Forecasting Bitcoin Prices with RSI and Twitter Sentiment: A Comparison Between Traditional and Transformer Models

Shakiba Sadat Mirbagheri, Claudio Sartori, Mehrzad Abdi khalife, Mina Farmanbar
shakiba.mirbagheri@studio.unibo.it University of Bologna | Ticker Bell Kft. Company | University of Stavanger
LinkedIn: shakiba-mirbagheri

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Universitetet i Stavanger

Introduction

Bitcoin's price is highly volatile and influenced by both technical and behavioral factors. Traditional forecasting models often rely on linear assumptions and fail to capture the non-linear dynamics of cryptocurrency markets. This study investigates whether **Transformer-based models** can provide better predictive performance by incorporating technical indicators like **RSI**, **MACD**, and **sentiment signals** extracted from Twitter.

Market Data and Preprocessing

We used 10 years of daily Bitcoin price data **(2015–2025)** from **Investing.com**. The dataset includes **Open, Close, High, Low, Volume**, and two engineered technical indicators: **RSI**, which captures price momentum, and **MACD**, which signals trend direction. Missing values were handled using **forward-fill**, and all features were normalized using **z-score**. The data was split chronologically to avoid leakage: **90% for training, 5% for validation, and 5% for testing**.

Models and Methodology

Goal: Identify the most effective model for Bitcoin price forecasting.

We evaluated multiple forecasting models to identify the most effective approach for Bitcoin price prediction. The models tested included:

- **Statistical** → ARIMA, Prophet
- **Machine Learning** → XGBoost
- **Deep Learning** → LSTM, mLSTM, sLSTM, xLSTM
- **Transformer-based** → PatchTST

The main contribution of **PatchTST** lies in its use of **patch-based representation** and **channel-independent attention**, making it especially suited for capturing non-linear temporal dependencies in financial data. Forecasting was performed using input windows of **24, 36, 48, and 60 days**. Performance was evaluated using both regression metrics **(MSE, MAE, RMSE, MAPE, R^2)** and classification metrics **(Accuracy, Precision, Recall, F1-score)**, the latter based on the direction of price movement.

Sentiment Analysis Pipeline

To investigate whether public sentiment could enhance forecasting performance, we extended our dataset by **incorporating Twitter data**. Bitcoin-related tweets were collected from **2015 to 2019**, offering a **textual snapshot of market sentiment** during that period.

We applied a multi-step preprocessing pipeline: tweets were cleaned by **removing** emojis, hashtags, mentions, URLs, and stopwords. Only English tweets were retained using a pretrained **XLM-RoBERTa model** for language detection.

Each tweet was then classified as **Positive, Neutral,** or **Negative** using **XLM-RoBERTa-base-sentiment**. Daily sentiment scores were averaged and merged with **Bitcoin price data** through **date alignment** to assess their influence on model performance.

Model Evaluation with Price History, RSI, and MACD

We benchmarked eight forecasting models using only historical Bitcoin market data from 2015 to 2025, including technical indicators RSI and MACD. All models were tested across four forecasting windows (24, 36, 48, 60 days). The 24-day input window consistently provided the most accurate results, while longer windows led to decreased performance due to noise in extended time horizons.

Model	MSE	R^2	F1-Score	RSI MAE
PatchTST	0.021	0.9881	0.786	3.47
sLSTM	0.04	0.9867	0.698	3.57
xLSTM	0.041	0.9835	0.698	3.62
mLSTM	0.049	0.9866	0.691	3.56
XGBoost	0.581	0.97	0.548	3.7
Prophet	0.06	0.97	0.578	3.7
ARIMA	0.605	0.968	0.543	3.75
LSTM	0.746	0.6614	0.539	20.49

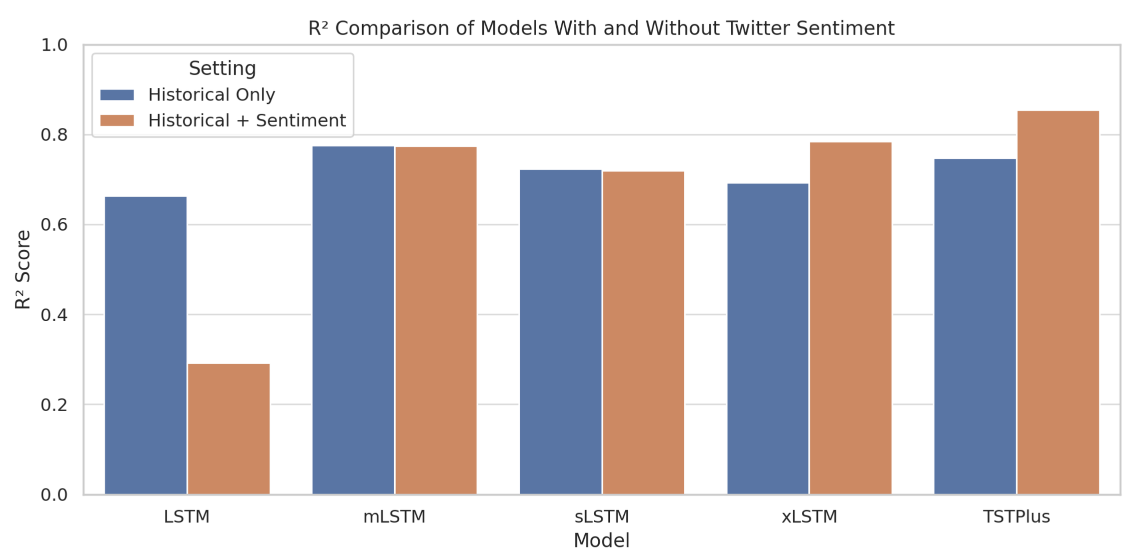
Among all models, PatchTST delivered the best performance, achieving an R^2 of 0.9881 and the lowest MSE of 0.021. It also demonstrated superior accuracy in predicting RSI values, with the lowest RSI MAE of 3.47. In contrast, LSTM performed the worst across all metrics, especially in predicting RSI, where its error exceeded 20, indicating poor generalization in volatile environments. These results confirm that Transformer-based architectures like PatchTST, when applied to pure market data enhanced with RSI and MACD, outperform both traditional and deep learning models in regression accuracy and robustness.

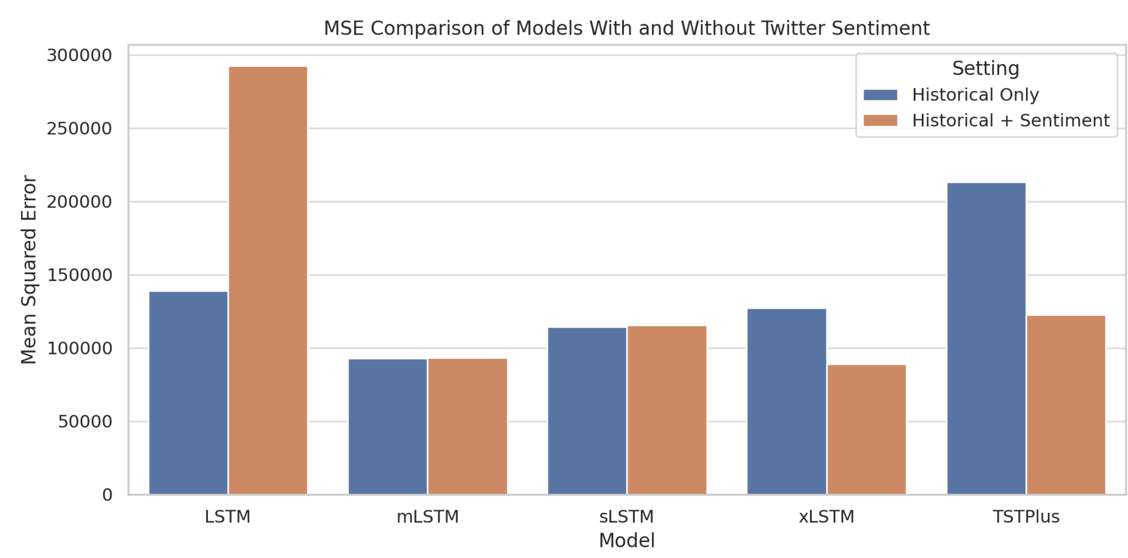
Model Results with Twitter-Enhanced Sentiment Integration

To evaluate the added value of social media sentiment, we compared two experimental setups: five years of historical Bitcoin price data (2015–2019) versus the same period enriched with Twitter-derived sentiment scores.

The results showed that combining sentiment with price data slightly improved the regression accuracy of some models. In particular, xLSTM and TSTPlus benefited from sentiment integration, achieving lower MSE and higher R^2. Conversely, LSTM performed worse when sentiment was added, indicating its limited ability to handle noisy textual signals. Other models, such as mLSTM and sLSTM, remained relatively stable.

Among all, TSTPlus emerged as the most robust and consistent model under both settings, highlighting the advantage of Transformer-based architectures in integrating heterogeneous data sources like social media signals.





Key Findings and Conclusion

This study compared a variety of forecasting models using Bitcoin market data, technical indicators (RSI, MACD), and Twitter sentiment.

Key findings:

- **Transformer-based models, PatchTST** and **TSTPlus** showed the best regression performance, with PatchTST reaching $R^2 = 0.9881$ and MSE = 0.021.
- **RSI and MACD** improved interpretability and boosted forecast accuracy.
- **Twitter sentiment** helped xLSTM and TSTPlus but worsened LSTM performance — emphasizing the importance of model robustness.
- **Recent data** (5-year, 24-day windows) outperformed longer histories, confirming their higher predictive value.
- **TSTPlus** proved to be the most stable across all setups, making it a strong candidate for future sentiment-aware forecasting.

Reference

GitHub: shakibaMB