Uncertainty Quantification for Machine Learning with Application to Financial Time Series (Python for Data Science)

Kota OKUDA, Ryunosuke TANAKA

December 17, 2025

Abstract

In financial time series forecasting, accurate point prediction is often insufficient for risk management and quantifying the associated uncertainty is equally critical. This project provides a framework for Uncertainty Quantification (UQ) using Long Short-Term Memory (LSTM) combined with Monte Carlo dropout. We apply this to the daily closing prices of corn futures to analyse several types of predictive variances. Specifically, we decompose the total uncertainty into three components: model (epistemic), data (aleatoric), and temporal diffusion. Our experiments investigate the impact of look-back windows, look-ahead horizons, and data amount on these components. The results indicate that whilst temporal diffusion becomes the dominant source of uncertainty in multi-step forecasting, the choice of look-back window affects model performance; excessively long histories (e.g., 120 days) may introduce outdated noise, whereas moderate windows (e.g., 60 days) offer a better balance. Furthermore, the derived uncertainty intervals dynamically expand during high-volatility periods, such as the COVID-19 pandemic, demonstrating the method's efficacy in capturing market risks.

The link for our `Python` implementation.
The link for our `Github` repository.

1 Introduction

In predictive inference applied in financial markets, the risk management of models is essential along with the accuracy. In particular, the uncertainty in statistical or machine learning models shall be quantified in a way that practitioners can discern several sources of variance inflation factors.

In most of the cases, the uncertainty quantification is overlooked, partially because the point prediction seems enough useful. However, if we mean by "useful" for decision making, the interval prediction provides the richer information in financial forecasting where the price change rapidly and drastically: for example, it might be better to say that the next price will be 460-520 USD with probability 80% than solely say 476 USD, since we recognise here that the price may not go below 460 USD with probability less than 20% (one-tail).

1.1 Uncertainties

There are two types of uncertainties:

- Aleatoric uncertainty: Represents the observational noise (e.g., market volatility, measurement errors). It cannot be reduced even if the amount of training data increases.
- Epistemic uncertainty: Represents the uncertainty in the model parameters due to a lack of knowledge or data. This can be reduced by adding more training data or improving the model.

1.1.1 Factors Influencing Uncertainty

We consider four types of uncertainties regarding the time series forecasting: look-back, look-ahead, data amount, and the model complexity. The first source of uncertainties, the look-back, represents the length of input data horizons. There is a general tendency that the longer length shrinks the variance of the output by including the larger past information (e.g., seven days look-back may encode the weekly effects, and the monthly periodic effects may be taken into consideration with 30 days.) The second uncertainty from the look-ahead, is related to the multi-step forecasting in traditional statistical time series prediction. It is expected that the longer step prediction makes the uncertainty rise due to the temporal diffusion and hence the variance inflates with the order of $O(\sqrt{t})$ where t for the t-step forecasting. The third one, regarding the data amount, comes from the lack of data in models and is related to the epistemic uncertainty. The last one, the model uncertainty, is also classified as an epistemic uncertainty. It is measured by the Monte Carlo dropout as an approximation of a Gaussian process, which is a Bayesian method in a nonparametric way.

We evaluate the total uncertainty with the last three components:

$$\sigma_{\text{total},t} = \sqrt{\hat{\sigma}^2_{\text{model},t} + \hat{\sigma}^2_{\text{data},t} + \hat{\sigma}^2_{\text{time},t}}. \tag{1}$$

For our reference, we assume that the standard deviation increases 0.05 per day for the temporal horizons in multi-step forecasting and hence

$$\hat{\sigma}^2_{\text{time},t} = (\alpha \cdot \sqrt{t})^2 = \alpha^2 t.$$

As for the other variances st time t, we use

$$\hat{\sigma}^2_{\text{model},t} = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{K}\sum_{k=1}^{K}(\hat{y}_{m,k,t} - \bar{y}_{m,t})^2\right), \quad \hat{\sigma}^2_{\text{data},t} = \frac{1}{M}\sum_{m=1}^{M}(\bar{y}_{m,t} - \bar{Y}_t)^2,$$

where

$$\bar{y}_{m,t} = \frac{1}{K}\sum_{k=1}^{K}\hat{y}_{m,k,t}, \quad \bar{Y}_t = \frac{1}{M}\sum_{m=1}^{M}\bar{y}_{m,t}.$$

Here, K indicates the number of iterations of Monte Carlo dropout which is later explained, and M represents the number of ensembles used to quantify the uncertainty between models. We could write that

$$\text{Var}(\hat{\sigma}^2_{\text{model},t}) = \sigma^2_{\text{data},t}.$$

As for the look-back, we present the three different length: 20, 60, and 120 days. We set $K = 20$ and $M = 5$.

1.2 Data

Dataset: Daily closing price series of Corn Futures (`ZC=F`) via Yahoo Finance API.

- Period: 1 Jan, 2015 – 1 Jan, 2024 ($\simeq$ 9 years).
- Frequency: Daily.
- Preprocessing: Standardised.

For the better prediction, we also use some features that are calculated directly from the raw price data to capture market trends and momentum. Features are: RSI (Relative Strength Index): a momentum oscillator that measures the speed and change of price movements; EMA20 (20-day Exponential Moving Average: a moving average that places a greater weight and significance on the most recent data points to identify the underlying trend direction over a short-term period (20 days); and BBL / BBU (Bollinger Bands Lower / Upper) volatility bands placed above and below a moving average. Furthermore, the calendar information (Cyclical Features) are incorporated. Month_Sin and Month_Cos are used which represent the month of the year transformed into sine and cosine values since time is cyclical (January follows December), and using simple numbers (1 to 12) breaks the continuity between 12 and 1.

1.3 UQ Methodology: MC Dropout

To quantify and decompose the uncertainties described above, one approach is to employ the Bayesian inference, which naturally updates the prior distribution to the posterior distribution, and hence it inherits the entire information on distribution. However, Bayesian methods are computationally costly.

Instead, we employ an Monte Carlo (MC) Dropout framework [Gal and Ghahramani, 2016], which quantifies the model uncertainty. In practice, it is equivalent to performing K times the network and averaging the results. From a theoretical perspective, this is the approximation of the posterior predictive expectation. Furthermore, it is shown that the neural network with dropout applied before every weight layer is mathematically equivalent to an approximation to the probabilistic deep Gaussian process. We perform the MC dropout with rate 0.3 and with $K = 20$ stochastic forward passes of the LSTM.

Algorithm 1: MC Dropout with LSTM for one-time sample

```
Input  : Input sequence $X = (x_1, \ldots, x_T)$, Initial states $h_0, C_0$, Dropout rate
         $p \in [0, 1)$, Parameters $\theta = \{W, b\}$
Output: Hidden state sequence $H = (h_1, \ldots, h_T)$

$z_x \sim \text{Bernoulli}(1 - p)$                              // Input mask
$z_h \sim \text{Bernoulli}(1 - p)$                          // Recurrent mask

for $t \leftarrow 1$ to $T$ do
    $\tilde{x}_t \leftarrow x_t \odot z_x$
    $\tilde{h}_{t-1} \leftarrow h_{t-1} \odot z_h$

    $v_{\text{in}} \leftarrow [\tilde{h}_{t-1}, \tilde{x}_t]$
    $f_t \leftarrow \sigma(W_f \cdot v_{\text{in}} + b_f)$
    $i_t \leftarrow \sigma(W_i \cdot v_{\text{in}} + b_i)$
    $\tilde{C}_t \leftarrow \tanh(W_C \cdot v_{\text{in}} + b_C)$
    $o_t \leftarrow \sigma(W_o \cdot v_{\text{in}} + b_o)$

    $C_t \leftarrow f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
    $h_t \leftarrow o_t \odot \tanh(C_t)$

return $H = (h_1, \ldots, h_T)$
```

Algorithm 2: MC Dropout Inference with LSTM: complete version

```
Input  : Input sequence $X = (x_1, \dots, x_T)$, Pre-trained Parameters $\theta = \{W, b\}$,
         Dropout rate $p \in [0, 1)$, Number of MC samples $K$ (e.g., $100$)
Output: Predictive Mean $\mu_Y$, Predictive Uncertainty $\sigma_Y^2$

for $k \leftarrow 1$ to $K$ do
    $z_x^{(k)} \sim \text{Bernoulli}(1 - p)$                // Input mask
    $z_h^{(k)} \sim \text{Bernoulli}(1 - p)$                // Recurrent mask
    Initialize states $h_0, C_0$
    for $t \leftarrow 1$ to $T$ do
        $\tilde{x}_t \leftarrow x_t \odot z_x^{(k)}$
        $\tilde{h}_{t-1} \leftarrow h_{t-1} \odot z_h^{(k)}$
        $v_{\text{in}} \leftarrow [\tilde{h}_{t-1}, \tilde{x}_t]$
        $f_t \leftarrow \sigma(W_f \cdot v_{\text{in}} + b_f)$
        $i_t \leftarrow \sigma(W_i \cdot v_{\text{in}} + b_i)$
        $\tilde{C}_t \leftarrow \tanh(W_C \cdot v_{\text{in}} + b_C)$
        $o_t \leftarrow \sigma(W_o \cdot v_{\text{in}} + b_o)$
        $C_t \leftarrow f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
        $h_t \leftarrow o_t \odot \tanh(C_t)$
    $y^{(k)} \leftarrow \text{OutputLayer}(h_T)$
$\mu_Y \leftarrow \frac{1}{K} \sum_{k=1}^{K} y^{(k)}$
$\sigma_Y^2 \leftarrow \frac{1}{K} \sum_{k=1}^{K} (y^{(k)} - \mu_Y)^2$
return $\mu_Y, \sigma_Y^2$
```

2 Results

2.1 Uncertainty variations in each components

we first observed the variation of each source of uncertainties. The Figure 1 suggests that the change in look-back days are not consistent with the length of days and specifically, 120 days might be too long in our situation. The uncertainty for look-ahead is, by design, growing as the stepsize of the forecasting is large. Also as expected, the smaller data amount and the smaller model uncertainty are related to the variance inflation.

However, the variation exists in the length of look-back days and variance and, too long history may introduce the outdated noise. It might be possible that the radical change quintessential to financial time series can only be adapted by looking back the short term past.

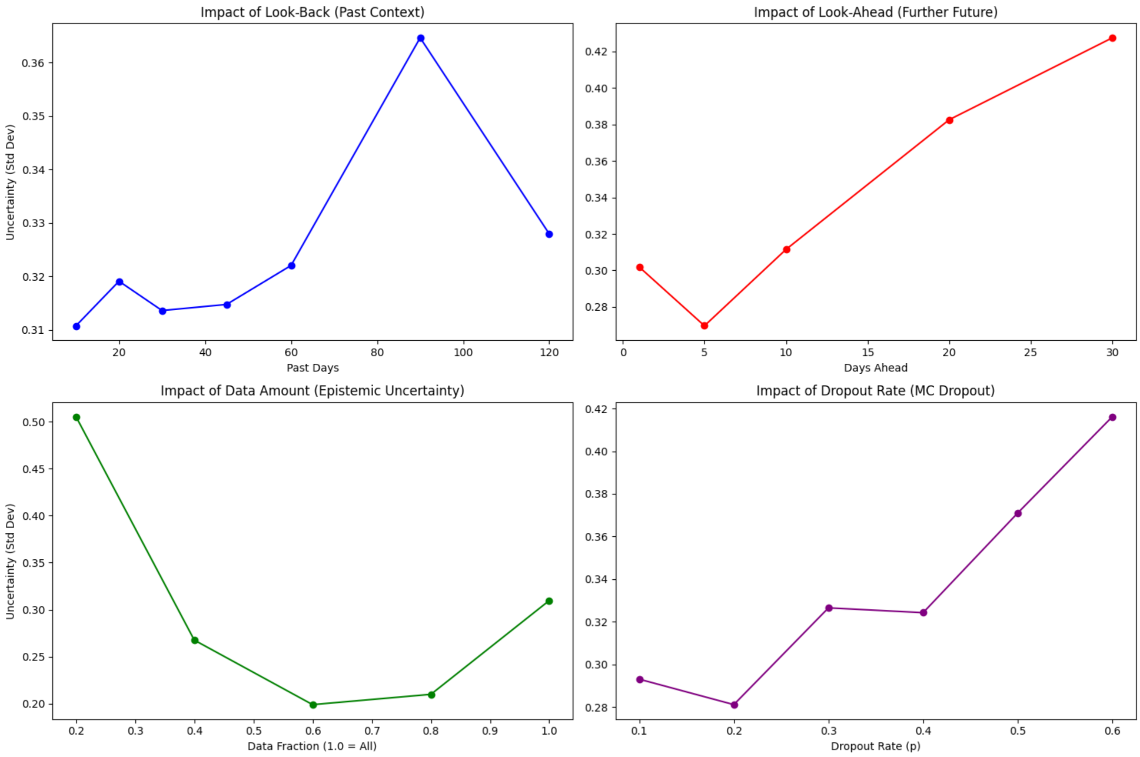

Figure 1: Variation of uncertainty with respect to: (Top Left) look-back days, (Top Right) look-ahead days, (Bottom Left) fractions of data amount, (Bottom Right) rates of neurons dropout.

2.2 Predictive Performance and Uncertainty Intervals

We henceforth fix the days of the look-ahead as 10 days and that of the look-back as 20, 60 and 120 days. Figure 2, Figure 3 and Figure 4 are the corresponding results of forecasting with the test data. The red solid line represents the ensemble mean prediction, while the shaded red area indicates the estimated total uncertainty interval ($\pm 2\sigma_{total}$). As suggested in Figure 1, 20 look-back days in Figure 2 tends to miss a prediction and the uncertainty is larger than that of the others. 60 and 120 look-back days in Figure 2 and Figure 3 are seemingly competitive, but the prediction with 120 look-back days tends to mispredict when the series changes dramatically. Overall, although the mean of the prediction is far from the correct point, the intervals might be reasonably seen as a risk measure: the uncertainty is large when the series changes quickly (e.g., during the COVID-19) and it stabilises after the pandemic. Thus, it can be said that the interval capture the "risky" market.

2.3 Decomposition of Uncertainty Components

The Figure 5 indicates the decomposition of the uncertainty into the three components in terms of time, data, and models in a fraction of 30 days. From the uncertainty in data, it can be said that there is an irreducible noise and structural ambiguity within the market data itself. Although the uncertainty in data and the models provides a default and stable range of variance

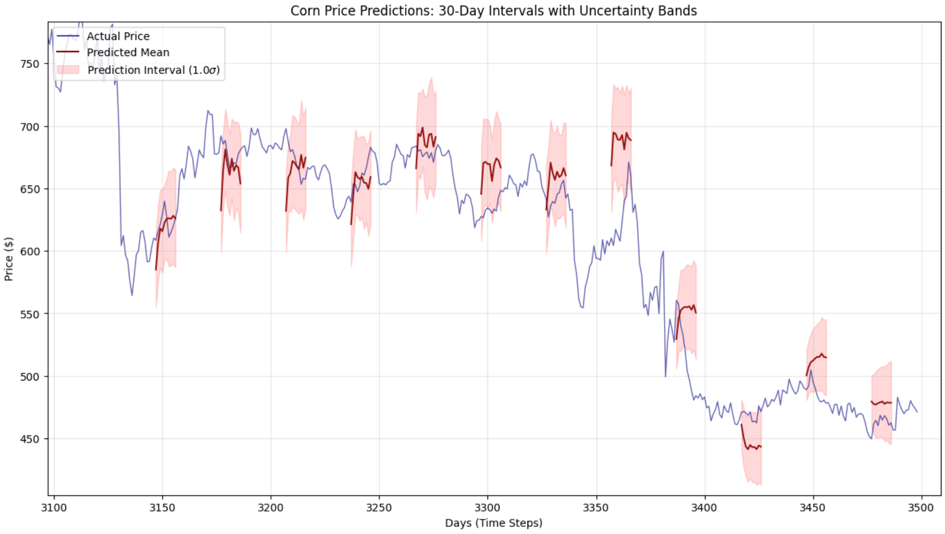

Figure 2: Prediction and uncertainty quantification for financial time series over the test-time with 20 look-back days.

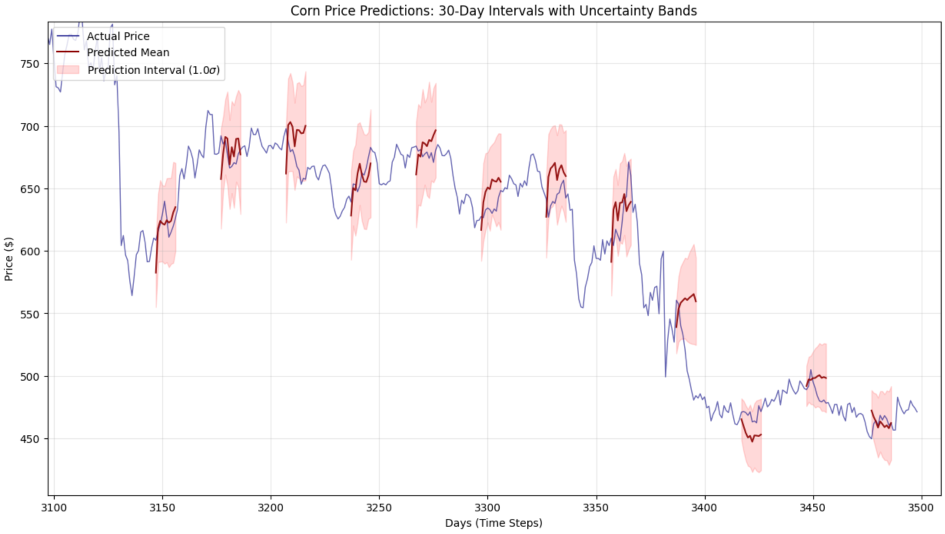

Figure 3: Prediction and uncertainty quantification for financial time series over the test-time with 60 look-back days.

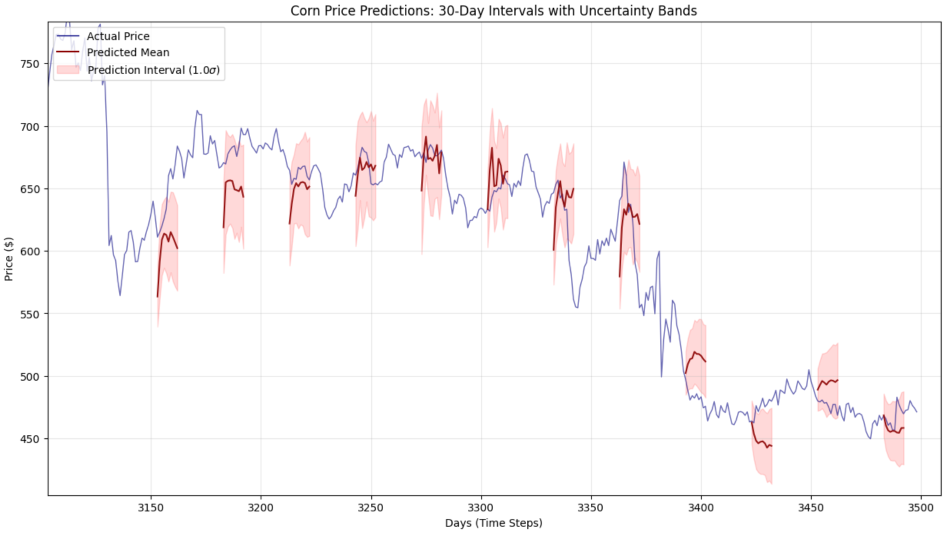

Figure 4: Prediction and uncertainty quantification for financial time series over the test-time with 120 look-back days.

during the entire 10 days, the variance from the temporal diffusion largely dominates the others. It might also be possible to improve models since the variance for models.

In our case, 80% of errors from the predicted price can cover 60 dollars of difference at 30-days ahead forecasting even though the point prediction is almost meaningless since the actual price changes rapidly and has larger volatility.

3 Discussion

In the scenario of financial time series forecasting where the series changes rapidly and drastically, the mere point prediction may not provide useful information. On the contrary, the probabilistic interval prediction provides the lower bound with certain probability, which can be useful for decision making of purchasing the commodities.

It is interesting to have observed that the longer look-back days might have not necessarily predict better in our cases. So, looking into the future, it might be possible to average the models (in our case, the model with 20, 60 and 120 look-back days) according to some local structures, e.g., in the crises or stable phases [McAlinn and West, 2019, Johnson and West, 2025] rather than to the global or entire test-time.

4 Comments

(Kota OKUDA) Throughout this project, we did not mention the computational burden, which, of course, should be taken into account in the real environment. We used GPU to accelerate the computation, but it would be very difficult and take very long time without it. Also, I originally

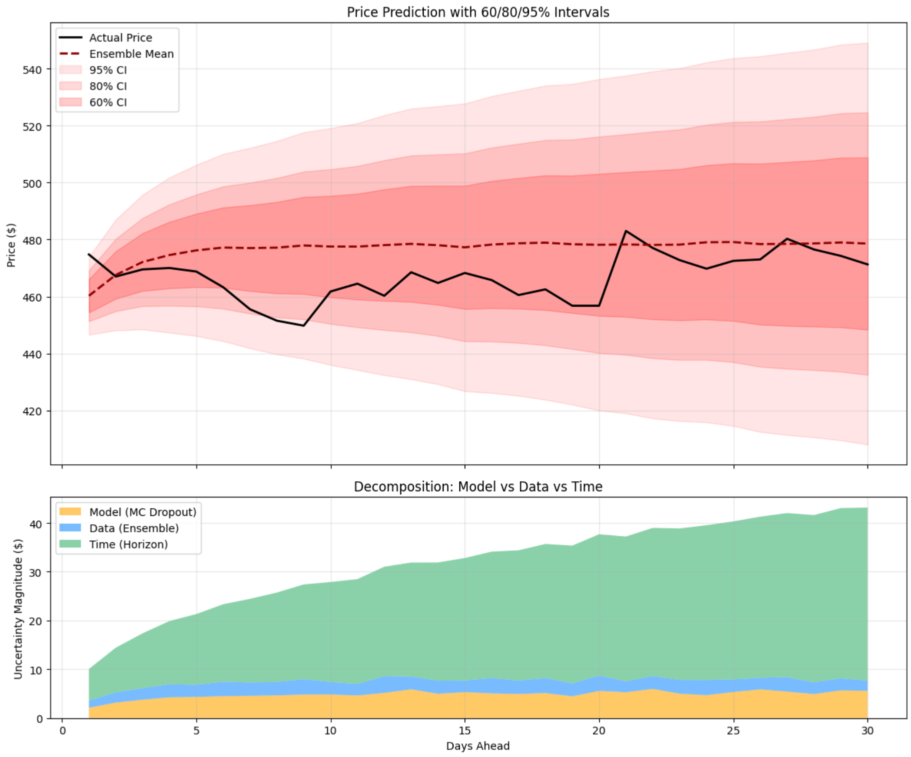

Figure 5: Decomposition of the uncertainty of 30 look-ahead days over the test-time.

thought that the (exact) Bayesian methods are the only way and did not pay much attention to the UQ methods in deep learning, but I am glad to learn that there is an approximation which asymptotically converges to the Bayesian posterior like Monte Carlo dropout. I am now curious about other UQ methods: for example, conformal prediction can be potentially useful even in Bayesian inference. However, I also have learnt the importance of decomposing the uncertainty and of emphasising "which uncertainty" I am trying to quantify.

(Ryunosuke TANAKA) In this project, we focused on corn futures and investigated how LSTM design choices, such as look-back window length and dataset size, affect predictive uncertainty. These effects may vary across commodities. Exploring how asset-specific volatility characteristics modulate these relationships would be both theoretically interesting and practically valuable. In addition, I found MC Dropout powerful in practice, as it provides a computationally efficient approximation to ensemble methods for uncertainty quantification.

References

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Matthew C Johnson and Mike West. Bayesian predictive synthesis with outcome-dependent pools. *Statistical Science*, 40(1):109–127, 2025.

Kenichiro McAlinn and Mike West. Dynamic bayesian predictive synthesis in time series forecasting. *Journal of econometrics*, 210(1):155–169, 2019.