Class Project Proposal – Transformer Model for Bitcoin Price Prediction

Course: Data 612 Deep Learning

Team Members: Sirui Zeng, Zhaoyang Pan, Yunlong Ou, Yuyun Zhen, Ruikang Yan

Date: June 27, 2025

1. Background and Significance

Transformer models have revolutionized sequence modeling, particularly in NLP tasks, due to their ability to handle long-range dependencies using self-attention mechanisms. Recently, their success has extended into time series forecasting, offering advantages over traditional models like ARIMA and RNNs/LSTMs, which struggle with vanishing gradients or limited receptive fields.

In line with the course's emphasis on modern neural network architectures (Lecture 4), we aim to apply a Transformer-based architecture to the problem of Bitcoin price prediction. This domain is well-known for its volatility and long-term dependencies, making it an ideal test case for self-attention models. Our objective is to demonstrate how Transformers can improve performance in forecasting complex, non-linear time series data such as cryptocurrency prices.

2. Statement of the Problem

Research Question: Can a Transformer-based neural network accurately forecast short- to medium-term Bitcoin closing prices using historical OHLCV (Open, High, Low, Close, Volume) data?

Sub-objectives:

- Evaluate the Transformer's performance on multiple timeframes (15-minute, 1-hour, 4-hour, daily)
- Compare results against baseline models such as LSTM and linear regression
- Interpret the model through attention-based visualization

3. Team Member Roles

All team members will collaborate on each stage of the project, including data collection and preparation, model development, evaluation, and report writing. We will work together on coding, experimentation, and documentation to ensure consistent understanding and shared responsibility throughout the project.

4. Dataset and Data Preparation

We will use the publicly available **Bitcoin Historical Dataset (2018–2024)** from Kaggle: https://www.kaggle.com/datasets/novandraanugrah/bitcoin-historical-datasets-2018-2024

This dataset provides historical candlestick data (OHLCV) for Bitcoin (BTC/USDT), sourced from the Binance exchange via API, covering the period from January 1, 2018, to the present. It includes data in four timeframes: 15-minute, 1-hour, 4-hour, and daily. Each record includes the following fields: open time, open, high, low, close, volume, close time, quote asset volume, number of trades, taker buy volumes, and a placeholder field (ignore). The dataset is automatically updated daily through a custom script.

5. Evaluation Metrics and Validation Strategy

To evaluate the performance of the Transformer model in forecasting Bitcoin prices, we will use the following metrics:

- Mean Absolute Error (MAE) – Measures average magnitude of errors in predictions.
- Root Mean Squared Error (RMSE) – Penalizes large errors more heavily than MAE.
- Mean Absolute Percentage Error (MAPE) – Useful for comparing forecast accuracy across different scales.
- R^2 Score (Coefficient of Determination) – Indicates how well the model explains the variance in the data.

We will split the dataset into a training set (80%) and a validation set (20%) using time-based splitting to preserve temporal order. The model will be trained using early stopping based on validation loss to prevent overfitting.

6. Deep Learning Framework and Implementation Plan

We will implement the Transformer-based time series forecasting model using the PyTorch framework. Specifically, we plan to use:

- torch.nn.Transformer for the core sequence-to-sequence modeling
- torch.optim.Adam as the optimizer
- scikit-learn for baseline models (Linear Regression, MinMaxScaler)
- Matplotlib and Seaborn for visualization of prediction results and attention weights
- If time permits, we will also experiment with other libraries such as PyTorch Forecasting or tsai to leverage pre-built time series transformer components.

We will ensure proper citation and documentation if any existing implementation is adapted for our project. In the final report and presentation, we will include summary tables comparing model performances and visualizations (e.g., attention weights, error plots) to highlight key findings.