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

Zhengri Cui*, Daniel Comerford*
*School of Computing, Dublin City University

***Abstract*—This paper explores a novel hybrid framework by combining numerical time series models with large language model (LLM)-based sentiment analysis. The first stage of this paper demonstrates the modelling of the RNN, LSTM and GRU models 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 the deep learning models outperformed the Naive Baseline Model, with LSTM and GRU showcasing superior MAE and MSE when compared to the RNN and Naive model. 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, these 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.**

I. Introduction

Price volatility has been one of Bitcoin's most challenging aspects when it comes to accurate forecasting. Bitcoin, unlike typical financial assets, has fluctuations that are heavily influenced by both quantitative indicators and the rapidly shifting sentiment from retail investors. Numerous machine learning (ML) and deep learning models have been proposed to forecast the price level or the log returns. However, many ignore sentiment from the public or rely on static sentiment score derived from models.

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 [1][2]. This allowed for the novel architecture proposed in this paper of integrating rich, dynamic sentiment with numerical time series models to allow the GPT to increase the prediction score without the need for handcrafted feature engineering or scoring systems.

This paper proposes 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 given the RNNs models accuracy metrics and predictions for a certain period. The first stage of this paper, involves modelling the RNN models to forecast the 30-day moving average (MA) for the Bitcoin daily returns using historical quantitative data and financial indicators. These predictions from the models were converted to directional signals (up/down). The second stage of this paper consisted of combining these signals with contextual text data from Reddit 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?

The framework was evaluated using standard regression (stage 1) and classifications (stage 2) metrics. 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.

II. Related Work

Previous studies on Bitcoin price movement forecasting have used various features which can be categorized as below: (1) Bitcoin price related variables such as OHLCV data, (2) technical indicators including MACD and RSI, 3) other cryptocurrencies like Ethereum (ETH) and Ripple (XRP), (4) macroeconomic and market indicators scuh as S&P500, (5) on-chain metrics such as hash rate, (6) public sentiment (e.g. Google trend, Tweets, Reddits, News). I. Georgoula et al.[3] report that in short-term analysis, Twitter sentiment and hash rate positively influence Bitcoin prices, while macroeconomic factors like S&P500 has a long-term effect on Bitcoin. This paper suggests that Bitcoin and S&P500 act as substitute assets, meaning a rise in S&P500 correlates with a decrease in Bitcoin prices.

Regarding model application, Recurrent Neural Networks (RNNs) have been actively used in many previous studies. A review of prior studies concluded that the application of LSTM significantly improves Bitcoin price prediction accuracy [4]. Neither financial time series forecasting using the traditional statistical method like ARIMA [5], nor approaches employing powerful machine learning algorithms like Random Forest [6], nor even the use of the more recent Transformer architecture [7][8] have outperformed LSTM. Moreover, LSTM has been served as a underlying model in studies utilizing advanced methodologies, including ensemble methods with three LSTM individual models across different time granularities [9], policy-based reinforcement learning techniques [10] and multi-stage CNN-LSTM architectures [11]. In a study on Bitcoin price prediction [12], GRU (Gated Recurrent Unit)

achieved lower RMSE and MAE compared to SVM and BiLSTM, while another study[13] 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, 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 [15]. In addition, sentiment-based prediction methods using LLMs have been explored. Such approaches have been mainly applied to stock market forecasting[16][17] and have not yet been used in Cryptocurrency market. In particular, one study utilised ChatGPT to analyse news headlines and predict stock price movements without any specialized training. Their method resulted in a cumulative return of over 650% between October 2021 and December 2023[18].

III. Methodology

The overall architecture and workflow employed in this study are displayed in Figure 1, encompassing all key steps from data collection to final prediction. The current workflow focuses on Bitcoin and Reddit sentiment, however, the framework is modular and extensible. It can easily accommodate other data sources and inputs such as APIs from from financial news, analyst reports or specific sector updates. This allows for 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, we use sequential deep learning models (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, 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:

A. Stage 1: Time Series Prediction via Sequential Models

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 the 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.

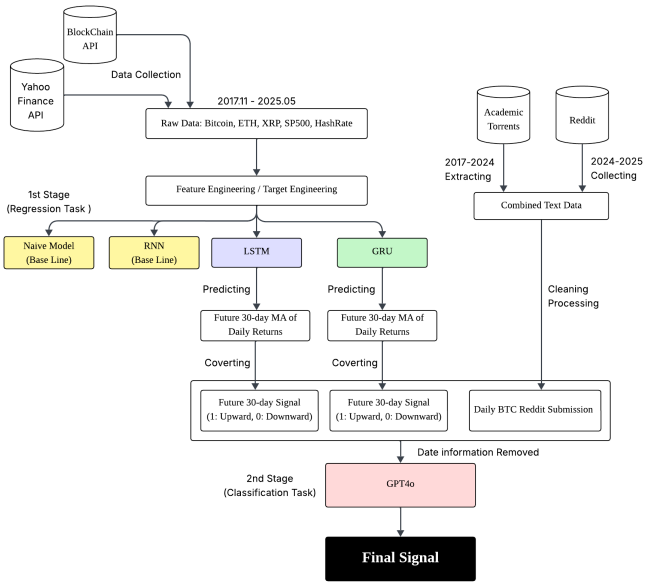


Fig. 1: Overall Architecture and Key Steps. While this study uses Bitcoin price data and Reddit sentiment, the modular architecture supports integration of additional data sources such as financial news APIs (e.g., FT, WSJ), social media platforms (e.g., Twitter), or domain-specific feeds, enabling broader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

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.

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, Log Returns and Fractional Differentiation were employed to stabilize the statistical properties of the data. In addition, moving average smoothing 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. The key transformation formulas are described below:

a) The Daily Percentage Change (DPC) is calculated as:

$$\text{DPC}_t = \left(\frac{V_t - V_{t-1}}{V_{t-1}} \right) \times 100 \tag{1}$$

where V_t represents the timeseries value at time t.

b) The Log Returns (LR) are computed as:

$$\text{LR}_t = \ln \left(\frac{P_t}{P_{t-1}} \right) \tag{2}$$

where P_t represents the asset Close price at time t.

c) Fractional Differentiation (FD): In simple terms, non-stationary raw data without any differencing is considered order 0 and fully differenced data is order 1. Fractional differentiation falls in between, where the amount of differencing is a fraction rather than a whole number. It works by applying a weighted sum of past values to each point of series, enabling the series to remember its historical patterns while achieving a stationary state. This method is recognized as an advanced technique for financial analysis[19] [14].

d) The 30-Day Moving Average (MA30) is calculated as:

$$\text{MA}_t = \frac{1}{30}\sum_{i=0}^{29} V_{t-i}$$

where V_t-_i represents the timeseries value at time $t-i$.

A total of 43 features were engineered, including the prediction target candidates. To clearly illustrate the differences among these transformations, we present only the visualization of BTC Close transformations in Figure 2.

The blue plot represents the non-stationary raw closing price of Bitcoin overtime. Kaabar et al. [14] argues that many deep learning models, including RNNs, inherently assume stationarity of time series data.

The green plot in Figure 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 yellow 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.

*3) **Feature and Target Selection**:* From the initial set of 41 features, a multi-stage feature and target selection process was conducted. The overall workflow is illustrated in Figure 3

a) Target Setting: 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.

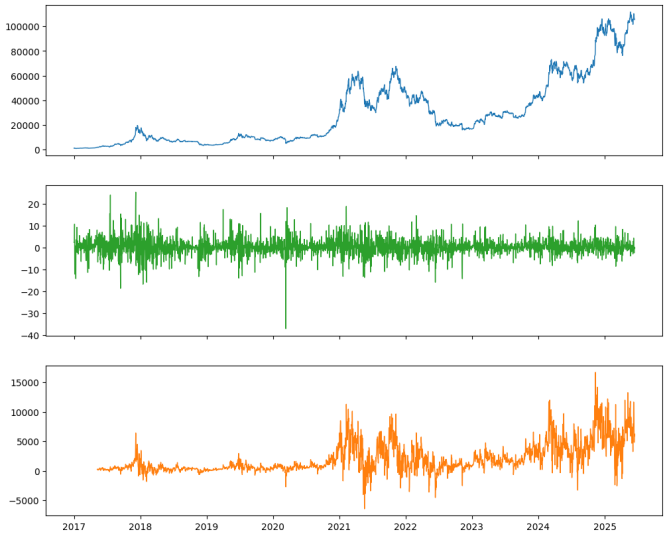


Fig. 2: BTC_Close vs BTC_DPC vs BTC_Close_FD

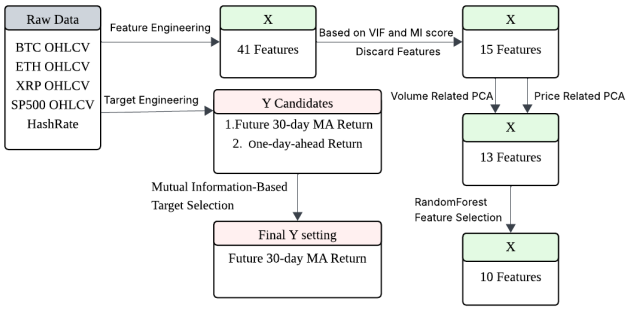


Fig. 3: Target and Feature Selection WorkFlow

To achieve this objective the following two candidates were considered as potential target variables:

- the 1-day ahead Bitcoin close return
- 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 [20][21]. 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.

b) Feature Engineering: The initial set of 41 features was reduced to 15 by considering both Variance Inflation Factor (VIF) and Mutual Information (MI) values. The initial feature selection criteria is as follows:

- Features whose MI values with the target are below 0.05 are removed.
- 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 because of their similar variance. Then within each group, only the feature with the highest MI was selected to reduce redundancy.

For example, in the S&P 500 family, the MI values with the target variable are as follows:

- SP500_RollingClose: 0.5019
- SP500_High: 0.4969
- SP500_Open: 0.4630
- SP500_Low: 0.4563
- SP500_Close: 0.4491
- SP500_Volume: 0.1118
- SP500_RollingVolumeChange: 0.0883
- SP500_DailyChange: 0.0000

Based on this criteria, among price-related features, only `SP500_RollingClose`, which had the highest MI value with the target was remained, while the other price-related OHLC features were removed. This effectively reduced the VIF. As reducing VIF is important because Random Forest for final feature selection will be employed at the next step. High VIF indicates multi-collinearity, which can distort feature importance ranking in the Random Forest model. Additionally, features with very low MI scores were discarded since its MI value is lower than 0.05. The same procedure was applied to other groups. The 15 initially selected features and their corresponding VIF values are summarized in the following Table I.

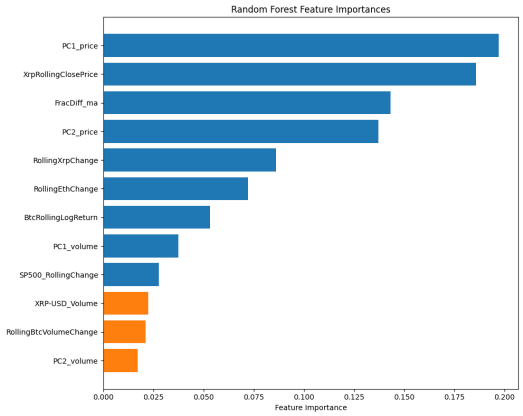
TABLE I: Initially Selected Features and Their VIF Values

Feature	VIF
RollingBtcClose	84.370654
SP500_RollingClose	41.204989
RollingHashRate	20.136093
EthRollingClosePrice	19.353293
SP500_Volume	17.102860
BTC_Close_FracDiff_MA	16.236423
RollingBtcVolume	15.132654
ETH_Volume	10.385493
XrpRollingClosePrice	8.422122
RollingBtcVolumeChange	5.310025
RollingEthChange	4.054758
XRP_Volume	3.755034
BtcRollingLogReturn	3.496564
RollingXrpChange	2.195537
SP500_RollingChange	1.431409

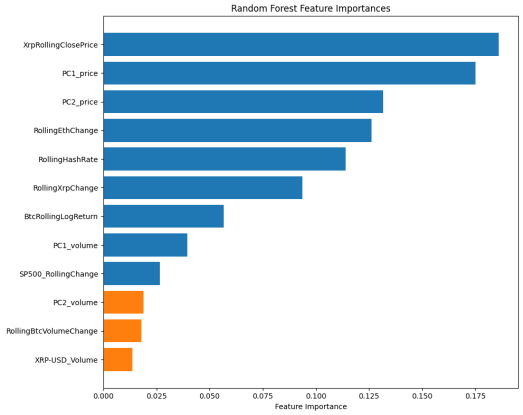
TABLE II: Features and Their VIF Values After PCA Reduction

Feature	VIF
RollingHashRate	12.219997
BTC_Close_FracDiff_MA	10.865961
RollingBtcVolumeChange	5.077509
XrpRollingClosePrice	4.420683
RollingEthChange	3.798718
BtcRollingLogReturn	3.519296
XRP-USD_Volume	3.494823
PC1_price	3.407892
PC1_volume	3.139093
RollingXrpChange	2.116676
PC2_price	1.557567
SP500_RollingChange	1.443153
PC2_volume	1.441213

c) Multicollinearity Reduction via PCA: Several features still exhibited VIF values exceeding 10, indicating that a high degree of multicollinearity remains in the dataset. Therefore, three price related features `RollingBtcClose`, `SP500_RollingClose`, `EthRollingClosePrice` and three volume related features `RollingBtcVolume`, `SP500_Volume`, `ETH_Volume` were reduced using PCA. After applying PCA to each group, the VIF values for all features are summarized in Table II.



(a) Excluding RollingHashRate



(b) Excluding BTC_Close_FracDiff_MA

Fig. 4: Importance rankings from two RF tests

d) Final Feature Selection Using Random Forest: 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 II, `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[22]. 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. The least important features were examined to conclude if they were consistent across both experiments and based on Fig 4, the last three were removed to finalize the 10-feature set. The final selected features and descriptions are in Table III:

TABLE III: 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

*4) **Modelling**:* Min-Max scaling 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. Hyperparameter tuning was conducted using only the training set 70% through Time Series Expanding Cross Validation with three folds. The hyperparameters that gave the lowest average validation loss (measured by MAE) across three folds were used as a reference. Subsequently, the models were retrained on the 70% training and 15% validation sets with the chosen set of hyperparameters. This process is designed to create a model that performs reliably and consistently well when predicting across different time periods in the time series data.

a) Naive Baseline Model: The target variable y is naively defined as the moving average of Bitcoin close returns over the past 30 days from the current time point.

b) Simple RNN Model [23]: Recurrent Neural Network (RNN) is a neural networks designed to model timeseries data by maintaining a hidden state that captures information from previous time steps. Therefore, in this study, the sequence length for the RNN model is limited to 5 or less to minimize the consequence of long-term dependency issue.

c) LSTM Model [24]: LSTMs incorporate input, forget and output gates, along with a cell state, to control how much past information should be remembered or forgotten. This gating mechanism allows LSTMs to address the vanishing gradient problem more effectively than simple RNNs. However, when dealing with large-scale datasets, the computational complexity increases, which may result in comparatively slower processing speeds.

d) GRU Model [25]: GRU integrates the forget gate and input gate of LSTM into a single update gate, while the reset gate adjusts the amount of previous information to forget. Additionally, the cell state and hidden state are merged into one, resulting in a simpler architecture. Due to this, GRUs tend to be computationally lighter than LSTMs, while achieving comparable performance, as demonstrated by many studies.

Table IV presents the hyperparameter search ranges used for each implemented model.

TABLE IV: Hyper-parameter tuning across different models

Model	**Parameter**	**Searching Interval**
Naive		
RNN	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
LSTM / GRU	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)
	doupout	0.2, 0.4, 0.6

*5) **Evaluation**:* When evaluated using regression metrics from Table V, RNN, LSTM and GRU significantly outperformed the Naive model; however, their performance was not clearly distinguishable based solely on these metrics.

TABLE V: Model Performance on Train & Test

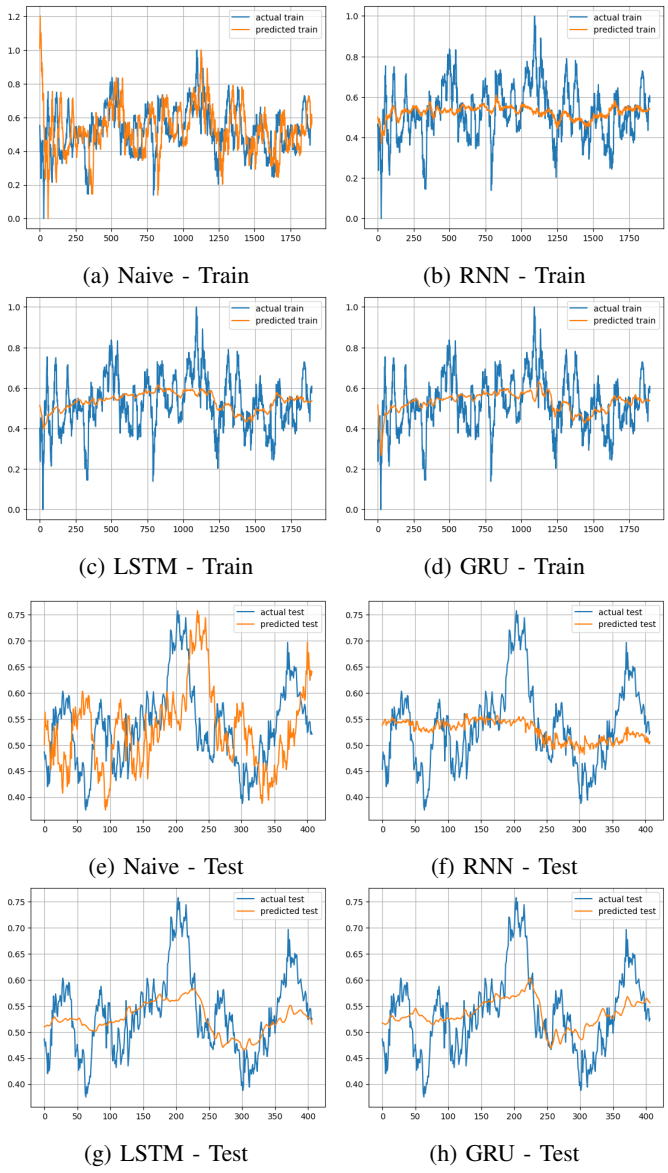
Model	Training Set			Test Set		
	MAE	MSE	SMAPE	MAE	MSE	SMAPE
Naive	0.147	0.035	29.10%	0.091	0.011	16.95%
RNN	0.098	0.016	**20.88%**	0.056	0.005	**11.5%**
LSTM	0.096	0.015	21.64%	**0.050**	**0.004**	12.01%
GRU	**0.095**	**0.015**	21.81%	0.051	0.004	11.82%

To further investigate, 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 were found between LSTM and GRU ($p = 0.3287$).

Moreover, from Figure 5, which presents the comparison between the actual values and the predictions made by the Naive, Simple RNN, LSTM and GRU models on both the

train and test datasets. It can be observed that LSTM and GRU more closely follow the actual value patterns than RNN.

Furthermore, in Fig 6 ROC analysis across varying thresholds also confirmed that GRU (AUC: 0.78) and LSTM (AUC: 0.76) outperformed RNN, which recorded an AUC of 0.68.



(a) Naive - Train (b) RNN - Train

(c) LSTM - Train (d) GRU - Train

(e) Naive - Test (f) RNN - Test

(g) LSTM - Test (h) GRU - Test

Fig. 5: Comparison of actual values and model predictions for training and testing data across Naive, RNN, LSTM and GRU models.

6) ***Results****:* Based on the overall evaluation, the models were ranked as:

$$\text{LSTM} \approx \text{GRU} > \text{RNN} > \text{Naive}$$

The final optimal thresholds in LSTM and GRU were determined using the training and validation sets to ensure consistency across different datasets.The binarized direction were then used as input features for the following stage.

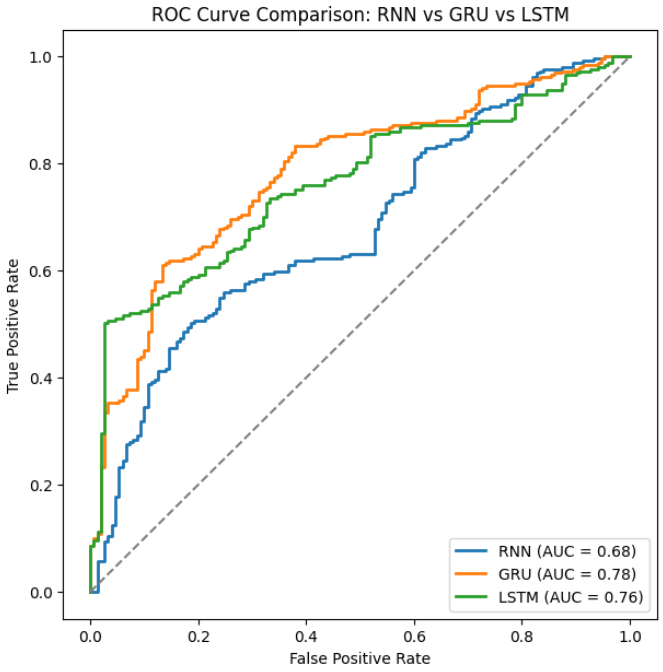


Fig. 6: ROC curve across different threshold

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

1) ***Reddit Text Extracting and Aggregation****:* 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 [26]. 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 overcome this limitation, a 3.2TB Reddit post archive was leveraged which included every post on Reddit from 2005-2024, was retrieved from Academic Torrents [27]. 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 used to build:

- **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.

*2) **Reddit Text Processing**:* After the extracting of the target subreddits was completed, the Reddit posts were pre-processed to prepare them for input into the GPT layer. This pre-processing involved various steps such as lower casing, stopword removal and normalisation [28]. 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.

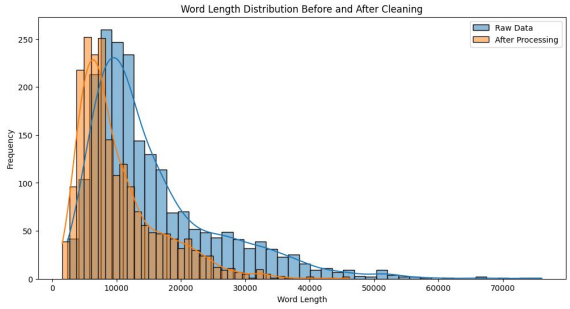


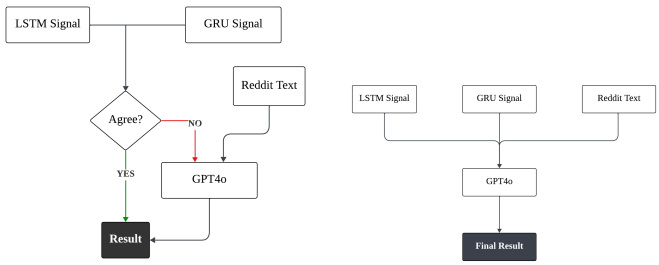
Fig. 7: Word length distribution before and after text processing

*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 [29] [26]. Two strategies were developed for the training and test sets to reflect real-world use cases and performance goals.

a) 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 8a
- **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 8b



(a) Selective GPT Involvement (Train)

(b) Full GPT Integration (Test)

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 [30].

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 [31].

b) 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 were 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 conclude the final decision. However, if the Reddit data was unavailable or was 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[31].

*4) **Results**:* We applied Selective GPT Involvement for train set and Full GPT Integration for test set, from table VI and VII, it can be observed in both dataset the LSTM-GRU-GPT4o model achieved significantly higher F1 scores (0.71 & 0.76) compared to LSTM and GRU models that did not use public sentiment. In training, the rule-based LSTM-GRU-GPT4o model performed a notable improvement in recall (0.77), although this also introduced potential over-reliance on LSTM and GRU signals. In contrast, during the test set, full integration of Reddit text considerably improved precision (0.85), showing GPT-4o's ability for contextual analysis to effectively filter false positives, thereby potentially helping to potentially reduce investment risk.

TABLE VI: Macro Average Performance on Train

Model	Accuracy	Precision	Recall	F1
LSTM(Converted)	0.65	0.65	0.64	0.64
GRU(Converted)	**0.66**	**0.67**	0.66	0.65
LSTM-GRU-GPT4o	**0.66**	0.66	**0.77**	**0.71**

TABLE VII: Macro Average Performance on Test

Model	Accuracy	Precision	Recall	F1
LSTM(Converted)	0.66	0.70	**0.71**	0.66
GRU(Converted)	0.70	0.68	0.68	0.68
LSTM-GRU-GPT4o	**0.73**	**0.85**	0.69	**0.76**

IV. Conclusion

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. The initial stage consisted of training the LSTM and GRU models for a regression task and then converting to directional trading signals for use in the next stages. The GRU achieved 70% accuracy and an F1 of 68%, while the LSTM achieved 66% for both metrics, both outperforming the Naive Baseline Models and single stage approaches.[32]. To further improve this predictive performance, Reddit-derived sentiment data was incorporated as an additional feature and processed in the GPT decision layer. This integrated framework resulted in an increase in F1 score as high as 76% in the test set, demonstrating the benefits of combining temporal modelling with sentiment data as well as the use of the GPT-4o layer for richer interpretation of unstructured data.

This study explores the potential benefit of utilizing a GPT-4o reasoning layer in a forecasting pipeline for volatile markets. 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.

V. Discussion

The aim of this study was to introduce a novel hybrid framework by combining numerical time series models with large language model (LLM)-based sentiment analysis. The target variable was clearly defined as the 30-day moving average for daily Bitcoin returns, based on daily inputs. 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. A key limitation of this framework is the interpret-ability or transparency of the GPT-4o layer. GPT-4o although powerful and demonstrates strong contextual understanding, it has a black-box nature and has limited explainability which is a major factor when carrying out financial modelling and making a decision based on these models. To partially address this black-box issue, the GPT-4o prompt was engineered to be deterministic by using a temperature of (0.0) and a strict prompt design of only being allowed three outputs of "increase", "decrease" or " I don't know" [30][31]. These constraints help to control the reasoning path in the GPT layer, minimize hallucinations and allowed more standardized analysis of the models outputs. While not fully transparent or explainable, the strategies used enhanced the models accountability and logic.

In addition to this, due to time constraints of the study, live testing of full model integration was not feasible. Instead, a backtesting approach was adopted to evaluate the model. Future work needs to be conducted for real-time experiments to assess the model further and fine tune parameters. Another 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.

VI. Acknowledgement

We would like to express our sincere gratitude to Prof. Marija Bezbradica and Dr. An P. N. Nguyen for their invaluable guidance and insightful feedback for this study.

The first author, Zhengri Cui, was primarily responsible for conceptual design, development of the first-stage models, initial coding for GPT-4o, Reddit text processing and manuscript preparation. The co-author, Daniel Comerford, contributed to conceptual design, exploratory data analysis (EDA) of first-stage features, hyperparameter optimization of the first-stage models, Reddit data collection, further implementation and experimentation of GPT-4o and manuscript writing.

REFERENCES

[1] P. Niszczota and S. Abbas, "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*, vol. 58, p. 104333, 2023, available at: https://www.sciencedirect.com/science/article/pii/S1544612323007055.

[2] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Llms and nlp models in cryptocurrency sentiment analysis: A comparative classification study," *Big Data and Cognitive Computing*, vol. 8, no. 6, 2024, available at: https://www.mdpi.com/2504-2289/8/6/63. [Online]. Available: https://www.mdpi.com/2504-2289/8/6/63

[3] I. Georgoula, D. Pournarakis, C. Bilanakos, D. Sotiropoulos, and G. M. Giaglis, "Using time-series and sentiment analysis to detect the determinants of bitcoin prices," *SSRN*, May 2015, available at SSRN: https://ssrn.com/abstract=2607167 or http://dx.doi.org/10.2139/ssrn.2607167.

[4] P. Boozary, S. Sheykhan, and H. GhorbanTanhaei, "Forecasting the bitcoin price using the various machine learning: A systematic review in data-driven marketing," *Systems and Soft Computing*, vol. 7, p. 200209, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772941925000274

[5] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401, available at: https://ieeexplore.ieee.org/abstract/document/8614252.

[6] J. Chen, "Analysis of bitcoin price prediction using machine learning," *Risk and Financial Management*, 2023, available at: https://doi.org/10.3390/jrfm16010051.

[7] I. Sonata and Y. Heryadi, "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.

[8] H. Zhao, M. Crane, and M. Bezbradica, "Attention! transformer with sentiment on cryptocurrencies price prediction," 01 2022, pp. 98–104.

[9] M. Shin, D. Mohaisen, and J. Kim, "Bitcoin price forecasting via ensemble-based lstm deep learning networks," in *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 603–608, available at: https://ieeexplore.ieee.org/abstract/document/9333853.

[10] F. Liu, Y. Li, B. Li, J. Li, and H. Xie, "Bitcoin transaction strategy construction based on deep reinforcement learning," *Applied Soft Computing*, vol. 113, p. 107952, 2021, available at: https://www.sciencedirect.com/science/article/pii/S1568494621008747. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621008747

[11] R. Dubey and D. Enke, "Bitcoin price direction prediction using on-chain data and feature selection," *Machine Learning with Applications*, vol. 20, p. 100674, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266682702500057X

[12] R. M. Pattanayak, M. C. Sai Raju, V. Vishnu, S. T. Vivek, and J. S. Rithwik, "Gated recurrent unit based deep learning model for bitcoin price prediction," in *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 2024, pp. 1–7.

[13] M. Yao, G. Sun, and J. Liu, "Gru prediction method for digital cryptocurrency prices," in *2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI)*, 2023, pp. 412–416.

[14] M. Kaabar *et al.*, *Deep Learning for Finance*. New York, NY: Apress, 2024. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4842-9232-0

[15] S. Levantesi, G. Piscopo, and A. Roviello, "Cryptocurrency in global dynamics: Analyzing the crypto volatility index and financial markets with machine learning," *Physica A: Statistical Mechanics and its Applications*, vol. 674, p. 130770, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437125004224

[16] C. Liu, A. Arulappan, R. Naha, A. Mahanti, J. Kamruzzaman, and I.-H. Ra, "Large language models and sentiment analysis in financial markets: A review, datasets, and case study," *IEEE Access*, vol. 12, pp. 134 041–134 061, 2024, available at: https://ieeexplore.ieee.org/abstract/document/10638546.

[17] M. Pelster and J. Val, "Can chatgpt assist in picking stocks?" *Finance Research Letters*, vol. 59, p. 104786, 2024, available at: https://www.sciencedirect.com/science/article/pii/S1544612323011583. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1544612323011583

[18] A. Lopez-Lira and Y. Tang, "Can chatgpt forecast stock price movements? return predictability and large language models," *SSRN*, 2023, available at: http://dx.doi.org/10.2139/ssrn.4412788.

[19] M. L. de Prado, *Advances in Financial Machine Learning*, 1st ed. Wiley Publishing, 2018.

[20] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.066138

[21] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLOS ONE*, vol. 9, no. 2, pp. 1–5, 02 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0087357

[22] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd ed. Livery Place 35 Livery Street Birmingham B3 2PB, UK: Packt Publishing Ltd., September 2017.

[23] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[25] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, D. Wu, M. Carpuat, X. Carreras, and E. M. Vecchi, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: https://aclanthology.org/W14-4012/

[26] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Llms and nlp models in cryptocurrency sentiment analysis: A comparative classification study," *Big Data and Cognitive Computing*, vol. 8, no. 6, p. 63, 2024. [Online]. Available: https://doi.org/10.3390/bdcc8060063

[27] stuck_in_the_matrix, Watchful1, and RaiderBDev, "Reddit comments/submissions 2005–2024 (zstd compressed ndjson)," https://academictorrents.com/details/20520c420c6c846f555523babc8c059e9daa8fc5, accessed 2025-07-17.

[28] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[29] P. Niszczota and S. Abbas, "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*, vol. 58, p. 104333, 2023. [Online]. Available: https://doi.org/10.1016/j.frl.2023.104333

[30] H. Zhang, S. Diao, Y. Lin, Y. R. Fung, Q. Lian, X. Wang, Y. Chen, H. Ji, and T. Zhang, "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)*. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 7113–7139. [Online]. Available: https://aclanthology.org/2024.naacl-long.394/

[31] A. Submission, "Time series augmented generation for financial applications," *ACL ARR May 2025 Submission*, May 2025, available via OpenReview. [Online]. Available: https://openreview.net/forum?id=O2CgjW8JZF

[32] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:18038866