Degree in Business Analytics

Bachelor's final project

ANALYTICAL FINANCE. APPLYING NLP and LLMs IN STOCK PRICE PREDICTION

Author
Elena Conderana Medem

Supervised by
María Coronado Vaca

Madrid
May 2025

Abstract

Stock price prediction remains a foundational and complex task in analytical finance, with the inherent volatility of markets and the limitations of technical indicators spurring the integration of sentiment analysis. Large Language Models (LLMs), such as encoder-only FinBERT, are garnering increased attention for their potential to develop more accurate and adaptive forecasting models in synergy with machine learning (ML) algorithms. This study investigates the impact of integrating FinBERT-derived sentiment from Reuters financial headlines alongside technical indicators on the predictive performance of Long Short-Term Memory (LSTM) models for IBEX-35 banking sector stock prices.

LSTM models were trained on an extensive set of technical indicators from May 2020 to May 2025, where closing price served as target variable of the research. Through extensive experimentation across various lookback windows and meticulous hyperparameter tuning, the findings consistently demonstrate that FinBERT-derived sentiment acts as a differential factor, capturing underlying market dynamics unobservable through traditional indicators alone, significantly improving LSTM model performance across metrics R^2 , $RMSE$, and MAE. The distinct contribution of this research lies in the application of LSTM models, Reuters news collection, and FinBERT sentiment extraction to the largely underexplored Spanish index, the IBEX-35, as well as solely including sentiment from company-specific news. Overall results underscore sentiment's significant potential for enhancing stock price prediction and warrant further exploration of its impact and interplay with other models within the IBEX-35.

Keywords: sentiment analysis; stock price prediction; encoder-only model; FinBERT; technical indicators; LSTM model; IBEX-35

Contents

List of Figures

List of Tables

Listings

Chapter 1

Forecasting the Unpredictable: Framing the Complexity of Stocks

What if the next financial shock is already buried in today's headlines? From seasoned professionals to casual traders, every investor faces the same uncertainty: how will stock prices fluctuate next? Markets rise and fall on the echoes of earnings reports, the tremors of geopolitical events, and the shifts of public sentiment. This unpredictable landscape turns forecasting into more than a mathematical challenge. It becomes a quest to understand human behavior, economic structure, and news impact.

Stock price prediction remains one of the most foundational and complex tasks in analytical finance (Kumbure et al. 2022). Volatility of stock prices is not the exception, but the rule (Chang et al. 2024) as illustrated by Figure 1.1. The graph depicts the market volatility of the IBEX-35 index in perspective, highlighting how disruptive events from political crises to scientific breakthroughs have repeatedly generated unpredictable price swings throughout the years. Hence, the real challenge for investors and researchers lies in designing adaptable forecasting models that are not only accurate, but, more importantly, capable of recognizing patterns and foreseeing market tendencies (Chang et al. 2024). In this direction, numerous statistical, ML, Deep Learning (DL), and hybrid approaches share a common goal: to find key variables that influence market dynamics with the potential to improve the chances of achieving higher risk-adjusted returns than the market (Chatterjee et al. 2021).

The pursuit of increasing predictive accuracy has led to the widespread adoption of ML and DL models in stock price prediction (D. Kumar et al. 2022). The ability of models like Support Vector Machines (SVM), Random Forests (RF) or LSTM models to uncover patterns in complex and noisy data, and to be nurtured with diverse input data makes them particularly appealing for these tasks (Chakravorty 2023; Chatterjee et al. 2021). Yet while these algorithms have greatly

Figure 1.1: Market volatility in perspective: The IBEX-35 index frequency of daily price movements over the past three decades. (BME 2017)

evolved since the introduction of neural networks (NN) for financial purposes in the early 1990s (Kamijo and Tanigawa 1990), their training on technical indicators often limits their ability to provide consistent robust forecasts, particularly when faced with unforeseen market shocks (Ma 2024). This limitation has motivated the integration of technical indicators with sentiment analysis derived from financial news, aiming to better capture market trends and achieve more accurate predictions (Oriol 2023).

In this context, the development of Natural Language Processing (NLP) has become crucial, enabling machines to understand human language and extract emotional cues from different financial texts such as news articles, earnings reports, and social media. Early sentiment analysis approaches based on lexicons and keyword frequency have been superseded by a new generation of DL sentiment models (Inserte et al. 2024). LLMs represent a significant leap forward, exhibiting a far greater precision in detecting subtle tone changes, recognizing the influence of the context and understanding complex linguistic structures compared to their predecessors (Inserte et al. 2024). However, the presence of specific jargon and nuances in the language requires further specialization in the sectors, which in the financial context has led to the emergence of tailored Financial Large Language Models (FinLLMs), such as FinBERT, FinGPT, and BloombergGPT (Li et al. 2023; Wang et al. 2024).

The synergy between traditional ML algorithms applied to numerical data and the ability of FinLLMs to extract underlying investor sentiment from financial news holds immense potential for the development of more accurate and adaptive forecasting models (Liu et al. 2023; Talazadeh and Perakovic 2024). Given that each market has unique characteristics due to its specific configuration, tailored approaches are necessary as even within a single market model performance can considerably deviate across companies within the same industry (Todorov and Sánchez-Lasheras 2023). Recognizing this phenomenon, this thesis aims to delve into the unexplored banking sector of the IBEX-35, which has been experiencing profound changes and shocks in recent years, such as the merger between Caixa and Bankia in 2021, or the takeover bid by BBVA for Sabadell in 2025. The objective of this research is to investigate and measure the differential impact of integrating financial news sentiment analysis, derived from the encoder-only FinBERT, a leading representative of foundational LLM architectures (Jun Gu et al. 2024), with a robust LSTM framework to improve stock price prediction accuracy within the Spanish financial market. The resulting analysis should help towards answering the question "Does integrating FinBERT-derived sentiment analysis improve the performance of LSTM models for IBEX-35 banking sector stock price prediction?"

1.1 From Hypothesis to Horizon: This Work's Purpose

The goal of this thesis is to analyze the impact of using sentiment in financial news extracted by a prominent LLM architecture, encoder-only through the FinBERT model, as input feature of an LSTM model to predict the stock prices of the companies in the banking sector within the IBEX-35 index. The first step consists of developing an LSTM model for each financial institution in the IBEX-35 using the historical price data of the previous five years to predict its short-term stock prices over a 1-day horizon, allowing to quantify the impact of sentiment analysis in daily predictions (Chen and Kawashima 2024).

After developing the benchmark model, a comprehensive dataset of news headlines related to both the IBEX-35 index and the specific banking companies comprising it (namely Banco Sabadell, Bankinter, Santander, BBVA, CaixaBank and Unicaja) will be collected from Reuters. News sentiment of the extended news dataset is extracted with the FinBERT model. The resulting sentiment metrics extracted from company-specific news will be correspondingly included as supplementary input features into the previously developed benchmark LSTM models, resulting in the creation of hybrid "LSTM+FinBERT" models for each banking institution.

As a last step, the initial models will act as benchmarks to assess the performance of the "LSTM+FinBERT" hybrid models. The resulting analysis should identify the different performance levels achieved by incorporating sentiment analysis from each LLM architecture, and quantify the impact of the inclusion of sentiment analysis in each of the cases, providing a clear answer to the research question "Does integrating FinBERT-derived sentiment analysis improve the performance of LSTM models for IBEX-35 banking sector stock price prediction?"

1.2 Motivation: Why Bring LLMs into the Market?

Stock exchanges are shaped not only by numerical data but by perception, narrative, and sentiment. News headlines and public commentaries, though not as fundamental as technical indicators, have an important impact on price movements and understanding of market trends (Oriol 2023). Any forecasting framework that seeks to reflect market reality in such a complex environment requires a deep understanding and integration of unstructured text data (R. Kumar and K S 2024). Traditional as well as modern models often rely purely on numerical features, remaining language an underutilized source of predictive insight (Liu et al. 2023).

The rise of LLMs, specifically of domain-specific variants like FinBERT, FinGPT, and BloombergGPT has made it possible to extract nuanced sentiment from financial text with unprecedented precision (Li et al. 2023). Their ability to interpret subtle shifts in tone and implications embedded in natural language is transforming qualitative content into structured input data for forecasting models (Liu et al. 2023). Despite their growing presence in numerous financial applications, the scope and unique composition of financial markets cause many potential applications of these LLMs to remain unstudied, a gap that this study will revise in section 2, focusing specifically on the Spanish stock market. While BERT approaches have been applied to the IBEX-35 (Consoli et al. 2022), the application of specialized FinLLMs in this market remains largely uninvestigated.

Furthermore, the distinct architectures of LLMs have led to different ways of processing textual data and extracting sentiment (Guo and Hauptmann 2024). Although generative pretrained transformer (GPT) frameworks are gaining prominence (Lee et al. 2025), this research focuses on encoder-only architectures, specifically on FinBERT, which has consistently demonstrated its ability to enhance stock price prediction across various indices (Jun Gu et al. 2024). Building on the limited exploration of the IBEX-35 and prior successes of BERT-based emotion classifiers in forecasting accuracy of this index (Consoli et al. 2022), this study aims to establish whether combining FinBERT with an LSTM model demonstrates superior ability to capture underlying market behavior within the novel IBEX-35 banking sector, compared to a plain LSTM benchmark.

1.3 How This Thesis Is Built and Reasoned

Although the technical implementation and results will be detailed in the following chapters, this section lays the conceptual and methodological foundation of the thesis. The thesis adopts a quantitative methodology rooted in a deductive reasoning approach. The deductive nature of the research derives from its structure, as the thesis is centered around quantifying the magnitude of the impact of the encoder-only FinBERT architecture on improving stock price predictions. To establish a comparative analysis, it is necessary to measure the change in accuracy with respect to a benchmark LSTM model trained solely on technical indicators: historical stock price, trading volume, highest price, lowest price, as well as trend and volatility measures (Chen and Kawashima 2024).

For this purpose, the use of a quantitative framework is particularly suitable, as it uses error metrics RMSE and MAE to objectively and rigorously compare the performance (Kim et al. 2023) of the benchmark and hybrid models. The hybrid models are created by incorporating sentiment scores as additional input features to the baseline LSTM models. Sentiment is extracted from IBEX-35 and its listed financial institutions relevant Reuters financial news headlines using FinBERT. This methodological approach is designed to answer the thesis' guiding question: "Does integrating FinBERT-derived sentiment analysis improve the performance of LSTM models for IBEX-35 banking sector stock price prediction?".

1.4 A Roadmap Through This Thesis

The thesis is structured into five main sections. The first presents the introduction, outlining the context of the problem, defining the research question, and setting the objectives that will guide the project. The second section is dedicated to reviewing the state of the art, focusing on the main lines of existing research on stock price prediction, sentiment analysis in finance, and the use of LLMs in combination with ML algorithms for forecasting purposes. The third part details the methodology, deepening on the selected techniques, LSTM, FinBERT, the data sampling and the overall implementation of the solution. The fourth block describes the obtained results, comparing the performance of the baseline model with the hybrid "LSTM+FinBERT" models and presenting the main findings on the impact of endcoder-only LLM architectures on stock price prediction in the

IBEX-35 banking sector. Lastly, comes the extraction of conclusions, derived from interpreting and discussing the practical implications of the results, as well as outlining lines of future work.

Chapter 2

Blueprints and Building Blocks: A Revision of the Evolving Architecture of Stock Price Prediction

2.1 The Elusive Crystal Ball: Navigating the Labyrinth of Stock Price Prediction

2.1.1 Decoding the Market's Pulse: An Introduction to the Stock Price Enigma

Financial markets are complex systems, where stock prices are influenced by company performance, economic indicators, market sentiment, and global events among others. The derived volatility of the confluence of these factors makes stock price prediction tasks notoriously challenging. The failure of traditional approaches based on technical analysis, which solely focus on historical price data and trading volume, to account for broader factors like market sentiment, investor psychology, and macroeconomic shifts, significantly impacts the accuracy and robustness of their predictions (Liu et al. 2023).

The limited accuracy of technical analysis can be explained with the adaptive market hypothesis (AMH), developed by Andrew Lo in 2004. The AMH builds on the efficient market hypothesis (EMH) developed by Eugene F. Fama in 1970, which suggested that stock prices reflect all available information and can therefore not be predicted using existing information (Lo 2004). Fama distinguished three forms of market efficiency. The weakest form presumes historical prices are of no help when forecasting future prices, since that information is already captured by the current stock price (Fama 1970). Hence, no approach based on technical analysis can be expected to consistently outperform the market (Kumbure et al. 2022). The strongest form states that stock prices fully reflect all publicly available information, including fundamental data about economic conditions, political events, or company-specific information, as well as insider information. This implies that no investor, regardless of their insider knowledge or active management, can achieve higher returns than the market (Fama 1970).

Over time, the EMH has been challenged by different calendar, technical and fundamental market anomalies, such as turn-of-year, low price to sales or weekend effects which generate momentum, overreactions and underreactions in the market (Latif et al. 2011). Even Eugene Fama himself considered the strong-form market efficiency rather extreme and not to accurately describe the real-world functioning (Fama 1970). The doubts raised on the ability of models based on investor rationality to capture market anomalies led to the emergence of behavioral finance (Shiller 2003). In this context, the AMH appeared acknowledging and explaining the existence of anomalies in financial markets. The AMH argues that investors act mainly rationally, though heightened market volatility can induce irrational behaviors (Lo 2004). Technical analysis may result less effective when confronted with rapid changes in the market, abnormal conditions or political upheavals, as historical data does not account for these changes (Chen and Kawashima 2024).

In this context, recent research has highlighted the growing importance of sentiment analysis in predicting stock prices (Chen and Kawashima 2024)(Lakshya et al. 2022). News sentiment provides insight into the market's reaction to current events, which may not be reflected in technical indicators derived from historical price and volume data, especially in unpredictable situations like political upheavals or natural disasters. Analyzing the underlying emotions in financial news can enhance prediction accuracy by reflecting market sentiment and investor behavior (Chen and Kawashima 2024).

While the integration of sentiment analysis with modern ML techniques is very promising to provide more accurate and reliable predictions (Kim et al. 2023), technical indicators constitute the backbone of forecasting models (Chiekezie and Toromade 2024). Research in stock price prediction based on historical data and economic indicators has been evolving from classical time series models to ML models (D. Kumar et al. 2022), although classical methods still remain essential benchmarks for evaluating the performance and reliability of the newer predictive systems (Chiekezie and Toromade 2024).

In the following section, we explore these classical methods in more detail. From time series models that capture temporal dependencies in stock prices, the focus will shift to several widely adopted ML algorithms, known for their flexibility in integrating a broader range of input variables.

2.1.2 Yesterday's Footprints, Tomorrow's Forecast? A Look at Technical Stock Prediction Methods

Stock price prediction has long been a key topic in financial modeling. For decades, time series and econometric models have been widely used for forecasting, though their limitations in handling complex market dynamics have led to the surge of alternative models (Bao et al. 2025). ML algorithms emerged to try to capture the volatility of stock price movements by focusing on modeling non-linear relationships and being more flexible in accommodating various types of input data (Kahan and Thomas 2024).

Whispers in the Time Series: Unraveling Patterns with Classical Models

ARIMA Model in Stock Price Forecasting The ARIMA (Auto-Regressive Integrated Moving Average) model is one of the most broadly used traditional models in time series analysis. It incorporates auto-regression (AR), differencing (I), and moving averages (MA) to identify past patterns in historical stock prices used to generate forecasts of future trends (Chatterjee et al. 2021), excelling particularly in short-term predictions (Ma 2024). However, its application to financial data often faces challenges, particularly due to the non-stationary nature of financial time series data, which can present trends or volatility clustering that ARIMA may not fully capture (Kahan and Thomas 2024). While it performs well in stable market conditions, the model struggles in the presence of sudden shocks or structural breaks. These limitations become most evident during economic recessions, natural disasters, or geopolitical crises, which require models that incorporate external factors beyond historical price data (Ma 2024).

GARCH Models for Volatility Forecasting To address some of ARIMA's limitations, such as the need to stabilize the time series sequences, the GARCH (Generalized Auto-regressive Conditional Heteroskedasticity) model has been introduced by some studies as an alternative to capture the heteroscedasticity of financial market fluctuations, a common feature in stock price data (Hu et al. 2020). Volatility clustering, the tendency for high and low volatility periods to be followed by similar conditions, and time-varying volatility are prevalent characteristics in stock markets. The ability of GARCH models to capture these phenomena has established them as cornerstones of volatility modeling. (Marisetty 2024)

While GARCH models manage to effectively capture the dynamics of major global stock indexes, traditional versions struggle with the asymmetric nature of financial volatility. The reliance on symmetry assumptions, which often deviate from real-world market behavior, limits and reduces their overall accuracy. Nonetheless, more advanced models like TGARCH or APARCH offer significant improvements by incorporating these asymmetric effects, which lead to enhanced forecast accuracy. However, even these models have limitations, particularly in accurately representing extreme market events like financial crises and working with complex data during model specification. (Marisetty 2024)

The Rise of the Algorithms: ML's Foray into Market Prediction

ML algorithms have garnered significant attention for stock price prediction due to their ability to model complex non-linear relationships and their adaptability to dynamic market conditions. Unlike the previous models, ML algorithms have two key advantages. For starters, their ability to combine multiple types of input data, such as historical price data, macroeconomic data, and social media and news sentiment, into a single predictive framework. The derived advantage is the increased capacity to adjust to the ever-changing landscape of financial market. (Chang et al. 2024) Some commonly used ML algorithms for price prediction tasks will be revised next.

MARS (Multivariate Adaptive Regression Splines) MARS is a regression technique that builds flexible models by fitting piecewise linear functions to the data, a particularly useful feature to model complex non-linear behaviors of markets. Another key strength is its ability to automatically select relevant features within the data and discard poor contributors to avoid overfitting. However, these strengths may turn disadvantageous in simple contexts where linear relationships dominate. Models may underperform as a consequence of mistaking random noise for structural change and inserting unnecessary knots, or creating redundant splits with highly correlated predictors. (Chatterjee et al. 2021)

Random Forest and SVM for Stock Price Prediction RF, an ensemble learning method, builds multiple decision trees and merges their predictions, significantly improving accuracy over single decision trees. Its use for stock price prediction is based on their ability to handle high-dimensional and imbalanced data (Chatterjee et al. 2021). Empirical findings indicate that the RF model demonstrates strong accuracy and stability when forecasting stock price movements, particularly in the long-term (Zheng et al. 2024).

Similarly, SVMs are used to identify the boundaries between different classes of data. In stock price prediction, boundary detection translates to identifying whether a stock will increase or decrease its price, being particularly useful to handle high-dimensional spaces where data points are not linearly separable and to identify non-linear correlations between characteristics in the data (Chakravorty 2023). Results show that SVMs effectively capture dataset diversity and exhibit strong generalization capabilities, thereby boosting the accuracy of stock price predictions. However, incorporating sentiment from financial-specific models like FinBERT can further enhance this accuracy, especially when combined with ensemble SVMs and a rolling window approach. This synergy leverages SVM's robust classification with FinBERT's nuanced sentiment analysis to outperform models using only historical data (Liu et al. 2023).

Nonetheless, a significant limitation of both algorithms lies in their substantial data requirements (Chakravorty 2023). While integrating diverse datasets encompassing news sentiment, financial reports, and social media activity can enrich feature spaces and potentially enhance predictive accuracy, the performance of these algorithms is also critically constrained by the careful selection and tuning of hyperparameters. This sensitivity to both data quantity and optimal parameterization presents a key challenge in achieving robust and reliable stock price forecasts (Chakravorty 2023).

XGBoost and Gradient Boosting for Market Prediction XGBoost is another powerful ML technique that has become popular in stock price prediction. It builds upon the principles of decision trees and gradient boosting to create robust predictive models that minimize errors in predictions through an iterative approach. Some of this model's most valuable features are its ability to handle missing data, control overfitting, and capture complex, non-linear interactions between features (Chang et al. 2024). XGBoost obtains fine and accurate forecasts when working with large high-dimensional datasets, where interactions between features are critical (Yifan Zhang 2022). When properly tuned, XGBoost consistently outperforms other ML algorithms like Prophet or ARIMA in stock price prediction tasks (Chang et al. 2024). Despite the aforementioned powerful capabilities, XGBoost's performance depends, as well as RF's and SVM's, on careful tuning of its hyperparameters, and its accuracy is heavily degraded when confronted with unexpected and unanticipated market changes brought on by uncertainties and speculation. (Chang et al. 2024)

LSTM for Long-Term Dependencies and Sequential Data LSTM networks are a class of recurrent neural network (RNN), whose design to learn long-term dependencies makes them particularly suitable for handling sequential data, such as stock price movements over time (Chatterjee et al. 2021). Although LSTM networks excel at modeling the temporal aspects of stock price movements, they do not come without challenges. Realizing their full potential demands high-quality historical data alongside careful hyperparameter tuning. Their effectiveness heavily relies on the dataset, as learning long-term dependencies requires significant and quality training data. LSTMs also inherently suffer from lagging, delaying the reflection of recent data changes. Despite these limitations, integrating external data like news and social media through NLP offers a promising approach to enhance prediction reliability. (Huang 2023)

2.1.3 Concluding Remarks

Despite the widespread adoption of traditional time series models such as ARIMA and GARCH in stock price forecasting, the complex and often non-stationary nature of stock markets reveals some of their inherent limitations. ARIMA's and GARCH's struggles with non-linearity, asymmetric volatility, and sudden market shocks highlight the need for more adaptive approaches (Marisetty 2024).

The emergence of ML algorithms offers a promising pathway to capture intricate market dynamics (Chakravorty 2023). Models like MARS, LSTM, RF, SVM, and XGBoost have demonstrated their ability to learn non-linear relationships and integrate diverse data sources, achieving accurate forecasts across various stock price indices. Despite these advancements, and while challenges such as the need for meticulous hyperparameter tuning, substantial high-quality data, and overfitting prevention exist, a more profound limitation shared by all of these algorithms is their vulnerability to unanticipated and abrupt market shifts, which are inherent to financial markets and can severely undermine their forecasting capabilities (Chang et al. 2024).

Therefore, to further enhance the accuracy and robustness of stock price predictions, the incorporation of information beyond historical price movements seems crucial (Huang 2023). Understanding the underlying sentiment driving market participants' decisions, often reflected in news and other textual sources, holds the potential to provide valuable insights that technical data might overlook or lack (Huang 2023). This brings us to the next critical area of exploration, the study of financial sentiment analysis and its integration into stock price prediction.

The following section delves into the growing importance of news and sentiment in market behavior. It explores how news dissemination can ripple through stock markets, the influence of market sentiment in trading decisions, and the evolution of sentiment analysis techniques. The incorporation of these techniques aims to unlock further potential for understanding the complexities of stock price dynamics and improve prediction accuracy.

2.2 The News Know Best? Unveiling the Sentiment Behind Stock Swings

Financial news have the power to foster market bubbles or trigger panic selling (Song 2023). Their influence on investors, whether positive or negative, shapes market expectations and drives stock price fluctuations as supported by the EMH (R. Kumar and K S 2024). Sudden information shocks, such as unexpected economic reports or geopolitical events, can trigger immediate reactions as investors quickly adjust their expectations and positions to limit their damage or maximize their benefit.

Analyzing this textual information offers a crucial pathway to understanding the collective mood and expectations that influence trading decisions and ultimately impact stock prices (Chen and Kawashima 2024). Early methods for deciphering sentiment relied on identifying positive or negative keywords within the text. However, the limitations of these simple approaches in capturing the nuances and context of language led to the development of more advanced techniques. These newer methods leverage the capabilities of ML to better understand the subtleties of sentiment expression in financial news, aiming for a more accurate and insightful measurement that can potentially enhance predictive capabilities. (Kirtac and Germano 2024) The following subsections delve deeper into these evolving methodologies.

2.2.1 Sentiment Foundations: From Keywords to Lexicons

The beginnings of sentiment analysis involved the usage of keyword-based and lexicon-based approaches to analyze textual data. Keyword-based methods use dictionaries where specific words or phrases are identified with predefined sentiment scores. Meanwhile, lexicon-based methods go a step further, utilizing financial lexicons, such as the Loughran-McDonald Financial Sentiment Lexicons, which allow for more accurate sentiment evaluation (Kirtac and Germano 2024).

Both methods help match market sentiment with price movements. Positive sentiment is often associated with rising prices, while negative sentiment tends to signal declines, particularly during extreme market conditions (Turner et al. 2021). Although these techniques have been foundational in sentiment analysis, their design limits their capacity to capture the context and nuances of financial language, which has led to the development of more advanced approaches (Jishtu et al. 2022).

2.2.2 Beyond Keywords: Harnessing the Power of ML

Given the notorious limitations of keyword and lexicon-based methods, more sophisticated ML techniques have emerged to efficiently capture sentiment. SVMs' and RFs' ability to handle noisy datasets, manage missing data, adapt to diverse data types, and generalize well to unseen data have proven successful when applied to classifying sentiment in news articles. (Khan et al. 2024) Further, LSTMs have also gained prominence in this area, excelling at learning long-term dependencies in sequential data like time-sequenced financial news. (Pandya et al. 2025) Despite their strengths, these methods encounter considerable limitations, particularly when dealing with long or nuanced financial texts that feature intricate long-term dependencies and specialized financial jargon (Pandya et al. 2025). Furthermore, sentiment analysis is often hindered by the difficulty of these algorithms to distinguish between neutral sentiment and potential biases introduced by imbalanced datasets, which can contain uneven distributions of positive or negative samples (Pandya et al. 2025).

To overcome some of these limitations, Large Language Models (LLMs) represent the next frontier in financial sentiment analysis (Inserte et al. 2024). Unlike traditional models, LLMs are capable of processing vast amounts of unstructured textual data and analyze the context of text from all sides. These characteristics enable them to understand human language with remarkable depth and offer a more refined approach to understanding financial language. (Kirtac and Germano 2024) The following section explores how LLMs, particularly those fine-tuned for financial data, are transforming financial analysis, improving prediction accuracy, and expanding their use in combination with different ML algorithms.

2.3 Bridging the Divide: Harnessing LLMs to Decipher the Rhythms of Stock Prices

2.3.1 The Language Alchemists: How Large Language Models are Transmuting Financial Understanding

The ability of LLMs, powered by architectures like BERT, GPT, and T5, to interpret and generate complex financial narratives has transformed the paradigm of financial Natural Language Processing (NLP) (Yang et al. 2024). The evolution of LLMs, driven by advancements in computational power, the availability of large-scale datasets, and the development of novel neural network architectures, has led to models with impressive capabilities. These models excel at understanding, generating, and reasoning about natural language, derived from their ability to extract valuable insights from noisy, unstructured data and to learn latent relationships within and across sentences. (Li et al. 2023) This also grants LLMs superior adaptation and flexibility, enabling them to handle multiple tasks such as sentiment analysis, summarization, and keyword extraction on financial documents simultaneously.(Yang et al. 2024)

The extensive pre-training of LLMs on vast corpora, which are collections of texts and audios in native languages organized into datasets, allows these models to leverage enhanced language understanding across various industries and domains, inferring sentiment, classifying risk and supporting market forecasts in specific fields. This feature is resembled in FinLLMs, which are specifically adapted LLMs for the nuances of financial language (Kim et al. 2023).

Speaking the Language of Finance: Navigating the Nuances of Financial Text

Financial language differs from general purpose language in its use of temporally grounded, jargon-heavy, and ambiguity-prone constructs. Terms like "bull," or "bear," vary notoriously by context. While in normal contexts both terms represent two types of animals, in finance they are used to denote the upward trend (bullish market) or downward trend (bearish market) of the market. (Chen and Kawashima 2024) FinBERT and BloombergGPT are two of the most popular examples of LLMs trained on financial documents such as SEC filings, earnings call transcripts, and regulatory reports. Both fine-tuned models have proven effective at capturing the subtle sentiment found in financial texts. (Inserte et al. 2024) Studies consistently show that domain-specific LLMs outperform general ones across natural language tasks like sentiment analysis, question answering, and summarization (Li et al. 2023). Arguably, the most notable challenge in financial NLP is that key sentiment cues are often embedded in implicit, domain-specific, non-obvious expressions, and numerical references, which causes generalist language models to fall short and underperform, due to their lack of understanding of financial texts (Li et al. 2023).

Forging FinBrain: The Emergence of Specialized Financial Language Models

The growing complexity of financial language and the limitations of general-purpose language models have spurred the development of specialized financial LLMs (FinLLMs) (Inserte et al. 2024). These FinLLMs are typically built upon two core architectural paradigms: encoder models like FinBERT or DeBERTa are optimized for understanding input text and generating dense representations for tasks such as classification, sentiment analysis, or risk scoring. Their compact design and low-latency performance make them ideal for real-time financial applications (Guo and Hauptmann 2024). On the other hand, decoder models such as StockGPT are designed for text generation. They are capable of producing detailed narratives, simulating hypothetical scenarios, and supporting more flexible, open-ended reasoning, though generally with greater computational demands. (Guo and Hauptmann 2024) Figure 2.1 illustrates the evolution from general language models like BERT and GPT towards fine-tuned models that excel at capturing financial nuances and gauging market sentiment.

FinBERT marked a pivotal moment in this transition and remains highly influential in financial sentiment analysis (Jun Gu et al. 2024). As an encoder model, it builds upon the transformer-based BERT architecture, which undergoes pre-training to learn general language understanding through tasks like masked language modeling and next sentence prediction. FinBERT was further fine-tuned on a substantial financial corpus (4.9 billion tokens from corporate filings, analyst reports, and earnings calls) (Kim et al. 2023). This specialization to understand domain-specific phraseology and subtle tone shifts has made it highly effective for tasks such as analyzing financial articles, news, and reports, and classifying investor sentiment (Liu et al. 2023).

Another significant leap came with BloombergGPT, a 50-billion parameter model trained from scratch on a mixed corpus of financial and general data (Wang et al. 2024). Unlike the adaptation approach of FinBERT, BloombergGPT was designed to balance domain specificity with broader linguistic competence.The model's improved capabilities in finance-related tasks do not come at the expense of general generative tasks, knowledge assessments, reading comprehension, and linguistic tasks. This has caused the model to perform well in both financial benchmarks and general language tasks, making it one of the first truly cross-functional FinLLMs. Cross-functionality, though, is computationally intensive and less suited for lightweight applications. (Li et al. 2023)

FinGPT represents another advancement, combining a LLaMA-based architecture with instruction tuning and LoRA adaptation. This decoder-based model can generate financial insights, answer analytical questions, and handle tasks beyond classification, such as simulating scenarios or summarizing market trends (Li et al. 2023). Meanwhile, StockGPT and InvestLM are examples of increasingly task-specific models. StockGPT is designed for forecasting and reasoning in trading scenarios (Mai 2024), while InvestLM is tuned to handle financial instructions across multiple text sources, such as stack exchange quantitative finance discussions and SEC filings (Lee et al. 2025). StockGPT, though relevant, is not explicitly shown in Figure 2.1.

The inherent differences in how encoder-only like FinBERT and decoder-only language models like FinGPT process and interpret text, influence the resulting sentiment analysis (Lee et al. 2025). The development of FinLLMs has unlocked new possibilities for integrating textual sentiment signals into quantitative forecasting frameworks, shifting away from isolated applications (Jun Gu et al. 2024; Liu et al. 2023). By more effectively capturing and utilizing market sentiment, these models offer a potential pathway to fight some of the limitations of traditional approaches in foreseeing market shifts. The potential synergy between open-source FinLLMs and ML algorithms to enhance stock price prediction will be explored in the following section.

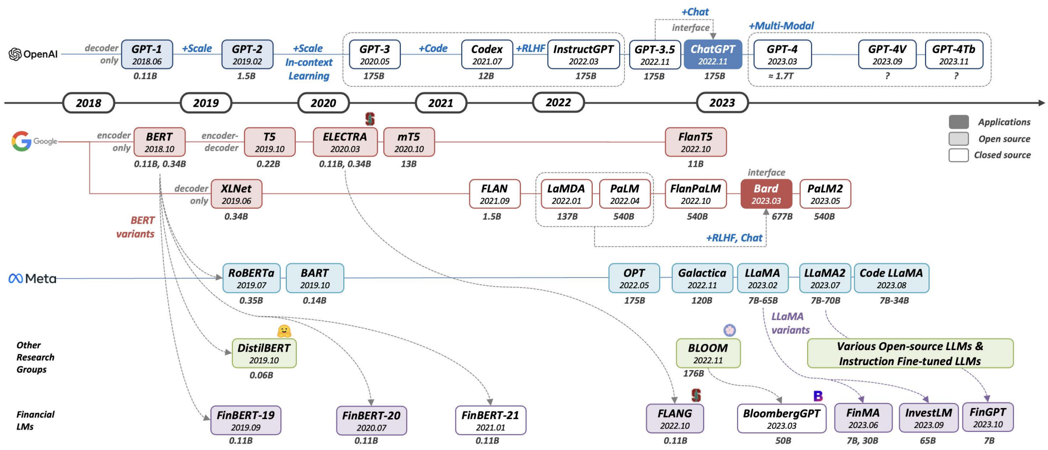

Figure 2.1: Evolution of Financial LLMs from general-purpose encoders to domain-specific fine-tuned models. Source: (Lee et al. 2025)

2.3.2 Synergies in Sight: How LLMs Improve Stock Price Prediction Performance

Given the potential of sentiment to offer valuable insights into market dynamics, a growing body of research focuses on integrating advanced stock price prediction mechanisms (Jun Gu et al. 2024; Oriol 2023), leveraging the power of LLMs to extract sentiment from news articles and combining these insights with technical indicators through ML algorithms. While numerous combinations of LLMs and ML algorithms exist, this section specifically focuses on pairings involving FinBERT with ML techniques as central case studies.

BERT and XGBoost

In the realm of stock return prediction, XGBoost models have established themselves as powerful tools. Their inherent capabilities in handling diverse feature types, mitigating overfitting, and managing missing values make them well-suited for the complexities and noise often present in financial datasets. However, the integration of sentiment information alongside traditional financial indicators has consistently demonstrated enhanced predictive performance. Interestingly, this improvement can be achieved even without the explicit use of domain-specific sentiment models, as evidenced by stock price predictions made on financial indicators and the analysis of financial news headlines and bodies of a BERT-based model trained on Amazon reviews (Oriol 2023). This finding suggests that even broadly trained language models can capture valuable sentiment signals relevant to advancing the accuracy and reliability of stock market forecasting when combined with robust prediction algorithms like XGBoost.

FinBERT and LSTMs

While LSTMs excel at capturing temporal dependencies within price data, the incorporation of FinBERT's sentiment analysis provides crucial contextual understanding from financial text. Comparing FinBERT-LSTM with Deep Neural Networks (DNN), which use a sequential arrangement of layers, where each layer integrates the output from its predecessor and forwards its own to the next layer, and LSTM models based solely on technical indicators, the LLM's ability to discern subtle linguistic nuances in news and identify meaningful information within complex or even neutral-sounding news furnishes the LSTM with valuable additional input (Jun Gu et al. 2024). The supplementary information found in the news headline and summary enables the hybrid model to generate forecasts that align more accurately with actual price movements, as demonstrated by significant reduction in error metrics (MSE, RMSE, and MAE) and the improved goodness of fit (higher R^2) observed when sentiment analysis results are included (Kim et al. 2023).

FinBERT and ensemble SVM

Recognizing the potential of FinBERT to enrich stock price prediction through sentiment analysis, researchers have also explored its integration with SVMs, a technique valued for its strong generalization ability and resilience to overfitting. However, to ensure real-world resembling, it is crucial to address the issue of look-ahead bias, a common pitfall where training data inadvertently includes future information, leading to inflated performance metrics (Liu et al. 2023). The rolling window approach provides an effective solution by training and evaluating the model, preventing the model from gaining an unrealistic advantage.

A key strength of employing FinBERT lies in its three-dimensional sentiment classification, which distinguishes between positive, negative, and, most importantly, neutral sentiments. This capability is vital for avoiding the misclassification of non-directional text, a limitation often encountered with binary sentiment analysis models, which restrains the prediction accuracy (Liu et al. 2023). While a potential drawback of FinBERT is its tendency to classify non-financial sentiment as neutral, the benefits of its domain-specific understanding outweigh this limitation. Empirical evidence suggests that the synergy between FinBERT and ensemble SVM models integrating the rolling window approach yields superior performance compared to standalone SVMs, as well as combinations involving simpler binary sentiment analysis or general-purpose sentiment analyzers like VADER (Liu et al. 2023).

Concluding Remarks

The previous studies underscore the significant influence of sentiment derived from financial news on stock price dynamics, offering valuable insights for enhanced risk management and more informed investment strategies. Integrating LLMs with ML algorithms paves a promising way for enhancing stock price prediction. As evidenced, even combining general-purpose LLMs like BERT with XGBoost offers encouraging results.

The successes of specialized FinLLMs like FinBERT combined with other ML models suggest a strong potential for further progress. While individual ML models already offer compelling results in stock price prediction, this research recognizes a valuable opportunity to delve deeper into the specific impact and analysis of integrating FinBERT with an LSTM framework. This combinations represents a valuable, complementary research opportunity to investigate the boundaries of a hybrid model within specific, novel settings.

2.4 Focusing on the Spanish Stage: Stock Price Prediction in the IBEX-35 Landscape

Prior research has extensively explored stock price forecasting across U.S. indices including the S&P 500, the Nasdaq, or the Dow Jones 500 (Jun Gu et al. 2024; Talazadeh and Perakovic 2024), leaving other markets such as the Spanish market, the IBEX-35 to be specific, comparatively unexplored. The following section therefore focuses on two distinct predictive approaches that researchers have applied to evaluate the accuracy of stock forecasting in the Spanish financial context.

Within the Spanish stock market, researchers have investigated the effectiveness of classical time series methods in the form of exponential smoothing techniques for predicting stock prices in the energy sector (Todorov and Sánchez-Lasheras 2023). Exponential smoothing techniques analyze historical price data, considering patterns in level, trend, and seasonality to forecast future values. Results show that the optimal exponential smoothing model can vary significantly between different stocks in the same industry and national market, highlighting the importance of selecting forecasting techniques that are well-suited to the individual characteristics of each stock's price history (Todorov and Sánchez-Lasheras 2023). While these traditional methods can provide reliable short-term forecasts, there is also a recognized potential for integrating more advanced ML techniques to further improve predictive accuracy in the complex and volatile dynamics of the IBEX-35.

In this context, another study explored the potential of leveraging emotional information extracted from a large corpus of Spanish news articles to improve the forecasting of daily fluctuations in the IBEX-35 index (Consoli et al. 2022). Recognizing the limitations of traditional time series models in capturing the influence of market sentiment, Neural Machine Translation (NMT) was used to translate Spanish news into English. This translation process was enhanced by integrating a pre-trained Spanish BERT model to ensure a richer understanding of the original Spanish text, which was then used for emotion classification via an English emotion classifier.

The resulting emotional features were then incorporated into DeepAR, an autoregressive recurrent neural network that excels at learning complex temporal dependencies (Consoli et al. 2022). The DeepAR model revealed a clear performance improvement when enriched with news-derived emotions, significantly outperforming benchmark models relying solely on technical indicators, such as DeepAR, moving average, and naïve methods (Consoli et al. 2022).

Evidence suggests that sentiment analysis of Spanish news holds the potential to enhance stock price prediction (Consoli et al. 2022). While LLMs have been extensively applied alongside ML algorithms in various markets, each market possesses unique characteristics given their unique blend of economic sectors, regulatory contexts, geographic locations, and company composition among others. In this context, the IBEX-35 is one such market that remains relatively underexplored. Furthermore, research has demonstrated that even within a single industry in the same market, optimal methods may exhibit varying degrees of effectiveness across individual companies (Todorov and Sánchez-Lasheras 2023). For this reason, applying one of the main LLM representatives, encoder-only FinBERT in combination with an LSTM model, represents a promising and novel approach. This unexplored combination can potentially provide valuable insights into the effect of financial news sentiment on stock price prediction within the banking industry of the IBEX-35, and shed light on how encoder-only LLM architectures contribute to sentiment classification and the subsequent stock price predictive performance.

Chapter 3

Foundations of Foresight: Constructing the Predictive Engine

This section outlines the experimental methodology employed to investigate the impact of news sentiment analysis on stock price prediction. It first presents the data source and acquisition for both the historical stock price and financial news articles datasets. Subsequently, the sentiment analysis methodology based on FinBERT is thoroughly described. Lastly, the LSTM foundational model and the derived hybrid model, "LSTM+FinBERT", are further detailed. The complete workflow is summarized in Figure 3.1.

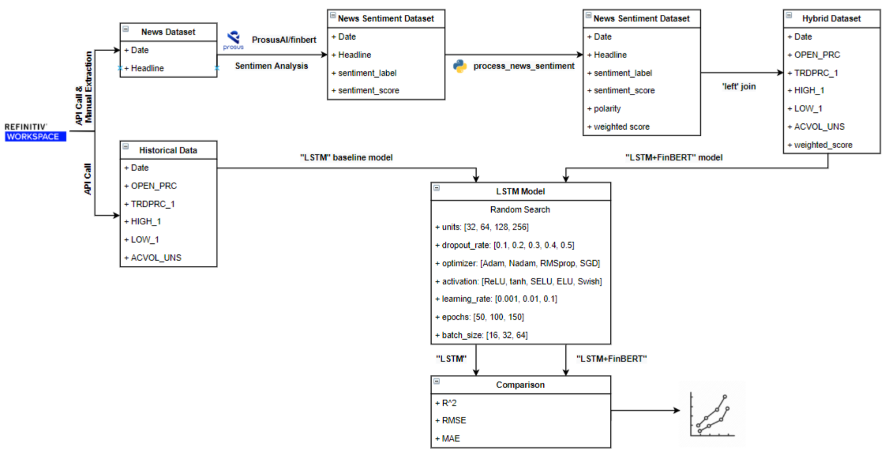

Figure 3.1: Workflow for FinBERT-Enhanced LSTM Stock Price Prediction.

3.1 The Information Tapestry: Weaving the Threads of Knowledge

The experimental design requires two primary data sources to address the thesis's objective of evaluating the impact of news sentiment analysis on stock price prediction: historical stock market data and financial news articles. Both sets of information were retrieved from the **Refinitv Workspace**, a comprehensive financial data platform that provides access to a broad spectrum of marked data, including historical prices, company financials, corporate events, financial news, and earnings estimates among others. Its validated reliability and extensive coverage makes it a rigorous source for data extraction (Inserte et al. 2024). To extract data, the Refinitiv Data Platform (RDP) provides a simple web-based API that grants access to numerous features, such as daily historical pricing by code value and news headline search.

3.1.1 Echoes of the Past: Unearthing Historical Market Narratives

To nurture the historical stock information, daily opening prices, highest daily prices, lowest daily prices and trading volumes were collected alongside daily closing prices for the banking companies constituing the IBEX-35. These companies are made up of Banco Bilbao Vizcaya Argetaria S.A. (BBVA), Bankinter S.A., CaixaBank S.A., Banco de Sabadell S.A., Banco Santander S.A. and Unicaja Banco S.A., and the extracted period comprised a five year span from May 20, 2020, to May 22, 2025, retrieved with the following function call to the RDP API:

Listing 3.1: Function to Fetch Historical Stock Data with RDP API Call.

```
def fetch_historical_prices(ric, days_back=365*10, dNow):
    sdate = dNow - timedelta(days=days_back)
    print(f"Fetching data from {sdate} to {dNow} for RIC:
       {ric}")

    close_price = rdp.get_history(
        universe=ric,
        interval="daily",
        fields=["OPEN_PRC", "TRDPRC_1", "HIGH_1", "LOW_1",
           "ACVOL_UNS"],
        start=str(sdate),
        end=str(dNow)
    )
```

```python

    close_price = close_price.dropna()

    return close_price
```

Where in the `rdp.get_history` API call:

- `universe`: Represents the specific financial instruments, identified by their unique Refinitiv Identification Code (RIC) value, for which historical data is requested. Specifically, `ric` corresponds to the aforementioned banking companies [BBVA.MC, BKT.MC, CABK.MC, SABE.MC, SAN.MC and UNI.MC].

- `interval`: Specifies the granularity of the historical data, set to "daily" to retrieve end-of-day summary statistics.

- `fields`: An array specifying the desired data attributes. `OPEN_PRC` denotes the opening price, `TRDPRC_1` the closing price, `HIGH_1` the daily high price, `LOW_1` the daily low price, and `ACVOL_UNS` the accumulated unadjusted trading volume.

- `start`: Defines the starting date of the data extraction period, represented by the variable `sdate`.

- `end`: Defines the end date of the data extraction period, represented by the variable `dNow`.

This extraction process resulted in six extensive datasets, each corresponding to one banking company, containing the daily open, close, high, low, and volume data for the specified five-year period. The decision to utilize a five-year historical period was informed by existing literature while aiming for a balanced scope. Studies have demonstrated that stock price prediction when combined with models like FinBERT can yield positive results across various timeframes, as evidenced by LSTM networks trained on two year (Liu et al. 2023) and eleven year periods (Jun Gu et al. 2024). However, to effectively capture key information from multiple dimensions and have a solid data foundation of media trends, stock market dynamics, and their interrelationships, while balancing data volume and computational efficiency, a five year historical period was selected, as it has empirically proven to contain enough information for LSTM networks to produce accurate S&P500 index predictions (Kim et al. 2023).

Table 3.1: Sample of Historical Stock Price Data of BBVA.

Date	OPEN_PRC	TRDPRC_1	HIGH_1	LOW_1	ACVOL_UNS
2025-05-13	13.0	13.23	13.245	12.945	8560914
2025-05-14	13.25	13.405	13.405	13.145	9666758
2025-05-15	13.3	13.345	13.375	13.28	6093306
2025-05-16	13.34	13.4	13.42	13.305	7457970
2025-05-19	13.42	13.465	13.55	13.35	5877843
2025-05-20	13.5	13.77	13.805	13.5	7496109
2025-05-21	13.665	13.695	13.9	13.645	7102438
2025-05-22	13.615	13.68	13.73	13.365	6376905

3.1.2 Whispers of the Market: Capturing the Pulse of Financial News

Financial news articles directly relevant to the six IBEX-35 banking companies were collected covering the five-year period from May 20, 2020, to May 22, 2025, consistent with the historical stock price data. The initial extraction of financial news headlines was performed using the RDP API, filtering for "Significant News" (Topic: `SIGNWS`) by `RIC` value, ensuring that only company related news were included, and explicitly restricting them to English language content (Language: `LEN`), with the following implementation of the RDP API:

Listing 3.2: Function to Retrieve and Store News Headlines

```
def fetch_financial_news(riclist, dNow, out_dir='data'):
    os.makedirs(out_dir, exist_ok=True)
    maxenddate = dNow - timedelta(days=365*2)
    compNews = pd.DataFrame()

    for ric in riclist:
        try:
            cHeadlines = rdp.news.get_headlines(
                "R:" + ric + " AND Language:LEN AND Topic:SIGNWS",
                start=str(dNow),
                end=str(maxenddate),
                count=10000
            )
            cHeadlines['cRIC'] = ric
            if len(compNews):
                compNews = pd.concat([compNews, cHeadlines])
            else:
                compNews = cHeadlines
```

```
        except Exception:
            cHeadlines = pd.DataFrame()
            pass
        file_path = os.path.join(out_dir, f'{ric}_news.csv')
        cHeadlines.to_csv(file_path, index=True)
        print(f"Saved {ric} news data to {file_path}")
    return compNews
```

The filtering criterion of exclusively selecting news articles in English was based on the fact that the sentiment analysis model FinBERT is pre-trained on English corpora. Processing news in other languages risks the appearance of inaccuracies and noise in the sentiment scoring, as non-English news were empirically confirmed to be classified as neutral. Translating headlines in other languages was also discarded due to the absence of a specialized financial translator in Python capable of capturing and maintaining domain-specific linguistic subtleties, which could cause the loss of context and nuance in the headlines.

A challenge encountered during the news collection, is that the RDP API restricts extraction of news to the last 15 months. Therefore, a hybrid approach was necessary to cover the entire five-year period, having to manually extract the articles beyond the 15-month window. During the preliminary data collection it was observed that the further back in time, the scarcer the `SIGNWS` articles, except for Banco Santander, which maintained a relatively high volume of significant news throughout the historical period. To ensure a sufficient representation of market sentiment across all banking companies, the scope of news topics for older dates was broadened to "Company News" (Topic: `CMPNY`), and "Economic News" (Topic: `MCE`) all of which related to the corresponding company. Instead of filtering by source the criterion was to extract the most relevant financial news to each of the companies. This process resulted in six distinct datasets, one for each banking company, comprising company-related financial news headlines, with varying amounts of total news per company as shown in Table 3.2.

Table 3.2: Summary of News per Bank

	BBVA	Bankinter	CaixaBank	Sabadell	Santander	Unicaja
Total News	9930	4354	6822	10211	6758	5191

3.2 The Language Lens: Decoding Sentiment with LLMs

To analyze the impact of sentiment analysis with LLMs on stock price prediction, the following section delves into encoder-only LLMs, particularly FinBERT which has been chosen as representative of this architectural design, due to its demonstrated ability to enhance stock price forecasting accuracy.

3.2.1 The Financial Lexicon Maestro: Deep Diving with FinBERT

Encoder-only LLMs generate a corresponding sequence of vector representations $\{h_1, \ldots, h_L\}$ to a given input sequence of text tokens $X = \{x_1, \ldots, x_L\}$. During the pre-training phase a subset of X's tokens are randomly masked, creating a corrupted input $\hat{X} = \{x_{\text{mask}} \text{ if } i \in M \text{ else } x_i \text{ for all } i = 1, \ldots, L\}$, where $M \subset \{1, \ldots, L\}$ denotes the indices of the masked tokens. The primary objective of the pre-training process is to predict these x_{mask} tokens, which are empty placeholders without inherent semantic meaning by maximizing their likelihood:

$$\log p(\{x_m\}_{m\in M}|\hat{X}) = \sum_{m\in M} \log p(x_m|X_{<m}, x_{\text{mask}}, X_{>m}) \approx \sum_{m\in M} \log p(x_m|h_m) \quad (3.1)$$

In Equation 3.1, $X_{<m} = \{x_1, \ldots, x_{m-1}\}$ and $X_{>m} = \{x_m, \ldots, x_L\}$ refer to the tokens preceding and succeeding x_m respectively. Maximizing the likelihood of the masked tokens x_m encourages the representation h_m to synthesize contextual information from both the left and right contexts ($X_{>m}$ and $X_{<m}$). Therefore, the Transformer's self-attention mechanisms learn to derive the meaning of h_m based on the similarities between the mask token and its surrounding contextual tokens. This bidirectional training mechanism is a core advantage of encoder-only models like BERT, enabling a more profound and contextually nuanced comprehension of linguistic flows compared to unidirectional approaches, as not only past or following information is uniquely considered, but both are relevant to understanding the tokens (Guo and Hauptmann 2024).

The training of FinBERT, a specialized LLM grounded in the BERT model, is carried out in two stages. First, it leverages the original BERT-base model, which was pretrained on general-domain corpora, and then continues its pre-training on a domain-specific financial corpus. This intermediate step, often referred to as domain-adaptive pre-training, is conducted on the TRC2-Financial dataset, a filtered subset of Reuters TRC2 containing over 29 million words related to financial news (Araci 2019). The objective of maximizing the x_m likelihood remains unchanged during this stage, allowing the model to adapt its representations to the specialized terminology and syntax of financial discourse.

Following this phase, FinBERT undergoes a supervised fine-tuning process on labeled datasets. As this thesis harnesses the capabilities of the pre-trained **ProsusAI/finbert** model, accessible via the Hugging Face Model Hub.[1], these capabilities are further refined through fine-tuning on the widely recognized Financial PhraseBank dataset by (Malo et al. 2014). The phrase bank covers a collection of 4840 senctences, annotated by researchers at Aalto University School of Business. The model produces softmax probabilities across three predefined sentiment categories: positive, negative, and neutral. To mitigate the loss of linguistic generalization during fine-tuning, FinBERT applies advanced training techniques to balance the retention of general language understanding with the acquisition of finance-specific patterns (Araci 2019).

3.2.2 Translating Theory to Code: The Digital Construction

To implement the sentiment analysis, the first step is to prepare the foundational environment. Since we're leveraging the `ProsusAI/finbert` model, it is necessary to load its corresponding `tokenizer` and `pipeline` for the subsequent data handling and NLP tasks. The `tokenizer` is FinBERT's personal linguist, as it is responsible for acquiring the vocabulary and tokenization rules the model was trained on. This step is paramount as it ensures that the financial news headlines are broken down into numerical tokens consistent with the model's training. This consistency is key to ensure an accurate interpretation and analysis of the raw data. After loading the pre-trained FinBERT model and its tokenizer, both elements are combined into a high-level `pipeline`. A powerful abstraction provided by HuggingFace Transformers that streamlines the entire process, from tokenization and model inference, to the post-processing of the results. The described loading process is instanced in Listing 3.3.

Listing 3.3: Loading of the FinBERT Sentiment Analysis Pipeline.

```
# Load FinBERT pipeline
model_name = "ProsusAI/finbert"
tokenizer = BertTokenizer.from_pretrained(model_name)
model =
    BertForSequenceClassification.from_pretrained(model_name)
finbert_sentiment = pipeline("sentiment-analysis",
    model=model, tokenizer=tokenizer)
```

[1]https://huggingface.co/ProsusAI/finbert

The second part of the implementation is dedicated to the extraction and processing of financial news headlines yielded through the `process_news_sentiment` function 3.4. Central to this function's operation is the nested `analyze_sentiment` function, which performs a predictive analysis to assess the sentiment embedded within the input text, the financial headlines to be specific. This analysis outputs a structured result with the sentiment label, either positive, negative or neutral, and the associated score, which quantifies how confident the model is in the assigned label. The closer to 1, the greater the degree of certainty in the model's prediction.

Each day possess multiple different headlines, therefore it is necessary to synthesize all sentiments to achieve a coherent daily representation. To address this, the designated approach was a `polarity` mapping, where each categorical sentiment label is translated into a numerical polarity value. Namely, 'positive' is assigned a value of 1, 'neutral' a 0, and 'negative' a -1. This transformation is essential for converting qualitative sentiment into a quantitative metric and computing the final `weighted_score`, as the multiplication between the derived `polarity` and the `sentiment_score`. This weighting mechanism ensures that positive and negative sentiment predictions, in which the FinBERT model exhibits higher confidence, exert a proportionally greater influence on the aggregated daily sentiment. Listing 3.4 contains the explicit implementation of the sentiment extraction process.

Listing 3.4: Function for News Headline Sentiment Analysis.

```
def process_news_sentiment(news_csv_path, finbert_model,
    label_to_polarity=None):

    if label_to_polarity is None:
        label_to_polarity = {'positive': 1, 'neutral': 0, 'negative': -1}

    def analyze_sentiment(text):
        try:
            result = finbert_model(text)[0]
            return result['label'], float(result['score'])
        except Exception:
            return 'ERROR', 0.0

    # Process the main file
    compNews = pd.read_csv(news_csv_path)
    sentiments = compNews['headline'].apply(analyze_sentiment)
    compNews['sentiment_label'] = sentiments.apply(lambda x: x[0])
    compNews['sentiment_score'] = sentiments.apply(lambda x: x[1])
    compNews['polarity'] =
        compNews['sentiment_label'].map(label_to_polarity)
    compNews['weighted_score'] = compNews['polarity'] *
        compNews['sentiment_score']
    compNews.to_csv(news_csv_path, index=False)
    print(f"Processed sentiment for {news_csv_path}")
```

The concluding section of this implementation defines the integration of the calculated daily sentiment scores with historical stock data, resulting in a unified dataset for the final financial forecasting. The function loads both the historical price data and the processed news sentiment data. As the financial news headlines are a combination of API and manually extracted samples, robust data parsing is implemented to avoid conflicts between potentially different date formats.

Following the date preparation, the news file is aggregated to derive a single `daily_sentiment_score` for each day by grouping the data by the extracted date and calculating the mean of the previously computed `weighted_score`. The aggregated daily sentiment data is then merged through a `left join` with the historical price data. This specific join is used to guarantee the retention of all historical stock price data points and prevent the loss of historical information, in case any day happens to be missing its corresponding news sentiment. Subsequently, any missing sentiment values are completed with `KNN interpolation` to smooth the sentiment data over time. KNN interpolation identifies patterns between similar data points, ensuring robust gap filling. This method was chosen for its proven ability to effectively leverage missing sentiment data in LSTM stock price prediction (Wabella 2024). The resulting augmented dataset is finally saved back to its original file path, rendering the integrated data for the development of the predictive models, which will be further studied in the upcoming section 3.3.

Listing 3.5: Function for Integrating Sentiment with Price Data.

```
def merge_sentiment_with_price(price_file, news_file):

    price_df = pd.read_csv(price_file, parse_dates=['Date'])
    news_df = pd.read_csv(news_file)

    # Robust date parsing
    news_df['date'] = pd.to_datetime(
        news_df['versionCreated'],
        errors='coerce',
        dayfirst=True,
        format='mixed'
    ).dt.date

    news_df = news_df.dropna(subset=['date'])

    daily_sentiment =
        news_df.groupby('date')['weighted_score'].mean().reset_index()
    daily_sentiment.columns = ['Date', 'daily_sentiment_score']
    daily_sentiment['Date'] = pd.to_datetime(daily_sentiment['Date'])
    price_df['Date'] = pd.to_datetime(price_df['Date'])
    price_df = price_df.merge(daily_sentiment, on='Date', how='left')
    first_valid_idx =
        price_df['daily_sentiment_score'].first_valid_index()
    if first_valid_idx is not None:
        price_df = price_df.loc[first_valid_idx:].reset_index(drop=True)
    price_df['daily_sentiment_score'] =
        price_df['daily_sentiment_score'].interpolate(method='nearest')

    price_df.to_csv(price_file, index=False)
    print(f"Updated {price_file} with sentiment columns.")
```

Table 3.3 illustrates the FinBERT-derived sentiment label distribution for each company's financial news headlines. The headlines show a predominant trend of neutral and positive sentiment, with these two categories maintaining broadly equivalent proportional representations among all companies. UNI.MC stands out as the only company exhibiting a higher presence of negative sentiment, indicating a more balanced weighting of the three sentiment classes within its financial news coverage.

Table 3.3: FinBERT Sentiment Scoring of Financial News.

Labels	Positive	Neutral	Negative
BBVA.MC	39.6%	45.6%	14.7%
BKT.MC	42.2%	40.6%	17.2%
CABK.MC	40.9%	43.1%	16.0%
SABE.MC	38.2%	46.7%	15.1%
SAN.MC	29.0%	51.3%	19.7%
UNI.MC	36.1%	33.3%	30.6%

3.3 The Convergence of Minds: Constructing the Hybrid Predictive Engine

3.3.1 Sequential Wisdom: The LSTM's Insight

LSTM models are a form of RNNs specifically designed to overcome some of the limitations of traditional ones, like the long-term dependency problem, where the nature of the *tanh* activation function can prevent earlier information from effectively propagating to later layers (Kim et al. 2023). LSTMs achieve an effective propagation by incorporating a sophisticated mechanism known as the cell state, which acts as a memory unit, allowing information to be carried across many time steps with minimal loss, making them highly effective in capturing long-term dependencies in sequential data. The cell state's contents are regulated by multiple gates, each of which is composed of a NN with a sigmoid activation function, that control the flow of information. The three primary gates are the input, forget, and output gates. Figure 3.2 illustrates the architecture of a typical LSTM network.

The core of an LSTM's operation involves updating the cell state C_t and hidden state h_t at each time step, leveraging the previous hidden state h_{t-1}, the previous cell state C_{t-1}, and the current input x_t. The processing of the inputs involves the aforementioned primary gates:

- **Forget Gate (f_t):** This gate determines which information from C_{t-1} should be discarded. Its output indicates how much of the old cell state to "forget" in a 0 to 1 range, where a value close to 1 preserves all information and a value near 0 discards all of it.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.2}$$

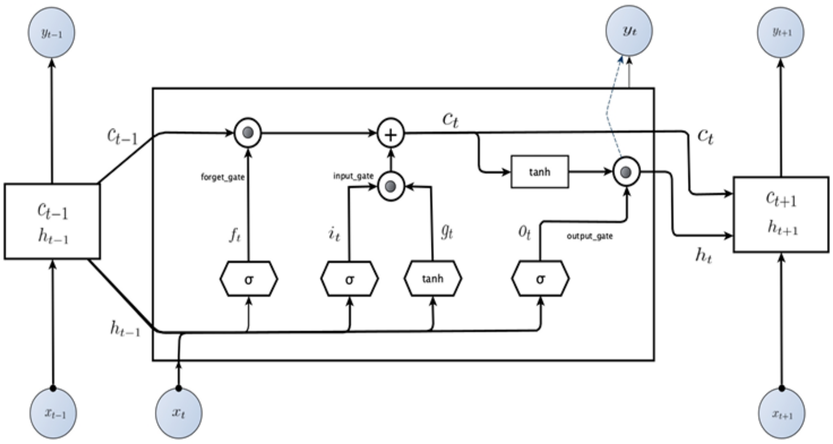

Figure 3.2: Architecture of LSTM. (Source: Kim et al. 2023)

Here, W_f is the weight matrix, b_f is the bias term, and $[h_{t-1}, x_t]$ represents the concatenation of the previous hidden state and current input.

- **Input Gate (i_t, g_t):** The input gate decides what new information from x_t should be stored in the cell state. It is composed of two parts, i_t (sigmoid-activated) determines which values to update, and g_t (tanh-activated) creates a vector of new candidate values.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.3}$$
$$g_t = \tanh(W_g[h_{t-1}, x_t] + b_g) \tag{3.4}$$

Here, W_i, W_g are weight matrices, and b_i, b_g are bias terms. C_t is updated by combining the forgotten old state with the new input:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot g_t \tag{3.5}$$

- **Output Gate (o_t):** The output gate controls which parts of the updated cell state will be outputted as the new hidden state (h_t), which subsequently serves as the input for the next time step.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.6}$$

Here, W_o is the weight matrix, and b_o is the bias term. h_t is derived by applying tanh to the cell state and then multiplying by the output gate's activation:

$$h_t = o_t \cdot \tanh(C_t) \tag{3.7}$$

3.3.2 Paving the Digital Path: Encoding the Prediction Machine

The proposed stock price prediction model was constructed with an LSTM architecture within the TensorFlow Keras framework. The model architecture is designed to capture sequential dependencies in the input data, which is comprised of basic price metrics, specifically closing price (`TRDPRC_1`), opening price (`OPEN_PRC`), price high (`HIGH_1`) and price low (`LOW_1`), trading volume (`ACVOL_UNS`), and a wide range of technical indicators to identify market trends and past patterns. The technical indicators include Exponential Moving Averages (`EMA_9`EMA_9) and Simple Moving Averages (`SMA_5`, `SMA_10`, `SMA_15`, `SMA_30`, `SMA_50`) over various periods, daily returns (`Returns`), and lagged values for open, high, low, and volume. Additionally, derived features such as `Open_Close_Ratio_Lag1`, `High_Low_Range_Lag1`, `Close_High_Ratio_Lag1`, and `Volume_Change_Lag1`, alongisde volatility (`Volatility_5`) and a normalized `Time_Index` are included. Such a wide range of technical indicators has been selected due to their demonstrated relevance in enhancing stock price prediction (Chen and Kawashima 2024; Fu and Yanbin Zhang 2024).

Moving averages are quantified using both SMAs and EMAs. While SMA calculates the average price of an asset over a defined period, assigning a uniform weight to each data point, EMA prioritizes more recent price data by giving greater weight to these observations. To try to capture more information from the historical context a wider range of SMA computations is included. The target variable for prediction is the next day's closing price, which represents the final consensus of traders and investors each day, and is considered the most important price in the stock market (Chen and Kawashima 2024). All features are scaled using `StandardScaler` prior to the sequence creation to normalize their range. The reason being the reduction of the model complexity, and the avoidance of overfitting (Kim et al. 2023) and of features with larger values dominating the prediction, thus generating biased results (Chen and Kawashima 2024).

After preprocessing the input data, all features are transformed into a sequence using a `lookback window`. The application of a lookback window implies that each input sample for the LSTM is composed of features from the past `lookback` days, enabling the network to learn from historical patterns over a defined period. Different window sizes allow to study how models perform with different predefined periods of historical data, allowing for the preservation of causality. This feature ensures that the model only uses data that would have been available before the prediction point, respecting the natural flow of time and mitigating the risk of an overly optimistic and unrealistic performance. In this thesis a total of seven lookbacks are used, namely [1, 5, 8, 10, 16 and 20], which have been succesfully used in short-term stock price prediction (Chen and Kawashima 2024; Kim et al. 2023). The extensive range of lookback window sizes was chosen to study different models and to mitigate the risks of both underfitting from short windows unable to capture long-term dependencies and overfitting from longer windows introducing noise or outdated patterns.

The pipeline of the model is composed of two LSTM layers each of which is followed by a dropout layer. The **first LSTM layer** contains a configurable number of LSTM `units`, to try different amounts to find the most suitable model. The initial layer is configured to `return_sequences=True` to output the sequence of hidden states for each time step in the input sequence. This configuration is necessary for stacking multiple LSTM layers, as it allows the subsequent layer to process the temporal features extracted by this initial layer. The `input_shape` is dynamically set to accommodate the `lookback` window and the number of input features. The activation function for this layer is also configurable and will be further explored in the hyperparameter tuning section 3.3.3. The **dropout layer**, a regularization technique that randomly deactivates a fraction of neurons during each training step, prevents the model from becoming overly reliant on specific connections, thereby reducing the risk of overfitting.

Subsequently, a **second LSTM layer** with half the number of units in the first layer is added. Unlike the first LSTM layer, it does not return sequences (`return_sequences=False`). This is a typical configuration for the final recurrent layer, as it outputs only the final hidden state representing the aggregated information from the entire input sequence. The activation function in this layer is also configurable, matching the hyperparameter options for the first LSTM layer. Here another **dropout layer** is applied to further enhance the model's generalization capabilities by introducing more regularization before the final output prediction. The final **output layer** is a dense layer with a single neuron and no explicit activation function, making it suitable for the regression task at hand, the prediction of the continuous value of stock closing prices. The model is compiled using the Adam optimizer and Mean Squared Error (MSE) as the loss function, which is a common choice for regression problems (Huang 2023).

3.3.3 Calibrating the Oracle: Optimizing the Model's Core

To optimize the model's performance and ensure robust training, a **Random Search** (RS) strategy was employed to explore different hyperparameter combinations to achieve the best possible models. RS has been selected as it generally outperforms Grid Search and requires less computational time (Kim et al. 2023). The grid of hyperparameters explored during this search is detailed in Table 3.4.

When modeling an LSTM model, the choice of hyperparameters is crucial as they significantly influence the performance of the model during training. In this case a total of 7 hyperparameters were used to find the best performing LSTM models: `Unit` determines the amount of units in the layer, the more units in the model the more complex and slower the learning process; `dropout rate` removes random neurons, preventing the model from overfitting; `optimizer` selects the optimization algorithm responsible of updating the model's weights; `activation function` selects the non-linear transformation applied by the layer; `learning rate` determines the step size at which the model learns, with high learning rates enabling faster learning, but at the cost of performance; `epochs` specifies the number of times the model processes the entire dataset; and `batch size` is used to select the number of small batches of data used to train the model. Larger batches result in faster training, but may cause memory issues. Conversely, a smaller batch size allows for more frequent weight updates, which can lead to better generalization. Finding the optimal values for these hyperparameters is essential for minimizing the selected loss function, the MSE. The values of the hyperparameters were selected based on their success in achieving high accuracy values to predict the S&P500 index with an LSTM network (Kim et al. 2023).

Table 3.4: Random Search Hyperparameter Grid

Parameter	**Grid**
units	[32, 64, 128, 256]
dropout_rate	[0.1, 0.2, 0.3, 0.4, 0.5]
optimizer	[Adam, Nadam, RMSprop, SGD]
activation	[ReLU, tanh, SELU, ELU, Swish]
learning_rate	[0.001, 0.01, 0.1]
epochs	[50, 100, 150]
batch_size	[16, 32, 64]

Two further features were implemented during the training process of the LSTM model. The `learning rate scheduler (ReduceLROnPlateau)` adapts the learning rate during training. If the validation loss does not improve for 3 consecutive epochs (`patience=3`), the learning rate is reduced in this implementation by a factor of 0.5 helping the model to fine-tune its weights when it approaches a minimum and improving convergence. The other element is the `early stopping`, which prevents overfitting and optimizes training time, as it halts the training if the validation loss does not improve for 7 consecutive epochs (`patience=7`). In the case of coming to a halt the weights from the epoch with the best validation loss are restored.

The data is split into 80% for training and 20% for testing before sequence creation. To ensure the reproducibility of results across different runs and environments, a custom `set_seed` function is implemented. The full implementation of the training process of the LSTM model can be found in Appendix **??**.

Sentiment Integration as Input Data to LSTM

Since the overall goal of the thesis is to measure the impact of including sentiment analysis in the stock price prediction model, sentiment scores of FinBERT need to be included in the training process. When sentiment analysis is enabled (`use_sentiment=True`), the model integrates the following sentiment features: `daily_sentiment_score_Lag1` to capture immediate influence, `sentiment_trend_3` and `sentiment_volatility_3` for observing sentiment momentum and dispersion over three days, `sentiment_ma_5` for smoothing out short-term noise and identifying sustained shifts, and `sentiment_diff_1` for detecting sudden daily changes. Lastly, interaction terms like `volatility_x_sentiment_lag1` between financial indicators such as volatility or volume change and the main lagged sentiment score are included to vary the sentiment impact based on prevailing financial market conditions.

3.4 The Acid Test of Foresight: Metrics of Predictive Performance

To evaluate the models various measures were used, though R^2 was selected as the primary metric to choose the definite model for each company, because it is a statistical measure that indicates how much of the variance in the dependent variable, the daily closing price, can be explained by the independent features of the model. It ranges from 0 to 1, where 0 indicates poor explainability and 1 perfect fit. However, a high value of R^2 does not always guarantee accuracy (Kim et al. 2023), therefore root mean squared error (RMSE) and mean absolute error (MAE) where chosen to measure how close the predicted values are to the actual stock prices.

RMSE is the square root of MSE, which measures the difference between the predicted and actual values by squaring and taking the average of both values. RMSE is chosen instead of MSE, because the latter is sensitive to the size of the prediction error, which is severely reduced when taking the square root, allowing for a more intuitive interpretation of the error size. MAE is an average measure that as RMSE reacts less sensitively to prediction error size and is less influenced by outliers compared to MSE. The higher the R^2 and the lower RMSE and MAE values are, the closer the average predictions are to the real stock prices. The corresponding equations to calculate these metrics are:

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.8}$$

$$\text{RMSE} = \sqrt{\frac{SSE}{n}} \tag{3.9}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.10}$$

$$\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{3.11}$$

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} \tag{3.12}$$

where SSE stands for error sum of squares and SST for total sum of squares, n is the number of data, y_i is the daily stock price of each of the banking companies in the IBEX-35, and $\hat{y}_i$ is the prediction value using the LSTM model.

Chapter 4

Harvesting Foresight: The Fruits of Predictive Labor

The performance of the LSTM model in predicting the stock prices of IBEX-35 banking companies was rigorously evaluated across various lookback windows with and without the integration of FinBERT sentiment scores. As summarized in Tables 4.1, and A.1 through A.5, the results reveal a complex interplay between the chosen lookback window, sentiment integration, individual company stock characteristics, and the underlying hyperparameter configurations. Overall, the models consistently achieved high prediction accuracies across all lookback windows and most company stocks, evidenced by high R^2 values and relatively low RMSE and MAE. These findings are explored in detail in the following sections.

4.0.1 Temporal Lenses: A Detailed Glimpse Through Lookback Windows

A granular examination of the results across different lookback periods highlights the varying impact of sentiment and of hyperparameter combinations on individual stock predictions. There is no single, unequivocally "best" lookback window that optimizes performance for all companies or all metrics simultaneously. Each lookback length presents unique characteristics and optimal model configurations. The Tables 4.1, and A.1 through A.5, present the best-performing results obtained from five iterations of random search hyperparameter tuning. The Figures A.1, and A.5 to 4.6 illustrate the model predictions alongside the real values corresponding to the best performers for each company. If a company does not appear in a specific lookback window, it indicates that the inclusion of sentiment did not improve the underlying baseline model. Though all lookbacks will be discussed, only the table corresponding to lookback 1 and the figures explicitly mentioned are included in this section, the remaining tables and figures can be found in Appendix

A.

Lookback 1

As shown in the line plots for Lookback 1 A.1 to A.4 and 4.1, and detailed in its corresponding metrics table 4.1, this shortest window takes only the previous day's information for predictions. The combination of very recent past information coupled with sentiment proves highly effective here. For instance, **SAN.MC** shows the most significant improvement, with its R^2 surging from 0.609401 without sentiment to 0.898193 with FinBERT, and a reduction in RMSE and MAE values by 0.25 and 0.18 respectively. These enhancements suggest that for SAN.MC, very recent sentiment is a powerful predictor.

BKT.MC also experienced a progression, with an R^2 gain of nearly 0.1 points and an equivalent reduction in RMSE and MAE. However, Figure 4.1 shows that this improvement for BKT.MC was mainly concentrated between July 2024 and January 2025, with the FinBERT-integrated model struggling slightly more in the final months of 2025 compared to the baseline. For **BBVA.MC, CABK.MC, and SABE.MC**, the improvement in their metrics was almost imperceptible, as they already exhibited very high baseline R^2 values (e.g., 0.941285 for BBVA.MC). The marginal R^2 improvements and consistent reduction in error metrics indicate that sentiment helps to slightly reduce the average prediction error, even for already well-modeled stocks.

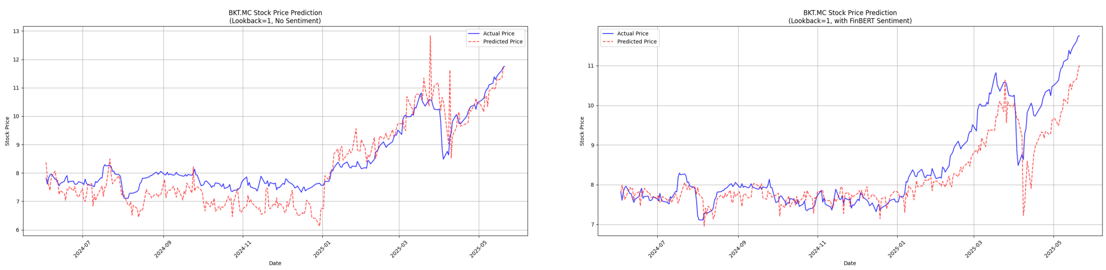

Figure 4.1: BKT.MC: LSTM vs. LSTM+FinBERT with Lookback 1.

Lookback 5

Plots A.5 to A.8 and 4.2 visualize the predictions for this lookback, and its Table A.1 quantifies the performance for this window. This lookback window continues to demonstrate strong performance, with **BBVA.MC and SAN.MC** showing substantial gains when combining technical indicators with FinBERT sentiment. BBVA.MC's R^2 improvements by 0.116662, and SAN.MC's by 0.189153 are accompanied with significant RMSE and MAE reductions, reaching a very high explana-

Table 4.1: Model Performance Comparison of LSTM with Lookback 1.

Company Stock	Sentiment Used	R^2	RMSE	MAE
BBVA.MC	No Sentiment	0.941285	0.362891	0.266345
	finbert	0.941485	0.363390	0.258979
BKT.MC	No Sentiment	0.689479	0.636557	0.516674
	finbert	0.777859	0.538401	0.387157
CABK.MC	No Sentiment	0.904591	0.239320	0.170612
	finbert	0.915616	0.225162	0.160908
SABE.MC	No Sentiment	0.856522	0.130504	0.097233
	finbert	0.896550	0.111670	0.076439
SAN.MC	No Sentiment	0.609401	0.516218	0.381220
	finbert	0.898193	0.264030	0.196551

tory power. Figure 4.2, for example, shows how SAN.MC forecesting experiences an overall improvement between the "LSTM" and "LSTM+FinBERT" models. **CABK.MC** shows an R^2 increase of 0.081526, while the remaining companies exhibit very slight improvements. **UNI.MC** is worth mentioning due to its very low MAE and RMSE values, the lowest in this batch, despite its R^2 reaching only 0.67 with FinBERT.

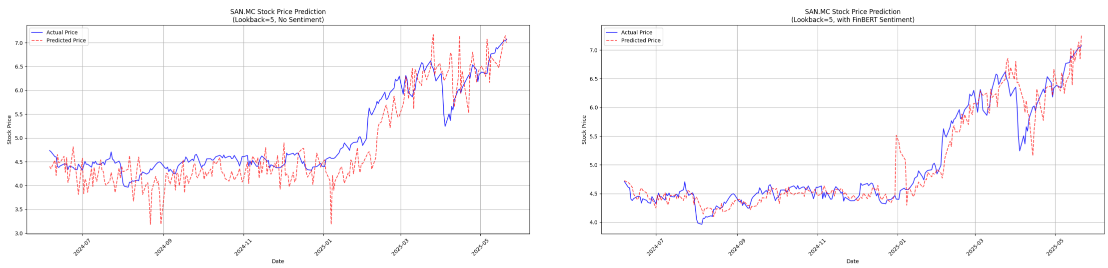

Figure 4.2: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 5.

Lookback 8

Plots A.9 to A.12 and 4.3 illustrate the model's predictive paths, with specific performance metrics provided in the corresponding Table A.2. With this lookback window both the baseline and FinBERT models are particularly effective forecasters, with the exception of **CABK.MC**, for which sentiment proves to be a vital feature. The baseline model for CABK.MC performs surprisingly poorly with an R^2 of 0.279084. However, with the integration of FinBERT sentiment, its R^2 skyrockets to 0.745272, yielding a massive 0.466188 improvement, and RMSE and

MAE correspondingly drop by more than 0.25. This indicates that for CABK.MC at this lookback, sentiment is not merely an enhancement but a fundamental requirement to develop a useful and powerful predictive model. This improvement is perfectly ressembled in Figure 4.3, wherethe sentiment-enhanced model's predictive performance closely aligns with the real values, outperforming the baseline model, except for April 2025 where it struggles a bit. **SABE.MC** also sees a significant R^2 improvement of 0.071112, along with good reductions in RMSE and MAE.

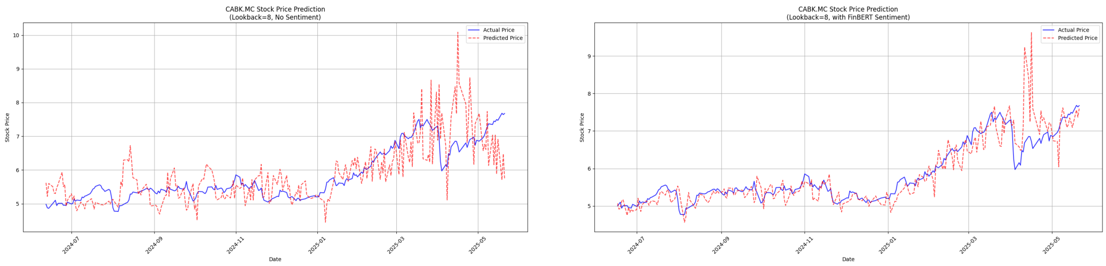

Figure 4.3: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 8.

Lookback 10

Figures A.13 to A.16 provide a visual understanding of the predictions, and the associated metrics Table A.3 presents the quantitative results. Unlike previous lookbacks where only one company was excluded (UNI.MC twice and SABE.MC once), in this instance, both **SABE.MC and BKT.MC** do not manage to improve their baseline models with the inclusion of sentiment.

SAN.MC continues to benefit notably, increasing its R^2 from 0.913682 to 0.943411, achieving the highest R^2 value observed. RMSE and MAE drop accordingly by 0.045914 and 0.033679. **UNI.MC**, appearing just for the second time, shows a strong R^2 for the first time with an improvement of 0.045426, highlighting the relevance of sentiment for this stock when using longer lookback windows. **CABK.MC**, with a baseline model already performing better than its best model in the previous lookback, is further enhanced by the addition of sentiment.

Lookback 16

The predictive outcomes are showcased in Figures A.17 to A.20, 4.4 and 4.5, and the performance metrics are detailed in Table A.4. This lookback period presents companies with baseline models facing considerable struggles, turning sentiment into a transformative factor. It is worth noting that for the first time all companies experience an improvement when taking into account sentiment.

BKT.MC exhibits an astonishing R^2 improvement of 1.128878, moving from a negative R^2 of -0.337510, that indicates the model performs worse than a simple mean prediction, to a more than respectable 0.791369 with sentiment. This transition from non-existent performance to capturing 80% of the variance in the data is one of the most compelling pieces of evidence for the power of FinBERT sentiment. Evidently the RMSE for BKT.MC dramatically decreases from 1.326464 to 0.523886. This improvement is evidenced in Figure 4.4 where the predictions evolve from an almost flat line to accurately mirror the real stock's movements. Similarly, **CABK.MC**'s baseline R^2 of 0.182898 improves up to 0.573194 with sentiment, but struggling particularly in accurately predicting the months of March, April and May 2025 4.5. **SAN.MC** is another company, that though less notably than the others, increases R^2 by 0.046148 points, along with small reductions in RMSE and MAE.

For **BBVA.MC and UNI.MC**, very high baseline R^2 values show only marginal improvements with FinBERT, which indicates that sentiment only acts as a fine-tuner. On the other hand, while **SABE.MC and SAN.MC** manage to improve their performance, the overall capability of the model remains poor despite having RMSE and MAE values lower than companies like BKT.MC, which possess a far greater R^2.

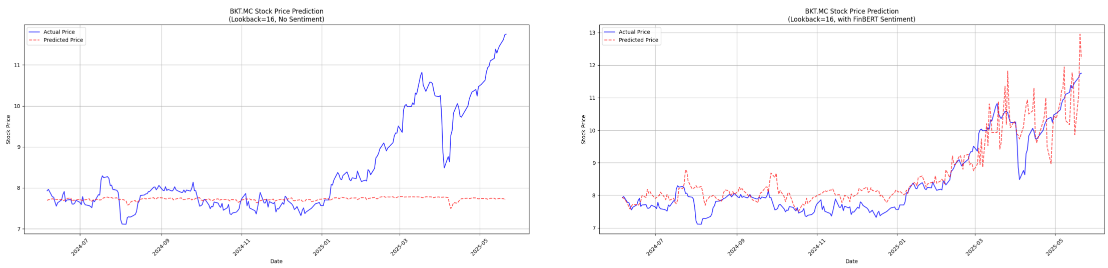

Figure 4.4: BKT.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

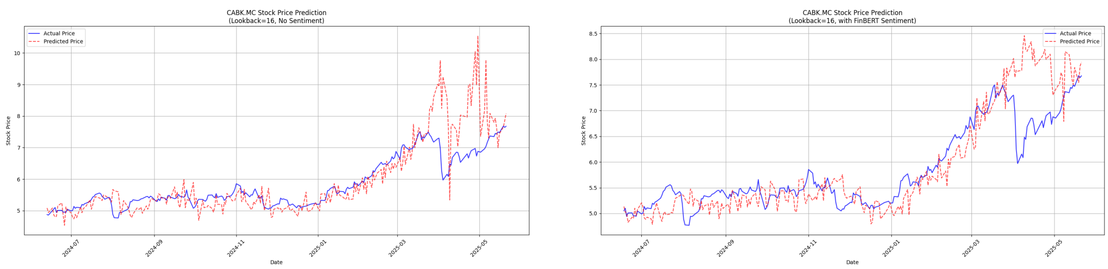

Figure 4.5: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

Lookback 20

The model's predictions for this lookback window are contained in Figures A.21 to A.25 and 4.6, and its metric results can be found in Table A.5. This longest lookback continues to highlight the stabilizing and performance-boosting effect of sentiment, especially for stocks that struggle with longer historical price data alone, as seen with the previous lookback window. **BKT.MC** shows a substantial R^2 improvement of 0.336189 with FinBERT, demonstrating its vital role for this stock over longer lookbacks. Its RMSE value also faces a significant reduction from 0.770927 to 0.388044. The improvement experienced by the inclusion of sentiment is particularly evident in April and May 2025, where the baseline model identified a sharp non-existent spike in Bankinter's stock prices. **BBVA.MC and CABK.MC** both show strong R^2 improvements of 0.051452 and 0.105331 respectively and considerable RMSE and MAE reductions, reinforcing sentiment's value for consistent performance. For **SAN.MC, UNI.MC, and SABE.MC**, their R^2 improvements are almost imperceptible, however, their baseline R^2 values are adequate, especially for SAN.MC working with an R^2 of 0.904007. The case of Unicaja is interesting, since despite the slight improvement in metrics, Figure 4.6 showcases a seemingly worse predictive approach for the months until January 2025. The following months however demonstrate a notable evolution with a fairly accurate representation of the real stock as opposed to the baseline model. The 2025 predictions are likely responsible for the light enhancement of R^2, $RMSE$ and MAE values in the case of UNI.MC.

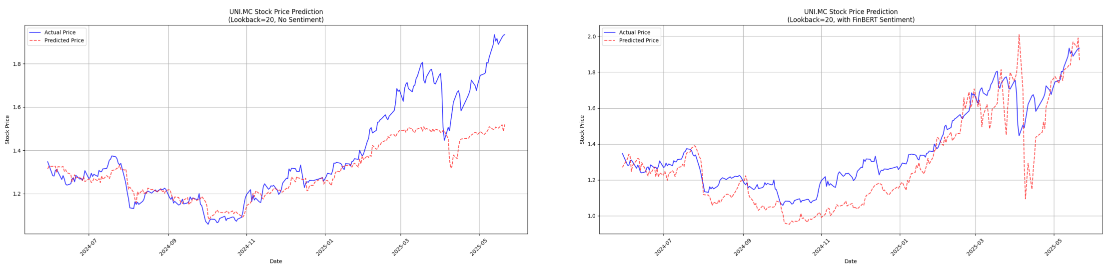

Figure 4.6: UNI.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

4.0.2 The Invisible Hand: The Art and Science of Parameter Optimization

A key factor in the consistently high performance observed across all lookbacks is the careful optimization of hyperparameter configurations. These configurations were obtained through random search hyperparameter tuning, allowing for the identification of the best possible combination for each individual model. The knowledge extracted from analyzing the hyperparameters of the best models for the different lookbacks including baseline and sentiment-augmented ones suggest that there is no single set of "best" hyperparameters that works universally, as optimal configurations vary significantly between companies and lookback periods.

For example, for **BKT.MC at Lookback 1**, the baseline model used `Units: 256`, `Dropout: 0.3`, `Optimizer: Nadam`, `Activation: relu`, `Learning Rate: 0.1`, `Epochs: 100`, `Batch Size: 32`. However, the "LSTM+FinBERT" model, which showed substantial improvement, used `Units: 32`, `Dropout: 0.2`, `Optimizer: RMSprop`, `Activation: relu`, `Learning Rate: 0.01`, `Epochs: 50`, `Batch Size: 64`. In this case incorporating sentiment changed the optimal learning dynamics, favoring a simpler model with fewer units, lower learning rate, and fewer epochs, which suggests that the additional information from sentiment simplifies the learning tasks. This pattern is not unique, **UNI.MC at Lookback 20**, for instance, exhibited a similar behavior where sentiment enabled a less complex model with fewer units and fewer epochs while still incrementing R^2.

In contrast, for **BBVA.MC across Lookbacks 1, 8, 10, 16, and 20**, the optimal hyperparameters for the baseline and sentiment models are almost identical. This implies that for BBVA.MC, sentiment consistently enhances the model's predictive power without fundamentally altering its optimal architecture or learning strategy. Specifically, for BBVA.MC at Lookback 1, both models used `Units: 128, Dropout: 0.3, Optimizer: SGD, Activation: relu, Learning Rate: 0.1, Epochs: 150, Batch Size: 32`. At Lookbacks 8, 10, and 16, they designated hyperparameters were `Units: 256, Dropout: 0.4, Optimizer: Adam, Activation: relu, Learning Rate: 0.001, Epochs: 100, Batch Size: 32`. The consistent behavior of BBVA.MC's hyperparameters suggests a robust price stability, where sentiment provides a direct enhancement without needing a fundamental shift in the model's learning approach.

While hyperparameter configurations are very diverse, certain features appear frequently in top-performing models. `Units` vary significantly from 32 to 256, however, smaller units (e.g., 32) are often seen in sentiment-augmented models where the sentiment signal might reduce the need for a highly complex network to extract patterns from raw price data. Within the `Optimizers` `Adam` and `RMSprop` are the most frequently chosen ones. Interestingly the choice of optimizer often correlates with the `learning rate`, `Adam` is frequently paired with a `learning rate` of 0.001 and `RMSprop` with 0.01. Lastly, among the `Activation Functions` `relu` is the most prevalent due to its computational efficiency and ability to mitigate vanishing gradients. `elu`, `tanh`, and `selu` are also used, but their appearance is more marginal.

The remaining hyperparameters do not show any prevalent trend or relationship with specific models, highlighting the importance of including a vast amount of hyperparameters to achieve the best configuration. The fact that the hyperparameters like `unit` change when sentiment is added indicates that sentiment is not just another input feature, but that it can fundamentally alter the learning task, leading to different optimal model complexities and learning strategies. This emphasizes the importance of performing separate hyperparameter searches for models with and without additional features like sentiment.

Chapter 5

Crystallizing Wisdom: Definitive Learnings from the Predictive Endeavor

The financial domain has extensively investigated the complex and volatile world of stock price prediction. To achieve more robust and accurate forecasts, numerous studies have explored the integration of sentiment analysis from financial news, employing different extraction methods like FinBERT or FinGPT, consistently highlighting the significant influence sentiment exerts on stock price prediction (Liu et al. 2023; Talazadeh and Perakovic 2024).

In this context, this thesis has rigorously investigated the impact of integrating **FinBERT-derived sentiment analysis** from Reuters financial headlines on the predictive performance of LSTM models for **IBEX-35 banking sector stock prices**. Although the primary objective was to evaluate sentiment's effect, the model was trained on an extensive set of technical indicators alongside the closing price, our target variable. The rationale behind not combining only sentiment with closing price resides in the fact that the aim of the study is to determine whether sentiment truly acts as a differential factor, capable of capturing underlying market dynamics not accessible through traditional indicators like `returns`, `volatility`, or other technical signals. The core of the thesis aims to answer the research question: **"Does integrating FinBERT-derived sentiment analysis improve the performance of LSTM models for IBEX-35 banking sector stock price prediction?"** Through extensive experimentation across various lookback windows and with meticulous hyperparameter tuning, our findings consistently support the affirmative.

The LSTM model was chosen for stock price prediction due to its proven suitability for sequential data analysis and as a consequence of similar researches having successfully managed to achieve consistent improvements in the prediction of other indices when integrating sentiment analysis to it (Kim et al. 2023). FinBERT was selected, because of its specialization in financial contexts, believing that its nuanced understanding of financial language would be crucial for extracting the most accurate sentiment scores, which have a direct impact on their contribution to the LSTM's predictive power. Our focus on the **IBEX-35 banking sector** addressed a notable gap in existing literature, since American indices like the S&P 500 and Nasdaq have been extensively researched (Jun Gu et al. 2024; Talazadeh and Perakovic 2024), but the IBEX-35 remains largely unexplored as stated in Section 2.4. Recognizing the unique characteristics inherent to each stock exchange, its industries, and constituent companies (Todorov and Sánchez-Lasheras 2023), the thesis focuses on the Spanish banking sector, an area which, to my knowledge, is unprecedented in academic study. The analysis is performed using a five-year dataset, covering the period from May 2020 to May 2025.

Therefore, this study's distinct contribution lies not in pioneering the individual methodologies, namely the usage of an LSTM algorithm for prediction, the news collection through Reuters, or the sentiment extraction with FinBERT, but rather in its **novel combination and application to the main Spanish index**. No prior research has investigated this specific intersection, with the closest approach using DeepAR in combination with an emotion classifier of Spanish news (Consoli et al. 2022). Another differentiating feature was the focus on **company-related and specific news**, rather than general news articles to focus specifically on the most relevant information to the prediction targets.

5.0.1 Pillars of Discovery: Key Contributions to Predictive Science

The integration of FinBERT sentiment generally led to **noticeable and often substantial improvements in model performance** across most companies in the different lookback windows. This enhancement is primarily evidenced by the **models incorporating FinBERT sentiment frequently demonstrating higher R^2 values**, indicating that a greater proportion of the variance in actual stock prices was explained. For instance, in Lookback 1, **SAN.MC** experienced a remarkable R^2 increase from 0.609401 without sentiment to 0.898193 with FinBERT, representing an improvement of 0.288792. Similarly, **BBVA.MC** for Lookback 5 showed a substantial R^2 improvement of 0.116662, and **CABK.MC** at Lookback 8, initially performing poorly with an R^2 of 0.279084, skyrocketed to 0.745272 after integrating sentiment into the LSTM model.

Alongside the improved R^2 values, **lower RMSE and MAE values were also consistent**. A reduction in the average magnitude of prediction errors means a closer alignment between predicted and actual stock prices. Following the previous example, SAN.MC at Lookback 1 saw its RMSE drop by 0.252189 and MAE by 0.184669 with sentiment. Even in rare instances where baseline performance was already exceptionally high like BBVA.MC RMSE's at Lookback 1 or UNI.MC's MAE at Lookback 20, sentiment contributed to fine-tuning, leading to marginal yet positive error reductions.

Perhaps one of the most compelling findings was the sentiment's ability to fundamentally transform models that initially struggled to understand the patterns in the data and provide accurate predictions. For companies like **CABK.MC** at Lookback 8 and, most strikingly, **BKT.MC** at Lookback 16, where baseline models exhibited surprisingly poor or even negative R^2 values (e.g., BKT.MC's baseline R^2 at Lookback 16 of -0.337510), the integration of FinBERT sentiment proved to be not merely an enhancement but a fundamental requirement for achieving a useful and powerful predictive model. BKT.MC's astonishing leap to 0.791369 with sentiment at Lookback 16 highlights the sentiment's capacity to convert poor performance into robust predictability.

Another insight derived from this study is the **diverse impact sentiment has on hyperparameter configurations**. Sometimes, it fundamentally alters the model's core, while in other cases, it simply enhances an existing stable architecture. The integration of **BKT.MC at Lookback 1** and **UNI.MC at Lookback 20**, for example, allowed to develop a simpler model architecture with fewer `units`, lower `learning rates` and `fewer epochs`. In contrast, for **BBVA.MC across Lookbacks 1, 8, 10, 16, and 20**, the optimal hyperparameters for both baseline and sentiment models remained very consistent. This dual behavior underscores that regardless of how sentiment impacts the model's structure or learning dynamics, its inclusion consistently aids performance, emphasizing the importance of performing separate and comprehensive hyperparameter searches for models with and without sentiment.

The overall high performance across lookbacks is attributed to the **careful optimization of hyperparameter configurations** through random search tuning. While no single set of "best" hyperparameters was universal, `Adam` and `RMSprop` were the most frequently chosen optimizers, often correlating with specific learning rates 0.001 and 0.01 respectively. `ReLU` emerged also as the most prevalent activation function due to its efficiency and ability to mitigate vanishing gradients. The observed variability in `Units` (from 32 to 256), epochs, and batch sizes across different stock-lookback combinations further highlights the adaptive nature required for optimal model fitting in this complex domain.

In conclusion, this study revealed that FinBERT sentiment analysis is a highly relevant feature in predicting stock prices for IBEX-35 banking companies. By integrating relevant and context-specific sentiment, LSTM models achieved superior performance, providing investors and analysts with more precise tools for risk management and informed investment strategies within the IBEX-35 banking sector. Ultimately, the findings provide a clear answer to the research question, demonstrating that FinBERT-derived sentiment analysis does indeed improve model performance, even if the degree of improvement varies from subtle enhancements at already high accuracy levels to significant, transformative gains in other instances. Regardless of the magnitude, the inclusion of sentiment has consistently proven to be beneficial in enhancing stock price prediction tasks.

Chapter 6

Uncharted Territories: New Trajectories for Predictive Discovery

While this study provides compelling evidence for the value of FinBERT sentiment in stock price prediction in the IBEX-35, it also lays the groundwork for several promising avenues of future research:

Firstly, while the LSTM model demonstrated superior forecasting capabilities for sequential data in this study, its limitations in handling extreme price drops or rapid spikes (Kim et al. 2023) warrant further investigation. To comprehensively evaluate model efficacy, future work should explore and compare the predictive capabilities of a **broader range of ML models** like XGBoost, SVM, and advanced hybrid models. A rigorous comparative analysis of their respective performances will offer deeper insights into the most effective modeling approaches for sentiment-augmented stock price forecasting and the context in which each is more robust and adequate.

Secondly, given the critical role of sentiment in our findings, an important future direction involves comparing **FinBERT's performance with other sentiment models**. This includes exploring alternative pre-trained language models, such as FinGPT or custom models trained on diverse financial corpora. Comparing the sentiment scores generated by these different methodologies and their subsequent impact on predictive performance will help assess the robustness and generalizability of FinBERT's specific contribution and identify whether other analyzers yield better results.

Thirdly, while this study focused exclusively on the IBEX-35 banking sector, future research should **expand the analysis to other industries within the IBEX-35 index**. Investigating sectors such as utilities, telecommunications, or retail will reveal whether the observed benefits of sentiment integration and specific hyperparameter dynamics are consistent across diverse economic domains or if their impact is sector-dependent. This broader scope will provide a more comprehensive understanding of sentiment's applicability across the Spanish market.

Furthermore, this study deliberately focused on short-term stock price forecasting. However, understanding long-term market trends holds significant value for investors. Therefore, a crucial future research avenue is to **attempt to apply our methodology to long-term predictions**. This would involve exploring different data aggregation strategies, considering a wider array of long-term macroeconomic factors, and adapting model architectures to capture more sustained price movements, thereby extending the practical utility of sentiment-driven insights beyond daily fluctuations.

Appendix A

The Atlas of Predictive Outcomes

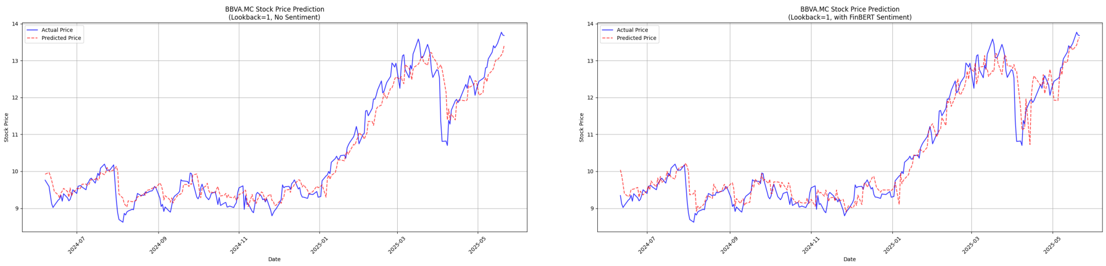

Figure A.1: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 1.

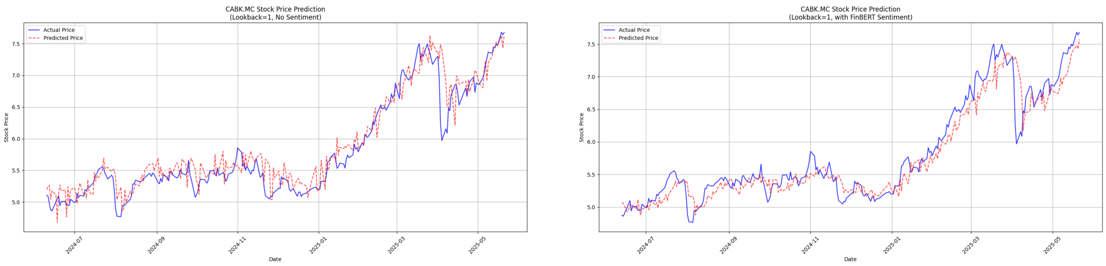

Figure A.2: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 1.

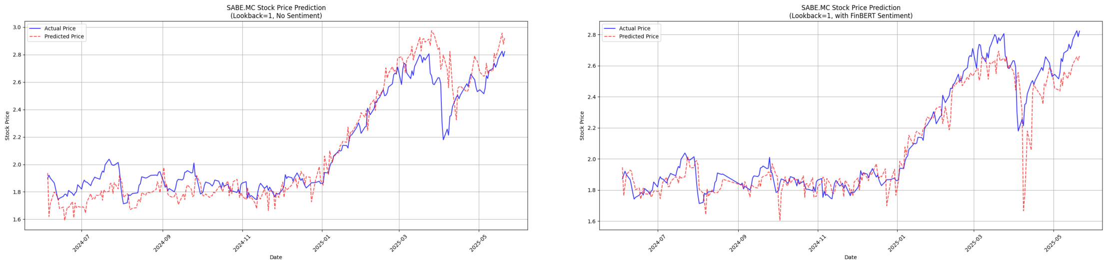

Figure A.3: SABE.MC: LSTM vs. LSTM+FinBERT with Lookback 1.

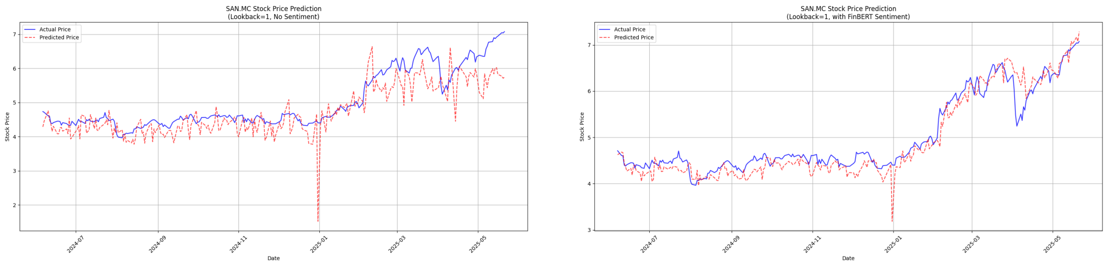

Figure A.4: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 1.

Table A.1: Model Performance Comparison of LSTM with Lookback 5.

Company Stock	**Sentiment Used**	R^2	**RMSE**	**MAE**
BBVA.MC	No Sentiment	0.803925	0.664249	0.573381
	finbert	0.920587	0.423756	0.339759
BKT.MC	No Sentiment	0.784687	0.530849	0.404281
	finbert	0.788968	0.525544	0.351257
CABK.MC	No Sentiment	0.735125	0.398907	0.321218
	finbert	0.816651	0.331575	0.247314
SAN.MC	No Sentiment	0.719508	0.437449	0.348058
	finbert	0.908661	0.250088	0.172841
UNI.MC	No Sentiment	0.636179	0.136228	0.092242
	finbert	0.672833	0.129183	0.090411

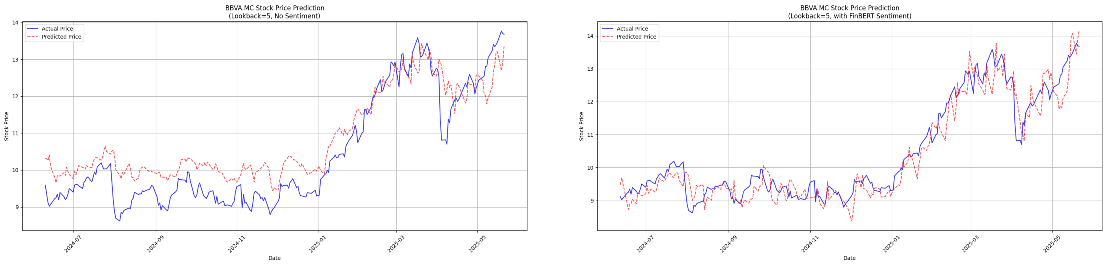

Figure A.5: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 5.

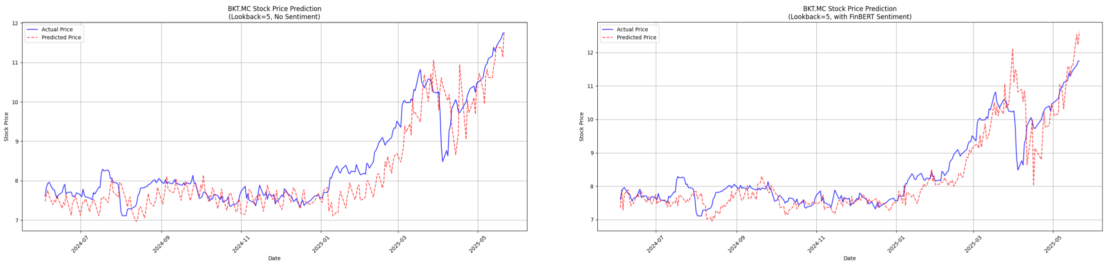

Figure A.6: BKT.MC: LSTM vs. LSTM+FinBERT with Lookback 5.

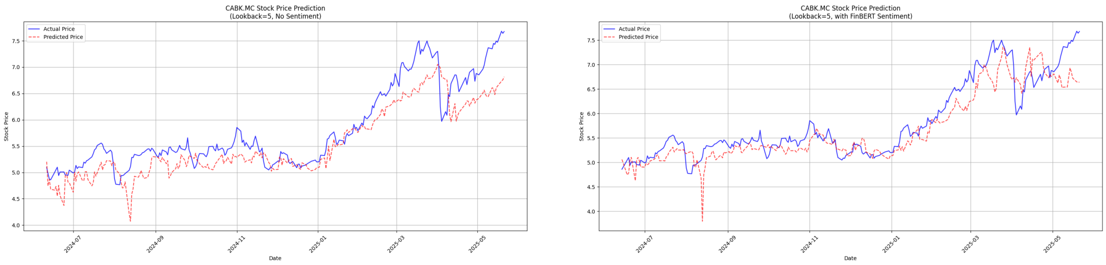

Figure A.7: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 5.

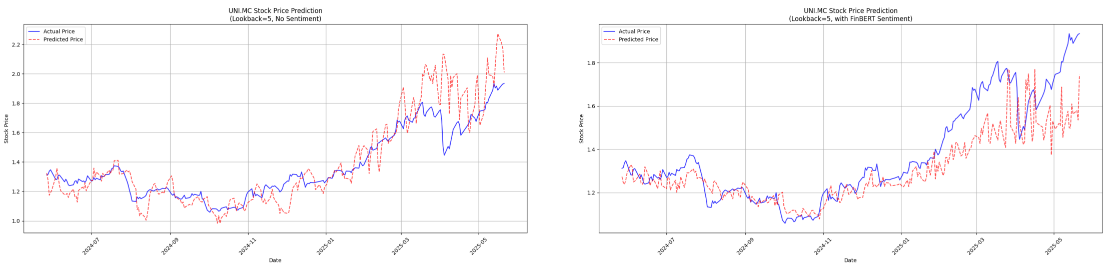

Figure A.8: UNI.MC: LSTM vs. LSTM+FinBERT with Lookback 5.

Table A.2: Model Performance Comparison of LSTM with Lookback 8.

Company Stock	**Sentiment Used**	$\mathbf{R^2}$	**RMSE**	**MAE**
BBVA.MC	No Sentiment	0.927186	0.404789	0.307838
	finbert	0.928830	0.401160	0.288617
BKT.MC	No Sentiment	0.831722	0.469786	0.374724
	finbert	0.839666	0.458563	0.364781
CABK.MC	No Sentiment	0.279084	0.658323	0.488279
	finbert	0.745272	0.390389	0.237709
SABE.MC	No Sentiment	0.836716	0.139612	0.099888
	finbert	0.907828	0.105527	0.082247
SAN.MC	No Sentiment	0.909555	0.248861	0.185093
	finbert	0.936849	0.208322	0.157441

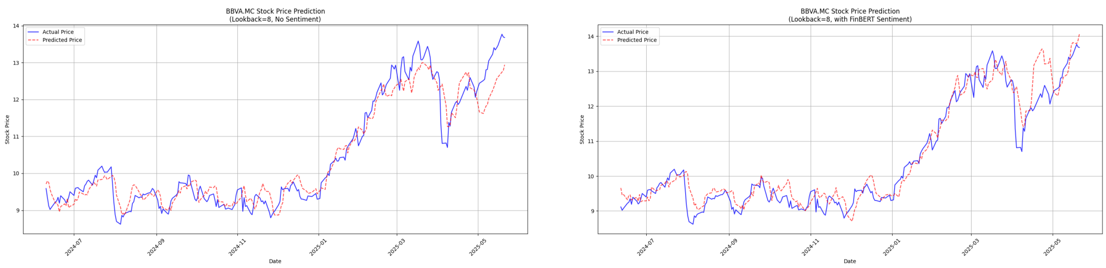

Figure A.9: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 8.

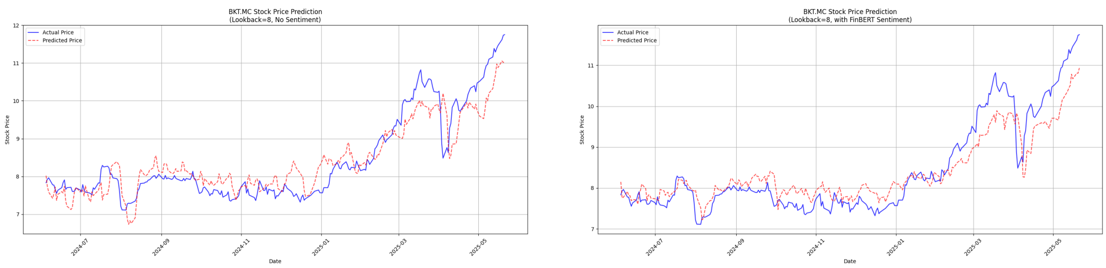

Figure A.10: BKT.MC: LSTM vs. LSTM+FinBERT with Lookback 8.

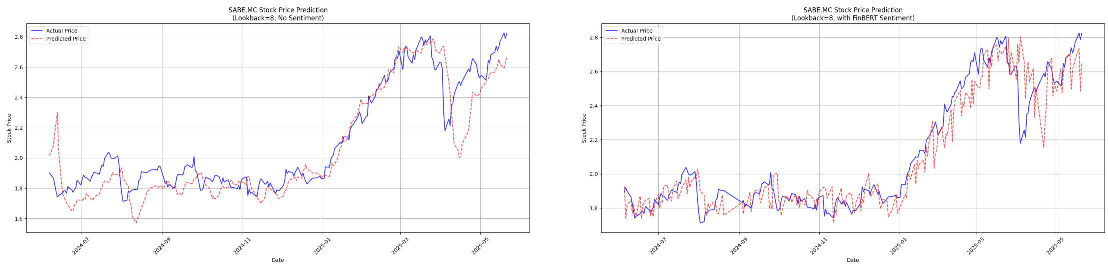

Figure A.11: SABE.MC: LSTM vs. LSTM+FinBERT with Lookback 8.

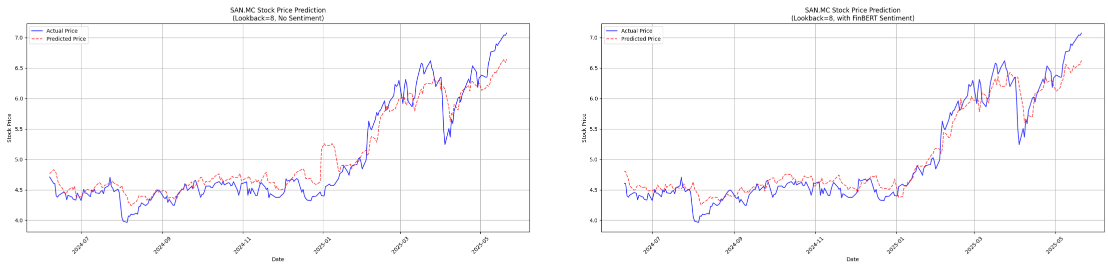

Figure A.12: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 8.

Table A.3: Model Performance Comparison of LSTM with Lookback 10.

Company Stock	**Sentiment Used**	$\mathbf{R^2}$	**RMSE**	**MAE**
BBVA.MC	No Sentiment	0.919962	0.424999	0.319221
	finbert	0.932286	0.391486	0.299153
CABK.MC	No Sentiment	0.787615	0.357321	0.281881
	finbert	0.859034	0.290413	0.216399
SAN.MC	No Sentiment	0.913682	0.243117	0.174892
	finbert	0.943411	0.197203	0.141213
UNI.MC	No Sentiment	0.821872	0.095500	0.076810
	finbert	0.867298	0.082428	0.062268

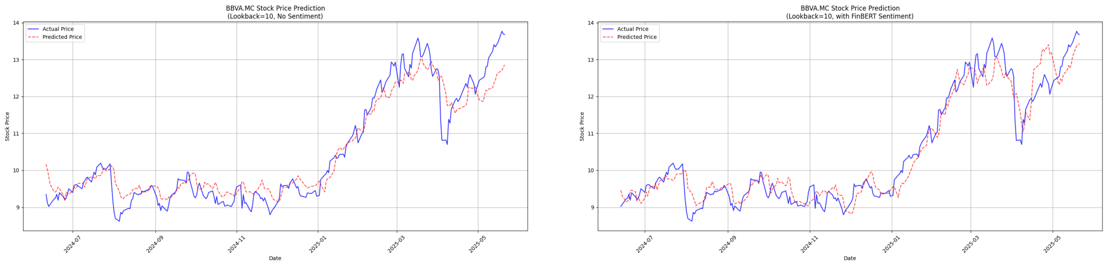

Figure A.13: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 10.

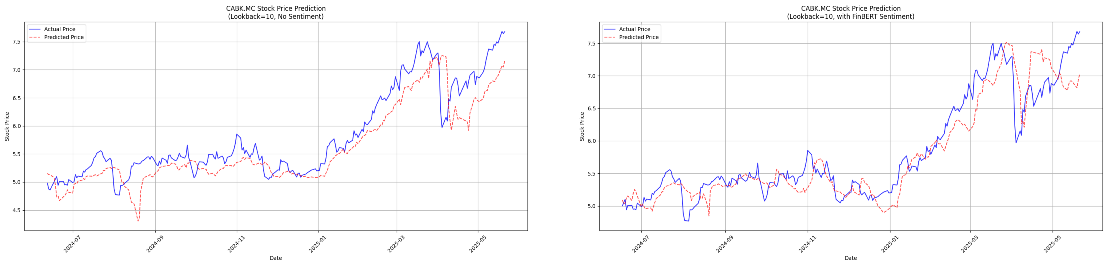

Figure A.14: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 10.

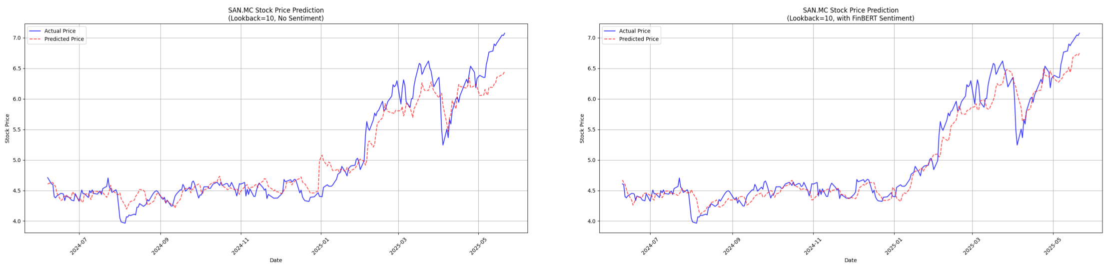

Figure A.15: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 10.

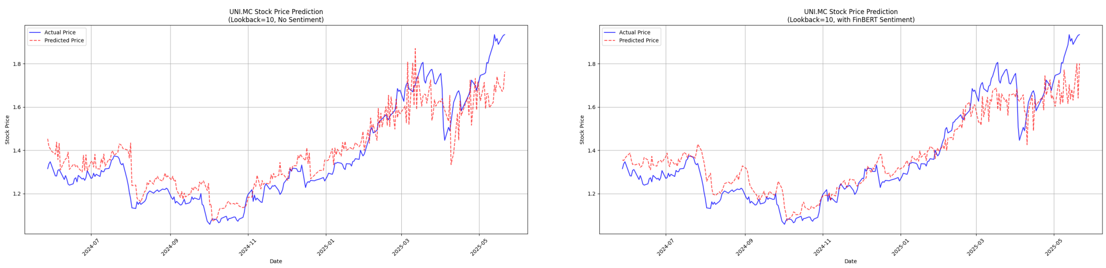

Figure A.16: UNI.MC: LSTM vs. LSTM+FinBERT with Lookback 10.

Table A.4: Model Performance Comparison of LSTM with Lookback 16.

Company Stock	Sentiment Used	R^2	RMSE	MAE
BBVA.MC	No Sentiment	0.930879	0.395345	0.286476
	finbert	0.933750	0.387319	0.307998
BKT.MC	No Sentiment	-0.337510	1.326464	0.816368
	finbert	0.791369	0.523886	0.397212
CABK.MC	No Sentiment	0.182898	0.700655	0.401543
	finbert	0.573194	0.505201	0.346027
SABE.MC	No Sentiment	0.167472	0.315672	0.214332
	finbert	0.200073	0.311762	0.283105
SAN.MC	No Sentiment	0.411842	0.636771	0.378531
	finbert	0.457990	0.612243	0.364090
UNI.MC	No Sentiment	0.898668	0.072166	0.056748
	finbert	0.911175	0.067566	0.051880

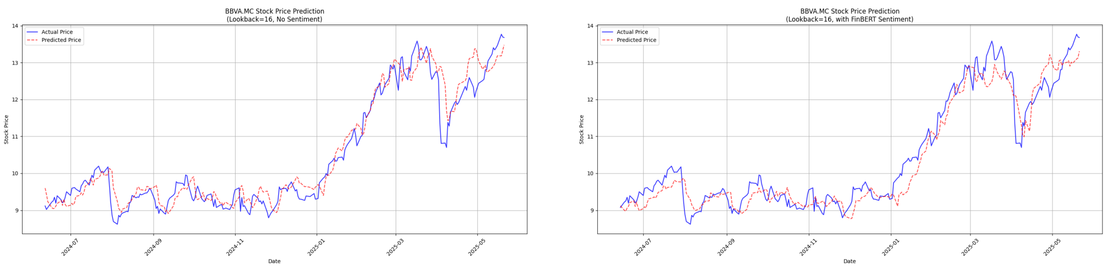

Figure A.17: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

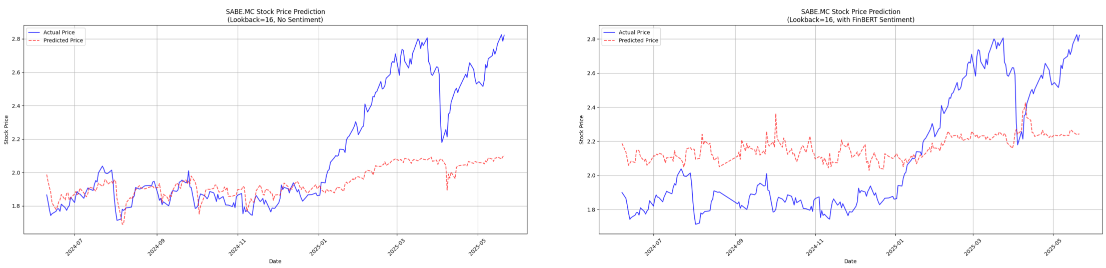

Figure A.18: SABE.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

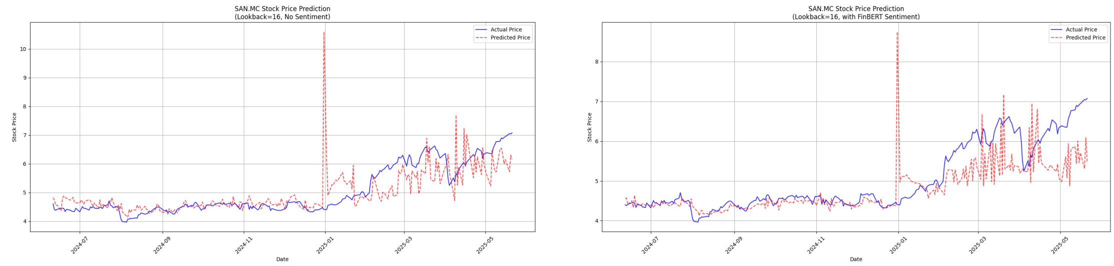

Figure A.19: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

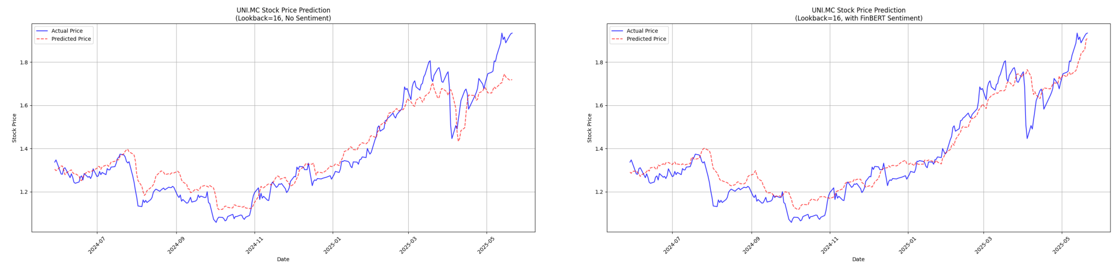

Figure A.20: UNI.MC: LSTM vs. LSTM+FinBERT with Lookback 16.

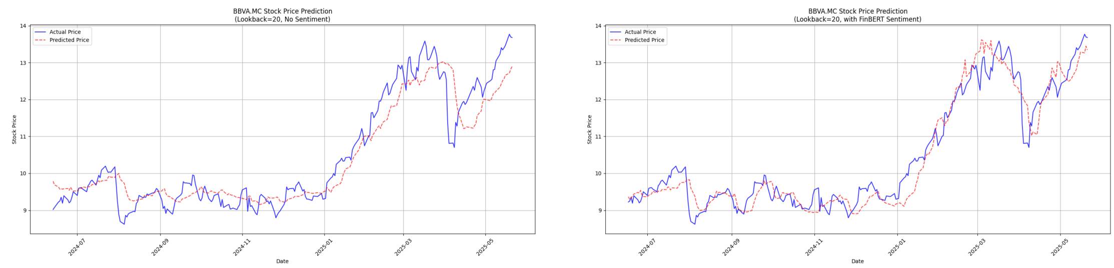

Figure A.21: BBVA.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

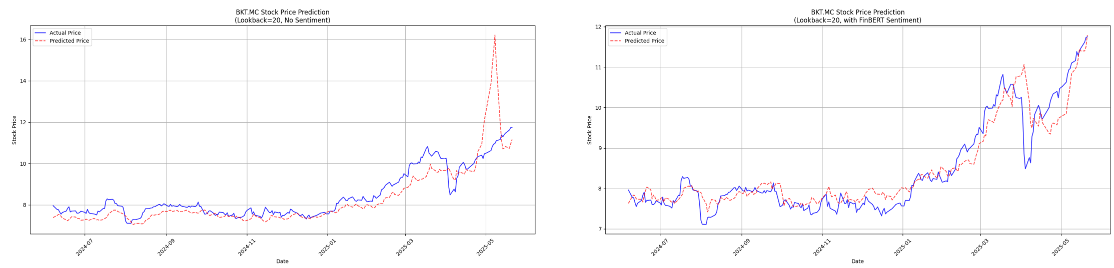

Figure A.22: BKT.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

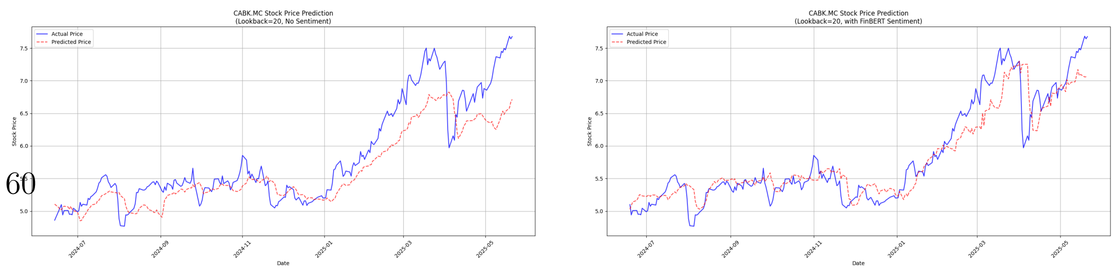

Figure A.23: CABK.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

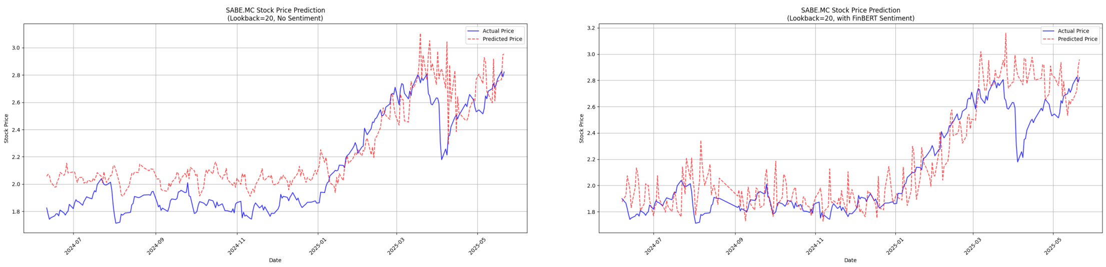

Figure A.24: SABE.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

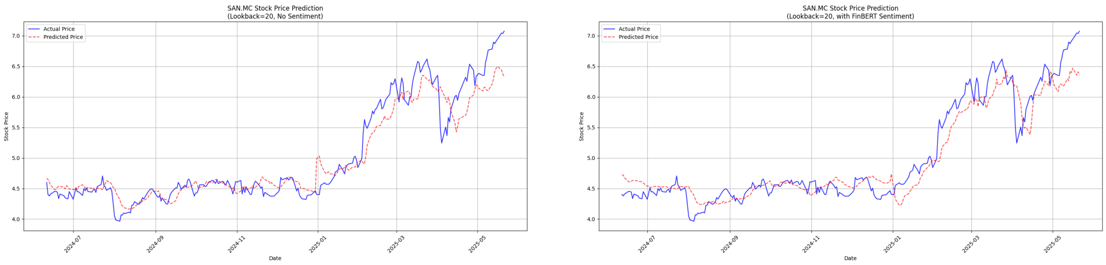

Figure A.25: SAN.MC: LSTM vs. LSTM+FinBERT with Lookback 20.

Table A.5: Model Performance Comparison of LSTM with Lookback 20.

Company Stock	**Sentiment Used**	$\mathbf{R^2}$	**RMSE**	**MAE**
BBVA.MC	No Sentiment	0.884565	0.511148	0.386861
	finbert	0.936017	0.380769	0.290570
BKT.MC	No Sentiment	0.549731	0.770927	0.458619
	finbert	0.885920	0.388044	0.267557
CABK.MC	No Sentiment	0.754359	0.383789	0.291370
	finbert	0.859691	0.289664	0.203822
SABE.MC	No Sentiment	0.649591	0.205018	0.172101
	finbert	0.681692	0.196663	0.145243
SAN.MC	No Sentiment	0.904007	0.257250	0.184338
	finbert	0.907271	0.253238	0.192276
UNI.MC	No Sentiment	0.656576	0.133117	0.085217
	finbert	0.669165	0.130654	0.101400

Appendix B

The Blueprint of the LSTM Engine

The complete code can be found in `https://github.com/Elconme/IBEX-Sentizer.git`.

Listing B.1: Python Code for LSTM Stock Price Prediction and Hyperparameter Tuning

```python
def set_seed(seed_value):
    np.random.seed(seed_value)
    tf.random.set_seed(seed_value)
    random.seed(seed_value)
    os.environ['PYTHONHASHSEED'] = str(seed_value)

# Create sequence of data for LSTM input
def create_sequences(X, y, lookback=1):
    Xs, ys = [], []
    for i in range(lookback, len(X)):
        Xs.append(X[i - lookback:i])
        ys.append(y[i])
    return np.array(Xs), np.array(ys)

# Build LSTM model
def build_lstm_model(input_shape, units, dropout_rate, optimizer_name,
    activation_function, learning_rate):
    if optimizer_name == 'Adam':
        optimizer = Adam(learning_rate=learning_rate)
    elif optimizer_name == 'Nadam':
        optimizer = Nadam(learning_rate=learning_rate)
    elif optimizer_name == 'RMSprop':
```

```
        optimizer = RMSprop(learning_rate=learning_rate)
    elif optimizer_name == 'SGD':
        optimizer = SGD(learning_rate=learning_rate)
    else:
        raise ValueError(f"Unknown optimizer: {optimizer_name}")

    model = Sequential([
        LSTM(units, return_sequences=True, input_shape=input_shape,
            activation=activation_function),
        Dropout(dropout_rate),
        LSTM(units // 2 if units > 32 else units,
            activation=activation_function),
        Dropout(dropout_rate),
        Dense(1)
    ])
    model.compile(optimizer=optimizer, loss='mse')
    return model

# Training of LSTM
def prepare_and_train_lstm(data, lookback=1, use_sentiment=False,
    units=64, dropout_rate=0.2, optimizer_name='Adam',
    activation_function='tanh', learning_rate=0.001, epochs=100,
    batch_size=16, verbose=0):

    data = data.copy()

    if 'Date' in data.columns:
        data['Date'] = pd.to_datetime(data['Date'], errors='coerce')
        data['day_of_week'] = data['Date'].dt.dayofweek
        data['month'] = data['Date'].dt.month
        data = data.sort_values(by='Date').reset_index(drop=True)
    else:
        raise ValueError("Missing 'Date' column for calendar feature
            extraction. Please ensure your CSV has a 'Date' column.")

    data['EMA_9'] = data['TRDPRC_1'].ewm(span=9,
        adjust=False).mean().shift(1)
    data['SMA_5'] = data['TRDPRC_1'].rolling(window=5).mean().shift(1)
    data['SMA_10'] = data['TRDPRC_1'].rolling(window=10).mean().shift(1)
    data['SMA_15'] = data['TRDPRC_1'].rolling(window=15).mean().shift(1)
    data['SMA_30'] = data['TRDPRC_1'].rolling(window=30).mean().shift(1)
    data['SMA_50'] = data['TRDPRC_1'].rolling(window=50).mean().shift(1)
    data['Returns'] = data['TRDPRC_1'].pct_change()
```

```
    data['Open_Lag1'] = data['OPEN_PRC'].shift(1)
    data['High_Lag1'] = data['HIGH_1'].shift(1)
    data['Low_Lag1'] = data['LOW_1'].shift(1)
    data['Volume_Lag1'] = data['ACVOL_UNS'].shift(1)

    data['Open_Close_Ratio_Lag1'] = (data['TRDPRC_1'].shift(1) -
        data['Open_Lag1']) / data['Open_Lag1']
    data['High_Low_Range_Lag1'] = data['High_Lag1'] - data['Low_Lag1']
    data['Close_High_Ratio_Lag1'] = (data['High_Lag1'] -
        data['TRDPRC_1'].shift(1)) / (data['High_Lag1'] -
        data['Low_Lag1'] + 1e-6)
    data['Volume_Change_Lag1'] = data['ACVOL_UNS'].pct_change().shift(1)

    data['EMA_9_Lag2'] = data['EMA_9'].shift(1)
    data['Returns_Lag2'] = data['Returns'].shift(1)
    data['High_Lag2'] = data['HIGH_1'].shift(2)
    data['Low_Lag2'] = data['LOW_1'].shift(2)
    data['Open_Lag2'] = data['OPEN_PRC'].shift(2)
    data['Volume_Lag2'] = data['ACVOL_UNS'].shift(2)

    data['Volatility_5'] =
        data['TRDPRC_1'].rolling(window=5).std().shift(1)
    data['Time_Index'] = np.arange(len(data)) / len(data)

    if use_sentiment:
        sentiment_cols = ['daily_sentiment_score']
        for col in sentiment_cols:
            if col not in data.columns:
                raise ValueError(f"Missing '{col}' column for FinBERT
                    sentiment. Cannot use sentiment analysis.")
        data['daily_sentiment_score_Lag1'] =
            data['daily_sentiment_score'].shift(1)
        data['sentiment_trend_3'] =
            data['daily_sentiment_score'].rolling(3).mean().shift(1)
        data['sentiment_volatility_3'] =
            data['daily_sentiment_score'].rolling(3).std().shift(1)
        data['sentiment_ma_5'] =
            data['daily_sentiment_score'].rolling(5).mean().shift(1)
        data['sentiment_diff_1'] =
            data['daily_sentiment_score'].diff().shift(1)

        main_sentiment_lag1_col = None
        if use_sentiment == True and 'daily_sentiment_score_Lag1' in
```

```
            data.columns:
            main_sentiment_lag1_col = 'daily_sentiment_score_Lag1'

        if main_sentiment_lag1_col and not
            data[main_sentiment_lag1_col].isnull().all():
            data['volatility_x_sentiment_lag1'] = data['Volatility_5'] *
                data[main_sentiment_lag1_col]
            data['returns_x_sentiment_lag1'] = data['Returns'] *
                data[main_sentiment_lag1_col]
            data['volume_change_x_sentiment_lag1'] =
                data['Volume_Change_Lag1'] * data[main_sentiment_lag1_col]
        else:
            if use_sentiment:
                print(f"Warning: Main sentiment column
                    '{main_sentiment_lag1_col}' not found or all NaN for
                    interaction terms. Skipping interaction terms.")

    data['Target'] = data['TRDPRC_1'].shift(-1)
    data = data.dropna()

    if len(data) < lookback:
        raise ValueError(f"Not enough data after dropping NaNs for
            lookback {lookback}. Data length: {len(data)}")

    feature_cols = ['EMA_9', 'SMA_5', 'SMA_10', 'SMA_15', 'SMA_30',
        'SMA_50', 'Returns',
                    'Open_Lag1', 'High_Lag1', 'Low_Lag1', 'Volume_Lag1',
                    'Open_Close_Ratio_Lag1', 'High_Low_Range_Lag1',
                        'Close_High_Ratio_Lag1', 'Volume_Change_Lag1',
                    'EMA_9_Lag2', 'Returns_Lag2', 'High_Lag2', 'Low_Lag2',
                    'Open_Lag2', 'Volume_Lag2', 'Volatility_5',
                        'Time_Index',
                    'day_of_week', 'month']

    if use_sentiment:
        feature_cols += ['daily_sentiment_score_Lag1',
            'sentiment_trend_3', 'sentiment_volatility_3',
                            'sentiment_ma_5', 'sentiment_diff_1']
        if 'volatility_x_sentiment_lag1' in data.columns and not
            data['volatility_x_sentiment_lag1'].isnull().all():
            feature_cols += ['volatility_x_sentiment_lag1',
                'returns_x_sentiment_lag1',
                'volume_change_x_sentiment_lag1']
```

```
    missing_cols = [col for col in feature_cols if col not in
        data.columns]
    if missing_cols:
        raise ValueError(f"Missing required feature columns:
            {missing_cols}. Please check your data.")

    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(data[feature_cols])
    y = data['Target'].values

    X_seq, y_seq = create_sequences(X_scaled, y, lookback=lookback)

    if len(X_seq) == 0:
        raise ValueError(f"No sequences created for lookback {lookback}.
            Check data length and lookback value.")

    split = int(len(X_seq) * 0.8)
    X_train, X_test = X_seq[:split], X_seq[split:]
    y_train, y_test = y_seq[:split], y_seq[split:]

    if len(X_train) == 0 or len(X_test) == 0:
        raise ValueError(f"Train or test set is empty after splitting.
            X_train: {len(X_train)}, X_test: {len(X_test)}")

    model = build_lstm_model(
        input_shape=(lookback, X_train.shape[2]),
        units=units,
        dropout_rate=dropout_rate,
        optimizer_name=optimizer_name,
        activation_function=activation_function,
        learning_rate=learning_rate
    )

    early_stop = EarlyStopping(patience=7, restore_best_weights=True,
        monitor='val_loss')
    lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.5,
        patience=3, verbose=0, min_lr=1e-6)

    history = model.fit(X_train, y_train, validation_data=(X_test,
        y_test),
                        epochs=epochs, batch_size=batch_size,
                            verbose=verbose,
```

```
                            callbacks=[early_stop, lr_scheduler])

    y_pred = model.predict(X_test).flatten()

    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    if verbose > 0:
        print(f"\n MAE: {mae:.4f}")
        print(f" MSE: {mse:.4f}")
        print(f" MSE  (RMSE): {rmse:.4f}")
        print(f"  R :  {r2:.4f}")


    return rmse, mae, r2, history.history['val_loss'][-1], model

# Random Search for hyperparameters
def run_random_search(data, num_iterations, lookback_value,
    use_sentiment, sentiment_model_type, seed):
    set_seed(seed)

    param_grid = {
        'units': [32, 64, 128, 256],
        'dropout_rate': [0.1, 0.2, 0.3, 0.4, 0.5],
        'optimizer_name': ['Adam', 'Nadam', 'RMSprop', 'SGD'],
        'activation_function': ['relu', 'tanh', 'selu', 'elu', 'swish'],
        'learning_rate': [0.001, 0.01, 0.1],
        'epochs': [50, 100, 150],
        'batch_size': [16, 32, 64]
    }

    param_combinations = []
    for _ in range(num_iterations):
        combo = {k: random.choice(v) for k, v in param_grid.items()}
        param_combinations.append(combo)

    results = []
    best_model_for_search = None
    best_r2_for_search = -np.inf

    print(f"\n--- Starting Random Search for {num_iterations} iterations
```

```
        (Lookback: {lookback_value}, Sentiment: {use_sentiment}, Model:
        {sentiment_model_type}) ---")

    for i, params in enumerate(param_combinations):
        print(f"\n     Iteration {i+1}/{num_iterations} with params:
            {params}")
        try:
            rmse, mae, r2, val_loss, current_model =
                prepare_and_train_lstm(
                data=data.copy(),
                lookback=lookback_value,
                use_sentiment=use_sentiment,
                sentiment_model_type=sentiment_model_type,
                units=params['units'],
                dropout_rate=params['dropout_rate'],
                optimizer_name=params['optimizer_name'],
                activation_function=params['activation_function'],
                learning_rate=params['learning_rate'],
                epochs=params['epochs'],
                batch_size=params['batch_size'],
                verbose=0,
                plot_results=False
            )
            result = {
                'Units': params['units'],
                'Dropout': params['dropout_rate'],
                'Optimizer': params['optimizer_name'],
                'Activation': params['activation_function'],
                'Learning Rate': params['learning_rate'],
                'Epochs': params['epochs'],
                'Batch Size': params['batch_size'],
                'RMSE': rmse,
                'MAE': mae,
                'R^2': r2,
                'Validation Loss': val_loss
            }
            results.append(result)

            if r2 > best_r2_for_search:
                best_r2_for_search = r2
                best_model_for_search = current_model

            print(f"     Iteration {i+1} completed. R^2: {r2:.4f}, RMSE:
```

```
                {rmse:.4f}")
        except ValueError as ve:
            print(f"    Iteration {i+1} skipped due to data error: {ve}")
        except Exception as e:
            print(f"    Iteration {i+1} skipped due to unexpected error:
                {e}")

    return pd.DataFrame(results), best_model_for_search
```

Appendix C

Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, **NO** está permitido su uso en la elaboración del Trabajo Fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, **Elena Conderana Medem**, estudiante de **Máster en Ingeniería de Telecomunicación y Business Analytics** de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "[**Título del trabajo**]", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código solo en el contexto de las actividades descritas a continuación [el alumno debe mantener solo aquellas en las que se ha usado ChatGPT o similares y borrar el resto. Si no se ha usado ninguna, borrar todas y escribir "no he usado ninguna"]:

1. **Interpretador de código:** Para realizar análisis de datos preliminares.

2. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.

3. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para qué se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: **17.06.2025**

Firma: ____________________

Bibliography

Araci, Dogu (2019). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. DOI: 10.48550/arXiv.1908.10063.

Bao, Wuzhida, Yuting Cao, Yin Yang, Hangjun Che, Junjian Huang, and Shiping Wen (2025). "Data-driven stock forecasting models based on neural networks: A review". In: *Information Fusion* 113, p. 102616. DOI: https://doi.org/10.1016/j.inffus.2024.102616.

BME (2017). *Evolución Histórica de la Cotización del IBEX 35*. URL: https://elpais.com/economia/imagenes/2017/01/13/actualidad/1484312173_786898_1484313696_sumario_normal.png.

Chakravorty, Amitabh (2023). "Applying Machine Learning Algorithms on Insider Trading Data to Predict Stock Prices". School of Information Technology - University of Cincinnati, pp. 1–12. DOI: 10.13140/RG.2.2.14023.06569.

Chang, Victor, Qianwen Xu, Anyamele Chidozie, and Hai Wang (2024). "Predicting Economic Trends and Stock Market Prices with Deep Learning and Advanced Machine Learning Techniques". In: *Electronics* 13, p. 3396. DOI: 10.3390/electronics13173396.

Chatterjee, Ananda, Hrisav Bhowmick, and Jaydip Sen (2021). "Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models". In: *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*. IEEE, pp. 289–296. DOI: 10.1109/MysuruCon52639.2021.9641610.

Chen, Qizhao and Hiroaki Kawashima (2024). "Stock Price Prediction Using LLM-Based Sentiment Analysis". In: *2024 IEEE International Conference on Big Data (BigData)*, pp. 4846–4853. DOI: 10.1109/BigData62323.2024.10825946.

Chiekezie, P and Adekunle Toromade (2024). "Forecasting stock prices and market trends using historical data to aid investment decisions". In: *Finance Accounting Research Journal* 6. DOI: 10.51594/farj.v6i8.1434.

Consoli, Sergio, Matteo Negri, Amirhossein Tebbifakhr, Elisa Tosetti, and Marco Turchi (2022). "Forecasting the IBEX-35 Stock Index Using Deep Learning and News Emotions". In: *Machine Learning, Optimization, and Data Science*. Vol. 13163. Giuseppe Nicosia et al. (Eds), pp. 308–323. DOI: 10.1007/978-3-030-95467-3_23.

Fama, Eugene F. (1970). “Efficient Capital Markets: A Review of Theory and Empirical Work”. In: *The Journal of Finance* 25.2, pp. 383–417.

Fu, Kui and Yanbin Zhang (2024). “Incorporating Multi-Source Market Sentiment and Price Data for Stock Price Prediction”. In: *Mathematics* 12, p. 1572. DOI: 10.3390/math12101572.

Guo, Tian and Emmanuel Hauptmann (2024). “Fine-Tuning Large Language Models for Stock Return Prediction Using Newsflow”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Franck Dernoncourt, F. et al. (Eds)., pp. 1028–1045. DOI: 10.18653/v1/2024.emnlp-industry.77.

Hu, Yuanwei, Zheng Tao, Dongao Xing, Ziyi Pan, Junyi Zhao, and Xiaoling Chen (2020). “Research on Stock Returns Forecast of the Four Major Banks Based on ARMA and GARCH Model”. In: *Journal of Physics: Conference Series* 1616.1, p. 012075. DOI: 10.1088/1742-6596/1616/1/012075.

Huang, Mingxuan (2023). “Effectiveness Validation of LSTM for Stock Prices Prediction on Four Stocks”. In: *Applied and Computational Engineering* 8, pp. 596–601. DOI: 10.54254/2755-2721/8/20230279.

Inserte, Pau Rodriguez, Mariam Nakhlé, Raheel Qader, Gaetan Caillaut, and Jingshu Liu (2024). *Large Language Model Adaptation for Financial Sentiment Analysis*. DOI: 10.48550/arXiv.2401.14777. URL: http://arxiv.org/abs/2401.14777.

Jishtu, Prajwal, Harshil Prajapati, and Jinan Fiaidhi (2022). *Prediction of the Stock Market Based on Machine Learning and Sentiment Analysis*. DOI: 10.36227/techrxiv.21692852.v1.

Jun Gu, Wen, Yi Hao Zhong, Shi Zun Li, Chang Song Wei, Li Ting Dong, Zhuo Yue Wang, and Chao Yan (2024). “Predicting Stock Prices with FinBERT-LSTM: Integrating News Sentiment Analysis”. In: *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing*. ACM, pp. 67–72. DOI: 10.1145/3694860.3694870.

Kahan, Nasir and Oliver Thomas (2024). “DEEP LEARNING IN FINANCIAL FORECASTING: UNLOCKING THE POTENTIAL OF STOCK MARKET PREDICTION”. In.

Kamijo, K. and T. Tanigawa (1990). “Stock price pattern recognition-a recurrent neural network approach”. In: *1990 IJCNN International Joint Conference on Neural Networks*, 215–221 vol.1. DOI: 10.1109/IJCNN.1990.137572.

Khan, Talha, Rehan Sadiq, Zeeshan Shahid, Muhammad Alam, and Mazliham Su’ud (2024). “Sentiment Analysis using Support Vector Machine and Random Forest”. In: *Journal of Informatics and Web Engineering* 3, pp. 67–75. DOI: 10.33093/jiwe.2024.3.1.5.

Kim, Jihwan, Hui-Sang Kim, and Sun-Yong Choi (2023). "Forecasting the S&P 500 Index Using Mathematical-Based Sentiment Analysis and Deep Learning Models: A FinBERT Transformer Model and LSTM". In: *Axioms* 12, p. 835. DOI: 10.3390/axioms12090835.

Kirtac, Kemal and Guido Germano (2024). "Sentiment trading with large language models". In: *Finance Research Letters* 62, p. 105227. DOI: https://doi.org/10.1016/j.frl.2024.105227. URL: https://www.sciencedirect.com/science/article/pii/S1544612324002575.

Kumar, Deepak, Pradeepta Kumar Sarangi, and Rajit Verma (2022). "A systematic review of stock market prediction using machine learning and statistical techniques". In: *Materials Today: Proceedings* 49 (part 8), pp. 3187–3191. DOI: 10.1016/j.matpr.2020.11.399.

Kumar, Rajesh and Sujit K S (2024). "Impact of sentimental factors on stock portfolio returns –an empirical analysis". In: *Heliyon* 10.9, e30217. DOI: 10.1016/j.heliyon.2024.e30217.

Kumbure, Mahinda Mailagaha, Christoph Lohrmann, Pasi Luukka, and Jari Porras (2022). "Machine learning techniques and data for stock market forecasting: A literature review". In: *Expert Systems with Applications* 197, p. 116659. DOI: https://doi.org/10.1016/j.eswa.2022.116659.

Lakshya, Lakshya, Prateek, and Divyasikha Sethia (2022). "Stock Price Prediction Using News Sentiment Analysis". In: *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pp. 1–6. DOI: 10.1109/CONIT55038.2022.9847747.

Latif, Madiha, Shanza Arshad, Mariam Fatima, and Samia Farooq (2011). "Market Efficiency, Market Anomalies, Causes, Evidences, and Some Behavioral Aspects of Market Anomalies". In: *Research Journal of Finance and Accounting* 2.

Lee, Jean, Nicholas Stevens, Soyeon Caren Han, and Minseok Song (2025). "A Survey of Large Language Models in Finance (FinLLMs)". In: *Neural Computing and Applications*. DOI: 10.1007/s00521-024-10495-6.

Li, Yinheng, Shaofei Wang, Han Ding, and Hang Chen (2023). "Large Language Models in Finance: A Survey". In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. Association for Computing Machinery, pp. 374–382. DOI: 10.1145/3604237.3626869.

Liu, Jin-Xian, Jenq-Shiou Leu, and Stefan Holst (2023). "Stock price movement prediction based on Stocktwits investor sentiment using FinBERT and ensemble SVM". In: *PeerJ Computer Science* 9, e1403. DOI: 10.7717/peerj-cs.1403.

Lo, Andrew (2004). "The Adaptive Markets Hypothesis". In: *The Journal of Portfolio Management* 30, pp. 15–29.

Ma, Heyang (2024). "Using ARIMA Model to Forecast the Stock Price of Microsoft". In: *Theoretical and Natural Science* 42, pp. 135–144. DOI: 10.54254/2753-8818/42/2024CH0224.

Mai, Dat (2024). "StockGPT: A GenAI Model for Stock Prediction and Trading". In: pp. 1–26. DOI: 10.2139/ssrn.4787199.

Malo, Pekka, Ankur Sinha, Pyry Takala, Pekka Korhonen, and Jyrki Wallenius (Apr. 2014). "Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts". In: *Journal of the American Society for Information Science and Technology*. DOI: 10.1002/asi.23062.

Marisetty, Nagendra (2024). "Applications of GARCH Models in Forecasting Financial Market Volatility: Insights from Leading Global Stock Indexes". In: *Asian Journal of Economics, Business and Accounting* 24.9, pp. 63–84. DOI: 10.9734/ajeba/2024/v24i91477.

Oriol, Juan Luis Ruiz-Tagle (2023). "Stock Forecasting Utilizing News Sentiment Data: A Comparative Study". Master Thesis. Universidad Politécnica de Madrid.

Pandya, Kush, Gaurav Kulkarni, Vrunda Patel, Eralwala Abbas Mohammadbhai, and Dr Mujtaba Ashraf Qureshi (2025). "Deep Learning Approaches for Financial Sentiment Prediction: Leveraging LSTM and Transformer Models". In: *IJIRT* 11.10.

Shiller, Robert J. (2003). "From Efficient Markets Theory to Behavioral Finance". In: *Journal of Economic Perspectives* 17.1, pp. 83–104. DOI: 10.1257/089533003321164967.

Song, Ziyu (2023). "Research on Investor Sentiment and Its Impact on the Market for Stocks". In: *Advances in Economics, Management and Political Sciences* 38.1, pp. 121–127. DOI: 10.54254/2754-1169/38/20231896.

Talazadeh, Saber and Dragan Perakovic (2024). *SARF: Enhancing Stock Market Prediction with Sentiment-Augmented Random Forest*. DOI: 10.48550/arXiv.2410.07143.

Todorov, Ivan and Fernando Sánchez-Lasheras (2023). "Stock Price Forecasting of IBEX35 Companies in the Petroleum, Electricity, and Gas Industries". In: *Energies* 16, p. 3856. DOI: 10.3390/en16093856.

Turner, Zane, Kevin Labille, and Susan Gauch (2021). "Lexicon-based sentiment analysis for stock movement prediction". In: *Journal of Construction Materials* 2.3. DOI: 10.36756/JCM.v2.3.5.

Wabella, Akinahom (2024). "Data Imputation: Numerical Analysis in Data Science". In: DOI: 10.13140/RG.2.2.20472.76800.

Wang, Shengkun, Taoran Ji, Linhan Wang, Yanshen Sun, Shang-Ching Liu, Amit Kumar, and Chang-Tien Lu (2024). "StockTime: A Time Series Specialized Large Language Model Architecture for Stock Price Prediction". In: DOI: 10.48550/arXiv.2409.08281.

Yang, Jingfeng, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu (2024). "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond". In: *ACM Trans. Knowl. Discov. Data* 18.6. DOI: 10.1145/3649506.

Zhang, Yifan (2022). "Stock Price Prediction Method Based on XGboost Algorithm". In: pp. 595–603. DOI: 10.2991/978-94-6463-030-5_60.

Zheng, Jiajian, Duan Xin, Qishuo Cheng, Miao Tian, and Le Yang (2024). "The Random Forest Model for analyzing and Forecasting the US Stock Market under the background of smart finance". In: *Proceedings of the 3rd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2024).* Atlantis Press, pp. 82–90. DOI: 10.2991/978-94-6463-419-8_11.