What's News with You: Price Forecasting and Sentiment Scores

Nico Rosamilia[‡]

February, 2025

Abstract

We adopt deep learning models to forecast stock prices and to compute news sentiment scores. The aim is to determine the predictive power of the sentiment for the Dow Jones Industrial Average index price. More precisely, we employ N-HiTS, a neural network architecture designed for time series forecasting, and RoBERTa, a powerful tool in sentiment analysis, to predict next-day closing prices exploiting sentiment scores and historical closing prices. We find that the N-HiTS model, when trained with comprehensive sentiment data from all news, significantly outperforms all the considered models across all key metrics. A model trained exclusively with ESG news sentiment improves upon the baseline model in the directional accuracy, delivering the best-predicting power of the price downward movements.

Keywords: Time series forecasting, sentiment analysis, ESG, deep learning.
JEL Classification: G11, G14, G17, G23.

[‡] Politecnico di Milano, School of Management, Milano, Italy. nico.rosamilia@polimi.it

1 Introduction

Stock price prediction is an ongoing issue in literature and investment management. Timmermann and Granger (2004) analyze the Fama (1965) Efficient Market Hypothesis (EMH) and its implication in financial forecasting, highlighting the need for adaptable models that respond quickly to new information and capture the complexities of financial markets. The authors criticize traditional models highlighting that every new model experience a "honeymoon" period when it effectively predicts the market, before the market itself adapts to the inefficiencies that drive the model predicting power.

Researchers utilize several different techniques to predict stock prices. Gandhmal and Kumar (2019) review fifty research papers that propose methodologies for prediction. The most common techniques in this field range from traditional statistical models, e.g., ARIMA in Ariyo et al. (2014), to machine learning techniques, e.g., support-vector machines (SVM) in Huang et al. (2005). Recent literature explores deep learning models for prediction and forecasting. Lu et al. (2021) adopt a deep learning model integrating Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Attention Mechanism (AM). The study conducts an extensive experimental evaluation using data from the Shanghai Composite Index, comparing the proposed model against traditional and other advanced machine learning models, and concludes that the CNN-BiLSTM-AM model outperforms other machine learning models in predicting next-day closing price of stocks.

Models for stock price prediction face the dilemma regarding the information relevant to improving the models predicting power. Different articles employ historical prices and technical indicators,[1] see Badge et al. (2012), Ticknor (2013), and Patel et al. (2015) among others. Sezer and Ozbayoglu (2018) build CNN-TA, which combines Convolutional Neural Networks (CNN) with Technical Analysis (TA), transforming time series financial data into 2-D image format. The model adopts 15 technical indicators to transform time series data into a 15x15 pixel image. Each pixel represents a specific attribute or indicator over 15 days. On the other hand, several papers incorporate sentiment information from news and social media. Numerous studies deal with the relation between news and stock prices, demonstrating that stock prices not only react to current economic conditions but also anticipate future productivity based on existing news, see Pearce and Roley (1984), Cutler et al. (1998), Beaudry and Portier (2006), Boudoukh et al. (2013) among others. These articles prove that stock prices react significantly to unexpected components of economic news, particularly monetary policy changes.

Recently, Environmental, Social, and Governance (ESG hereafter) ratings represent relevant investment factors for investors and companies. The relation between ESG and stock performance is ambiguous or not uniformly inclined in a defined direction. Indeed, while Gunnar Friede and Bassen (2015) and Khan (2019) show that high ESG ratings often correlate with superior financial performance, Rosamilia (2024) proves that ESG may be costly, diverting resources from short to long-term profitability and investments. In this regard, Capelle-Blancard and Petit (2019) finds that negative ESG events lead to a small but significant decrease in market value, whereas positive ESG news does not generally lead to an increase. Conversely, in this work, we aim to unfold the relevance of introducing ESG-type information in the prediction of stock prices, independently of the impact of sustainable disclosure on corporate value.

The technique used in this work builds on the model from Bollen et al. (2011). The authors propose that social media sentiment can correlate and predict the Dow Jones Industrial Average (DJIA) price. Utilizing two mood tracking tools, OpinionFinder and Google-Profile of Mood States (GPOMS), they analyze tweets

[1] A technical indicator is a mathematical calculation that uses historical price, volume, or open interest information of a security or contract, e.g., moving averages, rate of change, volume rate of change.

for positive vs. negative sentiments and six mood dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). Additionally, the authors utilize a self-organized fuzzy neural network (SOFNN) with historical information and sentiment to predict the next-day price of the DJIA. They find that specific mood dimensions, particularly "Calm", have a predictive relationship with DJIA price, suggesting that including specific public mood dimensions can significantly improve the accuracy of stock market predictions. The study challenges the EMH by showing that social media sentiment, an early indicator of public mood, can provide valuable insights into stock market movements. Nguyen et al. (2015) explore the predictive power of sentiment analysis derived from social media data on stock market movements. They develop a model that considers the overall sentiment from social media and focuses on the sentiments tied to specific topics related to companies. This approach represents a novelty in sentiment analysis methods, which generally assess the overall mood or sentiment without analyzing topic-specific sentiments. The aspect-based sentiment analysis method outperforms other methods by achieving higher accuracy in predicting stock movements, thereby highlighting the value of analyzing sentiments associated with specific topics. The authors compute the sentiment with an SVM model trained to classify unlabeled social media posts. In Mittal and Goel (2012) authors explore the correlation between public sentiment, as expressed on Twitter, and market sentiment, measured with the DJIA performance. Their study validates the concept of using social media sentiment in stock market predictions and suggests a practical application through portfolio management strategies based on predicted values.

From the above literature review, it is evident that the majority of studies deal with social media sentiment computed with lexicon-based approaches. On the contrary, this article investigates the predictive power of (ESG) news sentiment computed with a deep learning approach. Dealing with machine learning based applications, Vargas et al. (2017) explore deep learning methods, specifically Recurrent Convolutional Neural Networks, to predict the Standard & Poor's 500 index returns. The authors apply financial news title embeddings computed by word2vec and a set of technical indicators, highlighting the potential of deep learning for forecasting models in finance. While they run deep learning and train their model on a classification task, that is, their algorithm predicts if the price will increase (label [1,0]) or decrease (label [0,1]), we train our for a regression task, that is, our algorithm predicts the future stock price. A similar deep learning application comes from Mohan et al. (2019), which aim to improve the accuracy of stock price predictions by combining time series data with financial news articles. The study highlights the relevance of integrating news sentiment analysis with historical data to predict stock prices using all news. On the other hand, we add ESG-related news to predict the price of the DJIA, achieving similar results with a much smaller dataset. Similarly, Schmidt (2019) investigates the impact of ESG-related news sentiment on the stock market performance of DJIA constituents. The author employs a dataset of news articles from 2010 to 2018 and a dictionary approach to calculate a polarity-based sentiment index for ESG-related topics. He applies an Autoregressive Distributed Lag (ARDL) model that reveals significant effects of temporary and permanent changes in ESG news sentiment on the idiosyncratic returns of DJIA stocks. In his study, he shows that researchers can categorize stocks based on how investors react to ESG news. This suggests a link between investor behavior towards ESG news and stock financial performance. Similarly to the latter paper, we explore the ESG-related news prediction power for the DJIA price but with a deep learning approach for forecasting and sentiment computation. This approach improves the forecasting accuracy of Nyakurukwa and Seetharam (2023), which demonstrate that positive news has a greater effect in magnitude than the negative news on the Johannesburg Stock Exchange. Finally, Damrongsakmethee and Neagoe (2020) develop a predictive model using a variant of the Long Short-Term Memory (Deep LSTM) network to forecast stock

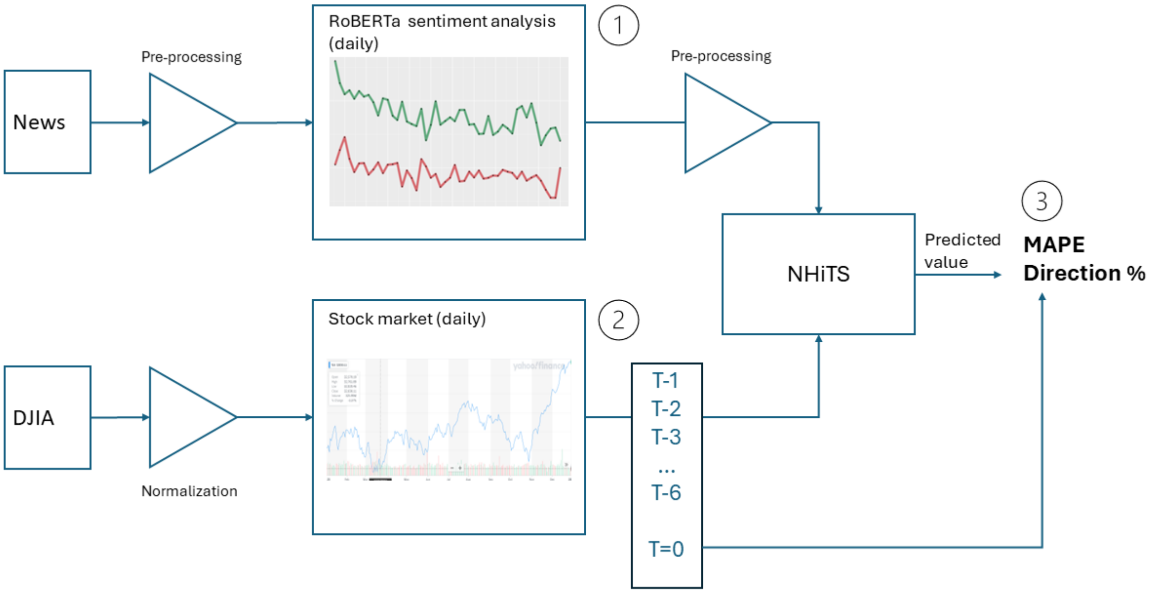


Figure 1: Diagram of the methodology: (1) finetuned version of distilled RoBERTa is used to compute the sentiment of news articles, (2) closing prices of DJIA are collected and normalized, (3) N-HiTS model is trained to predict the next day closing price of the DJIA using both historical information and sentiment scores.

prices.

In this work, we aim to improve the above results using a Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS hereafter) model, see Challu et al. (2022), incorporating sentiment information. More precisely, we investigate the prediction power of news sentiment, particularly ESG news, for the stock market. Unlike most research that relies on lexicon-based models, we adopt deep learning models to compute daily news sentiment and to predict the next-day closing price. To compute the sentiment, we utilize a distilled version of RoBERTa on a dataset of articles sourced from Refinitiv. The dataset includes news related to companies in the DJIA, eventually filtered specifically for ESG topics. Then, we use the last six historical prices of the DJIA and daily sentiments to predict the next closing price. To compute the sentiment score we weight each news score accordingly with the market capitalization of the company it refers to. This allows us to improve predicting performance. Finally, we compare the prediction results of the model when fed with all news versus only ESG news. The results agree with Capelle-Blancard and Petit (2019) in finding that the model trained with ESG sentiment best predicts downward movements of stock prices, suggesting a greater impact of ESG news on a decrease in stock prices.

The article is organized as follows: Section 2 outlines the methodology used, while Section 3 presents the main results of the study. Finally, Section 4 concludes.

2 Methodology

In the following, Section 2.1 shows the data, and then we move to the methodology, summarized in Figure 1. More precisely, Sections 2.2 and 2.3 focus on the sentiment computation and the price forecast, respectively.

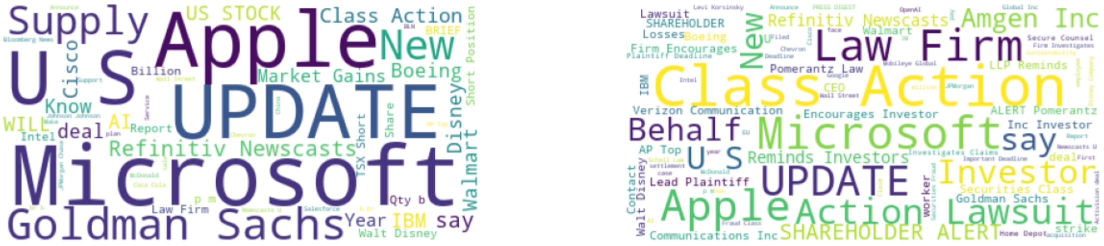


Figure 2: Word cloud of dataset containing all news (left) and ESG news (right)

2.1 Data Collection and Preparation

The study utilizes the historical daily closing prices of the DJIA and related news articles. The news dataset comprehends a general dataset containing all news articles related to DJIA and a specialized dataset containing exclusively news articles related to ESG issues. We source datasets from Refinitiv via its Python API filtering by news topic to extract these ESG-specific articles.

Both datasets span from January 1, 2023, to February 29, 2024.[2] The dataset of all news contains 17382 news articles, and the dataset of ESG news contains 7226 news articles. Figure 2 shows the word clouds of the titles of the two datasets of articles. We can appreciate that the more frequent words when all the news are considered are the names of the most important components of DJIA (e.g., Microsoft and Apple), while if we deal with only the titles of the ESG articles, some of the most important words are linked to legal topics (e.g., Class Action, Law Firm, Lawsuit), which are highly related to the Governance (G) and Social (S) pillars.

The textual news data required extensive cleaning to remove HTML tags, hyperlinks, and extraneous metadata that could interfere with the sentiment analysis. Then, we apply preprocessing techniques to both datasets to prepare them for the analysis:

- DJIA Time Series: we normalize the price data to facilitate comparative analysis and integration in predictive modeling.
- News Articles: text preprocessing included removing punctuation, stopwords, and non-vocabulary words, as well as converting all text to lowercase. These steps were essential for standardizing the text data for effective sentiment computation.

2.2 Sentiment Analysis

This section outlines the process used to compute the sentiment, starting from the news articles and arriving at a time series of daily sentiments.

We compute the sentiment with a distilled model version of RoBERTa, DistilRoBERTa, fine-tuned on a dataset of financial news.[3] Models similar to this one are extensively used in literature and in particular in

[2] The dataset length is constrained by news availability, which is limited to 14 months.

[3] The software is available here: https://huggingface.co/mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis

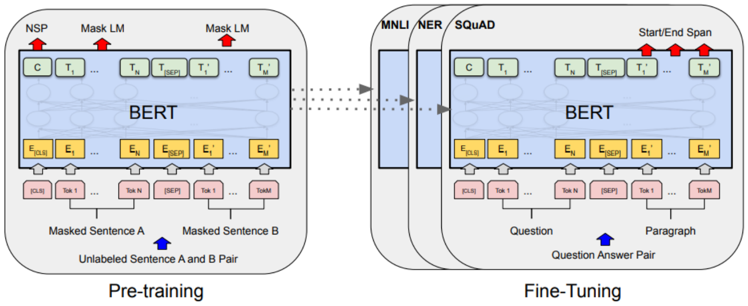


Figure 3: Overview of BERT training phases, Devlin et al. (2019)

financial contexts, e.g., Banerjee et al. (2024). In Bozanta et al. (2021) the authors compare the performances of various deep learning and pre-trained transformers models for sentiment analysis of tweets related to the stock market. Results show that RoBERTa outperforms traditional classifiers and that DistilRoBERTa offers an efficient alternative that facilitates quicker inference without substantial loss in accuracy. The following sections overview BERT, RoBERTa, and the knowledge distillation technique.

2.2.1 From BERT to RoBERTa

BERT (Bidirectional Encoder Representations from Transformers), Devlin et al. (2019), is a model developed by Google. Its architecture is based on the Transformer architecture, which involves a self-attention mechanism that makes the model focus on different parts of a sentence as needed per task.

Figure 3 shows the two phases of BERT training. The first one, the pre-training, involves two steps: first, a Masked Language Model (MLM) task and next a next-sentence prediction task. During this phase, the model is trained on a large corpus of text (BooksCorpus and English Wikipedia, which combined are approximately 3300M words). The second one is fine-tuning: during this phase, BERT specializes in one specific task, e.g., question answering or sentiment analysis.

RoBERTa (Robustly Optimized BERT Approach), Yinhan Liu et al. (2019), is an improvement of BERT from Facebook researchers that modified some training techniques to improve performances. It introduces several significant changes to the BERT pretraining approach that address undertraining issues and optimize performance:

- Longer Training Time: RoBERTa is trained for longer periods over more data.
- Removal of NSP (Next Sentence Prediction) Objective: RoBERTa developers remove this task, present in BERT, after discovering that it does not contribute significantly to the performance of downstream tasks.
- Dynamic Masking: Unlike BERT, which uses static masks in the MLM task, during the training of RoBERTa, masking patterns chenge dynamically.
- Training on Longer Sequences: RoBERTa processes longer sequences during training than BERT, which improves its ability to understand and generate context over longer stretches of text.

After these optimizations, RoBERTa demonstrates superior performance on several benchmarks. In Zhao et al. (2020), authors demonstrate that RoBERTa performs better than traditional lexicon-based methods in sentiment analysis tasks in a financial context; their work adopts the RoBERTa base model for sentiment analysis and key entity classification.

2.2.2 Distillation

Model distillation technique aims at compressing the knowledge from a large, cumbersome model (the teacher) into a smaller, more efficient model (the student) without significant loss of accuracy. This concept, first outlined by Hinton et al. (2015), leverages the teacher model's soft outputs (probability distributions) as training signals for the student model. This approach enables the student to learn both the hard targets and the probability relationships between classes learned by the teacher. In Mishev et al. (2020), authors analyze numerous methods for sentiment analysis in a financial context, and they conclude that the distilled version of RoBERTa retains the accuracy of the teacher model.

The fundamental concept of distillation is to train a student model to replicate the output behavior of a teacher model, particularly by learning from the teacher's output probabilities across various classes. These output probabilities, also known as "soft targets", convey nuanced information about the relationships that the teacher model has learned, providing richer information per training example compared to traditional "hard labels".

The key steps in the distillation process are as follows:

1. Training the Teacher Model: The teacher model is trained using standard supervised learning methods to achieve high accuracy. This model is usually larger and more complex and may even consist of an ensemble of multiple models.

2. Generating Soft Targets: The teacher model's output probabilities, or soft targets, are recorded for each input and are later used as targets for training the student model.

3. Training the Student Model: The student model is trained using a combination of the soft targets provided by the teacher model and the original hard labels from the training dataset.

An example of this process is the DistilRoBERTa model, which is a smaller and faster version of the original RoBERTa model. It was pre-trained on the same corpus as RoBERTa in a self-supervised fashion, with the base RoBERTa model acting as the teacher. In this technique, the student model is pre-trained on raw text without any human-labeled data. Instead, inputs and labels are generated automatically from the texts using the outputs of the RoBERTa base model. The primary training objectives are:

1. Distillation Loss: The student model is trained to produce output probabilities that closely match those of the RoBERTa base model.

2. Masked Language Modeling (MLM): This objective mirrors the original training objective of RoBERTa. The model takes a sentence, randomly masks 15% of the words in the input, processes the masked sentence, and then predicts the masked words. Unlike traditional recurrent neural networks (RNNs), which process words sequentially, or autoregressive models like GPT that mask future tokens, this approach allows the model to learn bidirectional representations of sentences.

3. Cosine Embedding Loss: The model is additionally trained to generate hidden states that closely resemble those of the RoBERTa base model, thereby encouraging the student model to align its internal representations with those of the teacher.

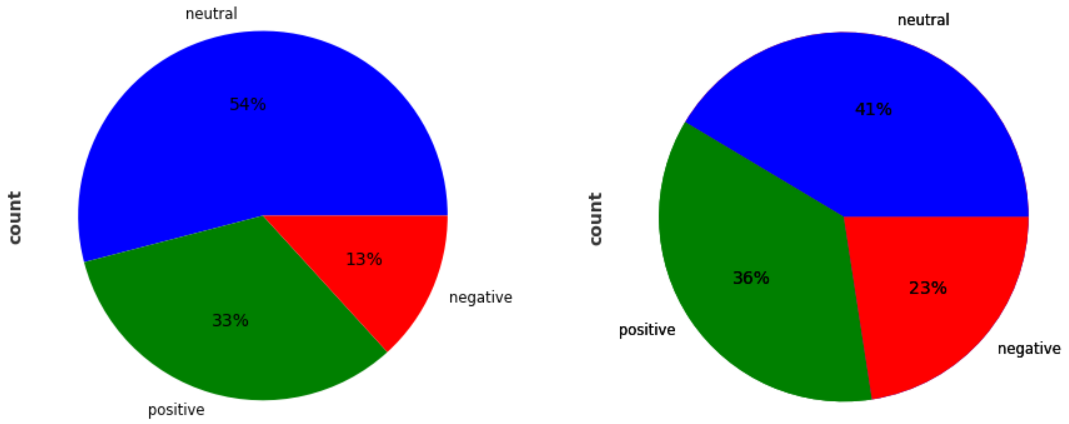


Figure 4: Sentiment labels for all news (left) and ESG news (right)

The resulting distilled model has 35% fewer parameters than the RoBERTa base model and, on average, runs twice as fast.

2.2.3 Sentiment Computation

We compute the sentiment analysis with a distilled version of the RoBERTa model fine-tuned for financial news sentiment analysis. Malo et al. (2014) construct the dataset to fine-tune the model. It comprises 4840 sentences from English financial news categorized by sentiment, humanly annotated by financial experts.

The model processes a concatenated input of the article's title and truncated text, limited by the maximum token count allowed by the model. It produces three values corresponding to three sentiment categories: positive, neutral, and negative. Each article is later labeled with the category that receives the highest value. Figure 4 represents the label distribution in the two datasets.

We implement a weighting mechanism to account for the varying impact of news on DJIA stock prices. Initially, each news item is linked to a DJIA company by searching for the company's name in the article's title and text using regular expressions. Subsequently, the sentiment score of the news is weighted based on the market capitalization of the associated company within the DJIA. If no company is linked to the article, we associate a weight equal to the mean of the weights.

The frequent occurrence of the same news reported by multiple outlets is an important issue to consider. We introduce an exponential decay mechanism to diminish the influence of similar articles within the same day, thus preventing these repeated articles from being overly weighted. Initially, we construct a TF-IDF (Term Frequency-Inverse Document Frequency) matrix for the titles. TF-IDF is a natural language processing technique used to evaluate the importance of a word in a document relative to a collection of documents. It combines two metrics: Term Frequency (TF) and Inverse Document Frequency (IDF).

The TF is defined as:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{1}$$

where $f_{t,d}$ is the frequency of a term in a document. The denominators represent the total number of terms in document d. This metric calculates how often a term appears in a document relative to the total number of terms. It helps to understand the importance of a term within a specific document. Furthermore, the

IDF is:

$$idf(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right), \tag{2}$$

where N is the total number of documents in the corpus D and $|\{d \in D : t \in d\}|$ is the number of documents in the corpus D that contain term t. The formula measures how important a term is within the entire corpus. A term that appears in many documents is less informative than one in fewer documents. Finally, the TF-IDF score

$$tf\_idf(t, d, D) = tf(t, d) \times idf(t, D). \tag{3}$$

reflects the importance of a term in a specific document relative to its importance in the entire corpus. This formula combines TF and IDF to assign a weight to each term in a document.

We compute the TF-IDF matrix which has as columns the terms inside the corpus, the titles of the articles in this case, and as rows the various articles. Each cell contains the TF-IDF score of a term concerning a specific article. Then, we compute the cosine similarity between all pairs of rows of this matrix. The weight of each article is later adjusted by applying an exponential decay based on the count of similar articles on the same day. We consider two articles similar if the cosine similarity between the two rows of the TF-IDF matrix is greater than 0.5.

Furthermore, we create a time series of sentiment by grouping the articles per day and computing a daily sentiment in the following way:

$$sentiment = positiveSentiment - negativeSentiment \tag{4}$$

where:

- *positiveSentiment*: is the sum of the positive scores of the articles with positive labels for a given day
- *negativeSentiment*: is the sum of the negative scores of the articles with negative labels for a given day

Figure 5 shows the resulting sentiment time series considering all news and ESG news respectively. Also, the three data points with the highest and the three with the lowest sentiment score are annotated with significant news of that day.

2.3 Price Forecasting

We forecast the next-day closing price of DJIA with N-HiTS (Neural Hierarchical Interpolation for Time Series). N-HiTS is a deep-learning model that utilizes several multilayer perceptions (MLP). An MLP is a modern feedforward artificial neural network, consisting of fully connected neurons with a nonlinear activation function, organized in at least three layers. This model presents several advantages: first, the model achieves state-of-the-art performance in time series forecasting tasks and has been specifically designed to handle and forecast sequential data. Also, the model's flexibility allows for the easy integration of sentiment as input. Additionally, it is specifically designed to predict time series. N-HiTS improves performance, reducing computation time, and the architecture of Neural basis expansion analysis for interpretable time series forecasting (N-BEATS). While in Yichen Liu et al. (2023) authors employ N-BEATS to predict stock prices, demonstrating better performance than traditional and deep learning models, in Jeffrey et al. (2023), authors utiliz eN-HiTS to predict stock prices, demonstrating the superior performance of N-HiTS compared to its predecessor.

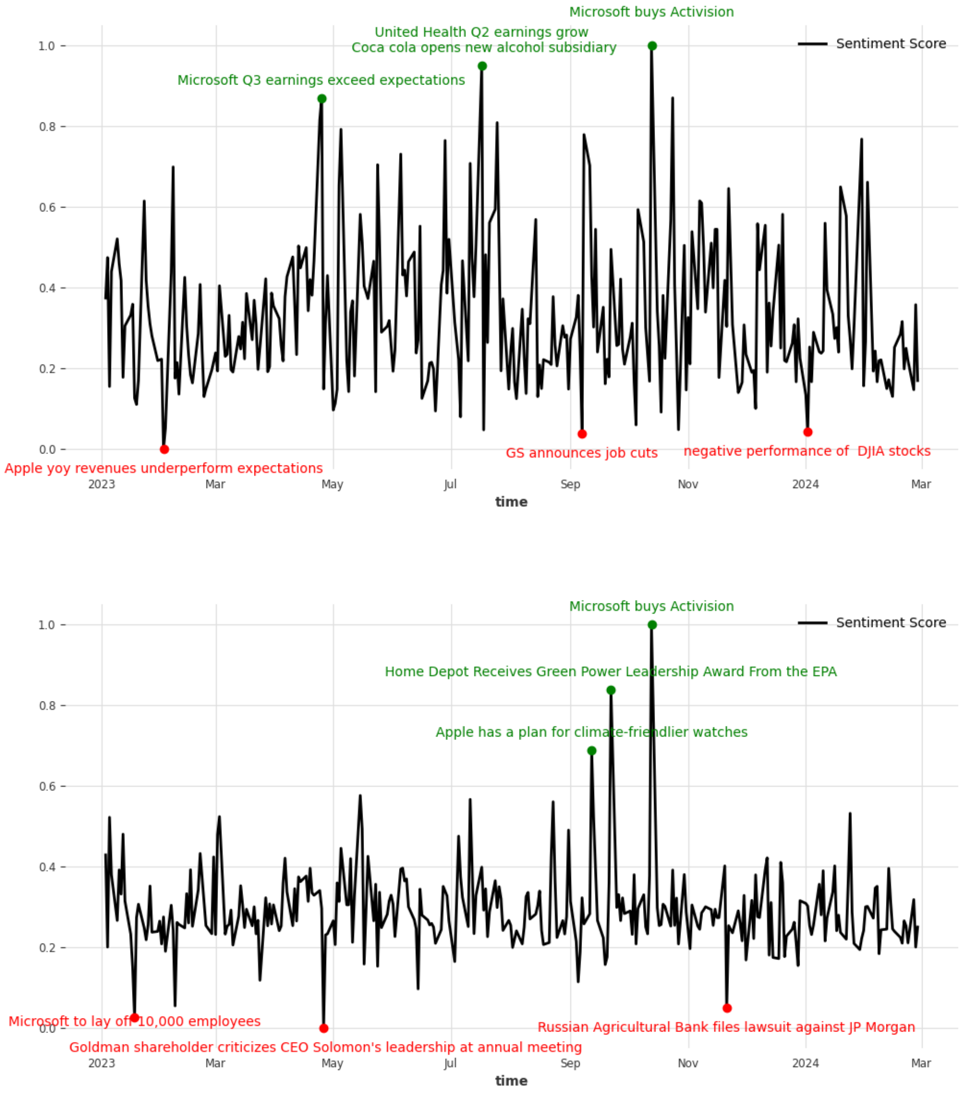


Figure 5: Sentiment time series of all news (above) and for ESG news (below)

This section explains the model used to predict the closing price, starting from N-BEATS and next moving to N-HiTS, and the hyperparameters tuning process.

2.3.1 From N-BEATS to N-HiTS

N-BEATS, introduced in Oreshkin et al. (2020), is a pure Deep Learning architecture for univariate time series forecasting. Before the publication of this model, pure statistical methods or ensemble of statistical methods and ML/Deep Learning models were common for time series forecasting tasks. While the former are fully explainable, the latter exhibit better performances in forecasting, at the cost of the loss of explainability (*black-box* models). N-BEATS demonstrates high performances and provides a way to construct an architecture that produces interpretable results similar to classical statistical methods, i.e., decomposition techniques seasonality-trend-level. Figure 6 shows an overview of the architecture.

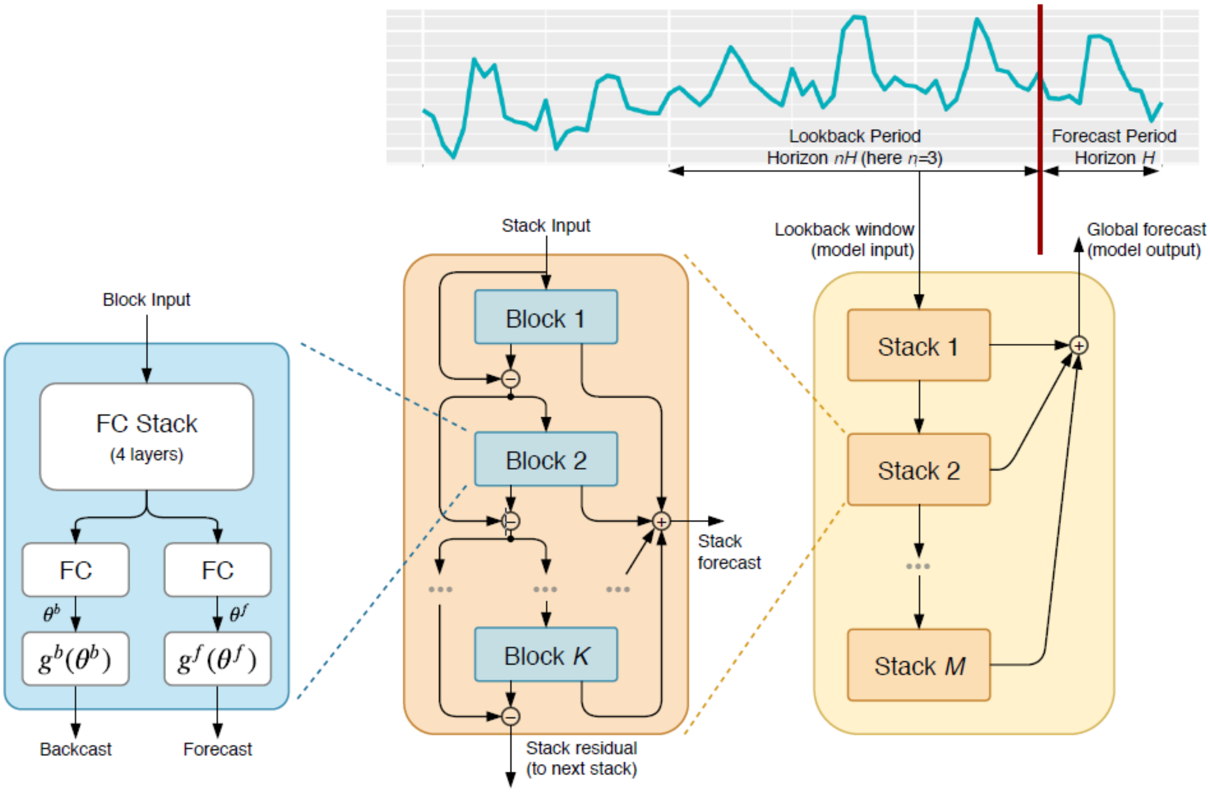


Figure 6: N-BEATS architecture, Oreshkin et al. (2020)

N-BEATS operates on the principle of basis expansion, decomposing the time series into interpretable components. The architecture is built with fully connected neural networks and applies a recursive strategy to forecast future values. The model's functioning presents stacks, the right part of Figure 6, which are essentially a collection of blocks that work together to refine the forecast, with each stack having a specialized purpose. N-BEATS typically utlizes two types of stacks: trend and seasonality stacks (the interpretable components). Each stack is responsible for capturing a different aspect of the time series data, and it is organized as a series of basic blocks (left part of Figure 6). In fact the model works recursively (center of Figure 6): the first block takes the input time series and outputs both a backcast and a forecast. The backcast is subtracted from the input time series, leaving a residual time series (the parts that the first block could not predict). This residual is next fed into the next block, which attempts to forecast the remaining residual. This process is repeated across several blocks, progressively refining the forecast.

More precisely, the basic block receives in input $\mathbf{x}$ (that for the first block is the input to the model, that is the lookback period of the time series to predict), and outputs two vectors: $\hat{\mathbf{x}}$, the block estimate of $\mathbf{x}$, and $\hat{\mathbf{y}}$, the block forecast of $\mathbf{y}$. The block consists of two parts. The first is a neural network that produces the forward, θ^f, and backward, θ^b, predictors of expansion coefficients. The second part consists of the backward, g^b, and the forward, g^f, basis layers that accept the expansion coefficients to produce the backcast $\hat{\mathbf{x}}$ and the forecast $\hat{\mathbf{y}}$. The model aims to predict the coefficients that optimize the partial forecast. Blocks are organized into stacks using doubly residual stacking principle. Unlike classical residual architecture, where the input of one stack is summed to the output of the following stack, this architecture has two residual branches, one for the backcast and the other for the forecast. The following equations describe the functioning:

$$\mathbf{x}_l = \mathbf{x}_{l-1} - \hat{\mathbf{x}}_{l-1}, \tag{5}$$

$$\hat{\mathbf{y}} = \sum_l \hat{\mathbf{y}}_l. \tag{6}$$

The idea is that every block passes as input to the next one the part of the input that it cannot explain, removing the part that it can approximate well, thus making the job of the next block easier. Equation (5) shows how the input is passed from one block to the next one. Each block produces a partial forecast that is summed first at the stack level and then at the overall network level, as explained in equation (6). The final forecast is the sum of all partial forecasts.

Yichen Liu et al. (2023) demonstrate the effectiveness of N-BEATS in predicting the S&P 500, achieving significant forecasting accuracy and exceeding the performance of other ML models. The authors employ a dataset of 500 stocks and the S&P 500 index; they predict the price of the next minute of the S&P using as input the prices of the last 15 minutes of the 500 stocks and the index. Regardless of the good results of this model, we underline some limitations:

- Lack of temporal dependencies: N-BEATS basic block uses an MLP that does not capture temporal dependencies as architecture such as LSTM, RNNs, and gated recurrent units (GRUs) do.
- Data requirement: every deep learning model training requires large data.
- Computational resources: when the number of stack and blocks increase, the number of parameters increases drastically, making the model computationally intensive to train.

Starting from N-BEAST architecture, authors in Challu et al. (2022) create a new N-HiTS model that improves performance and efficiency. The computational efficiency and the capacity of N-HiTS to capture both short-term and long-term tendencies of the time series is helpful in a time series forecasting cast in the financial context. Similar to N-BEATS, this model's block has an MLP that produces the coefficients to utilize with the basis function to output a backcast to clean the input for subsequent blocks and the forecast to hierarchically aggregate first at the stack level and later at the architecture level. The differences with the previous model lie in the following points:

- Multi-rate signal sampling: at the beginning of each basic block, there is a MaxPool layer with kernel size k_l. This layer has the effect that every block sees a different frequency of the input time series, and

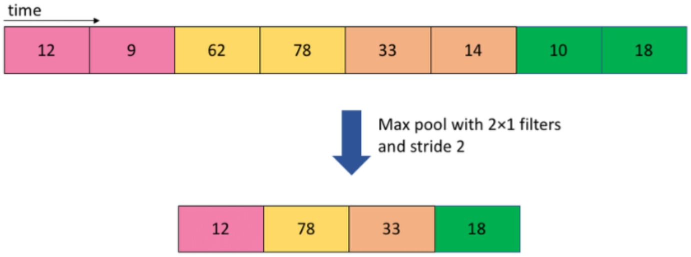


Figure 7: An example of how a max pool layer operates on a time series

blocks with larger kernel sizes results in filtering out high-frequency components from the input time series. Figure 7 shows a visual example of this process. This technique can be helpful in the financial context, as short-term fluctuations and long-term trends affect the daily movement of stock prices.

- Hierarchical interpolation: the multi-rate signal sampling results in a prediction dimensionality different than the prediction forecast horizon. The solution for this discrepancy is a linear temporal interpolation mechanism that allows the model to recover the initial sampling rate and to predict all the points of the forecasting horizon.

These improvements result in better performance and a more computationally efficient model. Indeed, Challu et al. (2022) demonstrate that N-HiTS is 1.26x faster and requires 56% of the parameters compared to N-BEATS.

2.3.2 Hyperparameter tuning and training

Hyperparameter tuning in ML is the process of selecting the optimal set of hyperparameters for a machine learning model to improve its performance. Hyperparameters are the configuration settings or parameters that define the structure of the model and control the learning process, but they are not learned from the data. Instead, they are set before training begins.

Hyperparameter deeply impact the architecture, the learning process, and the performance of the model. The hyperparameter space can become substantial and intricate in architecturally complex models like N-HiTS. Exploring such extensive and complex space can be computationally expensive and time-consuming. In particular, the hierarchical structure of N-HiTS makes this process more complicated, given the large amount of parameters that reciprocally influences each other (e.g., increasing the number of stacks would result in more blocks and, thus, more neurons). For this reason, we exploit an advanced hyperparameter tuning framework like Optuna (Akiba et al. (2019)), an open-source optimization framework designed to tune hyperparameters in ML models.

Table 1 presents the hyperparameters, the search space, and the optimal values selected. The *input window size* represents the number of past data points adopted to predict the next one. In this case, we employ the six previous closing prices and daily sentiments to compute the next closing prices. N-HiTS architecture possesses fully connected neural networks, the multilayer perceptrons, and the architecture of these multilayer perceptrons is defined by the hyperparameters *layer number* and *layer width*, that respectively represent the number of hidden layers of the multilayer perceptron and the number of neurons for each layer. Next, each architecture block has two MLPs to produce a backcast and forecast. The *blocks*

Parameter	**Search space**	**Chosen value**
input window size	[5,10]	6
stacks number	[3,10]	5
blocks number	[2,5]	3
layers number	[2,4]	3
layers width	[128,1024]	555
dropout	[0, 0.4]	0.32
batch size	-	64
epochs	-	100
patience	-	15

Table 1: Hyparameters, search space, and choosen values.

number hyperparameter defines the number of blocks. Blocks are subsequently organized into stacks, with the number of stacks defined by the hyperparameter, *stacks number*.

One issue with neural networks is overfitting, a phenomenon where the network performs exceptionally well on the training data but fails to generalize to new, unseen data. A common technique to prevent neural networks from overfitting is *dropout*, see Srivastava et al. (2014). This technique consists of temporarily removing some neurons from the network, along with its incoming and outgoing connections, thus avoiding the network becoming reliant on specific neurons and paths in the architecture that would next lead to overfitting. The *dropout* hyperparameter represents the percentage of neurons to drop during each training iteration. The batch size represents the number of training samples processed together in one forward and backward pass through the network. In this paper, the *batch size* hyperparameter is not optimized; instead, the largest batch size that did not cause memory errors was selected, since larger batch sizes require more memory. An epoch is a complete pass through the entire training dataset; at the end of one epoch, the neural network has seen every sample in the training dataset once. The *epochs* hyperparameters here define the maximum number of epochs for training. We adopt early stopping during training to prevent overfitting, resulting in less than 100 training epochs. Early stopping is another technique to avoid overfitting that consists of monitoring the performance of the model on a validation dataset during training and halting the training process when the performance on the validation set starts to degrade. The idea is to stop training the model before it begins to overfit the training data. When using early stopping, *patience* is the hyperparameter that indicates the number of epochs to wait for an improvement in the validation performance before stopping the training. As is standard practice, *batch size*, *epochs*, and *patience* were predetermined and not included in the hyperparameter search.

The model is then trained three times: the first time without sentiment data, the second time by adding the information on the sentiment of all news, and the third time by using only sentiment data of ESG news. We split data in the following way (see Figure 8):

- Train: 1st January 2023 - 31st December 2023;
- Test: 1st January 2024 - 29th February 2024.

3 Results

In this section, we present the numerical results of our algorithm's performance. The three evaluation metrics utilized to assess the accuracy of our algorithm are mean Absolute Percentage Error (MAPE), Symmetric

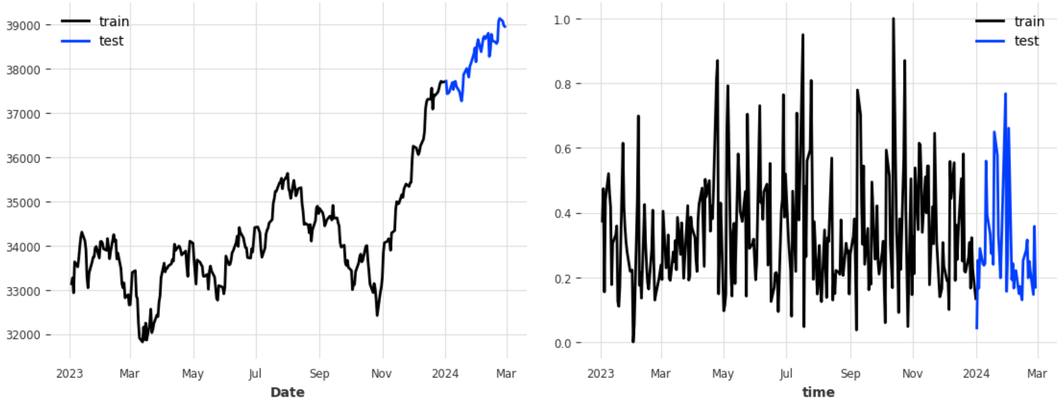


Figure 8: DJIA (left) and News Sentiment (right) Train Test split

Mean Absolute Percentage Error (sMAPE), and direction accuracy.

First of all, the MAPE and the sMAPE are defined, respectively, as

$$\text{MAPE} = \left(\frac{100}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| \right),$$

and

$$\text{sMAPE} = \frac{100}{n} \sum_{i=1}^{n} \frac{|P_i - A_i|}{(|A_i| + |P_i|)/2},$$

where:

- A_i is the actual value at time i.
- P_i is the predicted value at time i.
- n is the number of total observations.

Both MAPE and sMAPE provide a percentage measure of the prediction error. The difference is that sMAPE is symmetric because it treats overestimates and underestimates equally, unlike MAPE, which can disproportionately penalize overestimates.

The direction accuracy compares the predicted value, the output of the model, and the last closing price. Therefore, the label of the prediction is "up" if the predicted closing price is greater than the previous closing price, vice-versa the label is "down". The following formula computes the accuracy:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \tag{7}$$

In order to compare the accuracy of our methodology, we introduce four baseline models:

- The Naive Drift model fits a line between the first and last point of the training series and extends it in the future, that is, if our aim is to forecast y_T, knowing $y_1, y_2, \cdots, y_{T-1}$, the prediction is obtained by
$$\hat{y}_{T-1+h} = y_{T-1} + h \left(\frac{y_{T-1} - y_1}{T-2} \right),$$
setting $h = 1$.

Model	MAPE	sMAPE
Naive Drift	**2.42**	**2.42**
Naive Mean	59.24	84.20
Naive Moving Average	3.18	3.23
Naive Last	2.44	2.45

Table 2: Results comparison of baseline models

Model	MAPE	sMAPE	direction accuracy
Naive Drift	2.42	2.42	56.00%
N-HiTS with all news sentiment	**2.21**	**2.19**	**70.73%**
N-HiTS with ESG news sentiment	2.63	2.64	61.00%
N-HiTS with no news sentiment	2.45	2.46	58.50%

Table 3: Results comparison

- The Naive Mean prediction is given by the mean values of the historical data

$$\hat{y}_T = \frac{1}{T-1}\sum_{i=1}^{T-1} y_i.$$

- The Naive Moving Average forecasts using the last six prices, that is

$$\hat{y}_T = \frac{1}{6}\sum_{i=T-6}^{T-1} y_i.$$

- The Naive Last prediction adopts the last value of the training set, i.e., $\hat{y}_T = y_{T-1}$.

Table 2 shows a comparison among all the baseline models, the Naive Drift showing the best performances. Therefore, from now on, we consider this model as benchmark for our algorithm.

We would like to emphasize the critical importance of directional accuracy in trading. The ability to accurately predict the direction of stock market price movements is essential for both traders and investors, as it directly impacts future decisions to buy or sell assets, potentially leading to significant profits. Therefore, in analyzing the performance of our models, we will also focus on directional accuracy. Notice that, when considering the two best naive models—Naive Drift and Naive Last—the first model, as illustrated in Figure 9a, consistently predicts an upward movement across the entire test set. On the other hand, the Naive Last model is not well-suited for forecasting price direction, as it merely predicts tomorrow's price to be the same as today, offering no insight into future trends.

Table 3 shows the results on the three metrics (MAPE, sMAPE, and direction accuracy) for the baseline and the three different versions of the N-HiTS model, with the sentiment information of all news, with the sentiment information of ESG news only, and with no sentiment. First of all, we notice that the N-HiTS model trained with all news performs best according to all three metrics. N-HiTS trained with no news or only ESG news has worse MAPE and sMAPE but better direction accuracy than the baseline model.

To compare the four methodologies deeply, Figures 9a, 9b, 9c and 9d show the confusion matrix of the four models in predicting ups and downs. As we can see, the naive model (Figure 9a) always predicts "up" and therefore never correctly predicts "down". All three N-HiTS models improve from the baseline, and even though the N-HiTS trained with all news has the best overall accuracy score (Figure 9b), the one trained

Model	**# of Days with Positive Profit**	**Total Profit (Fixed $100/day)**	**Total Profit (Reinvesting Capital)**
Perfect Predictor	42	103.27$	175.92$
Naive Drift	24	21.05$	21.19$
N-HiTS all news sentiment	30	49.46$	59.28$
N-HiTS ESG news sentiment	26	30.46$	33.17$
N-HiTS no news sentiment	24	47.74$	58.33$

Table 4: Trading exercise

with only ESG news is the one that best predicts the "down" class (Figure 9c). For completeness, Figures 10a, 10b, 10c and 10d show the actual and predicted time series for all four models.

As a final comparison among models, we present a trading exercise in Table 4. In this exercise, we assume a trader buys (or short sells) $100 of the DJIA each day based on the model's prediction of price movement. Specifically, if the model predicts an increase (decrease) in price, the trader buys (short sells) accordingly. Table 4 reports the number of profitable trading days and the cumulative profits (and losses) in the second and third column, respectively, considering the baseline model as well as all the N-HiTS models. Additionally, we present results for a perfect predictor (i.e., it knows future prices), providing an upper bound for this trading exercise, as no model can perform better than this benchmark.

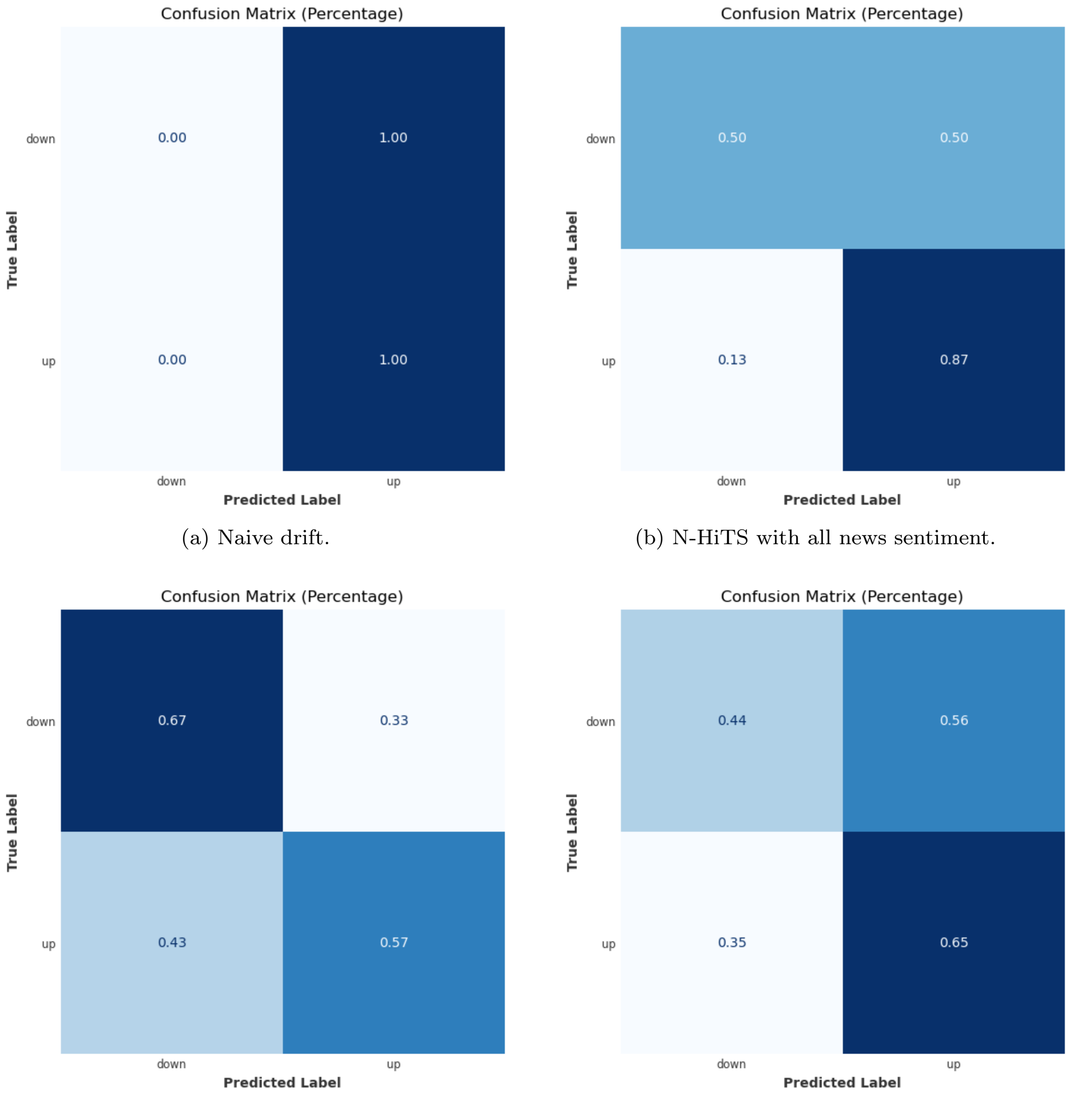
As expected, the number of profitable days aligns with the direction accuracy results, since profitable days correspond to the model correctly forecasting the direction of the market. In terms of cumulative profits, both the N-HiTS model with all news sentiment and the N-HiTS model without sentiment perform similarly, outperforming the model with only ESG news sentiment. This indicates that relying solely on ESG-related news for trading is limiting, as it primarily improves the prediction of downward movements. It is important to note that this result is influenced by the fact that the test set period occurs during a bull market, as shown in Figure 10. However, all N-HiTS models outperform the baseline model in terms of profits. A similar behaviour is obtained if we assume that the trader starts investing 100$, and each day reinvests the whole capital, also considering profits and losses, e.g., if on the first day there is a loss of 5$, the amount invested at the end of the day is no more 100$, but 100$-5$=95$. Results are reported in the last column of Table 4.

To conclude, in order to strengthen our results, we compare the N-HiTS model with news sentiment against another machine learning model, a Temporal Convolutional Network, trained on the same dataset. Our test results show a MAPE (sMAPE) of 8.00 (8.38), which is worse than all the N-HiTS models. This confirms the superiority of the N-HiTS architecture in price forecasting.

4 Conclusions

We employ deep learning models to compute the sentiment of news articles, particularly those related to ESG, and utilize the sentiment and historical price to predict the next closing price.

Current research demonstrates that news and social media sentiment have predictive power over closing prices. We run RoBERTa to compute the sentiment and N-HiTS to predict the closing price of the DJIA, training the model on a dataset from January 2023 to February 2024. Results indicate that deep learning models excel in prediction tasks and that including sentiment in time series forecasting enhances overall performance. The model trained with all news outperforms other models in all metrics. On the other hand, the model trained with ESG-only news best predicts the "down" stock movement. The latter finding suggests



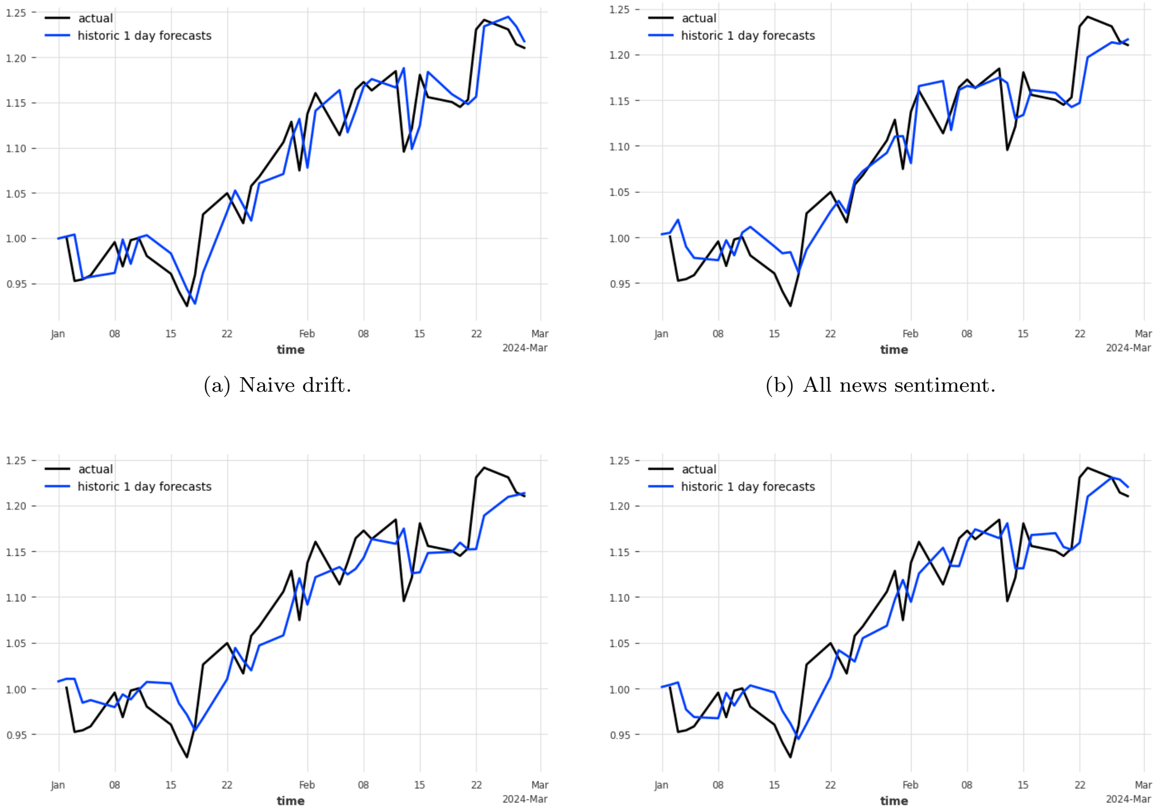
(a) Naive drift.

(b) N-HiTS with all news sentiment.

(c) N-HiTS with ESG news sentiment.

(d) N-HiTS with no sentiment.

Figure 9: Confusion matrices.



(a) Naive drift.

(b) All news sentiment.

(c) ESG news sentiment.

(d) No sentiment.

Figure 10: Actual and predicted time series.

that ESG weighs more on negative performances than on positive ones.

References

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework*. arXiv: `1907.10902 [cs.LG]`.

Ariyo, Adebiyi A., Adewumi O. Adewumi, and Charles K. Ayo (2014). “Stock Price Prediction Using the ARIMA Model”. In: *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112.

Badge, Jyoti et al. (2012). “Forecasting of Indian stock market by effective macro-economic factors and stochastic model”. In: *Journal of Statistical and Econometric Methods* 1.2, pp. 39–51.

Banerjee, Neelabha, Anubhav Sarkar, Swagata Chakraborty, Sohom Ghosh, and Sudip Kumar Naskar (2024). “Fine-tuning Language Models for Predicting the Impact of Events Associated to Financial News Articles”. In: *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing @ LREC-COLING 2024*. Ed. by Chung-Chi Chen, Xiaomo Liu, Udo Hahn, Armineh Nourbakhsh, Zhiqiang Ma, Charese Smiley, Veronique Hoste, Sanjiv Ranjan Das, Manling Li, Mohammad Ghassemi, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. Torino, Italia: ELRA and ICCL, pp. 244–247.

Beaudry, Paul and Franck Portier (2006). “Stock Prices, News, and Economic Fluctuations”. In: *American Economic Review* 96.4, pp. 1293–1307.

Bollen, Johan, Huina Mao, and Xiaojun Zeng (2011). “Twitter mood predicts the stock market”. In: *Journal of Computational Science* 2, pp. 1–8.

Boudoukh, Jacob, Ronen Feldman, Shimon Kogan, and Matthew Richardson (2013). “Which News Moves Stock Prices? A Textual Analysis”. In: NBER Working Paper Series 18725.

Bozanta, Aysun, Sabrina Angco, Mucahit Cevik, and Ayse Basar (2021). “Sentiment Analysis of StockTwits Using Transformer Models”. In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1253–1258.

Capelle-Blancard, Gunther and Aurélien Petit (2019). “Every Little Helps? ESG News and Stock Market Reaction”. In: *Journal of Business Ethics* 157.2, pp. 543–565.

Challu, Cristian, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski (2022). *N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting*. arXiv: `2201.12886 [cs.LG]`.

Cutler, David M., James M. Poterba, and Lawrence H. Summers (1998). “What Moves Stock Prices? (Spring 1989)”. In: *The Best of The Journal of Portfolio Management*. Ed. by Peter L. Bernstein and Frank J. Fabozzi. Princeton: Princeton University Press, pp. 56–64.

Damrongsakmethee, Thitimanan and Victor-Emil Neagoe (2020). “Stock Market Prediction Using a Deep Learning Approach”. In: *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: `1810.04805 [cs.CL]`.

Fama, Eugene F. (1965). “The Behavior of Stock-Market Prices”. In: *The Journal of Business* 38.1, pp. 34–105.

Gandhmal, Dattatray P. and K. Kumar (2019). “Systematic analysis and review of stock market prediction techniques”. In: *Computer Science Review* 34, p. 100190.

Gunnar Friede, Timo Busch and Alexander Bassen (2015). “ESG and financial performance: aggregated evidence from more than 2000 empirical studies”. In: *Journal of Sustainable Finance & Investment* 5.4, pp. 210–233.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). *Distilling the Knowledge in a Neural Network.* arXiv: 1503.02531 [stat.ML].

Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang (2005). “Forecasting stock market movement direction with support vector machine”. In: *Computers & Operations Research* 32.10. Applications of Neural Networks, pp. 2513–2522.

Jeffrey, Nathanael, Alexander Agung Santoso Gunawan, and Aditya Kurniawan (2023). “Check for updates Development of Multivariate Stock Prediction System Using N-Hits and N-Beats”. In: *Data Analytics in System Engineering: Proceedings of 7th Computational Methods in Systems and Software* 4, p. 50.

Khan, Mozaffar (2019). “Corporate Governance, ESG, and Stock Returns around the World”. In: *Financial Analysts Journal* 75.4, pp. 103–123.

Liu, Yichen, Chengcheng Zhong, Qiaoyu Ma, Yanan Jiang, and Chunlei Zhang (2023). “The S&P 500 Index Prediction Based on N-BEATS”. In: *Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023).* Atlantis Press, pp. 923–929.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* arXiv: 1907.11692 [cs.CL].

Lu, Wenjie, Jiazheng Li, Jingyang Wang, and Lele Qin (2021). “A CNN-BiLSTM-AM method for stock price prediction”. In: *Neural Computing and Applications* 33.10, pp. 4741–4753.

Malo, P., A. Sinha, P. Korhonen, J. Wallenius, and P. Takala (2014). “Good debt or bad debt: Detecting semantic orientations in economic texts”. In: *Journal of the Association for Information Science and Technology* 65.

Mishev, Kostadin, Ana Gjorgjevikj, Irena Vodenska, Lubomir T. Chitkushev, and Dimitar Trajanov (2020). “Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers”. In: *IEEE Access* 8, pp. 131662–131682.

Mittal, Anshul and Arpit Goel (2012). *Stock Prediction Using Twitter Sentiment Analysis.* URL: https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf.

Mohan, Saloni, Sahitya Mullapudi, Sudheer Sammeta, Parag Vijayvergia, and David C. Anastasiu (2019). *Stock Price Prediction Using News Sentiment Analysis.* IEEE Xplore. URL: https://ieeexplore.ieee.org/abstract/document/8848203/.

Nguyen, Thien Hai, Kiyoaki Shirai, and Julien Velcin (2015). “Sentiment analysis on social media for stock movement prediction”. In: *Expert Systems with Applications* 42, pp. 9603–9611.

Nyakurukwa, Kingstone and Yudhvir Seetharam (2023). “Investor reaction to ESG news sentiment: evidence from South Africa”. In: *EconomiA* 24, pp. 68–85.

Oreshkin, Boris N., Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio (2020). *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.* arXiv: 1905.10437 [cs.LG].

Patel, Jigar, Sahil Shah, Priyank Thakkar, and K Kotecha (2015). “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques”. In: *Expert Systems with Applications* 42.1, pp. 259–268.

Pearce, Douglas K and V. Vance Roley (1984). “Stock Prices and Economic News”. In: Working Paper Series 1296.

Rosamilia, Nico (2024). “Beyond the $\mathrm{E}C\mathrm{h}O_2$ Chamber:Efficient Sustainable Portfolios”. School of Management, Politecnico di Milano Working Paper.

Schmidt, Alexander (2019). “Sustainable News – A Sentiment Analysis of the Effect of ESG Information on Stock Prices”. In: *SSRN Electronic Journal: 3809657*.

Sezer, Omer Berat and Ahmet Murat Ozbayoglu (2018). “Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach”. In: *Applied Soft Computing* 70, pp. 525–538.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15, pp. 1929–1958.

Ticknor, Jonathan L. (2013). “A Bayesian regularized artificial neural network for stock market forecasting”. In: *Expert Systems with Applications* 40.14, pp. 5501–5506.

Timmermann, Allan and Clive W.J. Granger (2004). “Efficient market hypothesis and forecasting”. In: *International Journal of Forecasting* 20.1, pp. 15–27.

Vargas, Manuel R., Beatriz S. L. P. de Lima, and Alexandre G. Evsukoff (2017). “Deep learning for stock market prediction from financial news articles”. In: *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*.

Zhao, Lingyun, Lin Li, and Xinhao Zheng (2020). *A BERT based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts*. arXiv: 2001.05326 [cs.CL].