Forecasting and Segmentation of Bitcoin Price

Saadi Monib
FH Kaernten
edusaamon001@fh-kaernten.at

***Abstract*—Bitcoin's complex and volatile behavior poses a significant challenge for both short-term forecasting and broader market structure identification. This paper proposes a dual-task framework that segments the Bitcoin time series into discrete market states (e.g., tight/broad bull or bear channels, trading ranges) while simultaneously forecasting future price movements. Historical hourly data of BTC-USD is retrieved from the Coinbase exchange. A simple yet effective segmentation method is introduced by leveraging the average true range (ATR), log-price slope, and ratio thresholds to classify market regimes. For forecasting, the percentage change in key price points (open, high, low, close) over the next 24 hours serves as the primary prediction target. Multiple deep learning architectures—ranging from baseline LSTM and CNN-LSTM to Bahdanau attention-based LSTM—are trained to forecast the 24-hour horizon based on a 48-hour look-back window of standardized price and technical indicators. Furthermore, a segmentation model employing bidirectional LSTM is developed to classify the identified market states. Experimental results highlight that while CNN-LSTM and Bi-LSTM approaches deliver marginal performance improvements but capturing the inherent randomness and heavy-tailed distribution of Bitcoin price movements remains challenging.**

Keywords—Bitcoin Price, LSTM, Market Structure

I. Introduction

Bitcoin have revolutionized the financial landscape by enabling decentralized and secure transactions without the need for traditional intermediaries. Since its inception by Satoshi Nakamoto, Bitcoin has attracted significant attention from investors and speculators due to its potential for high returns. Bitcoin's price is volatile, with frequent spikes and sharp declines. For example, between 5th November 2024 and 11th November 2024, Bitcoin's price surged from $67,482 to $89,544 leading to a 32.69% change, followed by a decline to from $108,356 to $91,544 leading to a -15.52% change. Comparing these strong volatile moves to traditional financial instruments such as the S&P 500 or gold shows why there is critical need for accurate prediction models to aid traders and investors in making informed decisions.

Traditional time series forecasting methods, including Autoregressive (AR), Moving Average (MA), and Autoregressive Integrated Moving Average (ARIMA) models, have been widely applied to financial data. However, these models often fall short in capturing the complex, non-linear patterns which are often random patterns inherent in cryptocurrency markets. Recent advancements in deep learning, particularly with Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, have shown improvement in prediction accuracy. The unique characteristic of cryptocurrency markets poses a challenge because of lack of seasonality and extreme volatility.

This study proposes a dual-task deep learning framework that simultaneously segments Bitcoin's time series data into distinct market states and forecasts future price movements. By integrating segmentation and forecasting tasks, market dynamics can be better interpreted by the traders. We employ various deep learning architectures, including baseline LSTM, CNN-LSTM, and Bahdanau attention-based LSTM models, to forecast percentage changes in key price indicators over a 24-hour horizon based on a 48-hour look-back window. Additionally, we develop a bidirectional LSTM model for market state classification.

II. Literature Review

Bitcoin's unprecedented growth and extreme volatility have spurred extensive research into accurate price prediction methodologies. Traditional time series models such as Autoregressive Integrated Moving Average (ARIMA) have been widely utilized for financial forecasting **[1]**. However, these models often struggle to capture the non-linear and highly volatile nature of cryptocurrency markets. This limitation is addressed by proposing a novel forecasting framework that integrates Long Short-Term Memory (LSTM) networks with an AR (2) model in the literature **[1]**. Their approach demonstrates significant improvements in prediction accuracy metrics, including Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE), compared to conventional LSTM models. This study highlights the potential of hybrid models for forecasting and segmentation.

Additional research has explored various machine learning and deep learning approaches for Bitcoin price prediction **[2]**. Multiple methodologies, including Support Vector Machines (SVM), Bayesian Neural Networks, and Ensemble Neural Networks, noting that while these models offer improvements over traditional statistical methods, LSTM and Recurrent Neural Networks (RNN) consistently outperform them in capturing temporal dependencies and complex patterns in Bitcoin's price movements **[2]**.

III. Methodology

We started by collecting hourly BTC-USD price data from the Coinbase exchange. Unlike other studies that focus on predicting the exact opening or closing prices, our research predicts the percentage changes in these prices. This makes our model more flexible across different price levels and helps address issues with data that isn't stable over time. Instead of using min-max scaling, which is not feasible with ever increasing price of Bitcoin, we chose to work with percentage changes. This approach is more useful for traders because it helps them understand how much the market is likely to move. During preprocessing, we divided the data into different market states using Average True Range (ATR) and slope thresholds, calculated the percentage changes, added technical indicators, and standardized the features. We then developed three different deep learning models—baseline LSTM, CNN-LSTM, and attention-based LSTM—to capture the time-based patterns and complex behaviors in the volatile Bitcoin market. Additionally, we used both regular and bidirectional LSTM models to handle the segmentation tasks. The next sections will provide more details about the dataset, how we prepared the data, and the specific model architectures we used in this study.

A. Dataset

This study utilizes hourly Bitcoin (BTC-USD) price data sourced from the Coinbase exchange, spanning from 23rd July 2015 to 26th January 2024. Post pre-processing 82,944 records are available comprising of the Open, High, Low, Close, and Volume metrics. Figure 1 presents the trend of the raw BTC-USD price data over the specified period, highlighting the inherent volatility and dynamic nature of cryptocurrency markets.

Figure 1: Raw BTC-USD Price Data

B. Preprocessing

The preprocessing phase involves several critical steps to prepare the data for effective model training. Initially, the dataset is segmented into distinct market states—Tight Bull Channel, Broad Bull Channel, Tight Bear Channel, Broad Bear Channel, and Trading Range—based on calculated Average True Range (ATR) and slope thresholds. Mathematically, the ATR for a segment is computed using the following formula:

$$\mathrm{ATR} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{TR_i}$$

where TR_i(True Range) is defined as:

$$TR_i = \max\left(High_i - Low_{i,} \, |High_i - \, Close_{i-1}|, |Low_i - Close_{i-1}|\right)$$

Here, High, Low, and Close represent the high, low, and closing prices at time i, respectively. The ATR provides a measure of market volatility by averaging the true ranges over a specified window size. To classify each segment, the slope of the logarithm of typical prices is calculated to determine the trend direction. The typical price P is given by:

$$P_t = \frac{High_t + \, Close_t + Low_t}{3}$$

The slope m of log-transformed typical prices over a segment is computed as:

$$m = \frac{\log(P_{end}) - \log(P_{start})}{N - 1}$$

where $\log(P_{end})$ and $\log(P_{start})$ are the log transformed values of the typical prices at the start and end of segment. Lastly, a ratio R is calculated as:

$$R = \frac{|\log(P_{end}) - \log(P_{start})|}{\log(ATR)}$$

Based upon the slope m and ratio R, each segment is then classified as follows:

1. **Range**: If m < slope_threshold (manual value set by user and requires subjective judgment)
2. **Bull Channel:** If m > 0:
   a. Tight Bull Channel: If R > tight_ratio_threshold (manual value set by user and requires subjective judgment)
   b. Broad Bull Channel: If R < tight_ratio_threshold
3. **Bear Channel:** If m < 0:
   a. Tight and broad channel logic applies similar to bull channels

This segmentation process ensures that each segment is accurately categorized based on its volatility and trend strength. A single segment is allocated to the whole day i.e 24 hours or 24 rows. This way, a single sample for the training is composed by binning the data into a bin of 24 rows. This paper emphasizes forecasting percentage changes in the price. This approach enhances the model's ability to generalize across different price scales and mitigates issues related to non-stationarity. Percentage change features are computed for Open, High, Low, and Close prices, and additional technical indicators such as Moving Averages (MA20, MA50), Relative Strength Index (RSI), and ATR are incorporated to introduce temporal dependencies. Segmentation labels are one-hot encoded to facilitate classification tasks, and all features are standardized using the StandardScaler to ensure uniform scaling across inputs. Figures 2, 3, 4 and 5 illustrate the segmentation results, actual closing price, percentage change of closing price and the distribution of the percentage changes, respectively, showing the distinct patterns captured by the preprocessing pipeline.

Figure 2: Segmentation of Time Series Example with Range and Tight Bear Channel

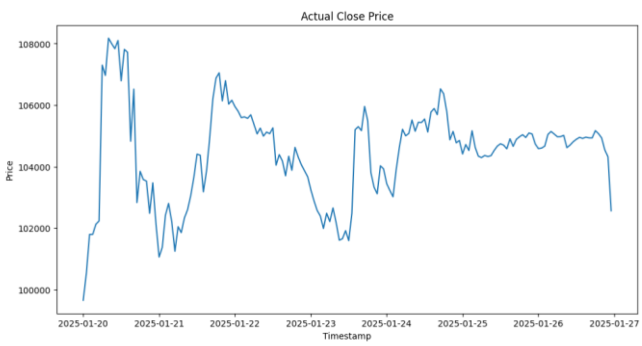

Figure 3: Actual Close Price

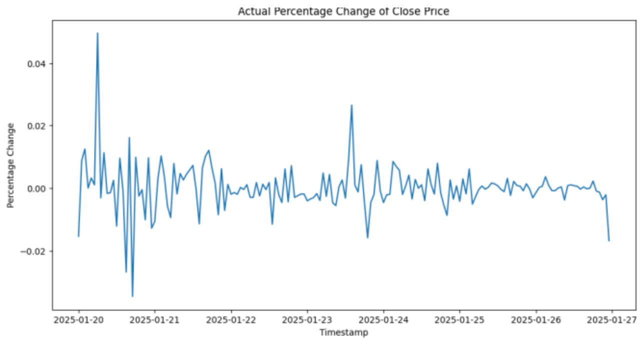

Figure 4: Percentage Change of Price along time axis

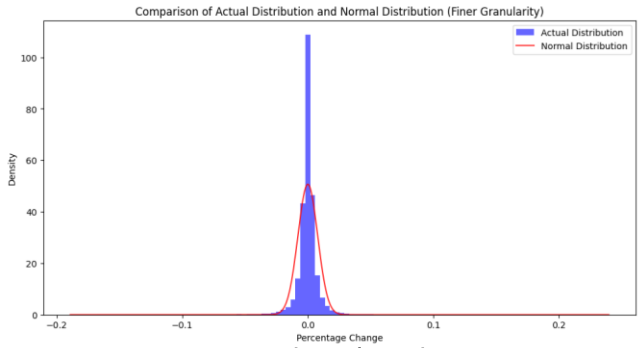

Figure 5: Distribution of Price Changes

C. *Model Architecture*

The forecasting framework comprises three deep learning architectures: a baseline Long Short-Term Memory (LSTM) model, a Convolutional Neural Network combined with LSTM (CNN-LSTM) model, and an attention-based LSTM model utilizing Bahdanau Attention. Additionally, for the segmentation task, both a baseline LSTM model and a bidirectional LSTM model are developed to classify the segmented market states based on the input features. All models are trained using the Adam optimizer and evaluated using loss functions and metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 for forecasting task and accuracy, precision, recall, and F1 score for the segmentation task.

The baseline LSTM model consists of three stacked LSTM layers with 32, 16, and 8 units respectively. These layers capture the temporal dependencies in the input data over a 48-hour look-back window. Following the LSTM layers, dense layers are employed to output the predicted percentage changes over a 24-hour horizon. This model serves as a fundamental benchmark to assess the effectiveness of more complex architectures.

The CNN-LSTM model integrates two Conv1D layers to capture local temporal patterns in the data before passing the output to four LSTM layers. The convolutional layers act as feature extractors, identifying relevant patterns that the LSTM layers can further process to learn long-term dependencies. This hybrid architecture aims to enhance the model's ability to learn complex and non-linear relationships in the highly volatile Bitcoin market.

The attention-based LSTM model incorporates a Bahdanau Attention mechanism to focus on relevant time steps in the input sequence, potentially improving forecast accuracy by dynamically weighting the importance of different segments of the input data. Attention mechanism allows the model to prioritize certain parts of the input data that are more influential in making accurate predictions.

The attention mechanism is defined by the following equations:

$$score(h_s, h_t) = v^T.\tanh(W_e.h_s + W_d.h_t)$$

where h_s is hidden state at time step s, h_t is the query or final hidden state of LSTM while the W_e *and* W_d are weight matrices. The he attention weights are computed using a softmax function applied to the scores, and the context vector is obtained as a weighted sum of the encoder outputs:

$$a_{t,s} = \frac{\exp(score(h_s, h_t))}{\sum_{k=1}^{T}\exp(score(h_s, h_t))}$$

And the final context is calculated as:

$$context_t = \sum_{s=1}^{T} a_{t,s}\, h_s$$

IV. EXPERIMENT

The experimental framework begins with the division of the preprocessed dataset into training, validation, and testing subsets. Specifically, 80% of the data is allocated for training, 10% for validation, and the remaining 10% for testing. Table 1 summarizes the distribution of data across these subsets, detailing the number of samples allocated to each phase of the experiment and the count of instances in each class.

Dataset	***X_Forecast***	***Y_Forceast***	***XSegment***	***Y_segment***
Training	(2763,48,13)	(2763,24,4)	(2763,24,13)	(2763,4)
Validation	(345,48,13)	(345,24,4)	(345,24,13)	(345,4)
Testing	(345,48,13)	(345,24,4)	(345,24,13)	(345,4)

Class	***Total Number of Row Instances***
Broad Bull Channel	0
Tight Bull Channel	17592
Range	51000
Tight Bear Channel	14304
Broad Bear Channel	48

Table 1: Data Split Overview and number of class labels

As the granularity of the dataset is hourly, and the binning of the segmented classes is done only for 24 hours, this is why we see very few instances for the broad channels. Broad bear/bull channels develop over a long time frame. For the tight channels, we see a greater number. The ever increasing price of bitcoin is also highlighted by greater number of bull channels compared to the bear channels. Finally, the highest instances are for the range which also supports the subjective opinion of traders that markets spend the majority of time in a range.

Subsequently, three distinct forecasting models—Baseline LSTM, CNN-LSTM, and Attention-Based LSTM—are trained using the training dataset. Each model is configured with specific architectures tailored to capture temporal dependencies and complex patterns inherent in Bitcoin price movements. The models are trained for up to 100 epochs with a batch size of 32, employing early stopping mechanisms to prevent overfitting by monitoring the validation loss. Adam is choosen as the choice of optimizer for all the models. For forecasting, the choice of loss fucntion is Hubor Loss because of the long tailed distribution we have for the dataset. The performance of each model is evaluated on the test set using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R². Additionally, segmentation models employing both Baseline and Bidirectional LSTM architectures are also used.

RESULTS AND DISCUSSION

Figure 6 (a combination of two figures) illustrates the training, validation loss curves and the forecasted vs actual for the best performing model.

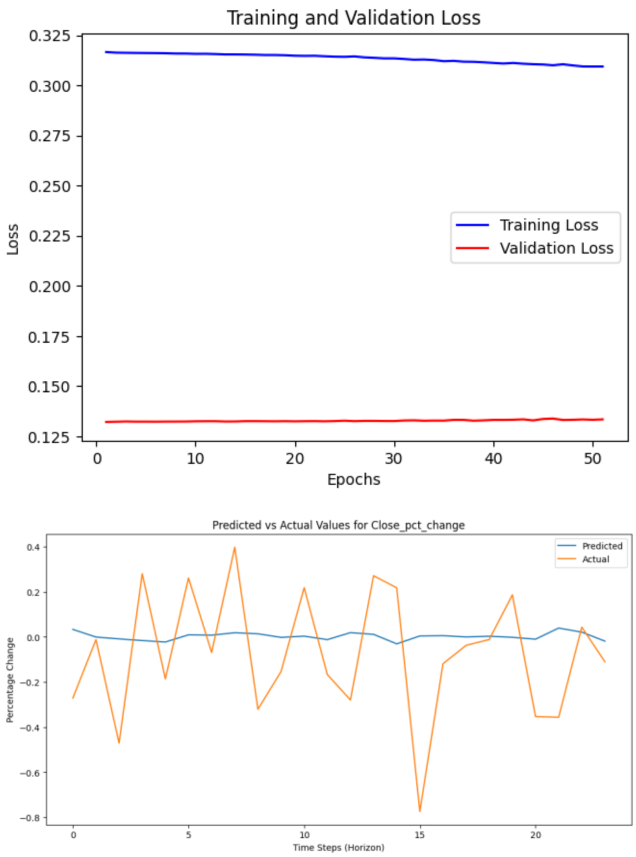

Figure 6: Training and Validation Loss Curves for CNN-LSTM Model

The evaluation results reveal that the CNN-LSTM model outperforms the Baseline and Attention LSTM models in terms of MAE, MSE, and R. Additionally, segmentation models results are also presented in Table 2. The Bidirectional LSTM segmentation model exhibits enhanced accuracy and F1 scores compared to its Baseline counterpart, underscoring the effectiveness of incorporating bidirectional layers in capturing market state dynamics. Scatter plot in Figure 7 further visualize the correlation between actual and predicted percentage changes.

Metric	***Base LSTM***	***CNN-LSTM***	***Attention LSTM***
Loss (Hubor Loss)	0.1952	0.1951	0.1957
MAE	0.4548	0.4545	0.4560
MSE	0.5069	0.5066	0.5080
R2	-0..001662	0.00000	-0.004293

Metric	***Base LSTM***	***Bi-Directional LSTM***
Loss (Cross Entropy Categorical)	0.2214	0.1600
Accuracy	0.9104	0.9508
Precision	0.6697	0.7192
Recall	0..6756	0.7017
F1	0.6687	0.7063

Metric	***CNN-LSTM Input to Bidirectional LSTM***
Loss (Cross Entropy Categorical)	2.332
Accuracy	0.6271
Precision	0.1562
Recall	0.2500
F1	0.1911

Table 2: Forecasting and Segmentation Model Performance Metrics

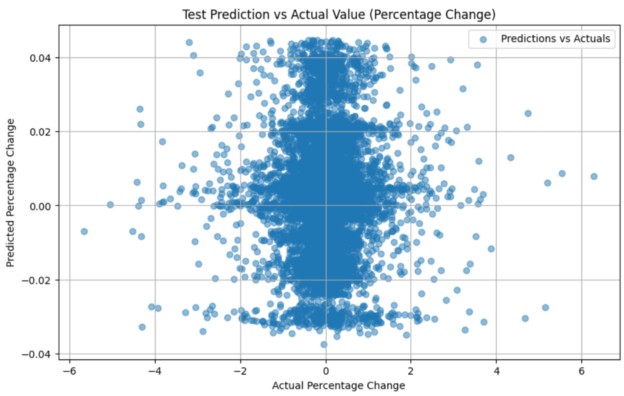

Figure 7: Actual vs. Predicted Percentage Change Scatter Plot for CNN-LSTM

The R-squarred metric shows that the model are not able learn the required pattern for the percentage changes for the forecasting task. Unlike predicting absolute prices, forecasting percentage changes presents a higher level of difficulty due to the inherently more random and volatile nature of percentage fluctuations, as illustrated in Figure 8.

This randomness poses significant challenges for the models to capture meaningful patterns, making accurate predictions more complex compared to forecasting absolute price values. Nonetheless, predicting percentage changes is more valuable for traders as it provides insights into the relative movement of prices, enabling better risk management and decision-making strategies. The next step in this direction could be using more complex model such as a transformer, and incorporating more technical indicators in addtion to the moving averages and RSI we have used. Furthermore, such forecasts should only serve as a directional bias for the trader. To actually make a product out of such models is difficult given the stochastic nature of the markets especially bitcoin. Techniques from risk management, stochastic optimzation can be explored for developing a model which can actually make a profit in the market.

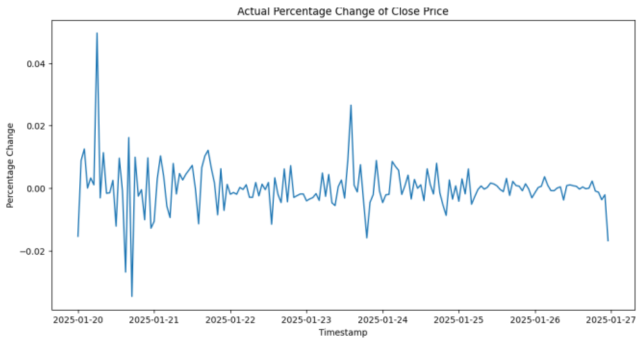

Figure 8: Actual Percentage Change of Close Price

References

[1] C. H. Wu, R. S. Lu, Y. F. Ma, and C. C. Lu, "A New Forecasting Framework for Bitcoin Price with LSTM," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Atlanta, GA, USA, 2018.

[2] F. Ferdiansyah, S. H. Othman, R. Z. R. Radzi, D. Stiawan, Y. Sazaki, and U. Ependi, "A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market," in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*