Uncertainty Quantification over Financial Time Series with Conformalized Risk Modelling

Rachmiel Andre Teo Ren Xiang
College of Computing and Data Science

Prof Bo An
College of Computing and Data Science

Cai Xinyu, PhD
College of Computing and Data Science

Abstract **–** Financial markets, particularly the cryptocurrency domain, exhibit high volatility, posing challenges in return prediction. Traditional machine learning models provide point estimates of returns but fail to quantify uncertainty effectively. In many financial tasks, machine learning models can suffer from the problem of overconfidence, which may lead to irreversible loss. In this study, we explore conformal prediction techniques to enhance predictive coverage, ensuring more reliable confidence intervals over each timestep. Specifically, we investigate the effectiveness of different conformal prediction methods on financial time series data, before proposing a novel method that can achieve the desired coverage guarantee with better performance. The entire process consists of 3 parts: training baseline models to predict cryptocurrency returns, using a subset to generate calibration scores, and finally using a quantile of these scores to generate intervals. Given a less than satisfactory baseline models, we generate synthetic data to meet the industry standard baseline models to reinforce the results from our original data.

Keywords – Conformal Prediction, Adaptive Conformal Inference, Volatility, Machine Learning, Uncertainty Quantification.

1 INTRODUCTION

1.1 CONFORMAL PREDICTION

The lack of uncertainty quantification of predictive models is a major barrier to the adoption of powerful machine learning methods, but at the same time, probabilistic forecasts are only valid asymptotically or upon strong assumptions on the data [1]. To reduce uncertainty, various conformal methods have been employed to quantify these risks.

Conformal prediction is best explained by a classification task. It works on a pretrained model and attempts to quantify the uncertainty of a prediction. Importantly, conformal prediction is not involved in the training stage of the pretrained model and is only applied during evaluation.

The process involves a 3-way split of the dataset, which includes train, calibration, and test data. Following the standard training of machine learning model, the calibration set produces a set of non-conformity scores, given by a score function s, e.g. Softmax. These conformity scores represent the size of the errors. Then we can compute a quantile $\hat{q}$ that approximates the $1-\alpha$ quantile of the scores. Specifically, we find the $\frac{\lceil (n+1)(1-\alpha)\rceil}{n}$ quantile. As such, the test data can now be used to generate a prediction set given by the equation

$$C(X_{test}) = \{Y: s(X_{test}, y) \leq \hat{q}\}.$$

s now takes in feature vector X_{test} and an arbitrary y value as inputs, and all outputs within the $\hat{q}$ threshold is considered as possible true values under a $1-\alpha$ guarantee. [2]

In other words, the aim is to achieve a guaranteed coverage under the chosen α, such that

$$P\big(Y_{test} \in C(X_{test})\big) \geq 1-\alpha$$

Moreover, in [2] it is proven that

$$1-\alpha \leq P\big(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})\big) \leq 1-\alpha+\frac{1}{n+1} \qquad (1)$$

This guarantees marginal coverage such that the probability that the prediction set contains the correct label is almost exactly $1-\alpha$.

Conformal prediction is easily translated to regression tasks. By defining a score function

$$s(x,y) = \frac{\left|y-\hat{f}(x)\right|}{\widehat{\sigma}(x)}.$$

Where $\widehat{\sigma}$ is the standard deviation to normalise the nonconformity scores. Then since we want the normalized errors $s(X_{\text{test}}, Y_{\text{test}})$ at test time to be less than $\hat{q}$ calculated on the non-conformity scores $s_1, \ldots, s_n$ at calibration time, we aim to achieve the following

$$P\left[\left|Y_{\text{test}} - \hat{f}(X_{\text{test}})\right| \leq \widehat{\sigma}(X_{\text{test}})\hat{q}\right] \geq 1-\alpha.$$

with

$$\mathcal{C}(x) = \left[\hat{f}(x) - \widehat{\sigma}(x)\hat{q},\, \hat{f}(x) + \widehat{\sigma}(x)\hat{q}\right]$$

as the predicted coverage. [2]

1.2 TIME SERIES

We now consider conformal prediction applied to time series data as our baseline experiment. Defining the prediction set to be

$$\hat{C}_n^{\alpha}(Z_{t+1}) = \hat{\mu}(X_{t+1}) \pm Q_{1-\alpha}(\{|y_i - \hat{\mu}(X_i)|\}_{i=1}^{n}) \quad (2)$$

We trained models $\hat{\mu}$ to predict the log returns of different cryptocurrencies i.e. $\log\left(\frac{p_{t+1}^{i}}{p_t^{i}}\right)$, before applying $\hat{q}$ to generate our desired marginal coverage.

However, due to the volatile nature of non-stationary financial time series data, the miscoverage rate, given by

$$M_t(\alpha) := P\left(S_t(X_t, Y_t) > \widehat{Q_t}(1-\alpha)\right)$$

fluctuates across timesteps. This makes conformal prediction inefficient, and there is a need to dynamically correct the predicted coverage over time. [3]

1.3 ADAPTIVE CONFORMAL INFERENCE

Isaac Gibbs and Emmanuel J. Candès propose adaptive conformal inference (ACI) [3] that tackles this problem by updating α values for every timestep.

Given the miscoverage rate, there is an alternative value of α, given by $\alpha_t^* \in [0,1]$ such that $M_t(\alpha_t^*) \approx \alpha$ obtained by re-estimating the score function and quantile function over time.

They introduce the algorithm called adaptive conformal inference represented by

$$\alpha_{t+1} = \alpha_t + \gamma\left(\alpha - \sum_{s=1}^{t} w_s\, err_s\right) \quad (3)$$

Which can achieve either approximate or exact marginal coverage if the new data is correctly calibrated.

1.4 INTRODUCING VOLITILITY

We propose a volatility-aware approach, called Volatility CP, to produce the desired intervals by introducing an additional model σ to predict volatility of cryptocurrencies. We chose a transformer model to leverage its strong performance in cross attention, capturing the strong correlation interdependence of cryptocurrencies on one another [4]. As such, for each timestamp, this model takes an input of shape (12, 100) which considers the 100 features for all 12 cryptocurrencies of our dataset.

This approach aims to enhance ACI by not only updating α to recalibrate non-conformity scores but also utilising volatility predictions from the σ model to generate better coverage. We employ conformal prediction by calculating the $1-\alpha$ quantile of

$$\frac{|y_i - \hat{\mu}(X_i)|}{\widehat{\sigma}(X_i)}, \quad (4)$$

which represents the typical size of error relative to volatility. Then we specify the calibration set $\hat{C}_n^{\alpha}(Z_{t+1})$ to now include the values within the adaptive interval, which we define as

$$\hat{\mu}(X_{t+1}) \pm \widehat{\sigma}(X_{t+1}) \times Q_{1-\alpha}\left(\left\{\left|\frac{y_i - \hat{\mu}(X_i)}{\widehat{\sigma}(X_i)}\right|\right\}_{i=1}^{n}\right) \quad (5)$$

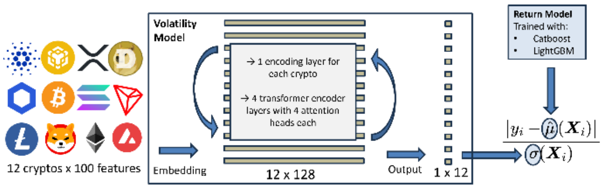
Volatility now influences the coverage calculation in 2 ways:

Firstly, it depends on the estimated volatility calculated at the current timestamp, $\widehat{\sigma}(X_{t+1})$.

Secondly, it scales that calculated volatility using the $1-\alpha$ quantile of standardised residuals as described in Equation (4).

By incorporating volatility correlation while adjusting for distribution shift, we expect it to further reduce miscoverage caused by distribution shift.

The following diagram shows the implementation of both the baseline log return model and volatility transformer model producing outputs to calculate Equations (4) and (5).


12 cryptos x 100 features
Volatility Model
→ 1 encoding layer for each crypto
→ 4 transformer encoder layers with 4 attention heads each
Embedding
12 x 128
Output
1 x 12
Return Model
Trained with:
• Catboost
• LightGBM
$\frac{|y_i - \hat{\mu}(\boldsymbol{X}_i)|}{\sigma(\boldsymbol{X}_i)}$

EVALUATION METRICS

Conformal Prediction focuses on 2 metrics to evaluate any conformal procedure.

Firstly, achieving the marginal coverage specified in Equation (1) at the desired α value reflects the validity of the method. Specifically, for a chosen confidence level $1-\alpha$, the constructed prediction intervals must contain the true outcome at least $(1-\alpha) \times 100\%$ of the time on average. This is the fundamental guarantee of conformal prediction, that coverage holds regardless of the underlying data distribution.

Secondly, conformal prediction also values the average interval size, as intervals that are too wide may cover everything but are practically useless. Therefore, average size can reflect the efficiency of the method. As smaller average size implies tighter intervals, they are preferred over other methods. provided the methods achieve marginal coverage.

As such, striking a balance between validity (coverage) and efficiency (size) is a key goal in conformal prediction methods.

2 METHODOLOGY

A pipeline running the experiments was constructed to preprocess data, train the baseline prediction models given by μ, and apply various conformal prediction methods. Namely, we experimented with conformal prediction, before improvement using the 2 different techniques: ACI and our proposed Volatility CP.

DATA

We utilized a vast dataset of 12 cryptocurrencies spanning 2016–2024 to train baseline models for return prediction and perform conformal prediction to evaluate predictive intervals.

The 12 cryptocurrencies used were: 1000SHIBUSDT, ADAUSDT, AVAXUSDT, BNBUSDT, BTCUSDT, DOGEUSDT, ETHUSDT, LINKUSDT, LTCUSDT, SOLUSDT, TRXUSDT and XRPUSDT.

While we used data from all 12 pairs to train the σ model in Equation (5), we only trained the μ return prediction model on BTCUSDT, ETHUSDT, DOGEUSDT.

Each cryptocurrency data comprised of over 1500 features, which was collected from a combination of data sources.

The data for each pair is split into training, calibration and testing sets.

DATA PREPROCESSING

Initially, given that our input X_i for each cryptocurrency is a feature vector of length exceeding 1500, we use feature selection to eliminate redundant or noisy features.

We opted to use CatBoost for this task. It builds an ensemble of decision trees and chooses the features that reduces the loss the most across all the splits they are used in. If a feature is frequently chosen at high-impact splits (e.g., near the root), it gets a higher importance score.

As such, training a CatBoostRegressor over 500 iterations helped us to extract the top 100 features using its get_feature_importance method.

$\hat{\mu}$ BASELINE MODEL TRAINING

The baseline models predict the log return on price in the next timestep.

We consider y_i as the target variable, the log return of the cryptocurrency between timestamp t and $t+1$, i.e, $y_i = \frac{p_{t+1}}{p_t}$. Then, given the 100 input features generated from data preprocessing, we train 2 baseline models to predict $\hat{\mu}$ using CatBoost and Light Gradient Boosting Machine (LightGBM).

CatBoost and LightGBM were chosen over alternatives such as Support Vector Machines (SVM) and Random Forest due to gradient boosting tree-based methods being highly optimised for speed and scalability. They can efficiently handle large-scale, high-dimensional data, and both support GPU acceleration, which significantly reduces training time. Moreover, certain features of these models make them favourable options.

CatBoost uses ordered boosting which helps prevent overfitting, while its symmetric trees ensure fast prediction speeds [5], which is crucial for making timely decisions in the dynamic cryptocurrency markets.

LightGBM uses histogram-based algorithms that speed up training and reduces memory usage, as well as a leaf-wise tree growth approach that tends to achieve lower losses than level-wise algorithms used by traditional decision tree algorithms [6].

To maximise performance, we employ Optuna for hyperparameter tuning. Some of these include tree depth, L2 regularisation, bagging temperature for CatBoost, and depth, L1 regularisation, L2 regularisation for LightGBM.

The models are trained to maximise Information Coefficient, which is the metric describing the correlation between the actual and predicted cryptocurrency returns. Stellar models for stock returns often have an IC of 0.05 to 0.1. [7]

$\hat{\sigma}$ MODEL TRAINING

This model was trained to implement our proposed Volatility CP method, given by Equation (5). The decision to use a transformer-based model lies on the fact that transformers can leverage self-attention to model cross-asset relationships, capturing interdependencies across multiple cryptocurrencies when predicting volatility.

Similar to our baseline models, this model's hyperparameters such as embedding sizes (d_model), number of attention heads (nhead) and number of transformer layers (num_layers) are tuned with Optuna, and the models are trained to maximise the information coefficient between the actual and predicted volatility.

CONFORMAL PREDICTION

This is the crucial step where our various conformal prediction techniques come into play. To generalize, we consider a range of α values, with 10 values ranging from 0.01 to 0.5.

Initially, all methods use the calibration data to calculate a certain $\hat{q}$ that would cover the true value $1-\alpha$ of the time. Next, we test our various techniques on test data.

Firstly, we employ standard regression conformal prediction on our test data, without accounting for any distribution shifts.

Subsequently, we employ ACI by updating α based on Equation (3), such that the value of $\hat{q}$ updates, leading to adaptive intervals.

Lastly, we use our novel Volatility CP method from Equation (5) to generate prediction intervals, which also updates α and $\hat{q}$.

To ensure robustness and consistency for ACI and Vol CP, we update the α every timestep using data from a lookback of 200 timesteps. The baseline μ models are also retrained every 1000 timesteps for ACI and Volatility CP to enhance adaptability.

3 RESULTS

3.1 VISUAL INFERENCE

The following are results that show marginal coverage and average size of the intervals produced by 3 conformal methods under an α of 0.06, on predictions of BTCUSDT log returns by a baseline CatBoost model.

Ideally, the best techniques should achieve marginal coverage as close to 0.94, while trying to minimize the average size as much as possible.

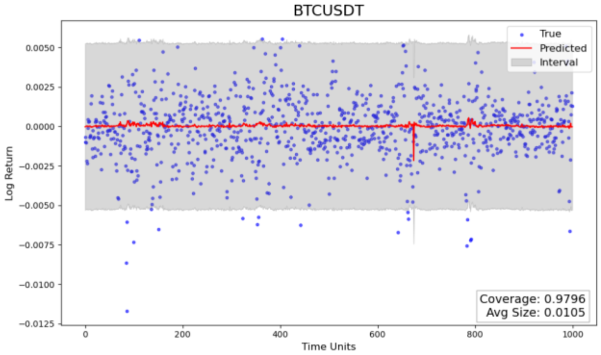


Figure 3.1, CP on predicted BTCUSDT log returns

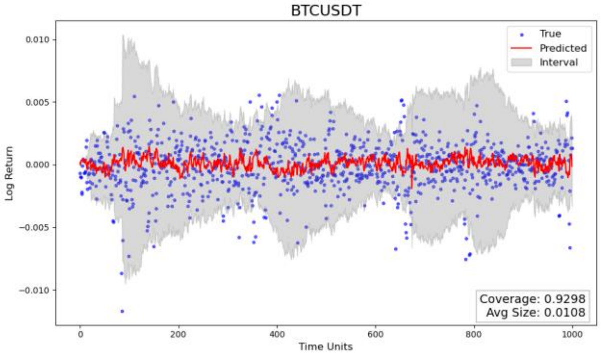


Figure 3.2, ACI on predicted BTCUSDT log returns

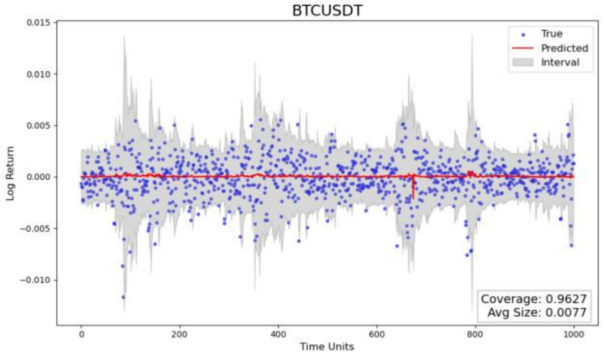


Figure 3.3, Volatility CP on predicted BTCUSDT log returns

Unsurprisingly, standard CP is inefficient due to its inability to account for distribution shifts, especially in the volatile environment of cryptocurrency markets. ACI and volatility CP adapt much better to recent data, with periods of volatility creating larger areas of coverage. Notably, the coverage of volatility CP is more responsive compared to the smoother coverage of ACI.

3.2 QUANTITATIVE INFERENCE

Quantitative comparisons provide a better understanding of how the individual CP techniques perform against each other.

Under our example $\alpha = 0.06$, ACI is the closest to the desired confidence level and target coverage of 0.94.

Table 3.1, Marginal coverage of conformal methods on log returns under a desired 0.94 coverage.

Symbol	Model	CP	ACI	Vol CP
BTC	CatBoost	0.9796	0.9298	0.9627
	LightGBM	0.9799	0.9359	0.9617
ETH	CatBoost	0.9743	0.9298	0.9555
	LightGBM	0.9741	0.9362	0.9553
DOGE	CatBoost	0.9850	0.9304	0.9649
	LightGBM	0.9846	0.9357	0.9636

However, Table 3.2 suggests that Volatility CP provides the smallest average size, i.e. has the highest efficiency.

Table 3.2, Average interval size of conformal methods on log returns

Symbol	Model	CP	ACI	Vol CP
BTC	CatBoost	0.0105	0.0108	0.0077
	LightGBM	0.0107	0.0112	0.0078
ETH	CatBoost	0.0122	0.0125	0.0090
	LightGBM	0.0121	0.0131	0.0091
DOGE	CatBoost	0.0226	0.0215	0.0158
	LightGBM	0.0227	0.0210	0.0157

The theme of ACI achieving the most exact marginal coverage and Volatility CP achieving smallest average size is consistent throughout all values of α, which is demonstrated in the subsequent subsections.

3.3 TRENDS IN MARGINAL COVERAGE

Comparing coverage rate against α, for both CatBoost and LightGBM in Figures 3.4 and 3.5, ACI is the best performing conformal method, achieving the closest coverage rate to the desired α threshold, with both CP and Vol CP showing signs of over-coverage, across all values of α,

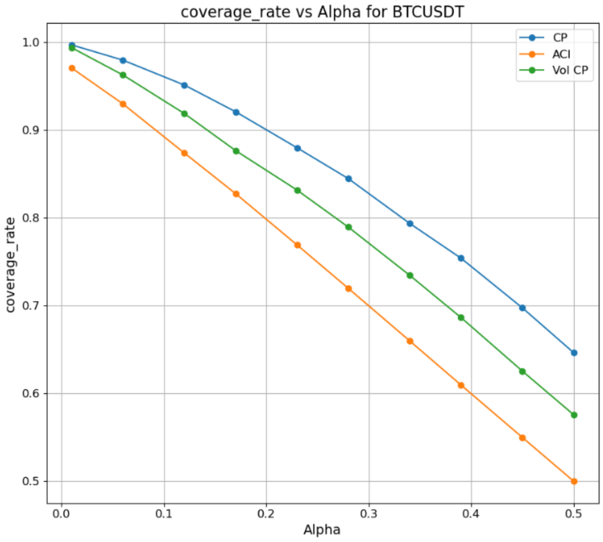


Figure 3.4, marginal coverage of various CP methods on predicted CatBoost BTCUSDT log return

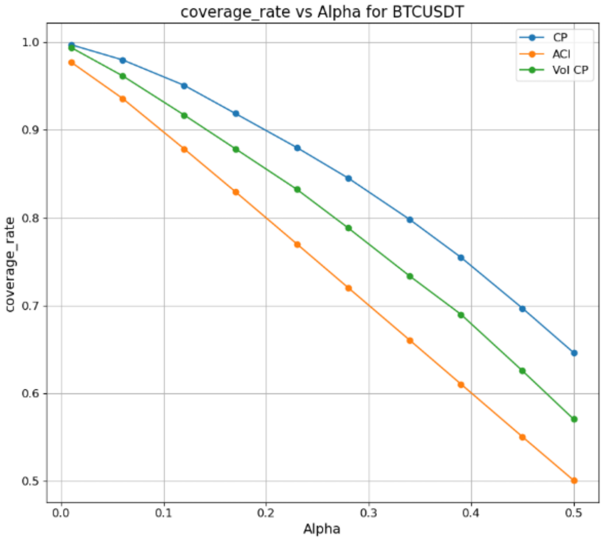


Figure 3.5, marginal coverage of various CP methods on predicted LightGBM BTCUSDT log returns.

3.4 TRENDS IN AVERAGE SIZE

From Figure 3.6 and 3.7, For both CatBoost and LightGBM, Vol CP achieves the smallest and most efficient intervals compared to CP and ACI across all values of α.

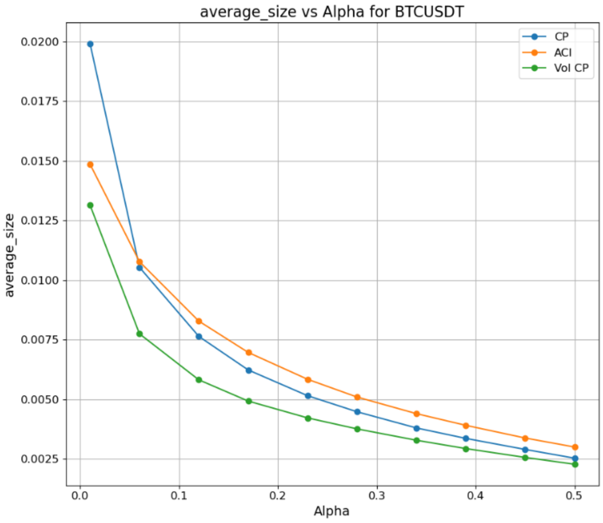


Figure 3.6, average size of various CP methods on predicted CatBoost BTCUSDT log returns

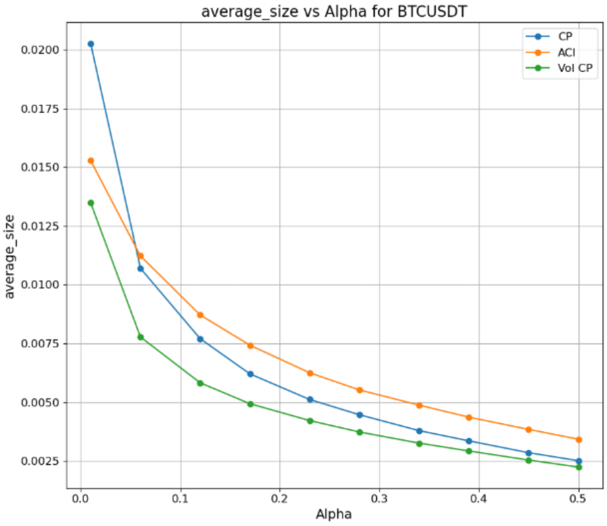


Figure 3.7, average size of various CP methods on predicted LightGBM BTCUSDT log returns

Among the 3 conformal methods we experimented on, ACI produced the best marginal coverage, over all 10 α values we specified from 0.01 to 0.5. However, the average size of intervals was larger than our proposed Vol CP method, which produced the most favourable interval sizes across all α values.

This is significant as results show that Vol CP achieves over-coverage, as seen by the points producing the green line in Figures 3.4 and 3.5. At every value of α, the coverage rate, i.e. $1 - \alpha$, is higher than the desired value of α. This is a positive sign moving forward for 2 reasons.

Firstly, the over-coverage suggests that the intervals are too large, and certain techniques can refine our Vol CP approach to reduce these intervals to achieve the exact marginal coverage that conformal prediction should guarantee.

Secondly, Vol CP has the case of over-coverage while simultaneously achieving the smallest average size, which means that post-refinement of Vol CP is likely to produce the valid conformal method with the smallest average size, which is our goal of this research.

3.5 SYNTHETIC DATA GENERATION

While our results were satisfactory, our μ baseline models produced low IC values of approximately 0, suggesting that our prediction returns were basically random did not meet the desired IC of between 0.05 and 0.1 of the best models. Therefore, to further test our post-processing method, we generated data to mimic real market features such that they could accurately produce an IC of around 0.10 after training.

The results from synthetic data generation can be found in the Appendix.

CONCLUSION

In this study, we used CatBoost and LightGBM models to predict log returns of 3 different cryptocurrencies, before applying conformal prediction by applying various conformal methods to generate prediction intervals such that they contain the true outcome at least $(1-\alpha) \times 100\%$ of the time on average.

We showed that a novel conformal method incorporating volatility can further improve risk quantification to generate more efficient prediction intervals of cryptocurrency log returns, provided we refine it to reduce over-coverage while maintaining the smallest average interval size.

Further studies can be carried out to investigate the performance of other conformal methods that have been proposed, such as a conformal proportional-integral-derivative (PID) controller [8].

ACKNOWLEDGEMENT

I would like to thank Professor Bo An and Cai Xinyu for this opportunity to participate in this URECA project, especially for their guidance and encouragement throughout the project.

This research project was supported by Nanyang Technological University under the URECA Undergraduate Research Programme.

REFERENCES

[1] Zaffran, M., Féron, O., Goude, Y., Josse, J., & Dieuleveut, A. (2022). Adaptive conformal predictions for time series. Proceedings of the 39th International Conference on Machine Learning, 162, 25834–25866. https://doi.org/10.48550/arXiv.2202.07282

[2] Angelopoulos, A. N., & Bates, S. (2022). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511. https://doi.org/10.48550/arXiv.2107.07511

[3] Gibbs, I., & Candès, E. J. (2021). Adaptive conformal inference under distribution shift. In Advances in Neural Information Processing Systems (Vol. 34, pp. 16988–16999). https://doi.org/10.5555/3540261.3540389

[4] Барановський, О., Кужелєв, М., Жерліцин, Д., Сердюков, К., & Сокирко, О. (2021). Cryptocurrency market trends and fundamental economic indicators: correlation and regression analysis. Financial and credit activity problems of theory and practice, 3(38), 249-261. https://doi.org/10.18371/fcaptp.v3i38.237454

[5] Parameter tuning | CatBoost (n.d.). https://catboost.ai/docs/en/concepts/parameter-tuning

[6] Features — LightGBM 4.6.0.99 documentation (n.d.). https://lightgbm.readthedocs.io/en/latest/Features.html

[7] Zhang, F., Guo, R., & Cao, H. (2020). Information coefficient as a performance measure of stock selection models. Wells Fargo & Company. https://doi.org/10.48550/arXiv.2010.08601

[8] Angelopoulos, A. N., Candès, E. J., & Tibshirani, R. J. (2023). Conformal PID control for time series prediction. arXiv. https://doi.org/10.48550/arXiv.2307.16895

APPENDIX

Results similar to our original experiments are observed when using synthetically generated data, as seen in Figures 4.1 to 4.4. ACI produces the most accurate marginal coverage and Vol CP has the smallest average size of intervals.

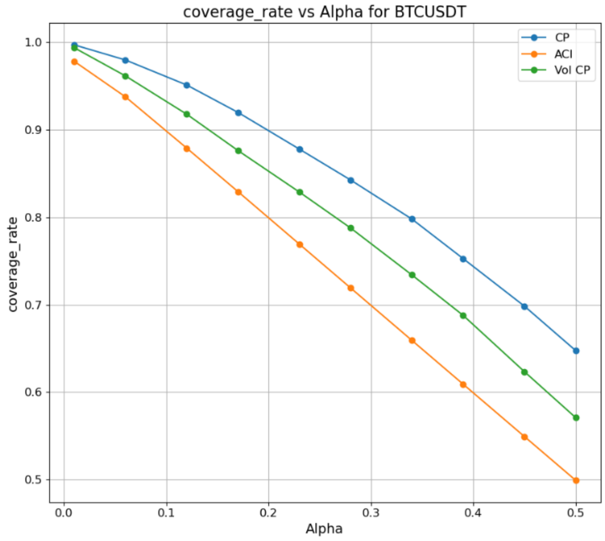


Figure 4.1, marginal coverage of various CP methods on predicted CatBoost BTCUSDT log returns

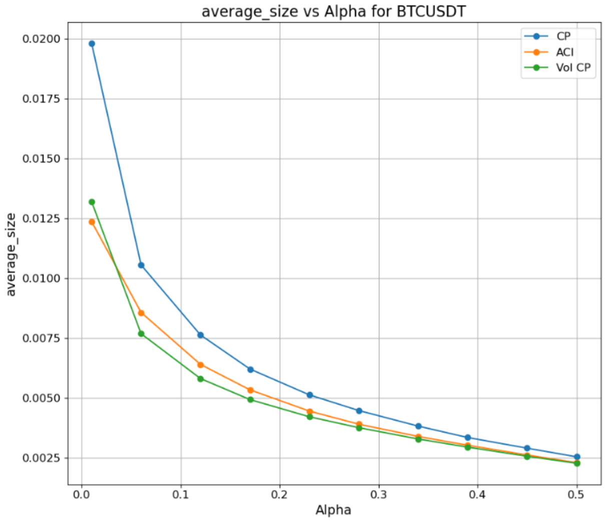


Figure 4.2, average size of various CP methods on predicted CatBoost BTCUSDT log returns

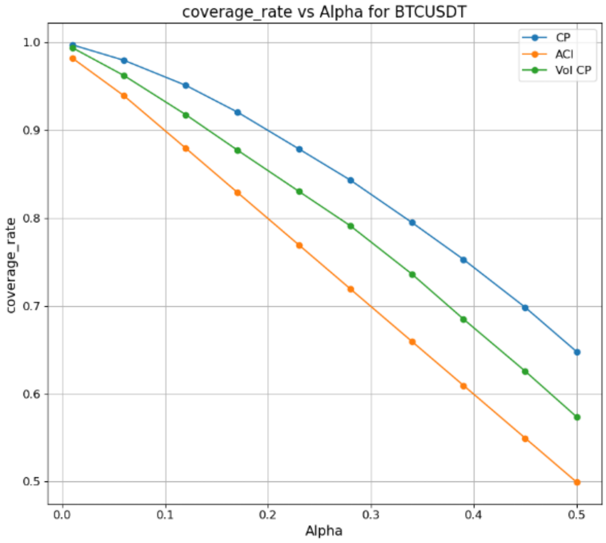


Figure 4.3, marginal coverage of various CP methods on predicted LightGBM BTCUSDT log returns

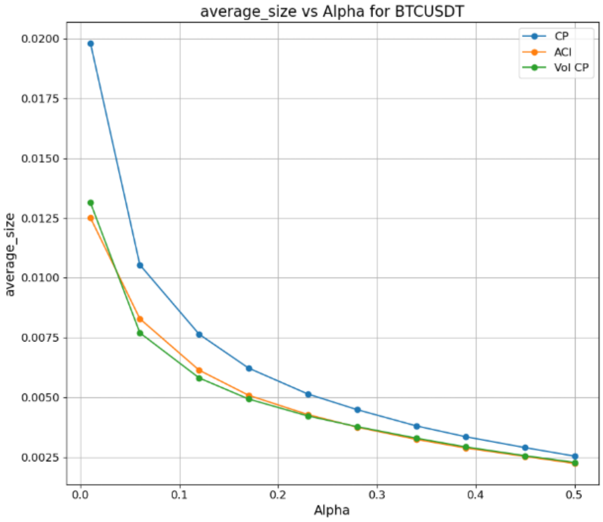


Figure 4.4, average size of various CP methods on predicted LightGBM BTCUSDT log returns