n. 3, p. 195–217, 2001.

CHENG, Feiyang et al. Does retail investor attention improve stock liquidity? a dynamic perspective. **Economic Modelling**, Elsevier, v. 94, p. 170–183, 2021.

CHENG, Runze; GAO, Jianjun. On cardinality constrained mean-cvar portfolio optimization. In: IEEE. **The 27th Chinese Control and Decision Conference (2015 CCDC)**. [S.l.], 2015. p. 1074–1079.

CREAMER, Germán G. Can a corporate network and news sentiment improve portfolio optimization using the black–litterman model? **Quantitative Finance**, Taylor & Francis, v. 15, n. 8, p. 1405–1416, 2015.

DING, Xiao et al. Deep learning for event-driven stock prediction. In: **Twenty-fourth international joint conference on artificial intelligence**. [S.l.: s.n.], 2015.

DU, Xin; TANAKA-ISHII, Kumiko. Stock embeddings acquired from news articles and price history, and an application to portfolio optimization. In: **Proceedings of the 58th annual meeting of the association for computational linguistics**. [S.l.: s.n.], 2020. p. 3353–3363.

FAIZAN, Muhammad Azfar. Multiobjective portfolio optimization including sentiment analysis. 2019.

FAMA, Eugene. The distribution of the daily differences of the logarithms of stock prices. **PhD diss., University of Chicago. Reprinted in the Journal of Business as "The Behavior of Stock Market Prices**, v. 38, n. 1, p. 34–105, 1964.

FAMA, Eugene F. Efficient capital markets: Ii. **The journal of finance**, Wiley Online Library, v. 46, n. 5, p. 1575–1617, 1991.

GERS, Felix A; SCHMIDHUBER, Jürgen; CUMMINS, Fred. Learning to forget: Continual prediction with lstm. **Neural computation**, MIT Press, v. 12, n. 10, p. 2451–2471, 2000.

GUPTA, Ishu et al. Hisa-smfm: Historical and sentiment analysis based stock market forecasting model. **arXiv preprint arXiv:2203.08143**, 2022.

HANAOKA, Gustavo. Seleção de carteiras de investimentos através da otimização de modelos restritos multiobjetivos utilizando algoritmos evolutivos. **Programa de Mestrado em Modelagem Matemática e Computacional**, CEFET-MG, 2014.

HARIHARAN, Gurushyam. **News Mining Agent for Automated Stock Trading**. [S.l.]: Citeseer, 2012.

HAYKIN, Simon. **Redes neurais: princípios e prática**. [S.l.]: Bookman Editora, 2007.

HEIDEN, Alexandre; PARPINELLI, Rafael. Applying lstm for stock price prediction with sentiment analysis. In: **Anais do 15 Congresso Brasileiro de Inteligência Computacional**. Joinville, SC: SBIC, 2021. p. 1–8.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HUTTO, Clayton; GILBERT, Eric. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: **Proceedings of the International AAAI Conference on Web and Social Media**. [S.l.: s.n.], 2014. v. 8, n. 1.

JIN, Zhigang; YANG, Yang; LIU, Yuhong. Stock closing price prediction based on sentiment analysis and lstm. **Neural Computing and Applications**, Springer, v. 32, n. 13, p. 9713–9729, 2020.

JING, Nan; WU, Zhao; WANG, Hefei. A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. **Expert Systems with Applications**, Elsevier, v. 178, p. 115019, 2021.

LIAGKOURAS, Konstantinos; METAXIOTIS, Konstantinos. Efficient portfolio construction with the use of multiobjective evolutionary algorithms: Best practices and performance metrics. **International Journal of Information Technology & Decision Making**, World Scientific, v. 14, n. 03, p. 535–564, 2015.

MAQSOOD, Haider et al. A local and global event sentiment based efficient stock exchange forecasting using deep learning. **International Journal of Information Management**, Elsevier, v. 50, p. 432–451, 2020.

MARKOWITZ, Harry. Portfolio selection. **The journal of finance**, Wiley Online Library, v. 7, n. 1, p. 77–91, 1952.

MCCULLOCH, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MITTAL, Anshul; GOEL, Arpit. Stock prediction using twitter sentiment analysis. **Standford University, CS229 (2011)**, v. 15, 2012.

NGUYEN, Thien Hai; SHIRAI, Kiyoaki; VELCIN, Julien. Sentiment analysis on social media for stock movement prediction. **Expert Systems with Applications**, Elsevier, v. 42, n. 24, p. 9603–9611, 2015.

PASCANU, Razvan et al. How to construct deep recurrent neural networks. **arXiv preprint arXiv:1312.6026**, 2013.

PATIL, Pratik et al. Stock market prediction using ensemble of graph theory, machine learning and deep learning models. In: **Proceedings of the 3rd International Conference on Software Engineering and Information Management**. [S.l.: s.n.], 2020. p. 85–92.

PAWAR, Anil Bhausaheb; JAWALE, MA; KYATANAVAR, DN. Fundamentals of sentiment analysis: concepts and methodology. In: **Sentiment analysis and ontology engineering**. [S.l.]: Springer, 2016. p. 25–48.

RANA, Masud; UDDIN, Md Mohsin; HOQUE, Md Mohaimnul. Effects of activation functions and optimizers on stock price prediction using lstm recurrent networks. In: **Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence**. [S.l.: s.n.], 2019. p. 354–358.

ROCKAFELLAR, R Tyrrell; URYASEV, Stanislav. Optimization of conditional value-at-risk. **Journal of risk**, v. 2, p. 21–42, 2000.

ROONDIWALA, Murtaza; PATEL, Harshal; VARMA, Shraddha. Predicting stock prices using lstm. **International Journal of Science and Research (IJSR)**, v. 6, n. 4, p. 1754–1756, 2017.

SCHMIDHUBER, Jürgen. Learning complex, extended sequences using the principle of history compression. **Neural Computation**, MIT Press, v. 4, n. 2, p. 234–242, 1992.

SCHUMAKER, Robert; CHEN, Hsinchun. Textual analysis of stock market prediction using financial news articles. **AMCIS 2006 Proceedings**, p. 185, 2006.

SMAGULOVA, Kamilya; JAMES, Alex Pappachen. A survey on lstm memristive neural network architectures and applications. **The European Physical Journal Special Topics**, Springer, v. 228, n. 10, p. 2313–2324, 2019.

SONG, Qiang; LIU, Anqi; YANG, Steve Y. Stock portfolio selection using learning-to-rank algorithms with news sentiment. **Neurocomputing**, Elsevier, v. 264, p. 20–28, 2017.

TABARI, Narges et al. A comparison of neural network methods for accurate sentiment analysis of stock market tweets. In: SPRINGER. **ecml pkdd 2018 Workshops**. [S.l.], 2018. p. 51–65.

VALLE-CRUZ, David et al. Does twitter affect stock market decisions? financial sentiment analysis during pandemics: A comparative study of the h1n1 and the covid-19 periods. **Cognitive Computation**, Springer, p. 1–16, 2021.

WALCZAK, Steven. Artificial neural networks. In: **Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction**. [S.l.]: IGI Global, 2019. p. 40–53.

XING, Frank; HOANG, Duc Hong; VO, Dinh-Vinh. High-frequency news sentiment and its application to forex market prediction. In: **Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)**. [S.l.: s.n.], 2020.

XU, Yumo; COHEN, Shay B. Stock movement prediction from tweets and historical prices. In: **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2018. p. 1970–1979.

YANG, Steve Y; MO, Sheung Yin Kevin; ZHU, Xiaodi. An empirical study of the financial community network on twitter. In: IEEE. **2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)**. [S.l.], 2014. p. 55–62.