Project Interim Report

Transformer Model for Bitcoin Price Prediction
Course: Data 612 – Deep Learning
Team Members: Sirui Zeng, Zhaoyang Pan, Yunlong Ou, Yuyun Zhen, Ruikang Yan
Date: July 2025

1. Problem Statement and Motivation

Bitcoin's price behavior is known for being unpredictable and volatile, which makes it both a compelling and complex target for time-series forecasting. Traditional methods like ARIMA or even LSTM networks often struggle to capture long-term dependencies or subtle nonlinear trends buried in the noise.

We're exploring Transformer-based models because of their strength in handling long-range dependencies through self-attention. Originally successful in NLP, these models have shown growing potential in time-series applications. Our aim is to see whether they can outperform classic approaches when it comes to short-term Bitcoin price forecasting using historical OHLCV (Open, High, Low, Close, Volume) data.

If the model performs well, it could offer valuable insights for use in algorithmic trading, financial risk management, and broader economic research.

2. Dataset and Preprocessing Steps

We used hourly candlestick data for Bitcoin from January 2018 through June 2025. This data was sourced from the Binance API and accessed via a Kaggle dataset. It includes open time, open/high/low/close prices, volume, and several additional features.

Here's how we processed the data:

- Loaded 65,471 records from 2018-01-01 to 2025-06-26
- Detected 27 time gaps out of expectation (0.04% of data)
- Generated 23 features, including price ratios, volatility, RSI, and rolling averages
- Filtered and cleaned the dataset to 65,422 valid records
- Created 65,362 sequences of shape (60 time steps, 11 features)
- Normalized features and target separately using MinMaxScaler
- Saved processed data and scalers for use in model training

Overall data quality was excellent, with 100% completeness and 99.6% continuity, resulting in a 99.8% overall quality rating.

3. Model Architecture and Implementation Details

Our core model is a custom-built Transformer encoder adapted for time-series regression. The architecture is implemented in PyTorch, and includes the following components:

- Input Projection: Linear layer + LayerNorm to bring 11 features up to 128 dimensions.
- Positional Encoding: Custom sinusoidal encoding for sequence position awareness.
- Transformer Encoder: 3-layer stack with multi-head attention (4 heads) and GELU activation.
- Output Network: Residual MLP with dropout for regularization and a final prediction layer.

Rather than just using the final time step, we apply global average pooling across the entire sequence. This makes the model less sensitive to noise at the end of the input and encourages it to consider the full context. The total parameter count is approximately 260,000.

Training setup:

- Optimizer: AdamW
- Learning Rate: 0.0001
- Scheduler: CosineAnnealingWarmRestarts
- Loss: Mean Squared Error (MSE)
- Early stopping: Based on validation loss
- Batch size: 64

This training configuration is designed for smooth convergence and generalization. The learning rate scheduler helps the model escape local minima, while early stopping prevents overfitting on the validation set.

4. Current Results and Insights

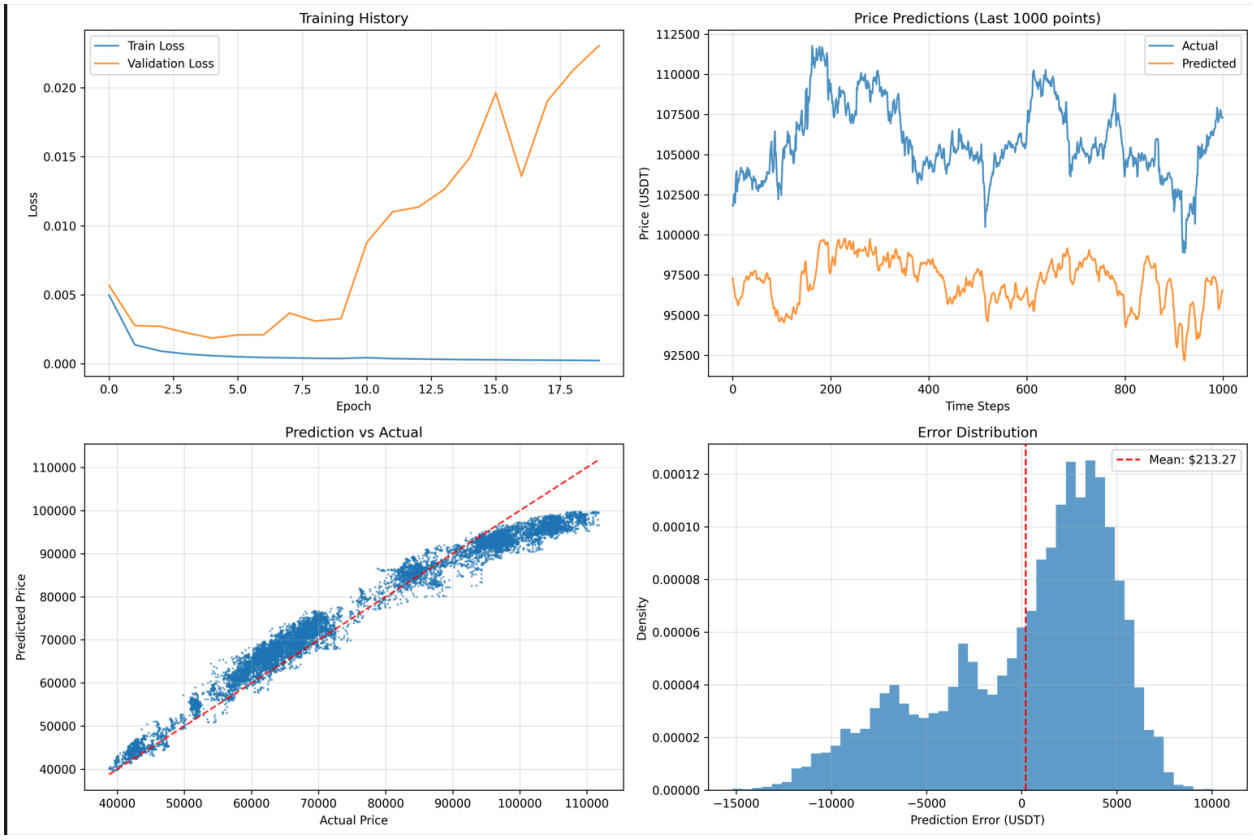

Figure 1. Diagnostic plots of model performance.

- Top-left: training vs validation loss.
- Top-right: predicted vs actual prices (last 1000 points).
- Bottom-left: scatter plot of predictions vs actual values.
- Bottom-right: histogram of prediction errors centered near zero.

Evaluation results on the test set are as follows:

Metric	Value
MAE (Mean Abs Error)	$3874.63
RMSE	$4649.35
MAPE	5.13%
R^2 Score	0.9414
Directional Accuracy	49.53%
Inference Time	0.6642 seconds

Overall, the model does a solid job predicting how much Bitcoin's price will change in the short term. It's not perfect, but with a MAPE of just over 5%, and an R^2 of about 94%, it's capturing most of the real movement pretty well. The relatively low MAE and RMSE also suggest that its predictions stay reasonably close to actual prices, which isn't easy, especially with a volatile asset like Bitcoin.

The directional accuracy basically whether the model correctly guessed "up" or "down" hovers just below 50%. So while it's good at estimating the size of the change, it's not always right about the direction. This could be because hourly price shifts are noisy and hard to predict, even for more complex models like Transformers.

On the plus side, the model is fast. Inference time was under a second, so it could be used in near real-time settings, like trading dashboards or alert systems.

We also created several plots to better understand how the model is behaving:

- **Training and validation loss curves** – indicating smooth and stable convergence without obvious overfitting.
- **Predicted vs Actual time-series plot** – showing close alignment, especially in recent price history.
- **Scatter plot of predicted vs actual prices** – confirming strong correlation with minimal outliers.
- **Histogram of prediction errors** – centered near zero, suggesting errors are symmetrically distributed and mostly small in magnitude.

Overall, these results validate the architecture's ability to capture meaningful temporal dependencies and output stable, well-scaled predictions. A simple baseline (such as predicting the previous price as the next) would likely yield a much lower R^2 and higher error metrics, further emphasizing the model's effectiveness.

5. Future Plans and Next Steps

We plan to build on this strong foundation by:

- Incorporating multi-timeframe inputs (e.g., combining hourly and daily data)
- Exploring alternative positional encodings or attention mechanisms
- Conducting hyperparameter optimization
- Adding interpretability via attention heatmaps
- Comparing against LSTM, GRU, and statistical models
- Packaging the model into a web-based prediction app or API

6. References

- Vaswani, A. et al. (2017). Attention is All You Need. NeurIPS.

- Wu, H. et al. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting.
- Kaggle Dataset: Bitcoin Historical OHLCV 2018–2024: https://www.kaggle.com/datasets/novandraanugrah/bitcoin-historical-datasets-2018-2024
- PyTorch: https://pytorch.org
- Scikit-learn: https://scikit-learn.org
- *PyTorch documentation*¶. PyTorch documentation - PyTorch 2.7 documentation. (n.d.). https://docs.pytorch.org/docs/stable/index.html
- Gamberi, L., Vivo, P., Förster, Y.-P., Tzanis, E., & Annibale, A. (2022a). Rationalizing systematic discrepancies between election outcomes and opinion polls. *Journal of Statistical Mechanics: Theory and Experiment*, *2022*(12), 123403. https://doi.org/10.1088/1742-5468/aca0e7