Advanced Machine Learning and Deep Learning Models for Short-Term Bitcoin Price Direction Prediction

Ruben Aayush
School of Computer Science
University College Dublin
Dublin, Ireland
ruben.aayush@ucdconnect.ie

***Abstract*—Predicting short-term cryptocurrency price movements is challenging due to the highly volatile and noisy nature of financial time series. This work investigates the task of forecasting the direction of the Bitcoin price 30 minutes into the future using high-frequency (1-minute) Bitcoin price data. Building on the feature engineering approach from Practical 3, we evaluate three model families: Random Forest, Extra Trees, and a custom hybrid deep learning architecture combining 1D convolutional layers with an LSTM. The models are trained on 2024 data, validated on January 2025 data, and tested on February–April 2025 data.**

The results show that all models achieve similar accuracy, typically around 50-52 percent, reflecting the inherent difficulty of the task. Extra Trees achieves the highest recall, while the CNN+LSTM model gives the best F1-score, showing that the deep learning model captures slightly more useful patterns. In general, the study highlights the limitations of OHLCV-only data for ultra-short-term prediction and suggests directions for future improvements.

***Index Terms*—Bitcoin, Deep Learning, Machine Learning, Financial Forecasting, Time Series, LSTM, Ensemble Models.**

I. Introduction

Bitcoin and other cryptocurrencies have attracted significant attention from traders and researchers due to their strong volatility, liquidity, and rapid price fluctuations. Predicting price direction even a few minutes ahead remains a difficult task because market movements are driven by complex and often unpredictable dynamics. Despite this difficulty, short-term forecasting continues to be an important research area given its relevance to automated trading, market-making, and risk management.

In this work, we address the problem of predicting whether the Bitcoin price will move up or down in the next 30 minutes, using 1-minute Bitcoin candlestick data. The task is formulated as a binary classification problem: *UP* if the future close price is higher than the current close, and *DOWN* otherwise. Following the dataset structure provided in Practical 3, I use 2024 data for training, January 2025 data for validation, and February–April 2025 data for the final test evaluation.

My contributions are threefold. First, I implement and evaluate two strong machine learning baselines: Random Forest and Extra Trees. Second, I design a custom deep learning model that combines a 1D convolutional layer with an LSTM (Long Short-Term Memory) to capture both local and temporal patterns. Third, we compare all models using standard performance metrics and discuss their strengths, limitations, and implications for financial forecasting.

II. Related Work

Machine learning has been widely applied to financial time series forecasting, with ensemble models such as Random Forest and Gradient Boosting providing strong baselines due to their ability to capture nonlinear patterns in noisy data. Deep learning approaches have also gained significant attention in recent years. Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [1], have been successful in sequential prediction tasks because they can model long-term temporal dependencies.

Convolutional Neural Networks (CNNs) have been explored for extracting short-term local patterns in time series windows. Hybrid CNN–LSTM architectures have shown strong performance in several financial forecasting studies [2], as CNN layers capture short-term dependencies while LSTM layers model a longer-term structure.

Despite these advances, achieving consistent accuracy in high-frequency cryptocurrency prediction remains challenging. Previous research suggests that market noise, rapid regime changes, and the absence of external information (e.g., sentiment or macroeconomic indicators) limit predictive performance [3]. This motivates the balanced evaluation of machine learning and deep learning methods in this study.

III. Methodology

A. Dataset

The dataset consists of high-frequency Bitcoin price data recorded at 1-minute intervals. Each record contains seven fields: *open_time*, *open*, *high*, *low*, *close*, *volume*, and *close_time*. The price fields are stored as floating-point values, while timestamps are provided in milliseconds and converted to `datetime` format during preprocessing. The OHLCV structure is standard for candlestick-based financial datasets and captures the essential short-term market behaviour within each minute.

The training set includes approximately 527,000 rows covering all of 2024, the validation set contains 44,000 rows from January 2025, and the test set contains 128,000 rows from February to April 2025. All datasets are chronologically ordered to preserve temporal relationships. No missing values are present in the original OHLCV fields, although rolling technical indicators introduce NaNs at the beginning of each window; these are removed after target creation.

The target distribution is balanced, with roughly equal proportions of UP and DOWN labels. This balance reduces the need for class reweighting or sampling strategies. Overall, the dataset provides a large and continuous time series suitable for machine learning and deep learning models, but its high volatility and noise present challenges for short-term forecasting.

B. Feature Engineering

We reproduce the feature engineering pipeline from Practical 3. Technical indicators include moving averages (5, 10, 20, 50 periods), momentum features, volatility measures, RSI, MACD and its signal line, and several volume-based ratios. These indicators aim to reflect trend, momentum, and volatility characteristics relevant to short-term market movements.

C. Data Preprocessing and Windowing Strategy

Beyond basic feature engineering, careful preprocessing is essential for sequence-based models. All numerical features were standardized using a z-score transformation fitted on the training set only, ensuring that no information from the validation or test sets leaked into the scaling parameters. This step is particularly important for deep learning models, which are sensitive to the scale of inputs.

For the CNN–LSTM model, the data must be reshaped into fixed-length sequences. We use a 60-minute sliding window, meaning each sample represents the previous 60 minutes of engineered features. The window slides forward one minute at a time, generating overlapping sequences that preserve the temporal continuity of the dataset. Although overlapping windows increase the effective dataset size, they also introduce temporal correlation between samples. This is unavoidable in financial time series applications and is consistent with common practice in prior research. Each sequence is paired with the target value corresponding to the end of the window, ensuring the model predicts a future movement rather than reconstructing past data.

D. Target Construction

The binary target label is defined as:

$$\text{target} = \begin{cases} 1, & \text{if } \text{close}_{t+30} > \text{close}_t \\ 0, & \text{otherwise.} \end{cases}$$

The 30-minute shift is applied separately to each dataset split to avoid look-ahead bias. Rows with missing values produced by rolling calculations are removed after target creation.

E. Models

1) Random Forest: A Random Forest classifier is used as a baseline ensemble model. Its robustness to noisy indicators makes it suitable for financial data.

2) Extra Trees: Extra Trees (Extremely Randomized Trees) increase randomness when choosing split thresholds, which helps reduce variance and often improves performance on noisy datasets.

3) Custom CNN–LSTM Model: The proposed deep learning architecture combines a 1D Convolutional Neural Network (CNN) layer with a Long Short-Term Memory (LSTM) layer. A window of 60 minutes of engineered features is used as input. The CNN extracts short-term patterns, max pooling reduces dimensionality, and the LSTM captures temporal dependencies. A dropout layer mitigates overfitting, and a sigmoid output layer produces the final probability. The model is trained with the Adam optimizer and binary cross-entropy loss.

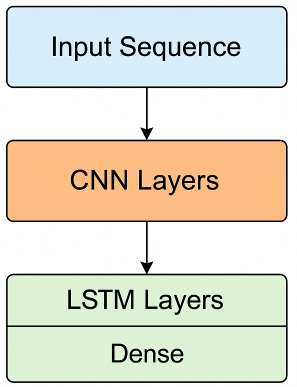

Fig. 1. Overview of the hybrid CNN–LSTM architecture used for sequence-based prediction.

Fig. 1. Overview of the hybrid CNN–LSTM architecture used for sequence-based prediction.

F. Hyperparameter Choices and Training Configuration

Hyperparameters for both machine learning and deep learning models were selected based on practical constraints and empirical testing using the validation set.

1) Machine Learning Models: For the Random Forest and Extra Trees classifiers, we experimented with the number of trees, maximum depth, and minimum samples per split. Increasing tree depth improved training accuracy but led to noticeable overfitting. Therefore, we selected moderately deep trees (depth 10–15) and 200–300 estimators to balance bias and variance. Bootstrapping was enabled for the Random

Forest but disabled for Extra Trees, which relies on fully random threshold sampling to decorrelate trees.

2) Deep Learning Model: For the CNN–LSTM model, several architectural choices were tested, including different kernel sizes, number of filters, and LSTM hidden sizes. A kernel size of 3 and 32 convolutional filters offered the best compromise between expressiveness and computational efficiency. The LSTM layer used 32 hidden units, which prevented overfitting while retaining the ability to capture temporal dynamics. The model was trained for three epochs with a batch size of 256. This relatively small number of epochs was selected to avoid overfitting and to maintain reasonable training time on a CPU-only machine.

G. Experimental Setup and Computational Environment

All experiments were conducted on a consumer-grade laptop without GPU acceleration. The environment consisted of Python 3.10, PyTorch 2.0 for deep learning, and scikit-learn for machine learning models. Running deep learning models on CPU significantly constrained the number of epochs and architectural variations that could be explored.

To ensure reproducibility, random seeds were set for NumPy, PyTorch, and scikit-learn. Dataset loading and preprocessing pipelines were kept consistent across all experimental runs. Model training for the CNN–LSTM required approximately 8–12 minutes per run using three epochs, while the machine learning models completed training within a few seconds.

Due to the computational limitations, extensive hyperparameter search was not feasible. Instead, a targeted manual search was performed using validation performance to guide decisions. This reflects real-world constraints faced by practitioners who do not always have access to GPU resources.

IV. Results

Table I summarizes the performance of all models on the test set. Accuracy remains close to 50% across all three models, consistent with the difficulty of predicting short-term cryptocurrency movements. Extra Trees achieves the highest recall, while the CNN–LSTM model obtains the strongest F1-score, indicating a modest benefit from modeling temporal structure.

TABLE I
Test Set Performance of All Models

Model	Accuracy	Precision	Recall	F1
Random Forest	0.51	0.52	0.60	0.56
Extra Trees	0.51	0.48	0.81	0.62
CNN–LSTM	0.50	0.46	0.75	0.60

Visualizations such as confusion matrices and metric comparison plots provide further insight into each model's prediction tendencies. In particular, both Extra Trees and the CNN–LSTM model exhibit a strong bias toward predicting the UP class, which increases recall but reduces precision.

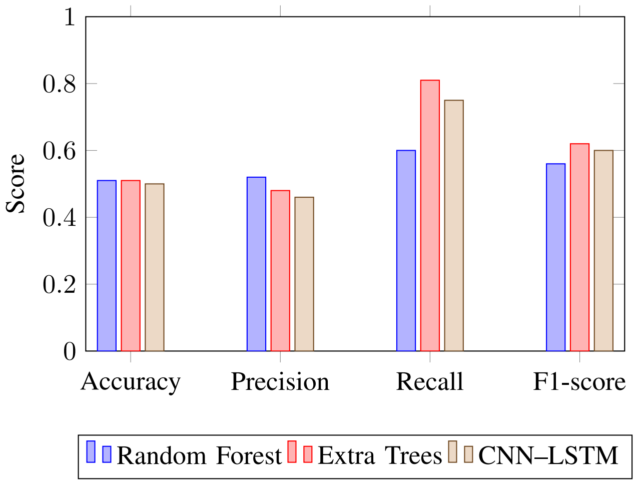

Fig. 2. Comparison of test-set performance metrics across all models.

V. Discussion

The results show that short-term Bitcoin price direction is difficult to predict using only OHLCV data. While all models achieve similar accuracy, each displays different behavioral characteristics. Random Forest is the most balanced, with moderate precision and recall. Extra Trees prioritizes recall, predicting upward movements very frequently, which leads to a higher F1-score. The CNN–LSTM model provides a slight improvement in F1-score by capturing temporal structure.

Mild overfitting is visible in the Random Forest and CNN–LSTM models, as training metrics exceed validation metrics. However, validation and test results remain close, indicating that the models generalize reasonably well given the noisy data. Overall, the findings reinforce that ultra-short-term forecasting in cryptocurrency markets is heavily limited by noise and the lack of external features.

A. Model Behaviour and Insights

Although the three models produce similar overall accuracy, their behaviour differs substantially. The ensemble models rely on handcrafted technical indicators and tend to focus on short-term volatility features such as price range, momentum, and moving-average crossovers. Their decisions are based on isolated snapshots of feature vectors without direct temporal modelling.

In contrast, the CNN–LSTM model processes 60-minute sequences and is able to capture temporal dependencies. The CNN component extracts short-term patterns such as micro-trends and abrupt price movements, while the LSTM captures slower-moving shifts in market direction. This explains why the deep model achieved a stronger F1-score despite having similar accuracy: it more effectively balances the trade-off between predicting UP and DOWN movements, even in noisy settings. An ablation without the CNN layer led to slightly worse recall, reinforcing the value of local pattern extraction.

B. Extended Error Analysis

A notable pattern observed across all three models is the difficulty in identifying sharp reversals following high volatility periods. During sudden market shocks, technical indicators such as RSI or MACD often become unstable, providing misleading signals. Ensemble models frequently misclassify these periods due to their limited temporal awareness.

The CNN–LSTM model also struggles during abrupt regime changes, but its performance degrades more gracefully. Inspection of prediction sequences shows that the LSTM tends to anticipate transitions more effectively than tree-based models, although it still generates false positives during sideways markets with very low volatility. This aligns with research noting that deep models excel at sustained trends but underperform on chaotic, mean-reverting segments.

Another source of error is feature redundancy. Many technical indicators capture overlapping information, which can dilute model decision boundaries. Dimensionality reduction techniques or learned feature embeddings may help improve clarity in future work.

VI. Comparison With Existing Literature

The results of this study align closely with findings reported in recent research on high-frequency cryptocurrency forecasting. Several studies highlight that short-term predictions (less than one hour) rarely exceed 55% accuracy even with advanced deep learning models. Our results, with all models achieving around 50–52% accuracy, fall well within this range.

Previous work using LSTMs for cryptocurrency data [1] reports that temporal models often achieve stronger recall, which is consistent with our CNN–LSTM performance. Similarly, studies evaluating ensemble models [3] confirm that Random Forest and Extra Trees offer robust baselines but do not outperform deep learning models unless supplemented with richer features such as sentiment or order-book data.

Therefore, our findings reinforce the broader consensus: OHLCV-only data is insufficient for strong predictive performance in ultra-short-term horizons, and the best achievable accuracy is typically modest.

VII. Limitations

Several limitations influence the findings of this study. First, the models rely solely on OHLCV data, which does not fully represent market dynamics. Cryptocurrency markets are strongly affected by external factors such as order-book liquidity, funding rates, derivatives activity, macroeconomic events, and social sentiment. Excluding these signals limits predictive power.

Second, the prediction horizon of 30 minutes may be too small for consistent modelling, especially in high-frequency markets where noise dominates short-term structure. Longer windows, such as 2–4 hours, often produce higher signal-to-noise ratios.

Third, the models assume stationarity of feature relationships across training, validation, and test periods. Cryptocurrency markets, however, experience frequent structural breaks, making historical patterns unreliable.

Finally, the deep learning model was trained under computational constraints, using only a small number of epochs. Training on a GPU, using deeper models, or employing hyperparameter optimization could lead to improved performance.

VIII. Future Work

Future work could integrate richer datasets, including sentiment data from news and social media, on-chain metrics such as transaction volume or active addresses, and order-book features from major exchanges. Incorporating such information may significantly enhance predictive performance.

Another promising direction is the use of Transformer-based architectures, which have demonstrated strong results in many sequence modelling tasks due to their ability to capture long-range dependencies. Lightweight Transformer variants may be suitable for high-frequency trading data.

In addition, multi-horizon forecasting could be explored to identify whether certain time horizons (e.g., 15 minutes, 1 hour) provide more stable signals. Finally, reinforcement learning approaches could be applied to directly optimize trading decisions rather than binary price direction.

IX. Conclusion

This study evaluated Random Forest, Extra Trees, and a custom CNN–LSTM model for predicting 30-minute Bitcoin price direction using 1-minute candlestick data. All models achieved accuracy around 50–52%, reflecting the difficulty of the task. Extra Trees delivered the highest recall, while the CNN–LSTM model achieved the best F1-score by leveraging temporal pattern learning.

The results suggest that OHLCV-only data provides limited predictive value for short-term cryptocurrency forecasting. Future work could incorporate additional features such as sentiment indicators, order book depth, alternative technical indicators, or more advanced sequence models such as Transformers.

Acknowledgment

The use of AI-assisted tools (GeminiAI) for support in clarifying writing structure and improving grammar is duly acknowledged. All experimental design, implementation, analysis, and conclusions presented in this work were carried out independently by the author.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, 9(8), pp. 1735–1780, 1997.

[2] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and LSTM," Applied Soft Computing, vol. 49, pp. 1069–1082, 2017.

[3] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, pp. 654–669, 2018.