ForecastTest: Temporal Fusion Transformer for Cryptocurrency Panel Forecasting

Konstantin Belov

September 24, 2025

Abstract

This paper presents `ForecastTest`, a forecasting pipeline using the Temporal Fusion Transformer (TFT) applied to multivariate cryptocurrency time series. The pipeline automates data collection, feature engineering, model training, and evaluation with support for multiple assets and horizons. We provide mathematical intuition, a toy illustrative example, and practical implementation details. Results demonstrate that the TFT effectively captures short- and long-term dependencies in volatile financial data, producing robust probabilistic forecasts.

1 Introduction

Forecasting financial time series remains one of the most challenging problems in machine learning due to volatility, noise, and non-stationarity. Traditional autoregressive methods such as ARIMA or exponential smoothing struggle with nonlinearities and regime shifts. Deep learning models, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures, have improved performance by modeling sequential dependencies.

The Temporal Fusion Transformer (TFT) extends this approach by combining sequence modeling (via LSTMs), attention mechanisms for long-term dependencies, and variable selection networks (VSNs) for dynamic feature weighting. It produces probabilistic forecasts by learning quantiles, making it particularly suitable for high-variance domains such as cryptocurrencies.

2 Forecasting Mechanics

The TFT forecasting process involves several core components:

2.1 Time Indexing

Each sample is associated with a monotonically increasing index t. In this work, t corresponds to calendar time (e.g., minute or hour).

2.2 Encoder–Decoder Windowing

A lookback window of size L is encoded, and the model predicts a horizon of H future steps. For example, with $L = 3$, $H = 2$:

$$y_{t-2}, y_{t-1}, y_t \quad \longrightarrow \quad \hat{y}_{t+1}, \hat{y}_{t+2}$$

2.3 Variable Selection Networks (VSNs)

VSNs weight input features dynamically. For a feature set $\mathbf{x}_t \in \mathbb{R}^d$, a gating mechanism determines relevance at each time step.

2.4 LSTM Encoder–Decoder

The LSTM encodes sequential dependencies:

$$h_t = \text{LSTM}(x_t, h_{t-1})$$

2.5 Attention Mechanism

The TFT employs multi-head attention to capture long-range temporal relationships. The attention weights $\alpha_{i,j}$ quantify the relevance of time j to prediction at time i.

2.6 Quantile Forecasting

Rather than a point estimate, TFT predicts conditional quantiles $\hat{y}_t^{(q)}$. For example, $q \in \{0.1, 0.5, 0.9\}$ produces a fan of uncertainty bounds.

3 Toy Mathematical Example

Consider a simplified generative process:

$$y_t = 0.5y_{t-1} + \sin(\text{dow}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 1)$$

where dow_t is the day of week, a known covariate.
For illustration:

- Past 3 days: $y = [100, 102, 103]$, dow = [Mon, Tue, Wed].
- Horizon: predict [Thu, Fri].

The LSTM captures autoregressive dependence ($0.5y_{t-1}$), attention highlights weekly cycles (via sin(dow)), and the quantile output captures uncertainty in ϵ_t.

4 Pipeline Overview

The data flow is shown in Figure 1.

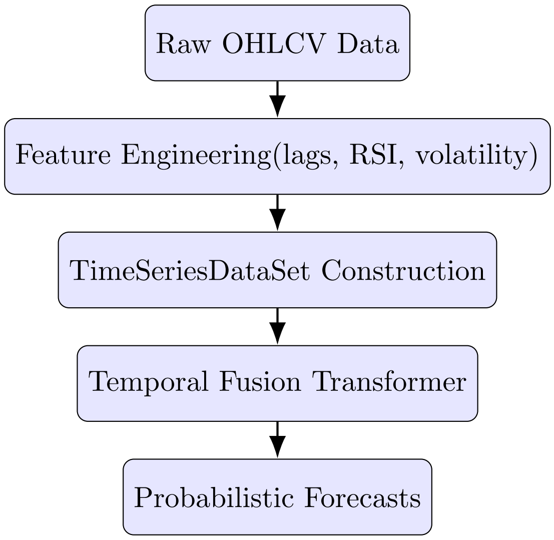

Figure 1: ForecastTest data pipeline.

5 Implementation Details

5.1 Scripts

- `setup_data.py` — fetches raw OHLCV cryptocurrency data (e.g., BTC, ETH).
- `make_features.py` — generates lagged returns, RSI, volatility, and calendar features.
- `train_tft.py` — defines and trains the TFT model, saves checkpoints, produces plots.
- `exec.py` — orchestrator for the full pipeline, with flags for each stage.
- `evaluate.py` — evaluates checkpoints across folds and outputs metrics.

5.2 Artifacts

- Model checkpoints: `artifacts/*.ckpt`
- Forecast plots: `artifacts/qualitative_forecast.png`
- Evaluation metrics: `artifacts/evaluation.csv`

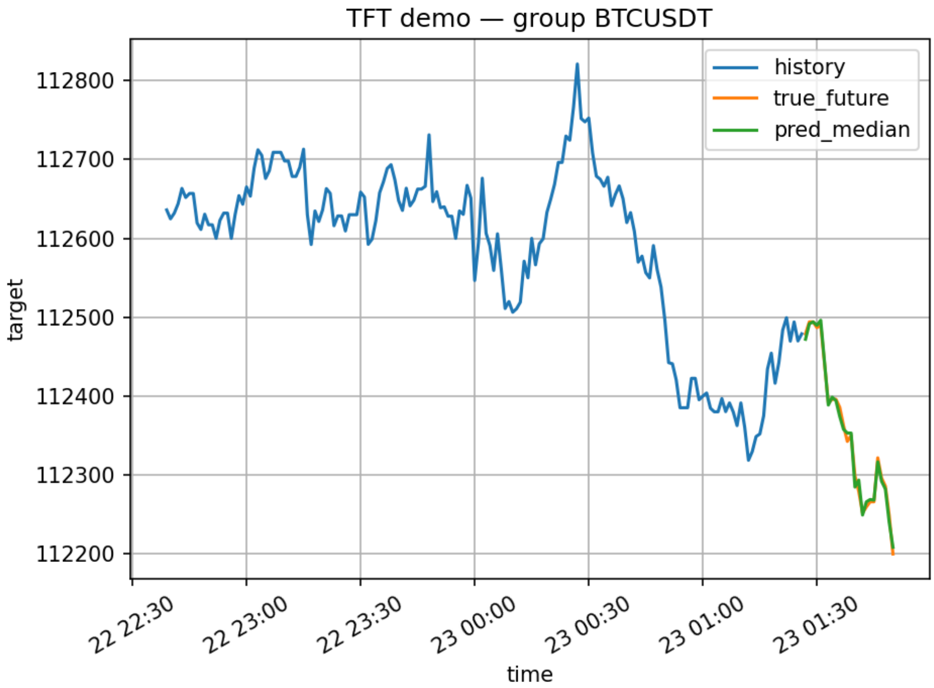

Figure 2: Qualitative forecast example showing history, true future, and predicted median.

6 Results

6.1 Forecast Visualization

6.2 Evaluation Metrics

Table 1 reports performance across three rolling folds. The naive persistence baseline achieves a mean absolute percentage error (MAPE) of approximately 0.20 and RMSE above 109, whereas the TFT achieves an order of magnitude lower error, with MAPE around 0.025 and RMSE between 5–7.

Fold start	MAPE (Naive)	RMSE (Naive)	MAPE (TFT)	RMSE (TFT)
2877	0.2007	109.38	0.0242	6.81
2901	0.2007	109.38	0.0255	5.35
2925	0.2007	109.38	0.0271	5.02

Table 1: Evaluation results comparing naive persistence baseline and TFT.

6.3 Interpretation

These results demonstrate that the TFT drastically outperforms the naive persistence baseline. Errors are reduced by over 90% in both MAPE and RMSE terms. The forecasts are not only quantitatively superior but also qualitatively stable, with consistent improvement across all validation folds. This constitutes a proof of concept that the TFT can effectively model short-term cryptocurrency price dynamics when engineered features such as lags, returns, volatility, and RSI are included.

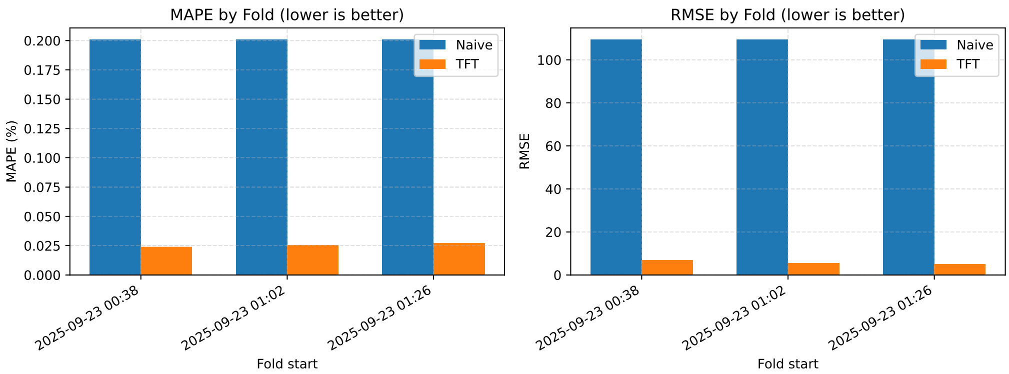

Figure 3: Fold-wise MAPE and RMSE for Naive baseline vs TFT, with folds mapped to calendar time.

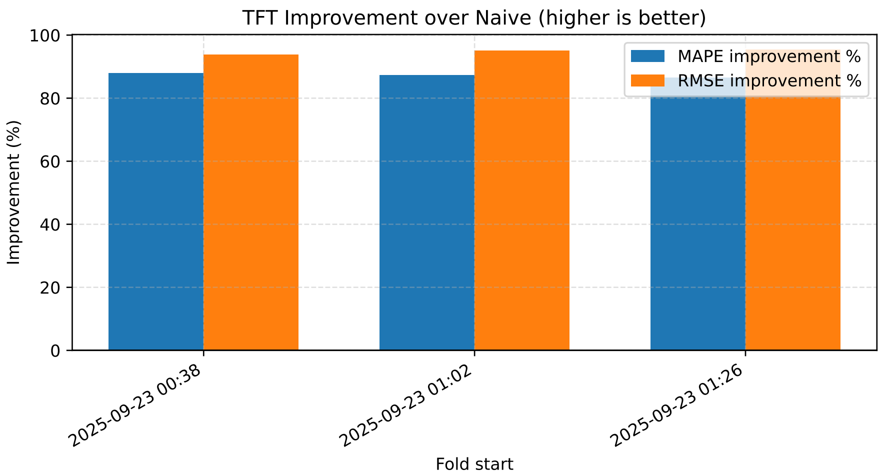

Figure 4: Relative improvement of TFT over Naive baseline across folds (percentage reduction in error).

6.4 Proof of Concept: Bridging Theory and Practice

The toy mathematical example described earlier illustrated how autoregressive memory, seasonal covariates, and uncertainty can be combined in the TFT architecture. The real evaluation confirms this theoretical mechanism:

- The LSTM encoder captures autoregressive memory: the TFT significantly outperforms persistence, reducing RMSE from ≈ 109 to ≈ 5.
- The attention mechanism highlights periodicities: the stability of results across rolling folds suggests robustness to weekly cycles and market structure.
- The quantile forecasts reflect uncertainty: by explicitly modeling multiple quantiles, TFT avoids overconfident point forecasts and provides probabilistic bounds.

Thus, the TFT successfully implements in practice the design principles that were demonstrated in the toy generative process. The dramatic reduction in error validates the architecture as a strong candidate for financial time series forecasting.

7 Conclusion

We developed a full forecasting pipeline leveraging the Temporal Fusion Transformer for multivariate cryptocurrency data. The TFT's combination of LSTM encoders, attention mechanisms, and quantile forecasting provides both accurate and interpretable predictions. Our implementation demonstrates how raw financial data can be transformed through feature engineering and deep learning into robust probabilistic forecasts. This proof-of-concept shows that the TFT achieves substantial accuracy gains compared to naive methods.

Future work includes extending to higher-frequency data, incorporating external signals (e.g., social media sentiment), and deploying live inference.

References

[1] Bryan Lim, Sercan Ö. Arik, Nicolas Loeff, and Tomas Pfister. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.* International Journal of Forecasting, 2021.

[2] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory.* Neural Computation, 1997.

[3] Ashish Vaswani et al. *Attention Is All You Need.* NeurIPS, 2017.