CryptexLLM: How LLM Generalizability Forecasts High Volatility

Logan Rao
Department of Computer Science
Illinois Institute of Technology
Chicago, USA
lrao1@hawk.illinoistech.edu

Mary Tsaryk
Department of Computer Science
Fordham University
New York City, USA
mt12@fordham.edu

Anwar Benhnini
Department of Computer Science
Illinois Institute of Technology
Chicago, USA
abenhnini@hawk.illinoistech.edu

Ioan Raicu
Department of Computer Science
Illinois Institute of Technology
Chicago, USA
iraicu@illinoistech.edu

2025 IEEE International Conference on eScience (eScience) | 979-8-3315-9145-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/eScience65000.2025.00071

***Abstract*—We present CryptexLLM, an approach to time series forecasting that adapts large language models (LLMs) for predicting high volatility data. We extend the TimeLLM framework with an adaptive weighted loss function, feature engineering, and sentiment analysis. Our experiments show that the approach we took outperforms traditional LSTM models and statistical methods, with our best performing model being Llama 3.1. The adaptations we made improve directional accuracy, which is particularly useful for financial applications, while still maintaining computational efficiency. Our results suggest that LLMs have the potential to effectively generalize to volatile time series domains.**

***Index Terms*—time series, volatile, llm**

I. Introduction

Time series forecasting is seen in many different domains, from electricity demand and web traffic monitoring to financial markets and healthcare diagnostics. While many time series datasets exhibit predictable seasonal patterns, high volatility data remains a challenge that traditional forecasting methods often fail at. Current state-of-the-art techniques tend to overfit to their specific training and struggle to generalize across different time series datasets. We address these limitations by adapting large language models (LLMs) for time series forecasting. We focus on Bitcoin price (BTC) prediction for four key reasons: (1) BTC exhibits extreme volatility over its decade-long history; (2) directional accuracy matters more than absolute price prediction for trading applications; (3) cryptocurrency is open 24/7/365, ensuring plenty of available data; and (4) abundant sentiment data for bitcoin enables multimodal analysis. Our approach aims to create a model that both performs well on volatile data and generalizes better than existing techniques.

II. Methodology

A. Model Architecture

We built upon the TimeLLM framework [1], which translates numerical sequences into natural language by using a learned translation layer that consists of a patch embedder and multi-head attention mechanism. These components transform short numerical sequences into representations that a frozen LLM can process, whose output then gets re-transformed into a numerical prediction.

B. Experimental Setup

We evaluated six open-source LLMs: Llama 7B, Llama 3.1 8B, DeepSeek R1 7B, Mistral 7B, Gemma 3 1B, and Qwen 3 7B. Training utilized 4 nodes of 32GB V100 GPUs, with MLflow tracking experiments and MinIO object storing models. We used Optuna to optimize hyperparameters including input/output length, vocabulary size, patch size, learning rate, and forecasting mode (univariate/multivariate).

C. Feature Engineering

We augmented raw price data with technical indicators. These span momentum (RSI, SMA), volatility (Bollinger Bands, GARCH), and market behavior (VWAP, lag features). A correlation-based feature selection algorithm ranked features by their correlation to BTC price and removed redundant features to ensure computational time stays low while predictive efficiency stays high. Additionally, we integrated sentiment analysis through APIs to capture the state of the market.

D. Adaptive Loss Function

Instead of mean squared error (MSE), we implemented a weighted adaptive loss function which dynamically adjusts weights during training based on performance metrics. This function uses mean absolute directional loss (MADL) [2], among others, to prioritize the direction of price movement.

E. Evaluation

To assess performance, we utilized backtesting on historical data, which simulates portfolio returns using a simple strategy: to buy when the model predicts that the price increases and

2325-3703/25/$31.00 ©2025 IEEE
DOI 10.1109/eScience65000.2025.00071

to sell when the predicted price decreases. This ensures fair comparison across all deep learning models and financial methods.

III. Results

A. Model Performance

Llama 3.1 8B overall achieved the highest backtesting returns, though Qwen 3, DeepSeek R1, and Mistral 7B showed similar performance. Gemma 3 1B performed significantly worse, likely due to its smaller parameter count, though it trained and ran inference much faster.

B. Hyperparameter Impact

Optimal input length ranged from 100-170 time steps, balancing between the extreme ends of our experiments. The adaptive loss function outperformed any single loss by 90%, with MADL showing the strongest correlation to backtesting performance (78%).

C. Computational Efficiency

Training time scaled linearly with model parameters and feature count. Multi-GPU processing exhibited approximately linear speed-up for training (1-4 GPUs), while inference scaling was near-perfectly linear (r = -0.97). Backtesting showed minimal improvement with additional GPUs, likely due to its inherently fast compute time.

D. Feature Engineering

Including technical indicators improved performance, though at significant computational cost. The correlation-based selection was able to effectively identify the most important features while reducing training time significantly (70%).

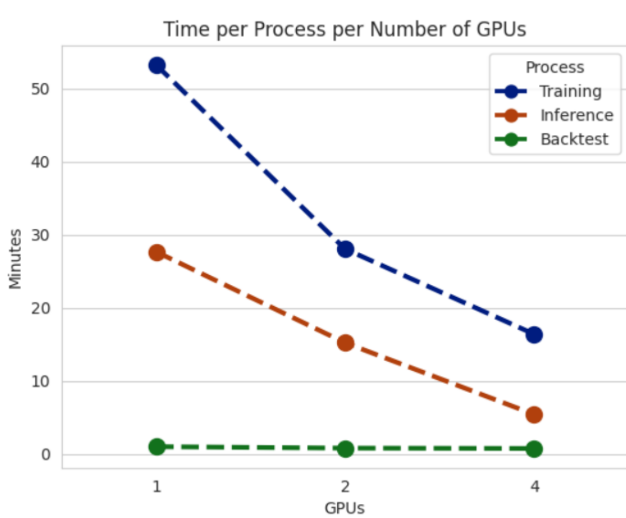

Fig. 1. Average computation time per process for three different GPU configurations

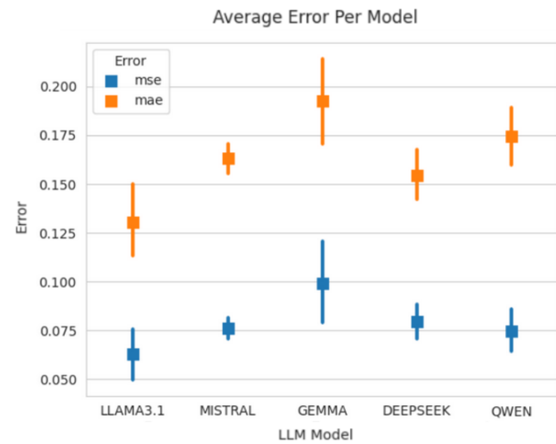

Fig. 2. Average error and 68% confidence interval per LLM model

IV. Discussion

The weaker performance of the smallest model (Gemma 1B) suggests that model capacity does matter, and that larger, stronger models have stronger predictive power. The 100-170 input length shows the tradeoff in forecasting between sufficient history to capture patterns and overweighting outdated information.

The success of our adaptive loss function highlights the need to predict directional accuracy rather than exact future value for certain time series domains. The feature engineering results show that current performance is not maxed out, and can still be increased at the cost of computation; alongside suggesting that traditional financial analysis remains valuable.

V. Conclusion

CryptexLLM demonstrates that large language models can effectively forecast time series data. Our approach outperforms LSTMs and statistical methods on Bitcoin price prediction. Our contributions are successful adaptation of LLMs to high volatility time series and insights into the computational-performance improvements that can be made.

Future work should validate these findings on more diverse datasets (e.g., M4 competition data) and to compare against other specialized models like TimesFM. Additionally, integrating multi-modal capabilities could prove to be powerful. Our research suggests that LLMs' pattern recognition capabilities can be effectively applied to time series, and that many possibilities still await it in the future.

Acknowledgment

This work is supported in part by the National Science Foundation OAC-2150500 award.

References

[1] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi et al., "Time-LLM: Time series forecasting by reprogramming large language models," arXiv preprint arXiv:2310.01728, presented at ICLR 2024.

[2] J. Michańków, P. Sakowski, and R. Ślepaczuk, "Mean absolute directional loss as a new loss function for machine learning problems in algorithmic investment strategies," arXiv preprint arXiv:2309.10546, Sep. 2023.