FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

I / We*, the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / our* work, except where acknowledged and referenced.

I / We* understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N. B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Matteo Sammut
Student Name

Signature

Student Name

Signature

Student Name

Signature

Student Name

Signature

ARI 5123
Course Code

Assignment
Title of work submitted

08/06/2025
Date

Integrating Daily Embeddings and Market Indicators for Deep Learning Algorithmic Trading

Matteo Sammut
matteo.sammut.20@um.edu.mt
University of Malta
Msida, Malta

Abstract

In this project we developed and tested the use of a deep learning method for end-of-day trading on the Standard & Poor's 500 (S&P 500) index. For this we developed a model capable of forecasting daily stock price movements by leveraging a combination of quantitative and qualitative data sources, in the form of technical indicators and financial article titles.

Keywords: Deep Learning,RCNN, CNN, LSTM, Prediction Model, Algorithmic Trading, Lexical Analysis

ACM Reference Format:
Matteo Sammut. 2025. Integrating Daily Embeddings and Market Indicators for Deep Learning Algorithmic Trading. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Algorithmic Trading is a very self explanatory term with deep reaching subdivisions. In essence it is taking stages of market trading and automatically determining stages of the trade by supplying ever expanding range of algorithms and models with various data sources[18]. In many markets, algorithms handle over 50% of all trades, with maturer markets such as the US stock exchange reaching up to 80%[23]. The main issue with most algorithmic trading models is that they suffer egregiously during severe market downturn like uncertain rare-events (black swans). Trend-following algorithms are the most susceptible as they rely solely on historical price data, which is problematic in extreme circumstances.

In recent years, it has become increasingly evident that machine learning techniques can identify complex, often non-linear structures within financial market data. Among the most promising deep learning architectures for this task is the Long Short-Term Memory (LSTM) network a type of recurrent neural networks. LSTMs initially gained traction over traditional approaches such as random forests and logistic regression, notably through their early application in [8]. Since then, LSTMs have continued to dominate the field of time series modelling due to their ability to capture long-range dependencies and temporal patterns in sequential data. This capability to learn complex models particularly well-suited for financial markets, where understanding the relationship between historical price movements and future trends is critical for predictive modelling. LSTMs' capacity to model time-dependent structures has led to significant advancements in forecasting accuracy, making them a popular choice for applications such as stock price prediction, algorithmic trading, and risk management.

While LSTMs excel at modelling sequences, Convolutional Neural Networks (CNNs) have been widely used to extract local patterns and spatial features from text data, including embeddings of article titles or sentences. As demonstrated in [5], CNNs are effective at capturing short-term dependencies and semantic patterns text. By extension, recurrent convolutional architectures (RCNNs) enhance this capability by combining the spatial pattern recognition of CNNs with the temporal modelling of RNNs, creating a powerful hybrid for text-based prediction tasks. For instance, [24] showed how an LSTM layer can be stacked on top of a CNN to model both local semantic features from news embeddings and their sequential relationships over time. This combination allows for a more comprehensive understanding of the impact of news events on financial markets.

Hence we propose an approach which takes in both technical indicators calculated from the previous three days, but also financial news titles to influence the prediction, ultimately attempting to predict the price movements. The main contribution of this work is the combination of a days worth of news articles into one vector representing the embeddings of all article titles henceforth referred to as a "day vector". After testing, we were able to show a mitigation of the effect of drawdowns, therefore having a positive performance in risk mitigation.

1.1 Aims & Objectives

Our aim is defined to develop and evaluate a hybrid deep learning framework that leverages market sentiment present

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

in financial news and technical indicators, to predict daily stock market movements of the S&P 500 and therefore maximise our returns while minimising the effects of drawdowns.

- To develop and implement a forecasting model using combining textual data and technical indicators, enabling the forecasting of future price movements
- Train, validate and test the model using historical price data from the S&P 500 along with a news articles dataset.
- Compare the performance of our model with other baseline models.
- Evaluate the models performance and showcase its real-world applicability through a simulated Buy-And-Hold strategy based on the output of our model.
- Demonstrate the models ability to minimiee drawdowns through testing.

2 Background and Literature Review

This section reviews the current body of research on the application of machine learning techniques in algorithmic trading, highlighting how artificial intelligence has increasingly shaped predictive models in financial markets. Particular attention was given to studies that involved the S&P 500 index, especially those that incorporate financial news as a predictive signal. The evolving integration of unstructured textual data with traditional market indicators is explored to contextualise the development of our proposed model. Finally, our approach is examined in light of existing methodologies, drawing comparisons with related models to evaluate its expected contributions and performance.

2.1 Econometric Foundations of Financial Market Forecasting

In financial markets, forecasting involves predicting future price movements based on a wide range of both quantitative and qualitative inputs. For individuals, especially before the modern era, this included examining historical price trends, trading volumes, macroeconomic indicators, and market sentiment derived from news and other public sources. When attempting to develop a model to predict market movements, we should also examine the interplay of these factors, capturing and reflecting the complex and often non-linear dynamics of financial systems. Several studies have emphasized the value of incorporating external economic indicators into predictive models, demonstrating improved performance when additional variables are included.

Despite this complexity, traditional financial theories like the Efficient Market Hypothesis (EMH) argue that markets are informationally efficient, meaning that all publicly available information is already reflected in asset prices. Consequently, EMH posits that price movements are essentially random and unpredictable, casting doubt on the feasibility of systematic forecasting [22]. Historically, econometric models like Autoregressive Integrated Moving Average (ARIMA) were the predominant tools for financial prediction. These models provided useful short-term forecasts by capturing time-series dependencies and volatility patterns with their mathematical simplicity. However, their reliance on linear assumptions and limited capacity to model complex relationships has led to growing interest in data-driven approaches. As financial markets have become increasingly influenced by unstructured data and dynamic global events, traditional models have struggled to adapt, paving the way for the adoption of machine learning and deep learning techniques better suited at capturing non-linear patterns and high-dimensional interactions [14].

2.2 AI & ML in Algorithmic trading

Machine Learning (ML) encompasses a suite of algorithms that enables systems to learn patterns from data and make decisions or predictions without being explicitly programmed. As one may imagine, in the realm of algorithmic trading, ML techniques are incredible assets and are employed to analyse vast amounts of financial data, identify trends, and execute trades with minimal human intervention. There exist two primary categories of ML,supervised and unsupervised learning. Both have found some degree of success in the space.

Supervised learning may be used when features can be associated with a known outcome. This may be beneficial when we are aware of a possible correlation between features and a target letting us relate them to an outcome. The affects of the general models like Random Forest and Support Vector Machines have been frequently applied to predict the stock index movement. In [15] the authors concluded that SVMs outperformed their other supervised models, with a hit rate of 68.44%. However, the choice of dataset in this study raises concerns that detract from the robustness and generalisability of the findings. Their dataset only covers a total time span of five years, which was then further split into a 75/25 train test set.

Unsupervised are used when there exists no target variable, usually they are employed when a hidden structure or pattern is presumed in the underlying data. A notable application is in pairs trading strategies, where unsupervised clustering methods such as k-means, DBSCAN, and agglomerative clustering are used to group stocks based on similarities in firm characteristics and return profiles. Research published in [10] and demonstrated that these clustering-based strategies significantly outperformed both the market and traditional benchmarks over a 40-year period.

In [9] the authors evaluated the effectiveness of various ML models in generating trading signals across multiple stock indexes. To assess the performance of these models, the study utilised independent subsets for evaluation and conducted sensitivity analyses to determine the robustness

of the estimations. The strategies derived from these models were compared against each other and against a traditional buy-and-hold benchmark, focusing on achieved levels of risk and return. The findings revealed that algorithmic strategies based on ML models outperformed passive strategies in terms of risk-adjusted returns. Buy-and-hold benchmarks provides a good baseline for performance comparison.

Recent research has demonstrated that some of the most effective predictive models in finance integrate supervised learning with unsupervised under-layers. In [19], it was shown that combining supervised learning models such as Random Forest and KNN with an underlying unsupervised GMM clustering technique and demonstrated superior predictive performance in buy-hold-sell decision strategies. Inspired by this hybrid framework, our model adopts a similar philosophy by incorporating unsupervised feature learning within a supervised prediction pipeline through the use of more powerful deep learning architectures. Specifically, the use of our RCNN architecture enables the model to extract rich semantic patterns from news data, enhancing its ability to learn from the unstructured representations produced by the Word2Vec NLP model.

2.3 Natural Language Processing & Word2Vec

Natural Language Processing (NLP) is essential for deriving actionable insights from unstructured textual data, including financial news, social media posts, and corporate reports. Through sentiment analysis and contextual understanding, NLP helps models assess market mood, which can exert a significant influence on financial decision making and price movements. The impact of this textual information becomes especially pronounced during periods of crisis, as highlighted in [2], where the authors found that the spread of misinformation during the COVID-19 pandemic deterred informed traders from acting on their private knowledge. This finding illustrates how misinformation can undermine market efficiency and increase drawdowns, emphasizing the role of news articles in times of heightened uncertainty.

Lexical analysis serves as the foundational step in NLP, involving the systematic decomposition of raw text into its most basic linguistic units, known as lexemes, they may consist of base forms or individual characters. This process results in the removal of irrelevant or redundant elements; such as punctuation, special characters, or stopwords, thereby eliminating unnecessary noise from the dataset. The resulting output is a tokenised representation of the original text, where each sentence is segmented into individual tokens. This refined textual format forms the basis for more advanced embedding techniques.

The word2vec representation, a popular embedding technique, transforms words into dense, low-dimensional vectors that capture semantic relationships based on contextual usage within a corpus. In our approach, we applied a pretrained Word2Vec model across the entire dataset to convert each word in a news title into an embedding vector. These word embeddings were then averaged to form a single composite vector representing the content of each article. Finally, all article vectors from a given day were combined to produce a unified day vector representation used as input to the model.

2.4 Our Model

Our proposed model leverages a hybrid deep learning architecture combining two LSTM networks,one processing technical indicators of the Standard & Poor's 500 (S&P 500) and another processing day vectors generated through a Word2Vec approach fed into a Recurrent Convolutional Neural Network (RCNN).

The decision to adopt a dual-LSTM architecture, stems beyond general interest and are related to the trajectory of the technology in academic works. Traditional machine learning approaches, such as logistic regression and random forests, though interpretable, often struggle to capture the intricate, non-linear relationships present in financial data. Earlier deep learning models, such as those explored by [5], demonstrated that using CNNs to process financial news can extract relevant patterns for stock movement prediction. However, their model primarily focused on intra-day predictions using individual articles, which can introduce high noise levels and may not align well with the needs of day traders who seek signals at a daily granularity.

Moreover, much of the existing work treats technical indicators and news data as separate input streams without leveraging the temporal dependencies across days in a cohesive manner. For example, while CNNs are powerful for extracting local patterns in text, they may not fully capture the sequential flow of market sentiment over time, which LSTMs are specifically designed to model.

Our approach directly addresses these gaps by combining daily news into an aggregated "day vector", with a word to vector and RCNN pipeline and sequential technical indicators in a unified framework. This setup enables the model to capture both the sequential dynamics of market indicators and the temporal evolution of news articles, making it more suitable for day trading strategies. Furthermore, by aggregating news at the day level, we reduce the noise inherent in intra-day article-level signals and focus the model's learning capacity on the broader market sentiment trends that are more relevant for day trading. Thus, our model provides possible a less noisy approach to market prediction, that may be the case in prior work.

3 Methodology

3.1 Datasets

While models are not yet capable of reliably predicting rare or extreme events, the incorporation of news-based algorithms may catch these events earlier, reducing overall drawdowns.

However, it is important to acknowledge that even with news-based insights, the speed of reaction may not always be sufficient, in periods of extreme volatility huge shocks may occur in a matter of minuets [17].

Naturally, the dilemma is always between the speed of reporting and the quality/bias of that reporting. We can have a real-time data stream of news by connecting an algorithm to X (twitter), but [1] found that of all news based posts; 25% contain misinformation and bias. Traditional news outlets are usually incentivised to be factually certain about their news, with most adhering to a level of impartiality compromising on reporting speed. This would naturally have an affect on the responsiveness of any model. Most solutions either avoid alternative media or attempt to corroborate it with available financial news [16].

The FNSPID dataset stood out as a large-scale financial news dataset relating S&P 500 companies, first described in [6] that contains 15.7 million article titles, each paired with a publication date ranging from 1999 to 2023. However, from our analysis of the dataset only 2007 till 2020 consists of usable data. By scraping data from the official Nasdaq news site, the authors achieved comprehensive temporal coverage, ensuring that nearly every day in the financial time series is represented. Unfortunately, the dataset had minimal preprocessing. The date index is unsorted, doing so reveals that the period 1999 till 2007 consist entirely of Russian titles, it is unclear why this is the case. Additionally, most columns contained predominantly null values, making the dataset unusable for other purposes. However, since both the article titles and dates were mandatory fields, the dataset was sufficiently robust and well-suited for the scope of this project.

For the technical layer, we utilised historical data from the S&P 500 index, sourced from Yahoo Finance, aligned with the date ranges defined in the FNSPID dataset. The S&P 500 was chosen not only for its availability and breadth as a benchmark of the U.S. equity market, but also due to its widespread use in prior academic studies focused on market prediction as mentioned previously. Its long-standing historical record makes it a suitable and realistic target for evaluating the effectiveness of trading models. The dataset was used to compute seven key technical indicators, which serve as structured, sequential inputs to the model's technical LSTM pathway.

3.2 Preprocessing Technical Layer

The technical layer in our model encapsulates seven key indicators derived from historical stock price and volume data, specifically using a three-day lookback window. This feature engineering process draws inspiration from the work of [25], which was one of the first where technical analysis is combined with news sentiment to predict daily stock movements. These indicators aim to capture a range of momentum, trend-based, and volume-based signals that traders and analysts traditionally use to inform trading decisions.

The first indicator, Stochastic %K (1), measures the current closing price relative to the range (high and low) over the lookback window.

$$\text{Stochastic } \%K = 100 \times \frac{\text{Close} - \text{Low}_{\min}}{\text{High}_{\max} - \text{Low}_{\min}} \tag{1}$$

This metric helps identify overbought and oversold conditions by comparing the closing price to recent price ranges. Relatively, the Williams %R (2) is a rescaled version of %K, inverting its scale to range from -100 to 0.

$$\text{Williams } \%R = -100 \times \frac{\text{High}_{\max} - \text{Close}}{\text{High}_{\max} - \text{Low}_{\min}} \tag{2}$$

Another momentum-based measure, Stochastic %D (3), is the moving average of the %K line, smoothing out short-term fluctuations and helping identify trend reversals.

$$\text{Stochastic } \%D = \text{SMA of Stochastic } \%K \tag{3}$$

The Accumulation/Distribution Oscillator (4) integrates both price and volume information to track the flow of money into or out of a security. It is calculated as the difference between the cumulative Accumulation/Distribution Line (AD Line) and its lagged value over the lookback window. The AD Line itself reflects how much volume is associated with upward or downward price movements.

$$\text{AD Line} = \sum \left(\frac{(\text{Close} - \text{Low}) - (\text{High} - \text{Close})}{\text{High} - \text{Low}} \times \text{Volume} \right) \tag{4}$$

Momentum (5) is a classic indicator that simply measures the change in closing price over the lookback period.

$$\text{Momentum} = \text{Close}_t - \text{Close}_{t-\text{lookback}} \tag{5}$$

This captures whether the price is accelerating upward or downward. The Disparity indicator (6) provides a percentage-based measure of the deviation between the current closing price and its simple moving average over the lookback window.

$$\text{Disparity} = \frac{\text{Close}}{\text{SMA}_{\text{lookback}}} \times 100 \tag{6}$$

Finally, the Rate of Change (7) normalizes price changes over the lookback window as a percentage, offering a relative momentum measure.

$$\text{ROC} = \frac{\text{Close}_t - \text{Close}_{t-\text{lookback}}}{\text{Close}_{t-\text{lookback}}} \times 100 \tag{7}$$

Together, these indicators form a compact yet rich representation of market dynamics over time. They capture price trends, momentum shifts, volatility, and volume-pressure signals that are considerably influential for day trading models.

While [25] were still successful in achieving a noticeable net profit with the use of SVMs, our approach integrates them into a sequential model via an LSTM which should give us an edge.

As made reference previously, our financial news we also utilized the FNSPID Financial News Dataset as the primary source of textual financial data to generate sentence and then day vectors. This dataset comprises a large collection of financial news articles, which were a crucial source of rich primary data for our level.

To prepare the data for embedding generation, we refined the large dataset to our dataset and dropped any unnecessary columns, we saved multiple sheets externally to not have to process the CSV on every run, the function for this preprocessing is included but is not referenced.

3.3 Model Implementation

This subsection outlines the setup and process of our numerical experiments designed to evaluate the model's performance. We used a time-series dataset spanning from January 1, 2007, to December 31, 2017, for training, with the entire year 2018 reserved for testing; a visual representation of these splits is viewable in figure . Historical S&P 500 technical data was , with news data coming from the previously mentioned Nasdaq Dataset. Our experiments found that using a shorter lookback window of 3 trading days yielded better results compared to longer sequences.

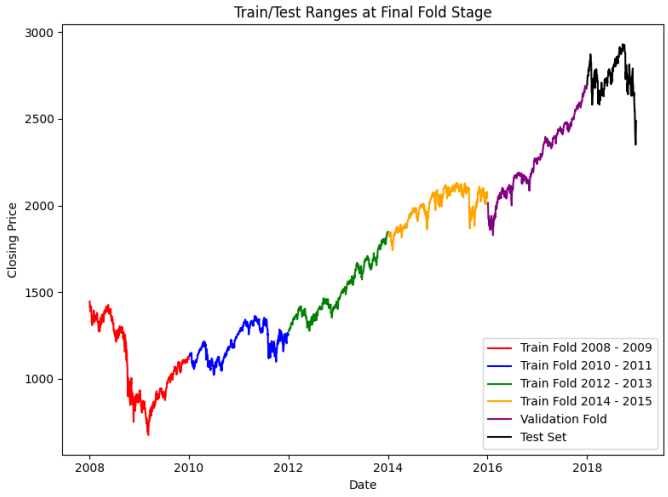

Figure 1. The different splits of the dataset(Source: Self)

We utilised the aforementioned Word2Vec model trained on the entire dataset of article titles to convert each token into a 100-dimensional embedding vector. To represent an article, the embeddings of its constituent tokens were averaged into a single sentence vector, and subsequently, all sentence vectors corresponding to a specific trading day were averaged to form a unified "day vector". These day vectors serve as the input to our RCNN-LSTM pipeline.

Consequently, each input sample for the technical features was shaped as (N,L,I), where N is the number of observations (2513 days in our case including lookback), L is the lookback period which was set statically to 3, and I is the input vector which was 7 and 100 for the technical and embedding layers respectively.

For the textual news component, the model first applies a 1D convolutional layer to sequences of word embeddings derived from a Word2Vec model. Combining these components has proven effective in natural language processing tasks for extracting local features and capturing short-term dependencies in textual data [3]. The use of a convolutional layer allows the model to detect informative phrases and sentiment patterns that may influence investor behaviour.

The CNNs output is then passed through a maximum-pooling layer, which reduces dimensionality while preserving the most salient features. Given the large number of parameters in this layer, the model is prone to overfitting. To address this, the dropout regularisation technique was implemented [21], setting it to a probability of 50% meant that during each training step, half of the neurons in the layer are randomly "dropped out" (set to zero), this reduces the number of retained neurons helping overfitting. Subsequently, the filtered feature maps are passed through a LSTM layer, which excels at modelling long-range dependencies in time series or language sequences [11]. This combination allows the model to understand important short-term textual signals and track how they evolve over time.

The final output from both streams—textual and technical—are concatenated into a single feature vector. This late fusion strategy ensures that both modalities retain their independently learnt representations before being combined. The concatenated vector is passed through a fully connected linear layer that outputs a softmax function, producing class probabilities. The architecture was designed for binary classification, reflecting a simplified market prediction task such as forecasting whether the stock price will rise or fall. The full architecture of the model is shown in figure 2.

The optimisation of model parameters was facilitated using the Adam optimiser. The Adam optimiser is a widely adopted gradient-based algorithm designed specifically for training deep neural networks. It maintains an exponentially decaying average of past gradients, allowing for a momentum-like approach for stochastic gradient descent[13]. A learning rate of 0.001 was chosen to balance convergence speed and stability.

To supervise the model's predictions, the Cross-Entropy Loss function was used, which is standard for classification tasks and particularly well-suited for evaluating the divergence between the predicted probability distribution and the true class labels. This setup enables the model to iteratively

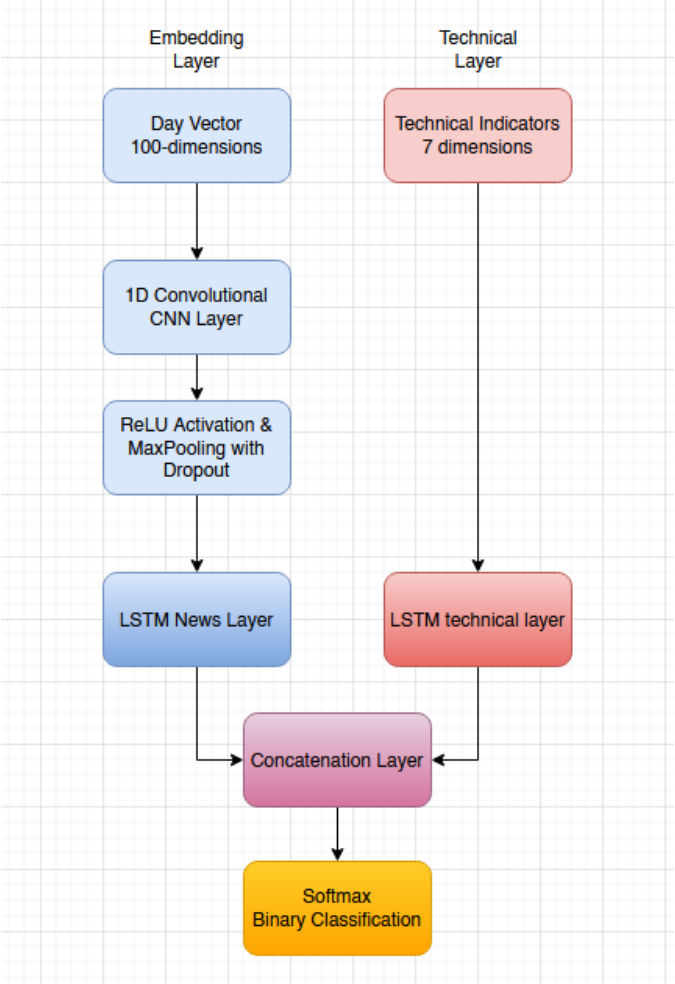

Figure 2. Full model architecture (Source: Self)

minimise classification error by adjusting weights based on the gradients computed during back-propagation.

The model was trained using K-fold time series split cross-validation to ensure robust evaluation across different temporal segments. For each fold, training was monitored using an early stopping mechanism, which halted the optimisation process if the validation loss failed to improve over a specified number of epochs. This technique helps avoid overspecialising to the training data and ensures better generalisation to unseen market conditions. Additionally, a prediction distribution penalty was introduced to discourage the model from becoming naively biased toward predicting only upward movements. If the proportion of up predictions exceeded predefined thresholds, a penalty term was added to the loss function. This adaptive regularisation acts as a soft constraint, ensuring the model learns meaningful patterns rather than exploiting the natural upward bias in the historical S&P 500 trend. After early stopping was triggered or training completed, a final evaluation of the models performance was conducted on the validation set.

3.4 Model Training and Validation

To train our model we made use of a training, validation and testing sets. For training and validation a K-Fold Time Series Split approach was used, which is similar to K-folds. In K-folds the model is trained on all folds (partitions) except one and tested on the remaining fold. This process is repeated per number of folds, with each fold used as the test set once. However, in time series problems, standard K-Fold may introduce data leakage by training on future data to predict past values, which is why it was preferred. Similar to K-fold we partition the data, but we never use any partition that occurs after the test partition chronologically. In our case we start with 2007 as the training set and 2008 and the validation set, followed by 2007 and 2008 as the training with 2009 as the validation etc. This methodology preserves the time dependency of the data, ensuring that future information does not leak into the past, mimicking a scenario where future events must be predicted based on past data.

For the test set, we did not expect to achieve exceptionally high accuracy. In stock price direction prediction, a commonly used benchmark is the naive strategy of random guessing, which typically results in an accuracy of around 50% [12]. Predictive models that consistently achieve accuracy between 55% and 65% are generally considered to perform very well. The results of the test set were tabulated in table 1.

For comparison, we used the tabulated metrics provided by [24] and show in figure, which tabulated the accuracy of a variety of baseline models.

Model	Training	Test
W-CNN	58,93%	57,22%
S-CNN	60,71%	60,96%
W-RCNN	59,76%	60,22%
S-RCNN	61,31%	61,49%
WI-RCNN	62,72%	61,29%
SI-RCNN	63,09%	62,03%
BW-SVM [36]	56,42%	56,38%
E-NN [29]	58,94%	58,83%
WB-NN [30]	60,25%	*
WB-CNN [30]	61,73%	60,57%
E-CNN [30]	61,45%	*
EB-NN [30]	62,84%	*
EB-CNN [30]	**65,08%**	**64,21%**

Figure 3. Comparison of baseline models on the S&P 500(Source: [24])

3.4.1 Evaluation Metrics. To thoroughly evaluate our model against the buy-and-hold baseline, we used the following financial performance metrics.

1. *Volatility:* Volatility is the standard deviation of returns and reflects the degree of variation in the portfolio's performance 8. Higher volatility typically implies higher risk. It is used in a number of subsequent formulas as is denoted through the use of σ.

$$\text{Volatility} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{8}$$

2. *Sharpe Ratio:* The Sharpe Ratio is a measure of risk-adjusted return. It evaluates how much excess return is achieved per unit of volatility σ_p. A higher Sharpe Ratio indicates better risk-adjusted performance 9.

$$\text{Sharpe Ratio} = \frac{E[R_p - R_f]}{\sigma_p} \tag{9}$$

3. *Sortino Ratio:* We also included the Sortino Ratio10, it improves upon the Sharpe Ratio by penalizing only downside volatility which avoids penalizing the model for high returns. This makes it more appropriate in financial settings where negative deviations are more relevant than total variance. It is denoted similarly to the Sharpe ratio, except for σ_d which is the standard deviation of the negative returns.

$$\text{Sortino Ratio} = \frac{E[R_p - R_f]}{\sigma_d} \tag{10}$$

4. *Cumulative Return:* This metric measures the total percentage gain or loss over the evaluation period 11.

$$\text{Cumulative Return} = \frac{V_f - V_i}{V_i} \times 100\% \tag{11}$$

where V_f is the final capital and V_i is the initial capital.

5. *Maximum Drawdown:* Maximum Drawdown (MDD) was of key interest in the case of our model, as it quantifies the largest observed loss from a peak to a trough in the capital curve 12, providing insight into downside risk.

$$\text{MDD} = \min\left(\frac{V_t - \max_{s \le t} V_s}{\max_{s \le t} V_s}\right) \tag{12}$$

where V_t is the value of the portfolio at time t, and $\max_{s \le t} V_s$ is the historical peak value up to that point.

4 Results and discussion

Table 1. Performance of the model on the 2014 Test Set.(Source: Self)

Metric	Value
Accuracy	0.5709
Precision	0.5869
Recall	0.8741
F1-Score	0.7022

From the results of the test set it might be surprising that we had an accuracy of 57% but a total recall is 87%, this accompanied by the F1-score indicated to us that the model tended to over predicts up. As the S&P 500 has naturally trended upwards for the past 50 years with a real return of 9% per year [4], a level of bias towards positive returns is introduced in the historic dataset in the form of class imbalance.

When comparing our results to the test result baseline models present in 3, we notice that our model had an accuracy somewhere in the middle to lower end of results. Despite this variety in the data aswell as time specific and bias in the results can all play an affect. Hence we required further analysis of our models performance through a practical example.

Following the test set, in order to have a practical example of the performance of our algorithm, we loaded our model and simulated its real-world use in a financial trading context through backtesting. In backtesting we use historical data to evaluate how our model would have performed, tracking capital over time based on predictions and actual market behaviour.

In our evaluation we use a rolling window approach where the move forwards day by day making prediction on the market, mimicking day-by-day decisions in a real scenario. We followed the predictions of the model exactly buying and holding when suggested by it. It should be noted we did not take into account any additional fees when implementing it such as transaction fees.

To establish a baseline for comparison, we implemented a buy-and-hold strategy which is a common and popular for the S&P 500 among long-term investors. By holding onto the investment over time and not reacting to short-term price fluctuations.

What should noted is that when we are predicting bad days we are simply sitting on the capital in cash, this results in no earnings if it ultimately rises. This missed opportunity is best shown in the confusion matrix in table 2. The alternative would have been to emulate a market timing strategy, such as those described in [7], to ensure our capital is always being utilised (invested). However, this would introduce additional factors that would influence the results, making it more challenging to isolate the direct impact of the model's predictions on performance.

Prediction	Actual Movement	Impact on Capital
TP	Up	Capital increases
FP	Down	No capital loss
TN	Down	No capital loss
FN	Up	Missed gain

Table 2. Confusion matrix of predictor outcomes and their effect on our trading capital. (Source: Self)

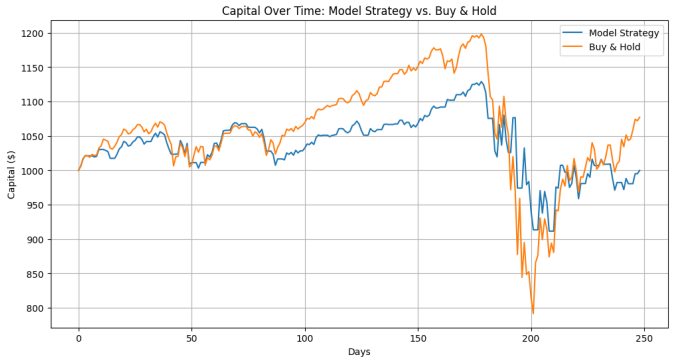

Figure 4. Performance of the model on 2019-05-31 till 2020-06-01.(Source: Self)

The model was evaluated on an out-of-sample test period spanning from May 2019 to June 2020, a time characterised by significant market volatility and uncertainty. Articles published during this period were particularly influential in shaping market sentiment. Capital trajectories were tracked for both the model-driven strategy and a passive buy-and-hold benchmark starting with $1,000 in capital. The trajectory of the capitals may be viewed in figure 4, with the tabulated results of the additional metrics in table 3.

Table 3. Performance Metrics: Model Strategy vs. Buy & Hold.(Source: Self)

Metric	Model Strategy	Buy & Hold
Cumulative Return	-0.02%	7.72%
Sharpe Ratio	0.00752	0.02494
Sortino Ratio	0.00764	0.02722
Volatility	0.01506	0.02065
Maximum Drawdown	-19.26%	-33.93%
Final Capital	$999.84	$1077.19

After tabulating the results we obtained final capital values of $999.84 and $1077.19, for our model and buy-and-hold strategy respectively. This translated into a cumulative return of **-0.02%** for the model and **7.72%** for the buy-and-hold strategy. Although the model underperformed in absolute return terms, its performance reveals several important strengths from a risk-adjusted perspective.

In particular, we wish to highlight that the model achieved a lower volatility of (1. 51%) compared to (2. 07%) achieved by buy-and-hold. Its maximum drawdown was significantly lower, just (0. 75%) versus (2. 49%) for the benchmark.

While the Sharpe and Sortino ratios were numerically low for both strategies, likely due to the short evaluation window and limited return volatility. Instead, we emphasise the reduced drawdown and overall smoother capital trajectory suggest that the model is better suited for strategies emphasising downside protection over aggressive growth.

While the model did not outperform buy-and-hold, it demonstrates the potential for risk-controlled allocation in uncertain conditions. This may be visualised in figure 4, although the model (blue line) missed some key spikes in gains, it correctly reduced the affects of drawdowns on the capital. These figures suggest that the model is particularly effective at limiting downside risk, an essential quality for strategies aiming to preserve capital during turbulent market conditions and black swans.

The likely explanation for this conservative risk profile lies in the model's ability to infer market uncertainty from news sentiment. Since the model was trained on embeddings of financial headlines, it may have learnt to interpret negative news signals and associate them as indicators of potential downside, often opting to hold rather than buy during such periods. This cautious behaviour, while limiting upside capture, offers clear benefits in drawdown protection and capital preservation. We also believe that during periods of instability, the news cycle and hence our day vectors are more refined in their overall negative sentiment.

4.1 Overfitting

We noticed in some experiments that the model is incorrectly fit if it simply mirrors the buy-and-hold strategy, as this would indicate that the model has not learnt any meaningful patterns from the data but instead has simply captured the underlying upward trend of the market, we may see a plot of this in figure 5. In this case, the model effectively becomes naive, as the buy-and-hold strategy itself requires no predictive skill, merely reflecting the historical tendency of the market to appreciate over time. This is mainly due to the class imbalance that exists for upward trends which is made evident in figure 1. If the model's predictions closely align with the buy-and-hold benchmark, we can infer that the model has likely overfit to the trend component of the data while failing to capture the day-to-day nuances or significant turning points driven by external factors, such as news events, sentiment shifts, or technical signals.

In practical terms, this means that while the model may achieve superficially high accuracy or recall, it is not offering any actionable insights beyond what a simple, passive investment strategy would provide. To counteract this we applied the previously mentioned penalty if the model converged on an naive approach, we found this approach to be successful at mitigating this affect. This only works as it is extremely improbable that a fold will only have positive trends.

5 Conclusion

This project explored the application of a hybrid deep learning architecture for predicting daily stock market movements

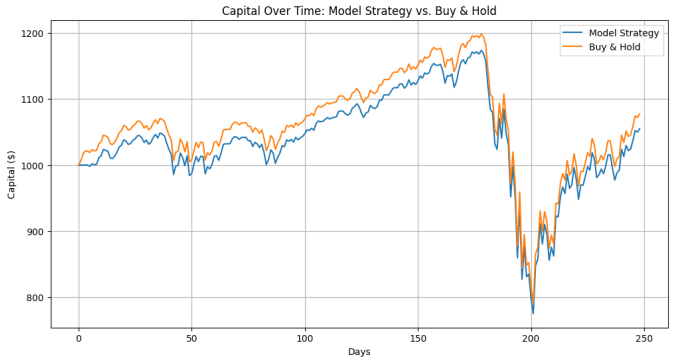

Figure 5. Example of the model on period 2019-05-31 till 2020-06-01 were the model has overfit to the general trend and has become naive.(Source: Self)

using both technical indicators and financial news. By embedding news headlines with Word2Vec and aggregating them into daily vectors, the model was able to capture market-relevant sentiment alongside traditional price-based signals. Although the model did not outperform the buy-and-hold benchmark in terms of cumulative return, it demonstrated strong risk mitigation characteristics, particularly through its substantially lower maximum drawdown. These findings highlight the model's potential for defensive trading strategies, where capital preservation is prioritised over aggressive gain-seeking.

The results also reinforce the value of incorporating unstructured text data into financial forecasting, especially in capturing market sentiment that may not yet be reflected in price, we propose several avenues for future work that may assist in this regard. Overall, the project demonstrates that deep learning models, when thoughtfully integrated with financial domain knowledge, can provide meaningful contributions to the development of intelligent algorithmic trading systems.

5.1 Future Work

We believe procurement of a better dataset would involve the development of a real-time data pipeline through automated web scraping. By continuously scraping financial news websites and aggregating headlines as they are published, this stream would preprocess text into new data points enabling the model to make use of evolving news narratives. Integrating this live data into the model would enhance for decision making on the next days trades.

Another promising extension of this work involves the integration of Named Entity Recognition to enhance the quality of textual inputs. NER is an NLP task that involves identifying and classifying key entities in text—such as people, organizations, locations, and financial instruments. In our context, NER can specifically leveraged to filter and highlight market relevant information from news articles by focusing on entities explicitly tied to financial events or companies listed on the S&P 500, allowing the model to weigh this entity based information differently. While prior work has mixed results from incorporating NER, with large scale prediction tasks not resulting in noticeable financial gains [20], we still believe the avenue would help the model understand the dataset further.

References

[1] Alexandre Bovet and Hernán A Makse. 2019. Influence of fake news in Twitter during the 2016 US presidential election. *Nature communications* 10, 1 (2019), 7.

[2] Cosmin Octavian Cepoi, Victor Dragotă, Ruxandra Trifan, and Andreea Iordache. 2023. Probability of informed trading during the COVID-19 pandemic: the case of the Romanian stock market. *Financial Innovation* 9, 1 (2023), 34.

[3] Yahui Chen. 2015. *Convolutional neural network for sentence classification.* Master's thesis. University of Waterloo.

[4] Elroy Dimson, Paul Marsh, and Mike Staunton. 2009. Triumph of the optimists: 101 years of global investment returns. In *Triumph of the Optimists*. Princeton University Press.

[5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction.. In *Ijcai*, Vol. 15. 2327–2333.

[6] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. 2024. FNSPID: A Comprehensive Financial News Dataset in Time Series. arXiv:2402.06698 [q-fin.ST]

[7] Todd Feldman, Alan Jung, and Jim Klein. 2015. Buy and Hold Versus Timing Strategies: The Winner Is... *Journal of Portfolio Management* 42, 1 (2015), 110.

[8] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research* 270, 2 (2018), 654–669.

[9] Jan Grudniewicz and Robert Ślepaczuk. 2023. Application of machine learning in algorithmic investment strategies on global stock markets. *Research in International Business and Finance* 66 (2023), 102052.

[10] Chulwoo Han, Zhaodong He, and Alenson Jun Wei Toh. 2023. Pairs trading via unsupervised learning. *European Journal of Operational Research* 307, 2 (2023), 929–947.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Firuz Kamalov, Linda Smail, and Ikhlaas Gurrib. 2020. Forecasting with deep learning: S&P 500 index. In *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*. IEEE, 422–425.

[13] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Vaia I Kontopoulou, Athanasios D Panagopoulos, Ioannis Kakkos, and George K Matsopoulos. 2023. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet* 15, 8 (2023), 255.

[15] Manish Kumar and M Thenmozhi. 2006. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper.*

[16] THORSTEN LEHNERT. 2024. Fake News and Options Trading. In *CREDIT 2024. THE FRONTIERS OF NEW RISKS: AI, DIGITAL AND SUSTAINABILITY TRANSITIONS.*

[17] Wujun Lv, Tao Pang, Xiaobao Xia, and Jingzhou Yan. 2023. Dynamic portfolio choice with uncertain rare-events risk in stock and cryptocurrency markets. *Financial Innovation* 9, 1 (2023), 73.

[18] Giuseppe Nuti, Mahnoosh Mirghaemi, Philip Treleaven, and Chaiyakorn Yingsaeree. 2011. Algorithmic trading. *Computer* 44, 11 (2011), 61–69.

[19] Gabriel Rodrigues Palma, Phil Maguire, et al. 2024. Combining supervised and unsupervised learning methods to predict financial market movements. *arXiv preprint arXiv:2409.03762* (2024).

[20] Rebecca J Passonneau, Tifara Ramelson, and Boyi Xie. 2014. Named entity recognition from financial press releases. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*. Springer, 240–254.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[22] Allan Timmermann and Clive WJ Granger. 2004. Efficient market hypothesis and forecasting. *International Journal of forecasting* 20, 1 (2004), 15–27.

[23] Cristiana Tudor and Robert Sova. 2025. An automated adaptive trading system for enhanced performance of emerging market portfolios. *Financial Innovation* 11, 1 (2025), 1–39.

[24] Manuel R Vargas, Beatriz SLP 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)*. IEEE, 60–65.

[25] Yuzheng Zhai, Arthur Hsu, and Saman K Halgamuge. 2007. Combining news and technical indicators in daily stock price trends prediction. In *International symposium on neural networks*. Springer, 1087–1096.

Acronyms

CNNs Convolutional Neural Networks
LSTM Long Short-Term Memory
ML Machine Learning
NLP Natural Language Processing
S&P 500 Standard & Poor's 500