Predicting Bitcoin (BTC-USD) prices using Twitter posts: A sentiment analysis approach

Computer Science

4026 words

Abstract

This study explores the potential of forecasting Bitcoin (BTC-USD) price movements by analysing sentiment from Twitter posts. Leveraging large language models, logistic classification and neural network models (written in python), this paper assessed the correlation between social media sentiment and Bitcoin price fluctuations. This paper's findings indicate a significant relationship, with the top model returning 239% for an investment over a three-month period, suggesting that public sentiment on Twitter can serve as a predictor for Bitcoin price movements.

Introduction

Bitcoin has emerged as one of the most volatile and widely traded digital assets since its introduction in 2009 [1]. Unlike traditional currencies such as the USD, which was historically tied to gold, Bitcoin is not backed by a gold reserve or any physical asset. Instead, it operates without a central authority, making its price highly susceptible to various external influences, including economic events, regulatory changes, and social sentiment. As financial markets become increasingly intertwined with digital discourse, social media platforms like Twitter have become powerful arenas where opinions, speculation, and news spread rapidly, often influencing asset prices within hours.

Understanding the impact of public opinion on Bitcoin's price movements is crucial for traders, investors, and financial analysts. Traditional market indicators such as technical analysis and fundamental analysis often fail to capture the immediate effects of mass sentiment shifts. However, advancements in artificial intelligence and natural language processing (NLP) now provide a means to extract and quantify emotions from social media posts, offering a new dimension to price prediction models. By integrating sentiment analysis with machine learning techniques, this research explores whether social sentiment on Twitter can serve as a reliable predictor for Bitcoin's short-term price fluctuations.

This study employs state-of-the-art sentiment classification models, such as XLM-RoBERTa, combined with a neural network-based prediction framework to analyse 2,000 tweets alongside historical Bitcoin price data. By assessing sentiment polarity and engagement metrics, this paper aims to determine whether market sentiment provides a tangible edge in forecasting price movements. If successful, this approach could lead to

more informed trading strategies, reduced risk exposure, and enhanced decision-making for market participants, potentially leading to increased profits.

The findings demonstrate a compelling relationship between social sentiment and Bitcoin's short-term price movements. Over a test period from September 30, 2024, to December 31, 2024, during which Bitcoin's price increased by 48%, the average return generated by the models was 97.7%, nearly doubling Bitcoin's performance in the same timeframe. Notably, the best-performing model achieved a return of 239% with a Percentage Correct Movement (PCM) of 62.3%. While it's possible that some outperformance resulted from statistical variance given the 486 models tested, the overall results suggest a tangible correlation between Twitter sentiment and Bitcoin price trends.

Technical Definitions

Bitcoin and blockchain technology

Bitcoin is a decentralized digital currency introduced in 2009 by an anonymous entity known as Satoshi Nakamoto. It enables peer-to-peer transactions without the need for a central authority, operating on a public ledger called the blockchain. Blockchain is 'an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way' [2].

Sentiment analysis and opinion mining

Sentiment analysis is determining the subjectivity, polarity (positive or negative) and polarity strength (weakly positive, mildly positive, strongly positive, etc.) of a piece of text [3]. In this paper, we focus on identifying the sentiment as positive or negative (polarity), but we do not evaluate how strong that sentiment is.

Opinion mining is performing the act of sentiment analysis across a large amount of data to find the general opinion of a given subject.

Twitter

Twitter is a microblogging service that allows users to communicate through short messages known as posts or tweets. These posts can include text, images, videos, and links, facilitating rapid information dissemination and public discourse [4].

Over fitting:

Over fitting in machine learning is a phenomenon when a model memorizes specific points in its training data which means that it performs very well on training data but not well on test data meaning it does not represent the general trends of the training data [5]. This typically occurs due to overcomplex models being used for basic trends in data [6].

Under fitting:

Under fitting in machine learning is when a model does not find a specific trend in the training data and therefore does not perform well on training data or test data. This typically occurs due to simple models being used on complex trends on data [6].

Natural Language Processing (NLP):

NLP is a field of AI that enables computers to understand, interpret and generate human language.

Large Language Model (LLM):

An LLM is a type of NLP model, that attempts to predict the next word from given previous words, an example of this is ChatGPT made by OpenAI.

Previous Research

Cardiff University demonstrated a method of sentiment analysis on tweets which puts the tweet through a Large Language Model (LLM) and then classifies its sentiment [7]. This proved to be able to pick up on sarcasm in tweets well which is important given that it was found that 12.80% of tweets have sarcasm in them [8]. Recognising sarcasm in tweets is crucial because if a model can't detect it, it will often misinterpret negative or critical messages as positive (or vice versa), leading to inaccurate sentiment analysis and flawed insights.

Previous research at Indiana University has demonstrated a correlation between social media activity and financial market trends. For instance, studies have shown that the average sentiment of tweets in any given day can predict the value of the Dow Jones Industrial Average in 3-4 days [9], which is a key indicator of the US stock market. However, due to the volatility of Bitcoin and its extremely sensitive price fluctuations, it is likely tweets will be able to help predict the price of Bitcoin in a shorter time frame such as 24 hours.

Methodology

Data collection was the initial step required, once this was completed the data could be processed. Subsequently, several models were designed and implemented, followed by their training. The models were then evaluated using both machine learning metrics and financial performance metrics to determine which performed best.

Data Collection Method

The required data from the Twitter posts was the datetime, number of likes, number of views, retweet count, number of followers as well as actual text of the post. Due to recent restrictions on web scraping and the high costs associated with official APIs, the tweets must be collected at a controlled rate using web scraping techniques. To maximise relevance, only tweets that had a minimum of 250 likes and had “Bitcoin” tagged were included in the dataset. This provided a dataset of approximately 2000 tweets which span over a two-year period from March 2023 – December 2024. Tweets were collected from different languages including English, Japanese, Hindi, Portuguese, French and Arabic. The required Bitcoin data was the open price, the high price and the low price for the BTC-USD exchange rate for every hour since 10th February 2023. This date was selected as it occurred one month after Elon Musk’s acquisition of Twitter, an event that may have influenced tweet sentiment by altering platform policies, shaping public discourse, or

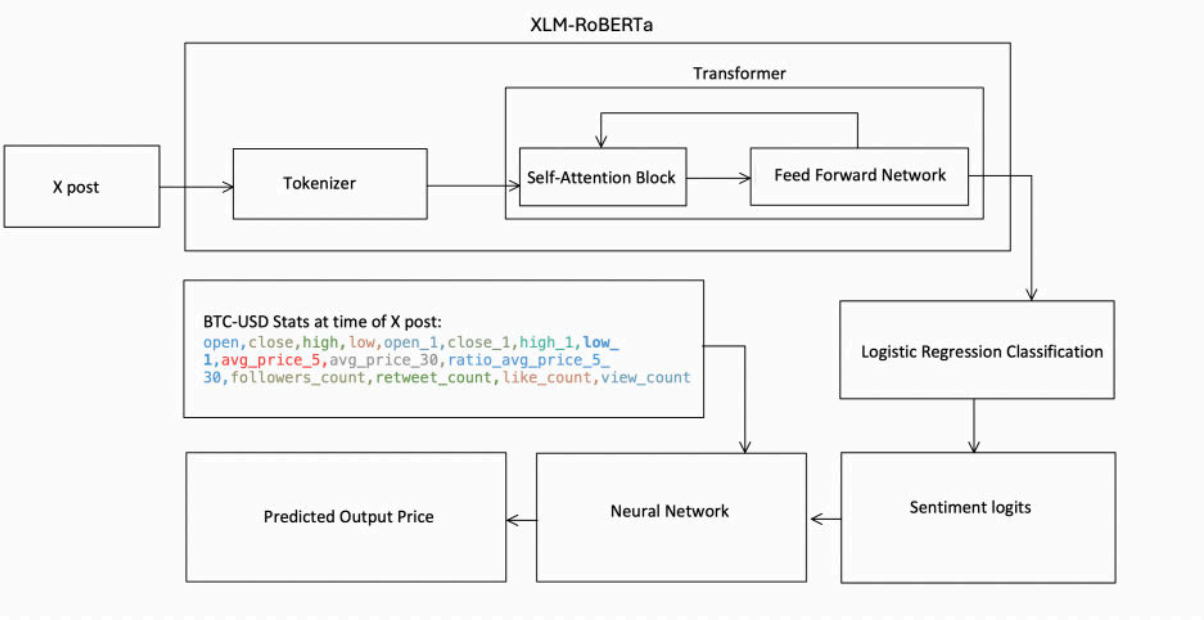
through Musk's own market-moving commentary on cryptocurrency. Bitcoin price data (BTC-USD) was sourced from Yahoo Finance via web scraping techniques [10]. Once both sets of raw data were collected, data processing needed to be performed. This involved assigning each tweet the Bitcoin price data for that time, as well as generating additional features, as depicted in Figure 1, which were generated to enrich the dataset [10].

Model Structure

To predict Bitcoin's price 24 hours ahead based on current price data and social media sentiment, the following steps are taken:

- **Sentiment Analysis:**
  - A tweet is processed using a large language model (LLM).
  - The tweet is classified into one of three sentiment categories: positive, negative, or neutral**.**
  - Classification is done using a logistic regression unit, which generates three sentiment logits, which consists of three probabilities between 0 and 1 which represent the probability the text is positive, negative or neutral.
- **Data Integration:**
  - The sentiment logits**,** Bitcoin price data, and tweet-related engagement metrics are collected.
  - These inputs are fed into a neural network for prediction.
- **Neural Network Optimization:**
  - The neural network takes all the processed and generated data and creates a prediction.
  - Various model architectures and parameters were trained and tested through iterative optimization.
  - The flexibility of architecture of the neural network allowed for performance improvement.

The general structure described of this approach is illustrated in Figure 1.



Flowchart to show structure

Figure 1

Sentiment Analysis XLM-RoBERTa:

To perform sentiment analysis, first the tweet is fed into a LLM which attempts to predict the next word in the sentence, it does this by passing the tweet through a tokenizer and then it is passed through a transformer (made up of a self-attention block and feed forward network). While predicting the next word is not the objective, the process it goes through to do this enables context to be transferred between words, allowing for better classification of sentiment.

Tokenizer:

The tokenizer separates the tweet's text into tokens. A token is usually a single word. Tokens are then put into an embedding table which returns a vector associated with the token. This vector represents the word in a high dimensional space (768 dimensions) where different directions encode different types of meaning.

For example, in the sentence 'The man and the woman were husband and wife,' each word is a token. The vector difference between 'man' and 'woman' is approximately equal to the vector distance between 'husband' and 'wife,' capturing the concept of gender direction in vector space.

The size, or magnitude, of a word vector also carries important information. For example, consider the words *like* and *love*. In vector space, both might point in a similar direction associated with positive sentiment or affection, but the vector for *love* would typically have a larger magnitude than *like*. This difference in size reflects the stronger, more intense emotion conveyed by *love* compared to *like*, even though they share a similar contextual meaning.

Transformer:

The transformer is made up of many iterations of a self-attention block and a feed forward network (FFN). The self-attention block is responsible for transferring meaning between words (adding context to a word) while the FFN is responsible for producing a prediction of the next word. The final output of the transformer is in the same format as its initial input, a token for each word entered.

Self-Attention Block:

The associated vector for each token is inputted and is then used to generate three subsequent vectors for each word: the value vector, the query vector and the key vector. The attention formula is then applied to the vectors.

A value vector holds the actual information associated with each word. A value vector is generated through matrix multiplication with a learned set of weights and the token's associated vector. This vector has 768 dimensions.

A query vector represents the information a particular word is seeking in the context of the sentence. It can be thought of as 'What information from other words is relevant to this word'. It is generated through matrix multiplication with the token's associated vector and a learnt set of 'query' weights which generates a lower dimensional vector. This vector has 64 dimensions.

A key vector represents the information it offers to other words. It can be thought of as 'Here is some context from this word that I can contribute to other words' It is generated through a matrix multiplication with the token's associated vector and a learnt set of 'key' weights which generates a lower dimensional vector which can be thought of as representing 'How I affect meaning'. This vector has 64 dimensions.

Attention Formula:

The attention formula is applied to the query, key and value vector, to generate a set of weights which are used to calculate a new value vector. This process transfers context between words[11].

Equation 1

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Where Q is the Query vector, K^T is the transpose of the Key vector, d_k is the number of dimensions of K and V is the value vector.

The operation QK^T calculates the similarity between each query vector and each key vector through a dot product. This results in a matrix where each entry reflects the degree of association between a specific query and a specific key. To stabilize training, these similarity scores are then scaled down by dividing by the square root of the number of dimensions in the key vector. Following this scaling, the SoftMax function is applied, transforming these raw similarity scores into a probability distribution. These probabilities, representing the attention weights, indicate how much each key vector

should be attended to for each query vector. Finally, these attention weights are used to compute a weighted sum of the value vectors. This weighted sum produces the output of the attention mechanism, effectively highlighting combining the information from the values that are most relevant to each query based on their similarity to the keys.

To illustrate this process, consider the sentence “There are many atoms in a mole.” If the model focuses on the word “mole,” it compares the query vector of “mole” to the key vectors of the other words in the sentence. Words like “atoms”, and “many” might receive higher similarity scores because they help clarify the context in which “mole” is used. In this case, “mole” refers to a unit of measurement in chemistry, not an animal or a spy. The similarity scores are scaled and passed through a SoftMax function to produce attention weights. The value vectors of “atoms” and “many” are then weighted more heavily in the final output for “mole,” enabling the model to infer that the intended meaning relates to quantity in a scientific context. This demonstrates how self-attention helps models disambiguate words with multiple meanings by gathering contextual clues from surrounding words.

Feed Forward Network (FFN):

The Feed-Forward Network (FFN) processes each value vector independently. The value undergoes a linear transformation using a learnt weight matrix and a bias vector expanding it into a 3072-dimensional space to capture richer features. The Gaussian Error Linear Unit (GELU) activation function is then applied to introduce non-linearity, improving the model’s ability to learn complex patterns [12]. After activation, a second linear transformation using a second learned weight matrix and bias projects the representation back to 768 dimensions. This processed token representation is then passed to the next head or used for final classification.

Logistic Regression Classification Unit:

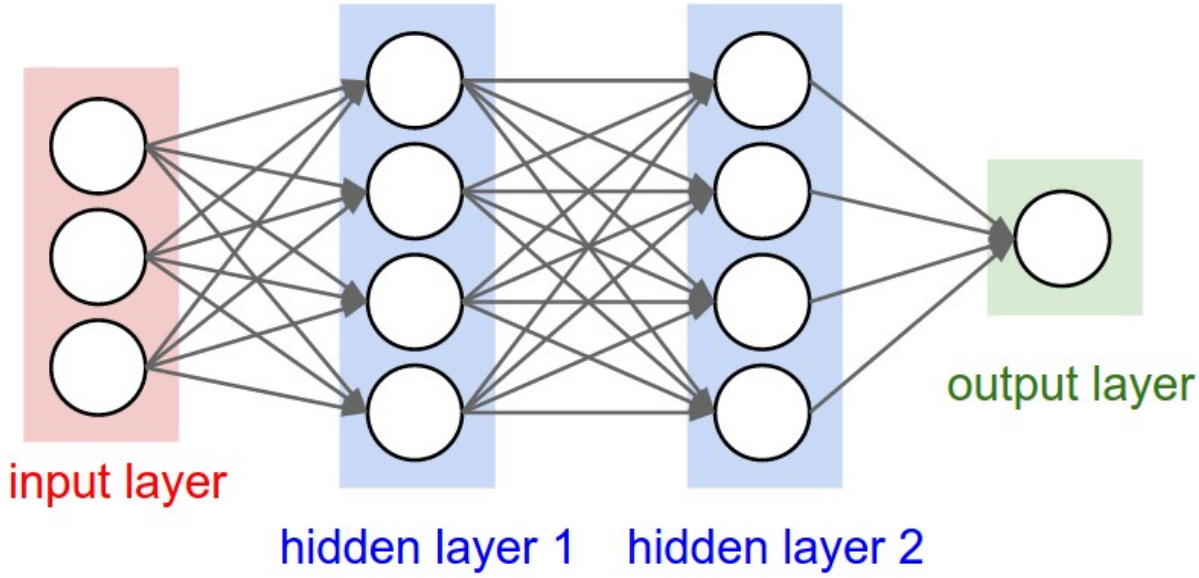
The logistic regression classification unit takes the output tokens from the transformer as inputs and a matrix multiplication is performed with 3 sets of learned weights, to give three values, which are passed through the sigmoid function to give outputs between 0 and 1 [13]. This outputs 3 sentiment logits, indicating positive negative and neutral sentiment.

Neural Network Model

The neural network takes the collected and processed data to make a prediction of the Bitcoin price. The specific inputs to the neural network can be seen in figure 1.

Neural Network Structure:

A neural network is made up of an input layer, several hidden layers and an output layer. This can be seen in figure 2.



Structure of Neural Network

Figure 2

The input layer takes the inputs and passes them through to the first hidden layer. Each hidden layer is made up of many nodes. Each node in a hidden layer multiplies each of the outputs of the previous layer each by a different learnt weight and summing these values along with a learnt bias produces a sum. Each node then passes their sum through an activation function which performs a non-linear mathematical function to produce an output for either the next hidden layer or the output layer. The output layer takes the outputs from the final hidden layer and like a single node in the hidden layers, it multiplies each of the outputs by a different learnt weight and add a learnt bias[14]. This then produces the output, in our case the predicted price of Bitcoin in 24 hours.

The weights are learnt by predicting the price with training data and then comparing that with the actual price in 24 hours and then comparing the two using a cost function. The backward propagation formula is then applied to each weight to update it, which can be seen in equation 2.

Equation 2

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial w^{(L)}}$$

Where C is the cost function, a is the output of a node, z is the sum of all the inputs multiplied by the weights with the bias added and w is the weight. The cost function used is Mean Squared Error (MSE) defined as:

$$C = (\hat{y} - y)^2$$

Where $\hat{y}$ is the predicted value of Bitcoin and y is actual price of Bitcoin.

This then gives us the formula to update the weights using formula from equation 2 and the derivative of the MSE function:

Equation 3

$$\Delta W = -\alpha \frac{1}{n} \sum_{k=0}^{n-1} a^{(L-1)} \sigma' \left(z_{(J)}^{(L)} \right) 2(a^{(L)} - y)$$

Where α is the learning rate, n is the number of samples of data, $a^{(L)}$is the output of the activation function from layer L, σ is the activation function and y is the price of Bitcoin.

This formula finds the average gradient of the cost with respect to the weight so that the model descends to a global optimum, which means that it finds the minimum MSE.

As shown, a neural network has many parameters that can be changed which, in turn, alters its performance, therefore, to find the best model this will iterate through many parameters. These are the parameters that will change, and the variables that will be iterated through:

Number of nodes in the 1st hidden layer: 128, 64, 32

Number of nodes in the 2nd hidden layer: 64, 32, 16

Number of nodes in the 3rd hidden layer: 16, 8, 4

Learning rate co-efficient: 0.03, 0.003, 0.001

Number of epochs trained: 3000, 5000, 10000

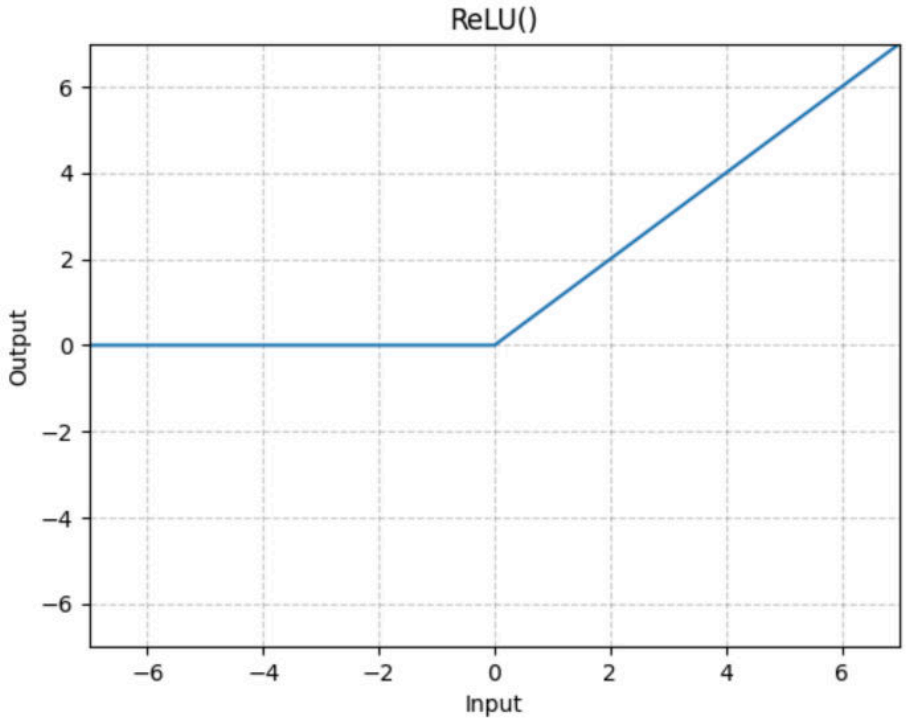
Activation Function: Relu, TanhShrink

It was decided to have 3 hidden layers for a balance, to enable the network to be able to capture the complexities of the data, while also not requiring large amounts of fine tuning and computational resources. The number of nodes in each layer effects the complexity of the model, therefore by testing different numbers of nodes in each layer it possible to ensure that the complexity of the model fits the complexity of the data. The learning rate co-efficient decides the speed of gradient descent. This means that with high learning rates convergence is quicker, but it can under fit [15]. Meanwhile low learning rates give slow convergence and can overfit [15]. Note the ‘Adam’ optimizer is used therefore the learning rate is not constant which gives better final local convergence of a model [16]. The number of epochs determines the number of times you train a model. A high number of epochs can result in overfitting, but a low number of epochs could result in underfitting therefore finding an optimal number of epochs is will greatly affect the quality of a model.

The two activation functions this paper will test are RELU and TanhShrink. RELU in recent years has shown to perform extremely well for predicting financial markets in neural networks particularly for forex trading [17]. However, one downside is that RELU can cause many dead neurons, this can be seen in the graph of figure 3, where dead neurons occur when a neuron's output becomes permanently zero from consistently negative inputs, halting its learning and limiting model performance.

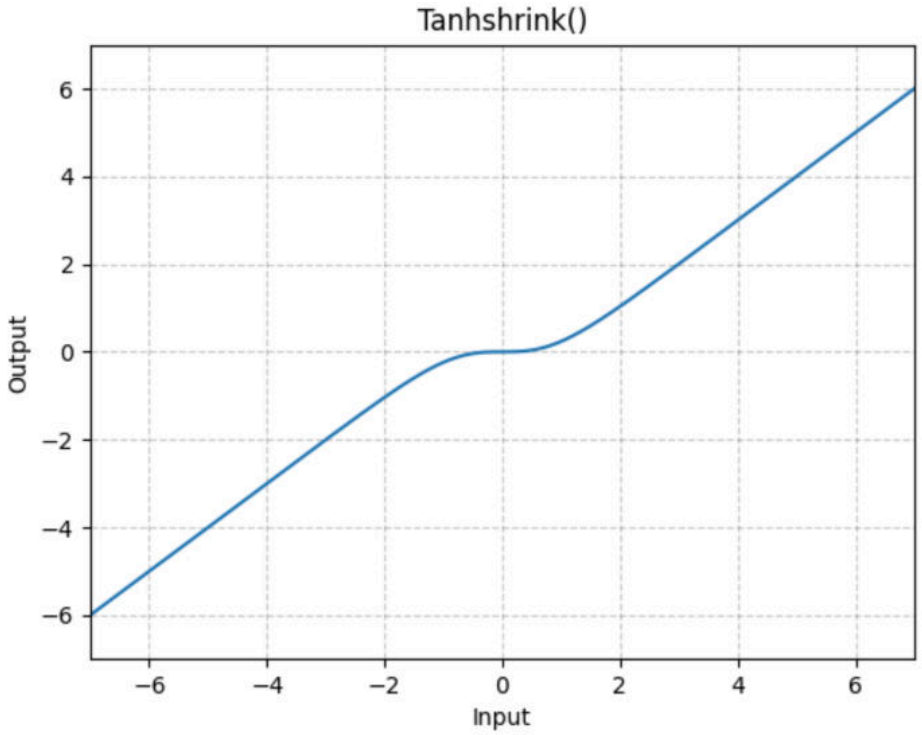
Therefore, a less well researched activation function TanhShrink will be assessed. The TanhShrink activation function was utilized to introduce non-linearities found in RELU but

reduce the number of dead nodes. The TanhShrink function is defined as f(x) = x - tanh(x), where tanh(x) is the hyperbolic tangent of x, and can be seen in figure 4 [18]. Testing every possible set of variables this gives us 486 models to test.



ReLu Graph

Figure 3



TanhShrink Graph

Figure 4

Results:

Each model was tested on 2 financial metrics, Percentage Movement correct and Ratio Made.

Percentage Correct Movement (PCM) is the proportion of price direction predictions that are correct. A prediction is considered correct if, for example, it forecasts the price to be higher in 24 hours than the current price, and this outcome occurs.

Ratio made (RM) refers to the multiplier that represents the gain or return on an investment or outcome. This can also be considered a Percentage Return (PR), calculated using the formula:

$$PR\% = (RM - 1) * 100$$

The test data was from 30th September 2024 – 31st December 2024. During this time Bitcoin went up by 48%[19], or as an RM this would be 1.480. Figure 7 shows the 486 models tested; the average RM by the models is 1.977 (97.7%) which would have returned a greater profit than Bitcoin over the test period. The model highlighted in green was the best performing model which had a ratio made of 3.393 (239%) and a percentage movement correct of 62.3%. This model used the TanhShrink activation function and had 32,16 and 4 nodes in the hidden 1st,2nd and 3rd layers respectively. This can be seen in figure 5.

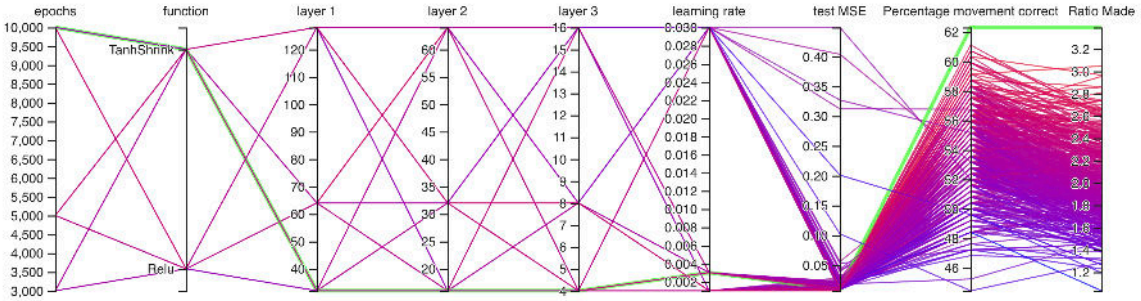


Figure 5

Analysis

The average model return of 97.7%, nearly double Bitcoin's 48.0%, indicates a relationship between Bitcoin-related sentiment on Twitter and its price movements. The best performing model returned 239%, which is extremely impressive, however, since I tested 468 models, this could also have potentially been achieved through random probability. One other factor that should be considered is that during the test period, Donald Trump became president and made comments about Bitcoin and the national reserve on Twitter. This meant that Bitcoin was heavily discussed on the platform. As a result, public sentiment affecting Bitcoin's price was captured in the dataset.

However, if, for example, a country like Russia were to buy a significant amount of Bitcoin without making it public, this could lead to a price increase that would not be reflected in the Twitter-based sentiment data, resulting in inaccurate predictions. This represents a limitation of the model used.

That said, it could also be argued that major market movements would still be indirectly reflected on Twitter through reactions from the large crypto community. This is supported by the fact that the official Bitcoin account has 7.7 million followers [20].

Improvements

There are a few possible methods to increase model performance. The biggest improvement is a larger dataset, the one used was only 2000 tweets which is very small for machine learning. This could easily reduce over fitting while also optimising for certain scenarios that may not have been in the limited dataset. A structure improvement would be to allow the neural network to be able to consider multiple tweets at once, this would allow for more data to be considered in each decision. To gain more consistency, a majority voting system of the top 100 models could be used. Where each model outputs a prediction which acts as a vote and the votes can be counted to predict a negative or positive return.

Conclusion

This study has demonstrated a correlation between sentiment on Twitter and Bitcoin price movements, supporting the hypothesis that social media sentiment can be a useful

predictor of short-term price fluctuations. By employing advanced sentiment analysis techniques through XLM-RoBERTa and integrating these insights into a neural network, I was able to develop predictive models that outperformed Bitcoin's baseline returns. The best-performing model achieved a Ratio Made (RM) of 3.393 with a Percentage Correct Movement (PCM) of 62.3%, indicating significant predictive power.

However, while the findings are promising, they also highlight potential limitations. The dataset of 2,000 tweets is relatively small for deep learning applications, and market-moving events, such as political decisions, may influence Bitcoin's price in ways not fully captured by Twitter sentiment alone. Future improvements should focus on expanding the dataset, incorporating multi-tweet analysis, and employing ensemble methods such as majority voting across top-performing models.

Despite these limitations, this approach presents an opportunity for traders to leverage social sentiment in Bitcoin price prediction, particularly given Bitcoin's unique market dynamics. While large financial firms may struggle to implement high-frequency trading due to Bitcoin's transaction times and liquidity constraints, individual traders could capitalize on sentiment-driven trends. This research underscores the importance of sentiment analysis in financial forecasting and suggests further exploration into integrating social media sentiment with traditional market indicators for enhanced predictive performance.

References:

[1] N. Gandal, J. Hamrick, T. Moore, and T. Oberman, "Price manipulation in the Bitcoin ecosystem," *J. Monet. Econ.*, vol. 95, pp. 86–96, May 2018, doi: 10.1016/j.jmoneco.2017.12.004.

[2] M. Iansiti and K. R. Lakhani, "The Truth About Blockchain," *Harvard Business Review*, Jan. 2017, [Online]. Available: https://www.researchgate.net/publication/341913793_The_Truth_About_Blockchain

[3] D. Osimo and F. Mureddu, "Research Challenge on Opinion Mining and Sentiment Analysis," *Eur. Comm.*, Sep. 2012.

[4] H. Kwak, C. Lee, H. Park, and S. Moon, "What Is Twitter, a Social Network or a News Media?," presented at the The 19th International World Wide Web Conference, Dec. 2024. doi: 10.1145/1772690.1772751.

[5] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 326–327, Sep. 1995, doi: 10.1145/212094.212114.

[6] S. Pothuganti, "Review on over-fitting and under-fitting problems in Machine Learning and solutions," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 7, pp. 3692–3695, Sep. 2018, [Online]. Available: https://www.researchgate.net/profile/Swathi-Pothuganti/publication/344882855_Review_on_over-fitting_and_under-fitting_problems_in_Machine_Learning_and_solutions/links/5f9696bd299bf1b53e45eba1/Review-on-over-fitting-and-under-fitting-problems-in-Machine-Learning-and-solutions.pdf

[7] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification," Oct. 26, 2020. doi: 10.48550/arXiv.2010.12421.

[8] A. Moreno-Ortiz and M. García-Gámez, “Corpus annotation and analysis of sarcasm on Twitter,” *J. Span. Assoc. Anglo-Am. Stud.*, Jun. 2022, doi: 10.28914/Atlantis-2022-44.1.11.
[9] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, Mar. 2011, doi: 10.1016/j.jocs.2010.12.007.
[10] L. Menon, *B0T1o1/Rouse*. (Mar. 26, 2025). Python. Accessed: May 09, 2025. [Online]. Available: https://github.com/B0T1o1/Rouse
[11] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Mar. 17, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
[12] M. Lee, “Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning,” *J. Math.*, vol. 2023, no. 1, p. 4229924, 2023, doi: 10.1155/2023/4229924.
[13] S. Narayan, “The generalized sigmoid activation function: Competitive supervised learning,” *Inf. Sci.*, vol. 99, no. 1, pp. 69–82, Jun. 1997, doi: 10.1016/S0020-0255(96)00200-9.
[14] M. Cilimkovic, “Neural Networks and Back Propagation Algorithm,” *Iconic Res. Eng. J.*, vol. 2018, 2015.
[15] L. Rice, E. Wong, and Z. Kolter, “Overfitting in adversarially robust deep learning,” in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 8093–8104. Accessed: Mar. 24, 2025. [Online]. Available: https://proceedings.mlr.press/v119/rice20a.html
[16] S. Bock and M. Weiß, “A Proof of Local Convergence for the Adam Optimizer,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8852239.
[17] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, “Mixture of Activation Functions With Extended Min-Max Normalization for Forex Market Prediction,” *IEEE Access*, vol. 7, pp. 183680–183691, 2019, doi: 10.1109/ACCESS.2019.2959789.
[18] “ReLU — PyTorch 2.6 documentation.” Accessed: Mar. 24, 2025. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html
[19] “Bitcoin USD (BTC-USD) stock price, news, quote and history,” Yahoo Finance. Accessed: Mar. 26, 2025. [Online]. Available: https://uk.finance.yahoo.com/quote/BTC-USD/
[20] “(2) Bitcoin (@Bitcoin) / X,” X (formerly Twitter). Accessed: Mar. 26, 2025. [Online]. Available: https://x.com/bitcoin

All Data and code can be accessed here in the repository:
https://github.com/B0T1o1/Rouse