

Bitcoin Prediction: A Hybrid Approach with Recurrent Neural Networks and GPT-based Reasoning

Zhengri Cui¹, Daniel Comerford¹, An Nguyen^{1,2}, and Marija Bezbradica^{1,2}

¹ School of Computing, Dublin City University, Collins Ave Ext, Whitehall, Dublin, Ireland
zhengri.cui2@mail.dcu.ie, daniel.comerford2@mail.dcu.ie

² ADAPT Research Centre, Dublin City University, Dublin, Ireland
ngocannnguyen.pham@dcu.ie, marija.bezbradica@dcu.ie

Abstract. This paper explores a novel hybrid framework by combining numerical time series models with large language model (LLM)-based sentiment analysis. In the first stage, we apply manual feature engineering and selection, followed by sequential modeling to predict the future 30-day moving average (MA) for Bitcoin daily returns. After running the models and comparing them the conclusion was made that LSTM and GRU showcasing superior MAE and MSE when compared to the Naive, RNN and CNN-LSTM models. These predictions from the GRU and LSTM were converted to directional signals (up/down) to be fed in further down the pipeline. In the second stage of this study, the predictions were integrated with Reddit data for each day and fed into the GPT-4o reasoning layer which was strictly prompted to intake the data without hallucination and predict based on the inputs. This novel architecture resulted in a directional prediction F1 score of 76% outperforming the standalone predictions from the time series models. Results suggest that incorporating an LLM-based sentiment analysis with time series predictions could lead to an increase in accuracy in such volatile markets.

Keywords: Time Series · Deep Learning · Cryptocurrency · Sentiment Analysis · Hybrid Models

1 Introduction

Price volatility has been one of Bitcoin’s most challenging aspects when it comes to accurate forecasting. Bitcoin, unlike typical financial assets, has extremely strong fluctuations that are heavily influenced by both quantitative indicators and the rapidly shifting sentiment from retail investors [4]. Numerous machine learning (ML) and deep learning (DL) models have been proposed to forecast the price level or the log returns. However, many of these models ignore public sentiment [3, 29, 33].

With the recent advancements in large language models (LLMs), specifically GPT-4o, the models have demonstrated the ability to reason over unstructured textual data in real-time [20, 28]. Such advancements allow for the novel architecture proposed in the present study, which integrates rich, dynamic sentiment with numerical time series models to allow the GPT to increase the prediction accuracy without the need for manually engineered features or scoring systems.

Specifically, we propose a hybrid framework that combines time series models with Reddit sentiment inputs which are processed via a GPT-4o reasoning layer for the final prediction. The GPT analyses the Reddit data and is provided with RNN models’ accuracy metrics and predictions for a certain period. Stage 1 of this study models recurrent neural network (RNN) architectures to forecast the 30-day moving average of Bitcoin daily returns using historical price data and financial indicators. These predictions from the models are converted to directional signals (up/down). Stage 2 combines contextual text data from Reddit posts with the RNN models’ accuracy metrics and input these into the GPT-4o reasoning layer for an enhanced prediction.

This research is guided by two central questions:

- **RQ1:** How accurately can the future 30-day price direction of Bitcoin be predicted using deep learning models?
- **RQ2:** Does incorporating Reddit-based sentiment reasoning layer via GPT-4o improve the directional predictions of Bitcoin price movements?

Regarding the data used in this study in stage 1, we utilize daily OHLCV³ of Bitcoin (BTC), Ethereum (ETH), Ripple (XRP) and the S&P 500 index, along with Bitcoin network Hash Rate. In Stage 2, we combine the predictions from Stage 1 with textual data from Reddit posts, which are then provided to the GPT-4o reasoning layer

The framework is evaluated using standard regression metrics for stage 1, including MAE, MSE and SMAPE. For stage 2, classification metrics such as Accuracy, Precision, Recall and F1 score are used. The final results shows that the combined approach outperforms the traditional deep learning models, achieving a F1 score of 76% which was an 8% improvement in the directional predictions in the models without the GPT layer.

2 Related Work

Previous studies on Bitcoin price movement forecasting have used various features which can be categorized as follows: (1) Bitcoin price related variables such as OHLCV data [2,3,29]; (2) technical indicators including MACD and RSI [1]; (3) other cryptocurrencies like ETH and XRP [5,17]; (4) macroeconomic and market indicators such as S&P500 [5,10]; (5) on-chain metrics such as Hash Rate [5,7,10]; (6) public sentiment such as Google Trends, Tweets, Reddit posts, and news [5,17,35].

Regarding model application, Recurrent Neural Networks (RNNs) have been actively used in many previous studies. A literature review conducted by Boozary et al. concluded that the application of LSTM significantly improves Bitcoin price prediction accuracy [4]. By contrast, neither financial time series forecasting using traditional statistical methods like ARIMA [30], nor approaches employing powerful machine learning algorithms like Random Forest [5], nor even the use of the more recent Transformer architectures [31,35] have outperformed LSTM. Moreover, LSTM has been served as an underlying model in studies utilizing advanced methodologies, including ensemble methods with three LSTM individual models across different time granularities [29], policy-based reinforcement learning techniques [18] and multi-stage CNN-LSTM architectures [7].

Another type of Recurrent Neural Networks is the so-called GRU (Gated Recurrent Unit), which has showed outstanding results in the existing literature. For instance, in a study on Bitcoin price prediction [21], this architecture achieved lower RMSE and MAE compared to SVM and BiLSTM, while another study [33] on Dogecoin prediction also reported that GRU, despite its simpler architecture, has better prediction performance compared to the LSTM model. This indicates the important role of RNNs in financial time series modelling. However, despite their impressive performance, RNNs typically assume that the time series is stationary; therefore, the presence of trends or seasonality can significantly degrade their performance [14].

More recently, alternative methodologies have been proposed that predict the Cryptocurrency Volatility Index as the target variable in order to reduce risk. Nguyen et al. [16] used a novel Deep Neural Network combining LSTM and MLP with an attention mechanism, which has been shown to outperform traditional approaches such as standard LSTM, Temporal Convolution Networks, Random Forest, Support Vector Regression and other statistical methods. In addition, sentiment-based prediction methods using LLMs have been explored, primarily in the context of stock market forecasting [17,23] and have not yet been used in the cryptocurrency market. In particular, one study utilised ChatGPT to analyse news headlines and predict stock price movements without any specialized training. This method resulted in a cumulative return of over 650% between October 2021 and December 2023 [19]. Another study conducted a live experiment with ChatGPT-4, showing that its ratings respond to earnings surprises and news events, resulting in a positive correlation with future earnings announcements and stock returns [23]. These works fill the gap by showing that LLMs can forecast financial markets without specialized training and respond to real-time events.

3 Stage 1: Time Series Prediction via Sequential Models

Before discussing the first stage of our study, we first recall the overall architecture and workflow employed in this study (as displayed in Fig. 1), encompassing all key steps from data collection to final prediction. We note that although the current workflow focuses on Bitcoin and Reddit sentiment, it is designed to be

³ OHLCV stands for Open, High, Low, Close, and Volume in financial markets.

modular and extensible. Therefore, it can readily incorporate additional data sources such as financial news APIs, analyst reports, or sector-specific updates. This allows the model to be used in a broader range of services such as equities, macroeconomics, or consumer predictions. Our methodology is structured into two stages. In the first stage (discussed in this Section 3), we use sequential deep learning models (CNN-LSTM, RNN, LSTM, GRU) to predict the 30-day moving average of daily percent returns based on quantitative time-series data. The predicted numerical values from LSTM and GRU are then converted into directional signals (1 for upward, 0 for downward). In the second stage (outlined in Section 4), these directional signals are combined with Bitcoin-related Reddit submission texts as new features. Under prompt engineering and temperature control settings, these new features are processed using the GPT-4o Reasoning layer to predict the 30-day directional signal (1 for upward, 0 for downward). Details are as follows:

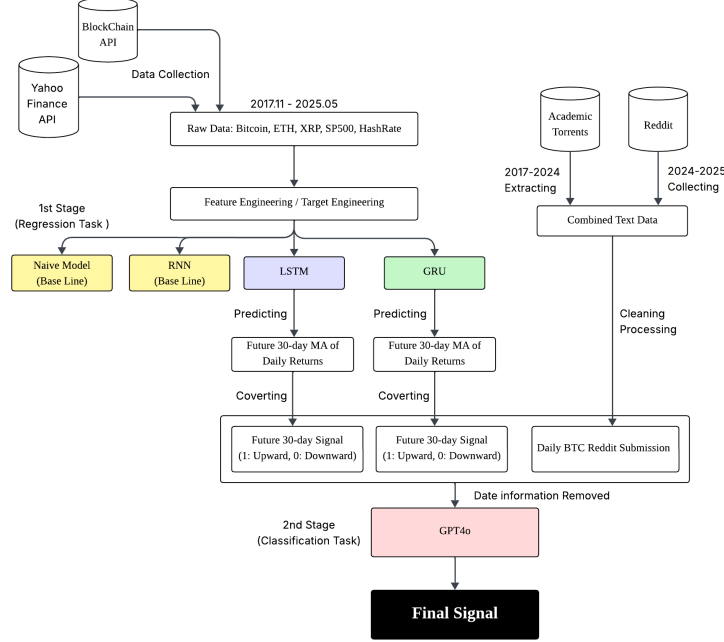


Fig. 1: Overall Architecture and Key Steps. While this study uses Bitcoin price data and Reddit sentiment, the modular architecture supports integration of additional data sources such as financial news APIs (e.g., FT, WSJ), social media platforms (e.g., Twitter), or domain-specific feeds, enabling broader applicability.

3.1 Data Collection and Preprocessing

Daily interval OHLCV (Open, High, Low, Close, Volume) data was collected for ETH, XRP and the S&P 500 index using the Yahoo Finance API and Bitcoin network Hash Rate via Blockchain API. Since Ethereum has a relatively short historical record on the Yahoo Finance, we made sure the common date range across all assets, from 2017-11-09 to 2025-06-15, when ETH data was consistently available. To address the missing data caused by its weekend market closing for S&P500 index, we applied forward-fill imputation, propagating Friday’s data to Saturday and Sunday.

On the other hand, Bitcoin, which serves as the features and target in our study, required a longer historical span to account for potential data loss during feature and target engineering. Therefore, BTC data was collected from 2017-01-01 to ensure sufficient data.

3.2 Feature and Target Engineering

Feature and target engineering play a crucial role in enabling effective prediction from raw financial time series. In our workflow, stationarisation techniques such as Daily Percentage Change [9], Log Returns [9] and Fractional Differentiation [14, 24] were employed to stabilize the statistical properties of the data. In

addition, moving average smoothing [26] was used to reduce short-term noise. These transformations were applied across only for the Close and Volume series of BTC, ETH, XRP and the S&P 500, as well as HashRate data.

A total of 43 features were engineered, including the 2 prediction target candidates. To clearly illustrate the differences among these transformations, we present only the visualization of BTC Close transformations in Fig. 2. The blue plot represents the non-stationary raw closing price of Bitcoin overtime. Kaabar et al. [14] argue that many deep learning models, including RNNs, inherently assume stationarity of time series data. The green plot in Fig. 2 represents the daily percentage returns of Bitcoin’s closing price. This transformation is similar to applying a full differencing, which primarily stabilizes the mean and variance over time, rather than directly using raw price levels, but exhibits high volatility and noise, which can lead to instability during model training. In contrast, the orange plot displays the result of applying fractional differencing to the closing price series. Exploratory experiments were conducted by changing the window size and fractional order to identify a transformation that ensure the timeseries is statistically stationary (ADF test p -value < 0.05), while preserving the highest correlation with the original series. Among the tested set, a window size of 130 and a fractional order of 0.48 provided a good balance between memory retention and stationarity. The p -value from the ADF test was 0.0416, indicating that the transformed series is stationary at the 5% significance level. This fractionally differenced series is used as one of the input features in modelling stage. From the initial set of 43 features, a multi-stage feature and target selection process was conducted. The overall workflow is illustrated in Fig. 3.

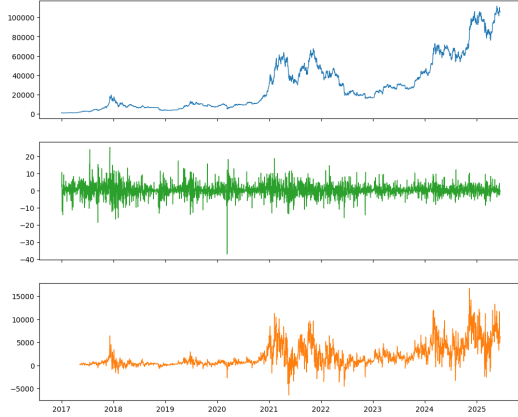


Fig. 2: Raw vs Percent Change vs Fractionally Differentiation

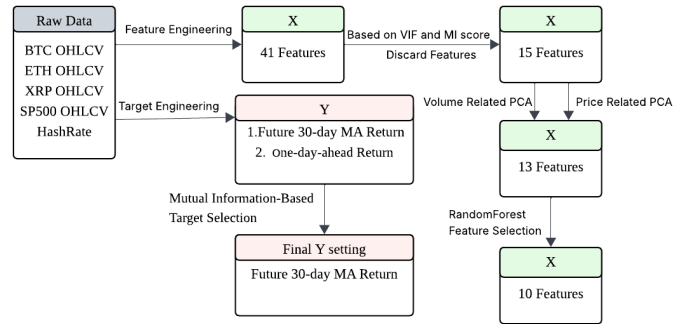


Fig. 3: Target and Feature Selection WorkFlow

Regarding target selection, this study aims to build a robust model for reliable forecasting in the volatile Bitcoin market, where stable model training is a necessary precondition, regardless of whether the target is short-term daily returns or long-term return trends. To achieve this objective, the following two candidates were considered as potential target variables: (1) the 1-day ahead Bitcoin close return and (2) the future 30-day moving average of returns.

The mutual information was calculated between input features and the continuous target variable using `mutual_info_regression` from Scikit-Learn. The method is nonparametric and based on entropy estimation using K-nearest neighbors distance [15,27]. The analysis revealed that the future 30-day MA of returns showed MI values exceeding 0.1 for 9 out of 15 BTC-related features, whereas the 1-day ahead BTC return exhibited significantly lower MI values, with the highest being just 0.0322. These findings suggest that short-term returns are heavily affected by noise and do not maintain meaningful information relationships even with its derived sources from BTC-related variables. Accordingly, to reduce noise and enhance both model stability and predictive performance, the future 30-day MA of returns was selected as the final target variable for this study.

Regarding input feature selection, the initial 41 input features were reduced to 15 using Mutual Information (MI) and Variance Inflation Factor (VIF). These selected features and their corresponding VIF are shown in Table 1. The criteria are as follows:

- Features with MI values below 0.05 were discarded.
- Features derived from the same asset and similar data types (e.g., price-based: Open, High, Low, Close, RollingClose; volume-based: Volume, RollingVolume) were grouped due to their similar variance. Within each group, only the feature with the highest MI was selected to reduce redundancy.

However, several features still exhibited VIF values exceeding 10, indicating that a high degree of multicollinearity remains in the dataset. Therefore, we reduce their multicollinearity using Principal Component Analysis (PCA) technique, resulting in 6 features being reduced to 4 based on price and volume groups. After applying PCA to each group, the VIF values for all features are summarized in Table 2.

Table 1: Initially Selected Features and Their VIF Values

Feature	VIF
RollingBtcClose	84.37
SP500_RollingClose	41.20
RollingHashRate	20.14
EthRollingClosePrice	19.35
SP500_Volume	17.10
BTC_Close_FracDiff_MA	16.24
RollingBtcVolume	15.13
ETH_Volume	10.39
XrpRollingClosePrice	8.42
RollingBtcVolumeChange	5.31
RollingEthChange	4.05
XRP_Volume	3.76
BtcRollingLogReturn	3.50
RollingXrpChange	2.20
SP500_RollingChange	1.43

Table 2: Features and Their VIF Values After PCA Reduction

Feature	VIF
RollingHashRate	12.22
BTC_Close_FracDiff_MA	10.87
RollingBtcVolumeChange	5.08
XrpRollingClosePrice	4.42
RollingEthChange	3.80
BtcRollingLogReturn	3.52
XRP-USD_Volume	3.49
PC1_price	3.41
PC1_volume	3.14
RollingXrpChange	2.12
PC2_price	1.56
SP500_RollingChange	1.44
PC2_volume	1.44

Table 3: Final Feature Set (X)

Feature (X)	Description
BTC_Close_FracDiff_MA	30-day MA of fractionally differenced BTC close price
RollingHashRate	30-day MA of BTC network hashrate
BtcRollingLogReturn	30-day MA of log returns of BTC close price
RollingEthChange	30-day MA of percentage change in ETH close price
XrpRollingClosePrice	30-day MA of XRP close price
RollingXrpChange	30-day MA of percentage change in XRP close price
SP500_RollingChange	30-day MA of percentage change in SP500 close price
PC1_price	PC1 from 30-day MA of BTC/ETH/SP500 close prices
PC2_price	PC2 from 30-day MA of BTC/ETH/SP500 close prices
PC1_volume	PC1 from RollingBTCVolume and ETH, SP500 volumes

In the final stage, Random Forest-based feature importance analysis was applied to reduce the 13 features to a final set of 10. Among the 13 features from Table 2, **RollingHashRate** and **BTC_Close_FracDiff_MA** still showed VIF values exceeding 10. As noted by Raschka et al., when two or more features are highly correlated, one feature may be ranked very highly, while the importance of the others may be distorted [25]. To address this issue, two separate Random Forest experiments were carried out, each excluding one of the two high-VIF features to ensure each group of VIF value is lower than 10. Based on feature importance and cross-experiment consistency, the last three were excluded, yielding a final set of 10 features, which are described in Table 3:

3.3 Modelling

Min-Max scaling [22] was applied to the prepared input features X and target variable y . The dataset was then split chronologically into train (70%), validation (15%) and test (15%) sets. In this stage, time series models including RNN, LSTM and GRU were employed. Additionally, a Naive Baseline Model was constructed by forecasting the future 30-day moving average of Bitcoin close returns as the same value as the past 30-day MA in order to compare model performance. To further validate the effect of manually denoising the input features before training, we also compared our approach against a state-of-the-art CNN-LSTM model [2], which incorporates convolutional and max-pooling layers for automatic denoising. In the

first comparison, the denoised features used in the other models were applied to the CNN-LSTM to evaluate performance differences resulting from the model architecture. In the second setting, 21 raw inputs, including the OHLCV data of BTC, ETH, XRP, S&P500, and the hashrate were used as features. Table 4 presents the hyperparameter search ranges used for each implemented model.

Table 4: Hyper-parameter tuning across different models

Model	Parameter	Searching Interval
Naive		
Simple RNN [8]	lags	2, 3, 4, 5
	epochs	500 with EarlyStopping
	hidden layers	2, 3
	neurons	4, 8, 16, 32, 64, 128
	batch size	16, 32
	learning rate	0.0001, 0.00005
	optimizer	Adam, RMSprop
CNN-LSTM [2]		
(denoised / raw input)	lags	7, 10, 14, 20
	epochs	100 with EarlyStopping
	convolutional layers	1, 2
	max pooling layers	1, 2
	filters	32, 64
	kernel size	3, 6
	batch size	16, 32
	lstm neurons	16, 32, 64, 128
	dense neurons	16, 32, 64, 128
	optimizer	Adam
	activation	Relu
	dropout	0.2, 0.3, 0.4
LSTM [11] / GRU [6]	lags	5, 6, 7, 8, 10, 12, 14, 16, 18, 20
	epochs	500 with EarlyStopping
	hidden layers	2, 3, 4, 5
	neurons	4, 8, 16, 32, 64, 128
	batch size	16, 32
	learning rate	0.0001, 0.00005
	optimizer	Adam, RMSprop
	activation	Relu, LeakyRelu(alpha=0.01)
	dropout	0.2, 0.4, 0.6

ROC Curve Comparison: CNN-LSTM(denoised) vs CNN-LSTM(raw) vs RNN vs GRU vs LSTM

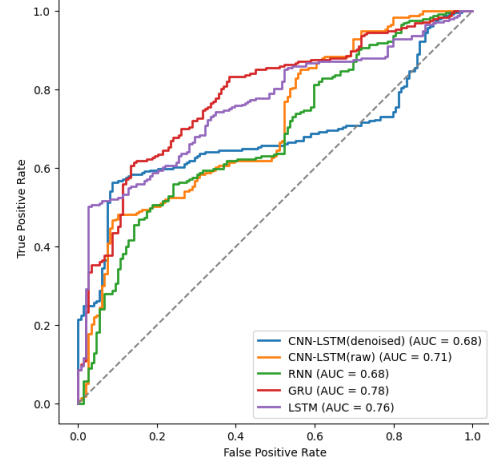


Fig. 4: ROC curve across different threshold

3.4 Evaluation

When evaluated using regression metrics from Table 5, CNN-LSTM, RNN, LSTM and GRU significantly outperformed the Naive model. For the state-of-the-art CNN-LSTM model, using the denoised features as input resulted in lower performance compared to using 21 raw features. This performance gap may be attributed to the additional smoothing introduced by the convolution and max-pooling layers, which led to further information loss. Interestingly, CNN-LSTM (raw input) achieved the lowest percentage-dependent metrics (SMAPE) on both the training and test sets, while LSTM and GRU demonstrated lower scale-dependent metrics (MAE and MSE). This occurs because SMAPE evaluates the relative distance between each predicted value and its corresponding ground truth, without taking the overall distribution into account. MSE and MAE, on the other hand, penalize errors according to their magnitude [12]. Therefore, the observed outcome is reasonable due to the inherent volatility of Bitcoin target value, which causes large variations in magnitude. Moreover, in Fig. 4, ROC analysis across varying thresholds also confirmed that GRU (AUC: 0.78) and LSTM (AUC: 0.76) outperformed RNN (AUC: 0.68), CNN-LSTM trained on raw input (AUC: 0.71) and CNN-LSTM trained on denoised input (AUC: 0.68).

Table 5: Model Performance on Train & Test

Model	Training Set			Test Set		
	MAE	MSE	SMAPE	MAE	MSE	SMAPE
Naive	0.147	0.035	29.10%	0.091	0.011	16.95%
CNN-LSTM (denoised)	0.103	0.017	22.64%	0.063	0.007	13.83%
CNN-LSTM (raw)	0.100	0.016	19.71%	0.055	0.005	10.28%
RNN	0.098	0.016	20.88%	0.056	0.005	11.5%
LSTM	0.096	0.015	21.64%	0.050	0.004	12.01%
GRU	0.095	0.015	21.81%	0.051	0.004	11.82%

To further investigate the performance differences among RNN, LSTM, and GRU, Diebold-Mariano test was conducted on the test dataset and the result showed that both LSTM and GRU statistically outperformed

over RNN ($p < 0.05$). However, no significant difference was found between LSTM and GRU ($p = 0.3287$). Fig. 5 further illustrates the comparison between actual values and the predictions generated by the CNN-LSTM(raw input), Simple RNN, LSTM, and GRU models on both the training and test datasets. It can be observed that LSTM and GRU more closely follow the actual value patterns than others.

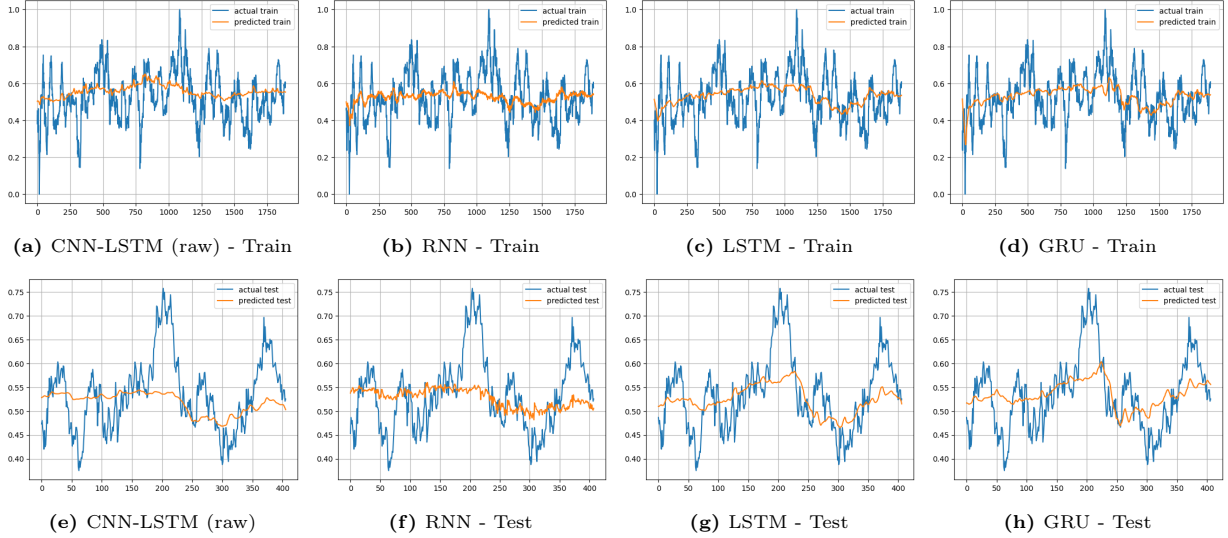


Fig. 5: Actual vs. predicted values on train and test sets for CNN-LSTM (raw), RNN, LSTM, and GRU models.

3.5 Results

Based on the overall evaluation, the models were ranked as:

$$\text{LSTM} \approx \text{GRU} > \text{CNN-LSTM (raw)} > \text{RNN} > \text{CNN-LSTM (denoised)} > \text{Naive}$$

These results answer our RQ1, suggesting that standalone LSTM and GRU models, when combined with appropriate manual feature engineering and selection, can outperform state-of-the-art CNN-LSTM models, which capture local patterns and perform automated denoising through their convolutional and pooling layers. The final optimal thresholds in LSTM and GRU were determined using both the training and validation sets to ensure consistency. The binarized direction were then used as input features for the following stage.

4 Stage 2: GPT-4o Fusion for Hybrid Bitcoin Direction Forecasting

4.1 Data Collection

The dataset used for the sentiment analysis for the GPT decision layer was sourced from Reddit. Reddit was selected due to its high volume of organic, real-time discourse, particularly in the financial and cryptocurrency communities [28]. Data collection occurred via two methods:

- **Live Reddit Data (PRAW API):** The PRAW library was used to scrape live data from Reddit. Submissions from a curated list of the top Bitcoin or cryptocurrency related subreddits (e.g., r/Bitcoin, r/Cryptocurrency, r/BitcoinMarkets). However, this PRAW API is limited to a certain time period in which it was allowed to be used and stopped scraping posts prior to February 2023.
- **Historical Reddit Archive (Academic Torrents):** To address this limitation, we utilized a 3.2 TB archive of Reddit posts encompassing all content from 2005 to 2024, obtained from Academic Torrents. These files were stored on a high speed SSD to facilitate the large files and allow for decompressing and parsing of the files to obtain the target subreddits. Monthly .zst files were decompressed using Zstandard CLI and parsed in 6 month batches due to the high volume of data in each file (approx 150GB per month).

Each post was parsed from JSON format, extracting title, selftext, subreddit and created_utc, which was used to create a date format in YYYY-MM-DD. There was 139 days missing in this dataset, where no Reddit data was available presumably down to some of the earlier days when Reddit wasn't as predominant as today and also some random missing days most likely due to the torrent file not having them. This parsed data was stored in a CSV file, and the number of data points is as follows:

- Train set: 1,894 data points (Torrent file)
- Test set: 407 data points (PRAW API)

Special care was taken to handle malformed JSON, Unicode/encoding issues and no standard characters, ensuring the highest data quality was achieved for downstream modelling.

4.2 Reddit Text Processing

After the extracting of the target subreddits was completed, the Reddit posts were preprocessed to prepare them for input into the GPT layer. This pre-processing involved various steps such as lower casing, stopword removal and normalisation [13]. This cleaned title and selftext was then concatenated for each day into a single cell per date. This resulted in the daily representation of Reddit sentiment. In addition, non-UTF characters, links, emojis and Markdown formatting were removed. To prevent prompt overflow, each daily document was capped at a certain character or token limit.

4.3 GPT-4o Prompt Design and Fusion Strategy

The model architecture was constructed as seen in Fig 1. The framework was designed to leverage the strengths of the time series models (LSTM and GRU), sentiment from Reddit and GPT-4o agent for predicting the directional movement of Bitcoin's 30-day moving average. GPT-4o was chosen for its advanced reasoning capabilities and its ability to understand complex textual data from Reddit, making it suitable for the model framework [20,28]. Two strategies were developed for the training and test sets to reflect real-world use cases and performance goals.

4.3.1 Train Set - Selective GPT Involvement

During the training phase (1,894 data points), the aim was to balance predictive power with the computational cost of running the GPT model across such a large dataset. The decision logic operated under the following rule based structure to combat this computational cost while also displaying the effectiveness of the model:

- If LSTM and GRU agreed on the prediction their consensus was accepted without evoking the GPT, as shown in Fig 6a
- If LSTM and GRU disagreed, GPT-4o was initialized to make a decision based on the model predictions, the model metrics and the Reddit sentiment, as presented in Fig 6b



Fig. 6: Selective vs. Full GPT Integration

This method was designed to display the benefit of using the GPT as a decision layer while trying to combat such a large dataset. In real deployment, the model would not be applied across such a large dataset spanning six years' textual data and could be utilised to its full capacity as seen in the test set. The GPT prompt included text from Reddit with the date removed, the metrics and also converted predictions from the LSTM and GRU and strictly instructed the model to return three valid JSON responses: "increase", "decrease" and "I don't know". This third response was included to allow the GPT to acknowledge uncertainty when it could not come to a conclusion and to restrict it not to hallucinate [34]. In addition to this, a temperature setting of 0.0 was used to make the model deterministic and eliminated any variability in output generation, providing a consistent and trustworthy evaluation [32].

4.3.2 Test Set - Full GPT Integration

In the testing phase, the system operated in full capacity to evaluate the full decision making capability of GPT-4o in isolation and mimic a practical deployment of the model. Regardless of whether LSTM and GRU agreed or disagreed, the GPT layer was always utilised. The setup for the GPT prompt was as follows:

- Converted signal from LSTM and GRU predictions.
- Reddit public sentiment (title and selftext for given date).
- Model Metadata (e.g., metrics like F1 score, precision, recall of converted LSTM and GRU).

GPT was prompted to use these three inputs of information to make the final decision. However, if the Reddit data was unavailable or were not conclusive, it would make the decision based on the metrics of the models.

To ensure deterministic, reproducible and reliable results, a temperature setting of 0.0 was used for the test set. This negated any ways of the GPT hallucinating and forced the GPT to only use the information provided to it [32].

4.4 Results

We applied selective GPT involvement for the train set and full GPT integration for the test set, from table 6, it can be observed that in both data sets the LSTM-GRU-GPT4o model achieved significantly higher F1 scores (0.71 & 0.76) compared to converted CNN-LSTM (denoised), CNN-LSTM (raw), LSTM and GRU models that did not use public sentiment. In training, the rule-based LSTM-GRU-GPT4o model showed a notable improvement in recall (0.77), although this also introduced a potential overreliance on LSTM and GRU signals. In contrast, during the test set, the complete integration of Reddit text significantly improved precision (0.85), showing the ability of GPT-4o for contextual analysis to effectively filter false positives, which potentially helped mitigating investment risk. Thus, our RQ2 is addressed through these findings.

Table 6: Macro Average Performance on Train and Test

Model	Train				Model	Test			
	Accuracy	Precision	Recall	F1		Accuracy	Precision	Recall	F1
CNN-LSTM (denoised)	0.58	0.58	0.56	0.54	CNN-LSTM (denoised)	0.54	0.67	0.62	0.53
CNN-LSTM (raw)	0.63	0.66	0.61	0.59	CNN-LSTM (raw)	0.60	0.69	0.66	0.59
LSTM	0.65	0.65	0.64	0.64	LSTM	0.66	0.70	0.71	0.66
GRU	0.66	0.67	0.66	0.65	GRU	0.70	0.68	0.68	0.68
LSTM-GRU-GPT4o	0.66	0.66	0.77	0.71	LSTM-GRU-GPT4o	0.73	0.85	0.69	0.76

5 Conclusion and Discussion

This study introduces a hybrid framework for Bitcoin price direction prediction combining both time series models with a GPT-4o reasoning layer integrated with sentiment from Reddit. This framework resulted in an increase in F1 score as high as 71% in the train and 76% in the test set, outperforming the state-of-the-art CNN-LSTM models, standalone LSTM and GRU models, which were initially trained for regression tasks and subsequently converted for classification. These results demonstrate the benefits of combining temporal modeling with sentiment data as well as the use of the GPT-4o layer for richer interpretation of unstructured data. While this framework is applied to Bitcoin in this study, the architecture in Fig 1 could be adapted to other domains such as stock trading, sports forecasting, and retail sales forecasting by switching the sentiment source and adjusting model inputs accordingly.

Further work could be implemented to use finer time resolutions such as gathering hourly data to predict the 24-hour MA returns. Moreover, the further combination of models trained on multiple temporal granularities as inputs to the GPT reasoning layer could enhance the predictive accuracy further. In addition to this, a constraint of the study was that the model could be susceptible to sentiment manipulation. A large trader could, in theory, flood Reddit with coordinated negative posts to influence sentiment and the model’s prediction. Future versions should consider countermeasures such as bot detection, source weighting, or cross-referencing sentiment signals. Additionally, various other time series models could also be added in to this large scale deployment. While the models performance is promising, the GPT-4o is subscription based and could pose challenges with computational cost and scalability.

Overall, this study lays the foundations for the potential combination of LLM-based sentiment processing with time series models, further research needs to be carried out in order to improve the framework’s interpret ability, increase data diversity and assess the real time performances.

References

1. Abdelatif Hafid, Mohamed Rahouti, L.K.M.E.M.A.S.: Predicting bitcoin market trends with enhanced technical indicator integration and classification models (2024), available at: <https://arxiv.org/abs/2410.06935>
2. Badar, W., Ramzan, S., Raza, A., Fitriyani, N.L., Syafrudin, M., Lee, S.W.: Enhanced interpretable forecasting of cryptocurrency prices using autoencoder features and a hybrid cnn-lstm model. *Mathematics* **13**(12) (2025). <https://doi.org/10.3390/math13121908>
3. Berger, T., Koubová, J.: Forecasting bitcoin returns: Econometric time series analysis vs. machine learning. *Journal of Forecasting* **43**(7), 2904–2916 (2024). <https://doi.org/10.1002/for.3165>
4. Boozary, P., Sheykhan, S., GhorbanTanhaei, H.: Forecasting the bitcoin price using the various machine learning: A systematic review in data-driven marketing. *Systems and Soft Computing* **7**, 200209 (2025). <https://doi.org/10.1016/j.sasc.2025.200209>
5. Chen, J.: Analysis of bitcoin price prediction using machine learning. *Risk and Financial Management* (2023), available at: <https://doi.org/10.3390/jrfm16010051>
6. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: Wu, D., Carpuat, M., Carreras, X., Vecchi, E.M. (eds.) *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. pp. 103–111. Association for Computational Linguistics, Doha, Qatar (Oct 2014). <https://doi.org/10.3115/v1/W14-4012>
7. Dubey, R., Enke, D.: Bitcoin price direction prediction using on-chain data and feature selection. *Machine Learning with Applications* **20**, 100674 (2025). <https://doi.org/10.1016/j.mlwa.2025.100674>
8. Elman, J.L.: Finding structure in time. *Cognitive Science* **14**(2), 179–211 (1990). [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E)
9. Fan, J., Yao, Q.: *The Elements of Financial Econometrics*. The Elements of Financial Econometrics, Cambridge University Press (2017), <https://books.google.co.kr/books?id=7N0cDgAAQBAJ>
10. Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D., Giaglis, G.M.: Using time-series and sentiment analysis to detect the determinants of bitcoin prices. *SSRN* (May 2015). <https://doi.org/10.2139/ssrn.2607167>
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (Nov 1997). <https://doi.org/10.1162/neco.1997.9.8.1735>, <https://doi.org/10.1162/neco.1997.9.8.1735>
12. Jierula, A., Wang, S., OH, T.M., Wang, P.: Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Applied Sciences* **11**(5) (2021). <https://doi.org/10.3390/app11052314>

13. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson, 3rd edn. (2023), <https://web.stanford.edu/~jurafsky/slp3/>
14. Kaabar, M., et al.: Deep Learning for Finance. Apress, New York, NY (2024), <https://link.springer.com/book/10.1007/978-1-4842-9232-0>
15. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Phys. Rev. E **69**, 066138 (Jun 2004). <https://doi.org/10.1103/PhysRevE.69.066138>
16. Levantesi, S., Piscopo, G., Roviello, A.: Cryptocurrency in global dynamics: Analyzing the crypto volatility index and financial markets with machine learning. Physica A: Statistical Mechanics and its Applications **674**, 130770 (2025). <https://doi.org/https://doi.org/10.1016/j.physa.2025.130770>
17. Liu, C., Arulappan, A., Naha, R., Mahanti, A., Kamruzzaman, J., Ra, I.H.: Large language models and sentiment analysis in financial markets: A review, datasets, and case study. IEEE Access **12**, 134041–134061 (2024). <https://doi.org/10.1109/ACCESS.2024.3445413>
18. Liu, F., Li, Y., Li, B., Li, J., Xie, H.: Bitcoin transaction strategy construction based on deep reinforcement learning. Applied Soft Computing **113**, 107952 (2021). <https://doi.org/https://doi.org/10.1016/j.asoc.2021.107952>
19. Lopez-Lira, A., Tang, Y.: Can chatgpt forecast stock price movements? return predictability and large language models. SSRN (2023), available at: <http://dx.doi.org/10.2139/ssrn.4412788>
20. Niszczoła, P., Abbas, S.: Gpt has become financially literate: Insights from financial literacy tests of gpt and a preliminary test of how people use it as a source of advice. Finance Research Letters **58**, 104333 (2023). <https://doi.org/10.1016/j.frl.2023.104333>
21. Pattanayak, R.M., Sai Raju, M.C., Vishnu, V., Vivek, S.T., Rithwik, J.S.: Gated recurrent unit based deep learning model for bitcoin price prediction. In: 2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS). pp. 1–7 (2024). <https://doi.org/10.1109/ICIICS63763.2024.10859990>
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
23. Pelster, M., Val, J.: Can chatgpt assist in picking stocks? Finance Research Letters **59**, 104786 (2024). <https://doi.org/https://doi.org/10.1016/j.frl.2023.104786>
24. de Prado, M.L.: Advances in Financial Machine Learning. Wiley Publishing, 1st edn. (2018)
25. Raschka, S., Mirjalili, V.: Python Machine Learning. Packt Publishing Ltd., Livery Place 35 Livery Street Birmingham B3 2PB, UK, second edn. (September 2017)
26. Raudys, A., Pabarškaitė, : Optimising the smoothness and accuracy of moving average for stock price data. Technological and Economic Development of Economy **24**(3), 984–1003 (May 2018). <https://doi.org/10.3846/20294913.2016.1216906>
27. Ross, B.C.: Mutual information between discrete and continuous data sets. PLOS ONE **9**(2), 1–5 (02 2014). <https://doi.org/10.1371/journal.pone.0087357>
28. Roumeliotis, K.I., Tselikas, N.D., Nasiopoulos, D.K.: Llms and nlp models in cryptocurrency sentiment analysis: A comparative classification study. Big Data and Cognitive Computing **8**(6), 63 (2024). <https://doi.org/10.3390/bdcc8060063>
29. Shin, M., Mohaisen, D., Kim, J.: Bitcoin price forecasting via ensemble-based lstm deep learning networks. In: 2021 International Conference on Information Networking (ICOIN). pp. 603–608 (2021). <https://doi.org/10.1109/ICOIN50884.2021.9333853>
30. Siami-Namini, S., Tavakoli, N., Siami Namin, A.: A comparison of arima and lstm in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1394–1401 (2018). <https://doi.org/10.1109/ICMLA.2018.00227>
31. Sonata, I., Heryadi, Y.: Comparison of lstm and transformer for time series data forecasting. 7th International Conference on Informatics and Computational Sciences (2024), available at: 10.1109/ICICoS62600.2024.10636892
32. Submission, A.: Time series augmented generation for financial applications. ACL ARR May 2025 Submission (May 2025), <https://openreview.net/forum?id=02CgjW8JZF>, available via OpenReview
33. Yao, M., Sun, G., Liu, J.: Gru prediction method for digital cryptocurrency prices. In: 2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI). pp. 412–416 (2023). <https://doi.org/10.1109/ICCBD-AI62252.2023.00076>
34. Zhang, H., Diao, S., Lin, Y., Fung, Y.R., Lian, Q., Wang, X., Chen, Y., Ji, H., Zhang, T.: R-tuning: Instructing large language models to say ‘i don’t know’. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2024). pp. 7113–7139. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024). <https://doi.org/10.18653/v1/2024.naacl-long.394>
35. Zhao, H., Crane, M., Bezbradica, M.: Attention! transformer with sentiment on cryptocurrencies price prediction. pp. 98–104 (01 2022). <https://doi.org/10.5220/0011103400003197>