Investment Sizing with Deep Learning Prediction Uncertainties for High-Frequency Eurodollar Futures Trading

Trent Spears, Stefan Zohren, and Stephen Roberts

Trent Spears
is a DPhil student within the Machine Learning Research Group and the Oxford-Man Institute of Quantitative Finance at the University of Oxford in Oxford, UK.
trent@robots.ox.ac.uk

Stefan Zohren
is an associate professor (Research) with the Machine Learning Research Group and the Oxford-Man Institute of Quantitative Finance at the University of Oxford in Oxford, UK.
zohren@robots.ox.ac.uk

Stephen Roberts
is the RAEng/Man Group Professor of Machine Learning at the University of Oxford and the Director of the Oxford-Man Institute of Quantitative Finance in Oxford, UK.
sjrob@robots.ox.ac.uk

KEY FINDINGS

- The authors model high-frequency Eurodollar Futures limit order book data using state-of-the-art deep learning to predict interest rate curve changes on small time horizons.
- They further augment their models to yield estimates of prediction uncertainty.
- In certain settings, the uncertainty estimates can be used in conjunction with return predictions for scaling bankroll allocation between trades. This can lead to clear trading outperformance relative to the case that uncertainty is not taken into account.

ABSTRACT

In this article, the authors show that prediction uncertainty estimates gleaned from deep learning models can be useful inputs for influencing the relative allocation of risk capital across trades. In this way, consideration of uncertainty is important because it permits the scaling of investment size across trade opportunities in a principled and data-driven way. The authors showcase this insight with a prediction model and, based on a Sharpe ratio metric, find clear outperformance relative to trading strategies that either do not take uncertainty into account or use an alternative market-based statistic as a proxy for uncertainty. Of added novelty is their modeling of high-frequency data at the top level of the Eurodollar futures limit order book for each trading day of 2018, whereby they predict interest rate curve changes on small time horizons. The authors are motivated to study the market for these popularly traded interest rate derivatives because it is deep and liquid and contributes to the efficient functioning of global finance—though there is relatively little by way of its modeling contained in the academic literature. Hence, they verify the utility of prediction models and uncertainty estimates for trading applications in this complex and multidimensional asset price space.

TOPICS

*Big data/machine learning, derivatives, simulations, statistical methods**

*All articles are now categorized by topics and subtopics. **View at PM-Research.com**.

In practice, it is common for investors to size investment positions based on conviction in trade ideas. Some traders highlight the importance of investment sizing at least with the view to leverage exposure to the positive right tail of the profit and loss distribution (Donnelly 2019). However, there is relatively little academic discussion regarding achieving the task in a data-driven way. Inspired by recent innovations in financial machine learning, we present a case for using uncertainty estimates gleaned from a prediction model for the purpose of investment sizing within a trading strategy. Furthermore, we show how this can improve relative investment performance.

The trading strategy depends on a financial asset price prediction model at its core. In this context, and in the era of big data, deep learning models often present as the state of the art. This is particularly true for models given high-frequency limit order book data, whereby many millions of pieces of unique information may be collected and processed according to data-intensive algorithms. In recent work, prediction models for financial time series have been improved on by deep convolutional neural networks (CNN) and long short-term memory (LSTM) networks (Chen et al. 2016; Borovkova and Tsiamas 2019). A state-of-the-art CNN-LSTM with inception (CNN-LSTM Inc) was shown to be of utility in formulating financial prediction as a classification problem (Zhang, Zohren, and Roberts 2018b).

We build from this work to model high-frequency data at Level 1 of the Eurodollar futures[1] limit order book. Important variations include our framing of prediction as a regression problem for modeling outright price changes and both our input and output data covering a multidimensional asset price space. Furthermore, this approach complements recent literature advancing interest rate derivative price prediction (Kondratyev 2018; Gonzalvez et al. 2019).

However, although there is ongoing innovation and success in applying deep learning prediction models in finance, less has been shown by way of retrieving meaningful prediction uncertainties. This is despite innovations and demonstrable utility within a diverse range of application domains, including computer vision, medicine, and astrophysics (Kendall and Gal 2017; Leibig et al. 2017; Cobb et al. 2019a). Of course, prediction uncertainty can arise from a variety of sources, although two stylized facets of uncertainty are often modeled. On one hand, there is aleatoric uncertainty, which we model explicitly as expressed through a (spatial) heteroskedastic variance–covariance matrix of the predicted prices. On the other hand, and less directly, we model epistemic uncertainty as induced by the model specification.

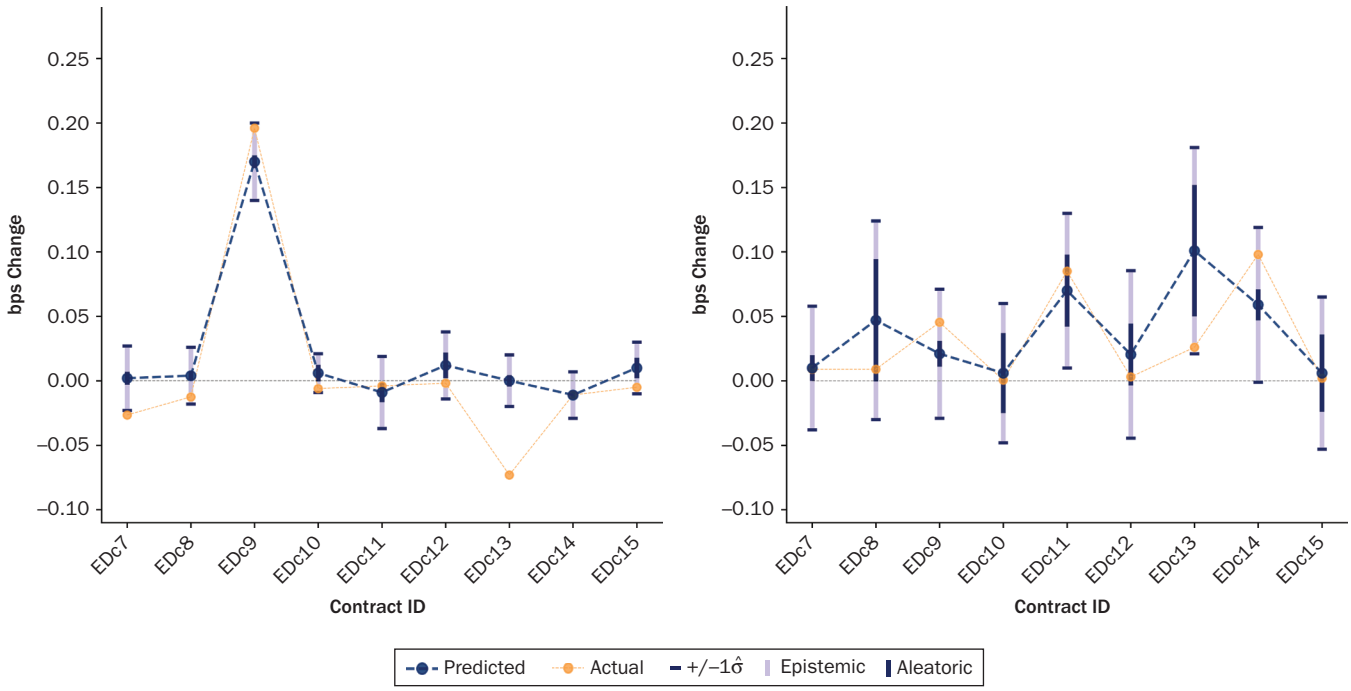
In the context of deep learning, pseudo-Bayesian approximations to epistemic uncertainty have been proposed. For example, it has been shown that epistemic uncertainty can be cheaply approximated in deep learning models with dropout (Gal and Ghahramani 2016c). Given that dropout is often a component of deep learning models for financial prediction, and given the relative ease by which dropout sampling can be implemented, this approximation seems a practical, yet relatively underexplored, component of prediction uncertainty for financial applications; refer to Zhang, Zohren, and Roberts (2018a) for an early attempt.

In any case, the utilization of model predictions and uncertainty estimates—such as displayed in Exhibit 1—can become apparent in the details of a trading strategy. A proof of concept of potential usefulness can be found in improving the standard and preliminary performance metrics by which trading strategies are typically evaluated. In particular, we target improvement in the Sharpe ratio (Sharpe 1994). Some authors have improved on a vanilla long–short trading strategy by directly incorporating a

[1] Here, we note the class of derivative in focus is the popularly traded interest rate derivatives, whose reference contract interest rate applies to a hypothetical USD-denominated unsecured bank funding deposit—rather than a foreign exchange derivative on the EUR/USD currency pair.

EXHIBIT 1

Interest Rate Curve Predictions with Uncertainty Estimates



NOTES: We use deep learning models of changes in a subset of the interest rate curve for a collection of liquid Eurodollar futures contracts and recover uncertainty estimates around predictions. Predictions for the next event time are shown as dark blue circles, versus the actual realized outcome in orange. Total prediction uncertainty is depicted via the one standard deviation horizontal dark blue bars, which comprise of an estimate of aleatoric (epistemic) uncertainty in proportion to the vertical part shaded dark blue (mauve). The left-hand image depicts a prediction that is more certain, and corresponds more closely to the target outcome, relative to the right-hand image.

Sharpe measure in the loss function (Choey and Weigend 1997; Lim, Zohren, and Roberts 2019) or by applying parameterized decision rules for investment sizing (Towers and Burgess 1999). Our approach is closer in spirit to the latter: With the addition of reference uncertainty estimates for model-based beliefs on price returns, we can optimize the Sharpe metric in a way that is inspired by how some successful traders intuitively adjust trade sizes in the real world—by scaling into high conviction trades (corresponding to lower uncertainty) offering sufficiently large reward with relatively more bankroll.

In the subsequent sections, beginning with the second section, we specify the dataset and preprocessing steps, the model, and the trading strategy. Results and discussion are offered in the third section. We conclude in fourth section and offer avenues for future research.

DATA AND MODELING CHOICES

Data Commentary

Seeking to model high-frequency Eurodollar futures prices, and hence a component of an interest rate curve, we have collected Thomson Reuters data for continuous contract EDc1 to EDc15 for each US trading day of 2018. These data reflect Level 1 limit order book best bid–ask-volume quotes, with some additional trade execution data. The first 15 contracts are a subset of the 44 market-traded contracts available

EXHIBIT 2

Statistics for Number of Quotes, Trades, and Volume

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Quotes	2.2	3.0	3.0	3.0	2.2	1.0	157.5	242.6	243.8	308.8	287.1	286.2	317.3	264.2	274.8
Trades	0.3	0.3	0.1	0.1	0.0	0.0	2.0	1.8	2.2	1.9	2.3	1.8	1.6	1.5	1.8
Volume	23.7	17.2	4.7	2.1	0.2	0.0	37.3	34.5	48.7	28.7	24.5	19.5	22.1	10.9	11.7

NOTES: Median Daily Number of Quotes, Trades, and Total Volume Executed for Eurodollar Futures Contracts EDc1 to EDc15. Data recorded on each (US) trading day of 2018. Units are in 000s.

SOURCE: Thomson Reuters.

EXHIBIT 3

Summary Statistics for Daily Microprice Changes

	EDc7	EDc8	EDc9	EDc10	EDc11	EDc12	EDc13	EDc14	EDc15
Min	0.3	1.2	1.4	1.6	1.7	1.8	1.3	1.1	1.6
25th perc.	2.3	2.8	3.1	3.4	3.6	3.8	4.0	4.0	4.0
50th perc.	3.0	3.7	4.3	4.6	4.8	5.0	5.1	5.1	5.3
75th perc.	4.4	5.3	5.9	6.5	6.9	6.9	7.2	7.3	7.5
Max	16.2	17.1	20.6	22.1	23.3	24.9	25.8	26.2	26.2
σ (daily Hi-Lo)	2.4	2.6	2.8	3.0	3.2	3.2	3.3	3.4	3.3

NOTE: Summary statistics for the yield data used in these notes: difference in contract microprice levels (bps) high and low on each (US) trading day of 2018; percentiles and daily standard deviation.

SOURCE: Thomson Reuters.

through CME group. In terms of the number of quotes and trades and total volume traded, we observe greater trading activity in contracts EDc7 to EDc15 relative to the earlier contracts; see Exhibit 2 for reference. Owing to this market activity, we focus our analysis on EDc7 to EDc15.

A key step of our data preprocessing is to construct a time series of microprices for each contract c, $P^c := \{P_t^c : t \in T^c\}$, such that

$$P_t^c := \frac{\text{Ask volume}_t^c \cdot \text{Bid price}_t^c + \text{Ask price}_t^c \cdot \text{Bid volume}_t^c}{\text{Ask volume}_t^c + \text{Bid volume}_t^c}$$

following the definition of microprice given by Gatheral and Oomen (2010). Exhibit 3 displays various percentile levels for the median daily difference in the high versus the low in P^c, offering some intuition around daily variability by contract. This stands in contrast to the median daily bid–ask spread of 0.5 bp for each contract throughout 2018.

A second key step is to align and down-sample $\{P^c : c \in [\text{EDc7}, \ldots, \text{EDc15}]\}$. Aligning over index sets T^c is essential given the irregular temporal sampling across contract microprices, and down-sampling conveniently filters for sufficiently small microprice fluctuations, significantly reducing the size of the aligned dataset. We define a cut-off parameter M and set it to 0.1 bp, and for each trading day we construct a time series of curve observations $C := \{\boldsymbol{P}_t = (P_t^c : c \in [\text{EDc7}, \ldots, \text{EDc15}]) : t \in T\}$ for a common temporal index set T, according to the following rules:

- For a new trading day, record an initial curve observation at the first time that a microprice exists for each contract.

- Given an observation time t^*, find the earliest time $t > t^*$ such that $|P_t^c - P_{t^*}^c| \geq M$ for at least one contract c and record an observation of each contract's most recent microprice at or equal to t. If such a time does not exist, revert to 1 for the next trading day.

Finally, we construct a time series $D := \{(\boldsymbol{D}_t, \boldsymbol{Y}_t) = ((\boldsymbol{P}_{t+k} : k \in \{-99, \ldots, 0\}), \boldsymbol{P}_{t+1}) : t \in T\}$ consisting jointly of a rolling window of 100 sequential curve observations[2] and the curve observation at the next event time. Furthermore, for each member of D, we implement a window normalization whereby each microprice subseries in $\boldsymbol{D}_t$ is shifted by its empirical mean within the window and scaled by the standard deviation of the set of all (shifted) window values. Similarly, we then normalize the corresponding members of $\boldsymbol{Y}_t$ using these same statistics (computed over the backward-looking window). To be clear, we note that no step of our data preprocessing introduces look-ahead bias.

Model Specification

Given our dataset construction, we model the prediction target $\boldsymbol{Y}_t$ as a multivariate normal random variable with parameters dependent on our window observations:

$$\boldsymbol{Y}_t \sim \text{MVN}(\mu(\boldsymbol{D}_t), \Sigma(\boldsymbol{D}_t))$$

For brevity, we continue by writing $\mu_t := \mu(D_t)$ to denote the mean vector of dimension $1 \times c$ and $\Sigma_t^A := \Sigma(\boldsymbol{D}_t)$ to denote the variance–covariance matrix of dimension $c \times c$ (with a superscript A to denote aleatoric uncertainty; Kendall and Gal 2017).

We consider two model architectures for estimating μ_t and Σ_t^A. With respect to μ_t, we are inspired to adapt the CNN-LSTM Inc of Zhang, Zohren, and Roberts (2018b), and we seek to compare trading strategy performance to a more straightforward two-hidden-layer multilayer perceptron (MLP). Further details on the architecture are offered in "Modeling Notes" in the appendix, and a schematic of the model architecture is depicted in Exhibit 4. A key point of interest, and novel for financial applications, is the network branching, whereby we estimate our targets over two output heads. We recall Nix and Weigend (1994), who appended an auxiliary output to a network architecture to yield a heteroskedastic variance estimate in conjunction with predicting the mean of a univariate Gaussian. This was extended recently for the multivariate case (Dorta et al. 2018). Requiring that Σ_t^A be a symmetric positive definite matrix, its inverse is also symmetric positive definite and can be expressed as its Cholesky decomposition. Hence, writing $(\Sigma_t^A)^{-1} = \boldsymbol{L}_t \boldsymbol{L}_t^T$ for a lower triangular matrix $\boldsymbol{L}_t$, as per Dorta et al. (2018), and with minor notional updates for our specific problem, the network loss function $\mathcal{L}$ has been shown to be

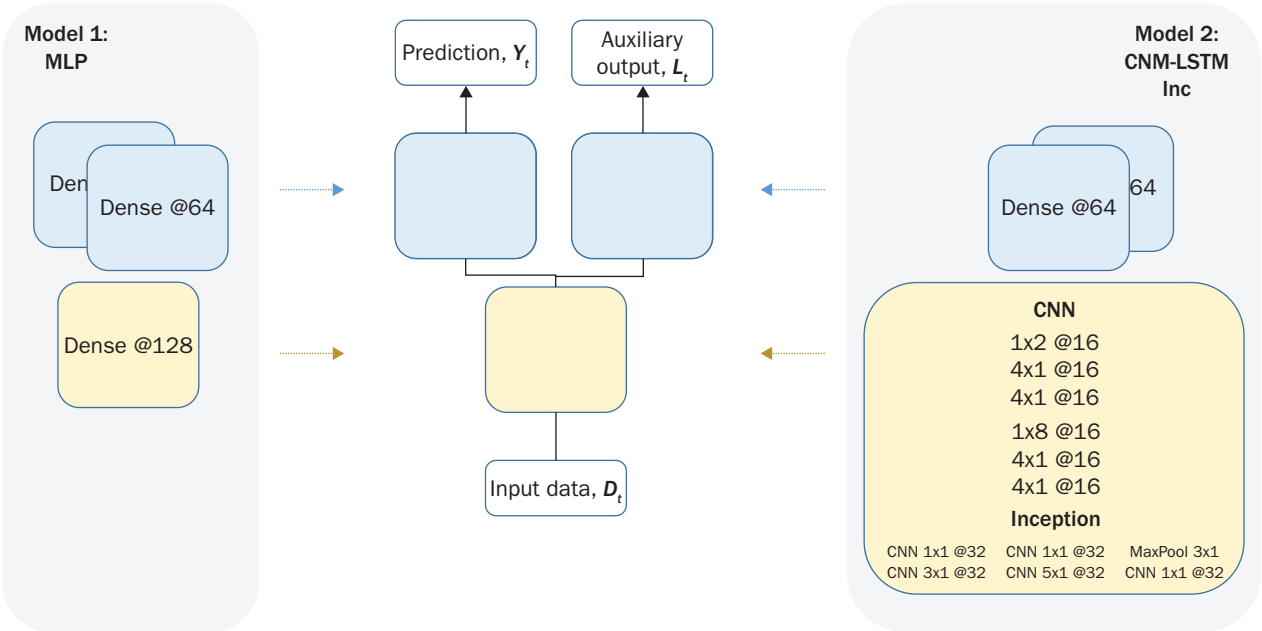
$$\mathcal{L} = -2\Big[\sum_{i=1}^{c} \log l_t^{ii}\Big] + (\boldsymbol{Y}_t - \mu_t)^T \boldsymbol{L}_t \boldsymbol{L}_t^T (\boldsymbol{Y}_t - \mu_t) \tag{1}$$

Here l_t^{ii} denotes the ith diagonal entry of $\boldsymbol{L}_t$. It is interesting to note that Σ_t^A is implicitly learned through the loss function, without reference to explicitly labeled variance–covariance data. We estimate $\boldsymbol{L}_t$ as output to each model, applying a linear activation function after the final layer. The diagonal elements are then exponentiated, ensuring the positive definiteness and hence invertibility of $(\Sigma_t^A)^{-1}$ so that Σ_t^A is recoverable.

[2] The value 100 is chosen for comparison with work by Zhang, Zohren, and Roberts (2018b) and has been seen to work well in practice. Note that no extensive optimization of this parameter has been performed here.

EXHIBIT 4

Model Architectures for Predicting Curve Moves with Uncertainty



NOTES: The MLP specification was chosen such that it had a similar number of parameters to the CNN-LSTM Inc—the latter model has nearly 100,000 parameters, while the former has about 133,000. Anecdotally, we determined that the branching architecture, featuring a common module before branching (shaded yellow above), improved fit significantly relative to the exclusion of a common module.

There are various proposals for approximating epistemic uncertainty for deep neural network models, including pseudo-Bayesian approximations. One popular technique applies to models estimated with dropout. Recall that dropout was first presented by Hinton et al. (2012) as a regularization method in an attempt to prevent neural network overfitting and improve model performance on out-of-sample data. The intuition for implementing dropout is to randomly omit some proportion of hidden weights (by setting them to 0) for each training iteration of a network and to rescale the final model weights by this proportion at test time. This was originally interpreted as yielding an effect similar to calculating an ensemble of models (depending on which parameters were omitted) and taking their average result as a better predictor than any one model on out-of-sample data. In the context of modern deep learning, dropout is popular among some practitioners for its perceived regularization benefit and for its relative ease of implementation. The key to retrieving the epistemic uncertainty approximation is the technique of dropout sampling (Gal and Ghahramani 2016c). This technique is applied by generating many model predictions for a single input datum with random dropout sampling enabled at test time. It has been shown that one can naturally retrieve an approximation to epistemic uncertainty for predictions, calculating the estimate in a straightforward manner by averaging (Gal and Ghahramani 2016c).

Hence, we have that, for N stochastic dropout samples associated with a single model prediction, our parameter estimates are calculated as

$$\widehat{\mathbb{E}\boldsymbol{Y}_t} := \hat{\mu}_t = \frac{1}{N}\sum_{n=1}^{N}\hat{\mu}_{t,n} \qquad \text{and} \qquad \hat{\Sigma}_t^A = \frac{1}{N}\sum_{n=1}^{N}\hat{\Sigma}_{t,n}^A$$

with the expectation estimate according to Gal and Ghahramani (2016c) and where the covariance estimate conforms to Russell and Reale (2019). Again, as done by Russell and Reale (2019), an estimate of the epistemic uncertainty of the prediction is given by

$$\hat{\Sigma}_t^E = \frac{N}{N-1} \cdot \left(\frac{1}{N} \sum_{n=1}^{N} \hat{\mu}_{t,n} \hat{\mu}_{t,n}^T - \hat{\mu}_t \hat{\mu}_t^T \right)$$

where we have made an additional adjustment to yield an unbiased sample covariance estimate. Hence, the total prediction uncertainty estimate can be written

$$\widehat{\mathrm{Var}(\boldsymbol{Y}_t)} := \hat{\Sigma}_t = \hat{\Sigma}_t^A + \hat{\Sigma}_t^E$$

Trading Strategy

In this section, we present a trading strategy that depends on our model outputs $\hat{\mu}_t$ and $\hat{\Sigma}_t$ for selecting α_t, the investment size for a particular trade at each decision time t.

In the literature, a typical backtest given a classification or regression model often amounts to selecting a trading decision based on one of three choices: sell one unit, do nothing, or buy one unit, respectively corresponding to $\alpha \in \{-1, 0, 1\}$. However, a more flexible range of permissible investment sizes, such as the case of continuous $\alpha \in [-1, 1]$, could lead to trading strategies showing nontrivial improvement across real-world performance outcomes. When interpreting $\hat{\mu}_t$ and $\hat{\Sigma}_t$ as true parameters of the underlying return distribution, one can intelligently scale relative investment sizing depending on a predefined trading objective. An example objective that we present is to maximize the out-of-sample Sharpe ratio trading performance metric. Assuming additive rather than multiplicative returns, and further assuming the simplified setting whereby trade opportunities occur a uniformly equal number of times for each estimated pair of distribution parameters, we show in "Choosing α—Sketch Intuition" in the appendix that the optimal investment size is given by

$$\alpha_{t,i} = \frac{\hat{\mu}_{t,i}}{\hat{\sigma}_{t,i}^2}$$

Here, $\mu_{t,i}$ denotes the ith element of $\hat{\mu}_t$, and $\hat{\sigma}_{t,i}$ denotes the square root of the ith diagonal element of $\hat{\Sigma}_t$.

To assist intuition, we chart the corresponding α-surface in the right-hand panel of Exhibit 5, in which we have made the additional arbitrary choice to rescale the surface such that the point $(\mu, \sigma) = (0.3, 0.1)$ corresponds to $\alpha = 1$.

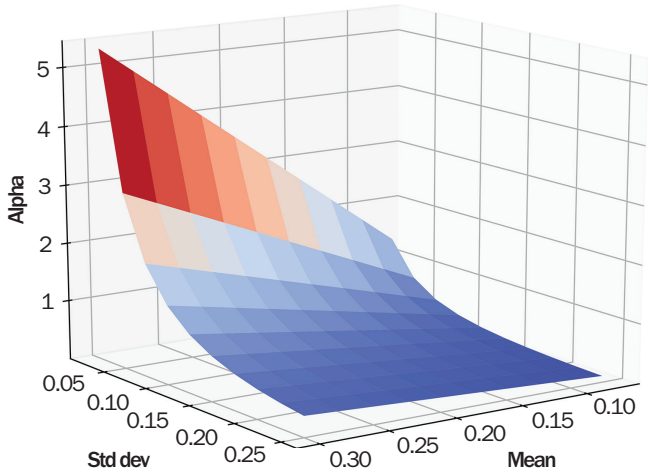
There are two pleasing outcomes of this approach relative to Zhang, Zohren, and Roberts (2018a), who performed early work using an approximation to Bayesian uncertainty estimation for trading given the predictions of a deep classification model. First, we do not increasingly penalize a trade based on increased uncertainty alone but, rather, based on risk-adjusted returns. This corresponds to real life, in which high uncertainty for a trade is not necessarily undesirable outright—a trade proposition should have its estimated uncertainty considered in conjunction with the expected return size. Second, we do not depend on an ad hoc rules-based approach for setting α_t and, instead, select it in an arguably more intuitively pleasing way.

EXHIBIT 5

Measures of Model Fit (left); Scaled α-Surface (right)

	Loss		Mean (Std dev.) SE	
Month	MLP	CLI	MLP	CLI
7	–4.91	**–6.74**	**0.483*** (2.335)	0.491 (2.275)
8	–1.83	**–5.64**	0.744 (2.691)	**0.719*** (2.482)
9	–1.83	**–4.93**	**0.738*** (2.593)	0.756 (2.474)
10	–1.22	**–4.33**	0.647 (2.141)	**0.615*** (2.012)
11	–4.08	**–6.66**	0.519 (2.093)	**0.474*** (2.011)
12	–	–	–	–
Avg	–2.77	**–5.66**	0.626 (2.371)	**0.611** (2.251)

*p ≤ 0.0001 for each one-tailed test of least model MSE by month.



NOTES: **Left:** Validation loss and standard error statistics by month for MLP and CNN-LSTM Inc models. Each model is estimated using the full year's data, up to but excluding the validation month. **Right:** α-surface of the second section, with α set to 1 for $(\mu, \sigma) = (0.3, 0.1)$.

RESULTS

Given that we have one full year of data, we partition by month and evaluate and verify our models by training on months 1 to $M - 1$, validating using month M, and finally testing our trading strategy using month $M + 1$, for $M \in \{7, 8, 9, 10, 11\}$. This designation is arbitrary but somewhat logical within, say, a trading platform that may be updated on a monthly basis.

All models are estimated using Keras (Chollet 2015). We trained with the objective of minimizing validation loss and implemented an early stopping rule with a patience of 15, restoring the best model weights. Further modeling details are given in "Modeling Notes" in the appendix. Of important note, we trained the MLP and CNN-LSTM Inc with diagonal variance–covariance matrixes to compare to the proposed full variance–covariance case. Also, for practicality, we tuned an L2 weight regularization parameter and selected a uniform dropout rate by grid search over the validation data for the MLP with diagonal covariance model. Hence, these parameter values were, respectively, set to 1e-8 and 0.1 for each of the four models.[3] Other important details of a linear Bayesian baseline model, which we offer as a relevant benchmark when discussing out-of-sample trading performance, are included in "Modeling Notes" in the appendix.

In the left-hand table of Exhibit 5, we see that the validation data loss is uniformly lower for CNN-LSTM Inc relative to the MLP model, on average differing by a factor of 2.0. On a mean-square-error basis, the CNN-LSTM Inc outperforms the MLP for three out of five months for a one-tailed *t*-test of least MSE, with *p*-values ≤ 0.0001 in each case. On average across all months, the CNN-LSTM Inc MSE is 2.5% smaller. The outperformance of the CNN-LSTM Inc model in these respects supports the claim of the usefulness of this model for the prediction task.

[3]Where applicable, dropout layers are included as per Gal and Ghahramani (2016c); for convolutional and recurrent layers we follow Gal and Ghahramani (2016b) and Gal & Ghahramani (2016a), with the exception that we add no dropout within the inception layer. We calculate 30 dropout samples for each prediction. Furthermore, we found that trading strategy performance showed a negligible difference when using 60 samples.

EXHIBIT 6

Monthly Sharpe Performance for MLP and CNN-LSTM Inc Models for a Set of Investment Sizing Strategies on Out-of-Sample Data

		MLP				CNN–LSTM Inc			
		Rlsd vol	Base	Alea	*Al + Ep*	Rlsd vol	Base	Alea	*Al + Ep*
	7	–	–	–	–	–	–	–	–
Diagonal	8	0.87	0.81	0.88	**0.89**	**1.01**	0.88	0.97	0.98
Σ	9	1.99	**2.32**	1.98	2.00	1.39	1.64	**1.86**	1.81
	10	1.76	**2.47**	1.96	1.98	1.75	**2.45**	2.25	2.23
	11	**1.33**	1.14	1.21	1.21	**1.31**	1.20	1.23	1.25
	12	2.69	2.90	2.98	**2.99**	2.57	**2.99**	2.95	2.94
	Cuml	1.29	1.40	1.42	**1.44**	1.24	1.27	1.37	**1.38**
	7	–	–	–	–	–	–	–	–
Full	8	0.89	0.83	0.95	**0.96**	0.95	0.82	1.01	**1.02**
Σ	9	1.84	**2.19**	2.12	2.12	1.65	1.85	**2.17**	2.15
	10	1.71	**2.47**	1.84	1.88	1.73	**2.46**	2.45	2.41
	11	**1.34**	1.12	1.24	1.24	**1.27**	1.15	1.18	1.20
	12	2.65	2.87	2.87	**2.87**	2.57	**2.94**	2.91	2.91
	Cuml	1.25	1.38	1.39	**1.42**	1.27	1.26	**1.42**	1.41

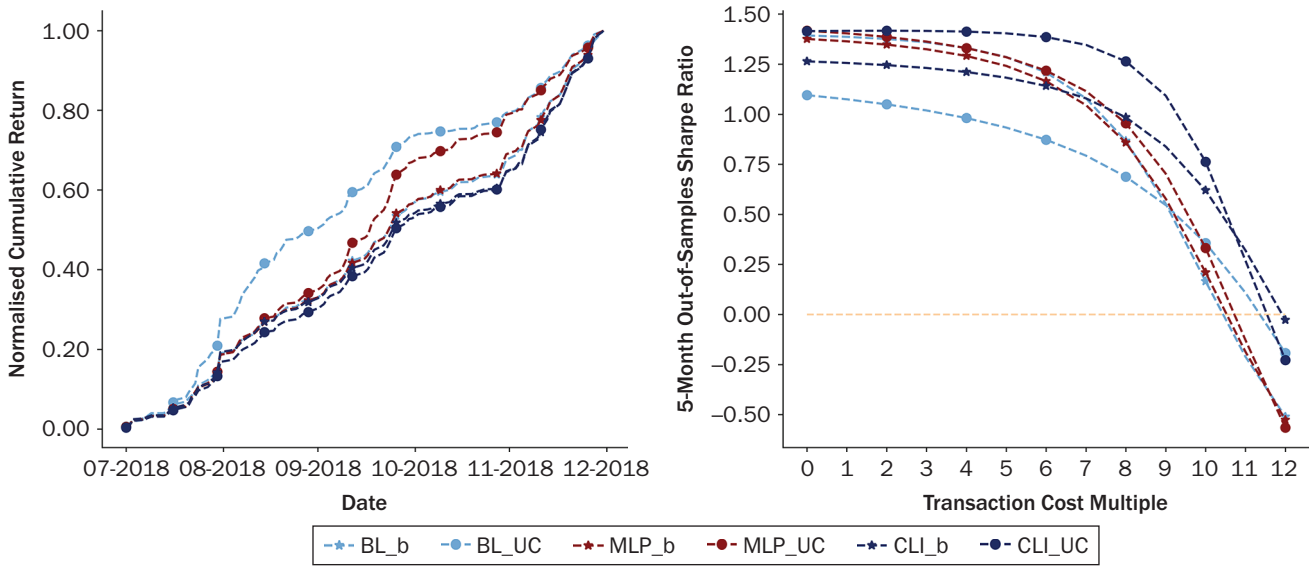
NOTES: Cumulative five-month Sharpe is shown in the last row of each table. Investment size is scaled by realized window volatility (Rlsd vol), the aleatoric uncertainty estimate (Alea), and the sum of aleatoric and epistemic uncertainty (*Al + Ep*). Base has no investment sizing by uncertainty.

Next, we discuss the results for the trading strategy described in "Trading Strategy," focusing on the monthly and cumulative Sharpe performance on out-of-sample data. First, for all strategies, we note that we have defined a trading threshold of 0.1 bp such that any trade entered must have a predicted (absolute) asset price change greater than or equal to the threshold. This threshold was chosen to be equal to the cutoff parameter defined in "Data Commentary." Next, we define a strategy called *Base* that enters a trade position $\alpha \in \{-1, 1\}$ depending on the sign of the predicted price change (respectively, lower or higher, and corresponding to short or long). We also analyze a trading strategy that takes the notion of prediction uncertainty into account as proposed in "Trading Strategy" and explore two additional substrategies whereby we estimate uncertainty by alternative methods. For the strategy *Rlsd vol*, the realized volatility of the input data window is used as a proxy for uncertainty and serves as an alternative benchmark. For *Alea*, aleatoric uncertainty is instead used, and for *Al + Ep*, we use the sum total of aleatoric and epistemic uncertainty as per "Model Specification." We do not present results for an analogous strategy based on epistemic uncertainty alone, on finding cases of unreasonably large α corresponding to an uncertainty estimate for a particular prediction very close to zero.

The out-of-sample Sharpe ratios are shown for the performance of each model by investment sizing strategy in Exhibit 6. For comparison, we show the results for both the diagonal (top row) and full (bottom row) variance–covariance model variants. For three of the four subtables, we notice that the cumulative five-month Sharpe performance is greatest for the *Al + Ep* strategy, with Al second greatest in each case (with outperformance between 0.7% and 2.2%). For the fourth subtable, Al is greatest and marginally ahead of *Al + Ep* (by 0.7%). More strikingly, the relative outperformance across *Al + Ep* strategies compared with Base ranges from 2.9% to 12.7%, and we observe a minimum 11.3% outperformance relative to Rlsd vol. These results offer a preliminary demonstration of the utility of incorporating estimates of

EXHIBIT 7

Cumulative Returns Paths (left); Strategy Sharpe Ratio Given Transaction Costs (right)



NOTES: **Left:** Out-of-sample normalized daily cumulative basis point return for Base and *AI* + *Ep* long–short trading strategies for the baseline, MLP, and CNN-LSTM Inc models. **Right:** Five-month out-of-sample Sharpe ratio for Base and *AI* + *Ep* long–short trading strategies for the baseline, MLP, and CNN-LSTM Inc models, accounting for transaction costs. The *x*-axis varies by multiple of transaction costs, where a multiple of 1 represents a cost of $\frac{1}{20}$ of the trading threshold of 0.1 bp.

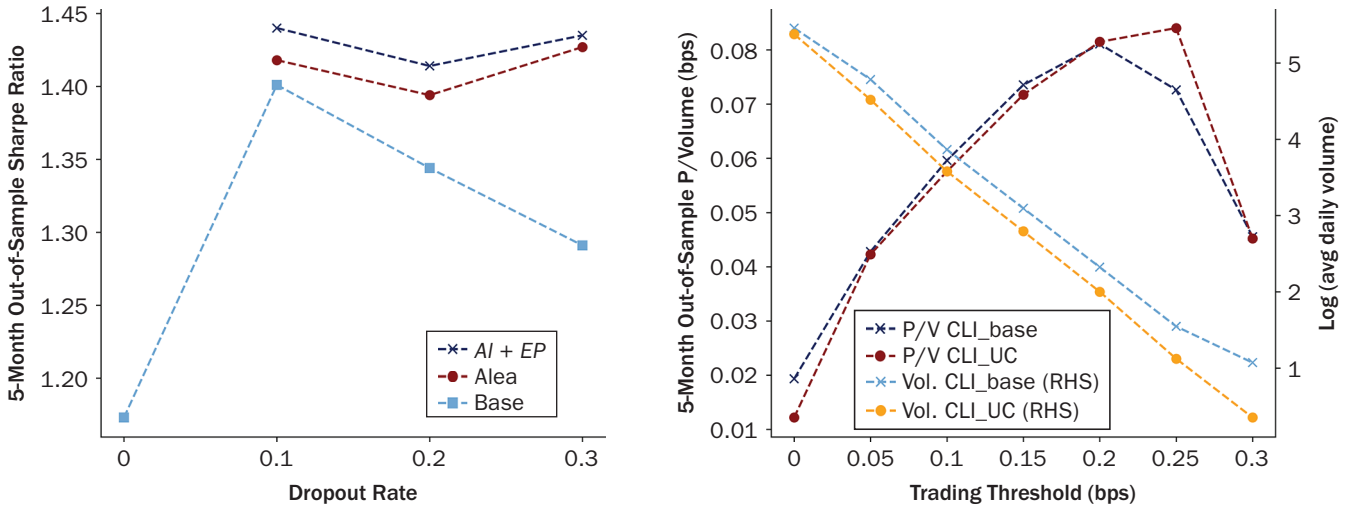
both aleatoric and epistemic uncertainty in the context of investment sizing within our high-frequency trading setting.

Of interest, we highlight both what appears to be the outperformance of the diagonal MLP *AI* + *Ep* strategy versus any other and the apparent strong performance of the Bayesian OLS Base strategy, as shown in the appendix in Exhibit A1. To rationalize this, we first consider the sample paths of returns as in the left-hand panel of Exhibit 7. We observe that the out-of-sample daily cumulative (basis point) return of the presented strategies appears mostly without outliers and exhibits relative smoothness, and we do not attribute any outperformance as being due to any obvious lucky or outsized earnings. Instead, we next consider the stylized impact of transaction costs for comparing model and strategy outperformance.

First, we define an element of total net transaction cost to be equal to 0.005 bp per unit volume. Based on our trading threshold of 0.1 bp, this corresponds to 5% of the minimum prediction size such that $|\alpha_t| > 0$ for any trade. We accrue the transaction cost in proportion to volume transacted at each time step. To be clear, we only trade the delta adjustment to achieve the required position of size α_t for the specific asset, relative to the existing position on the day's most recent trade time. From the right-hand panel of Exhibit 7, we see the five-month out-of-sample Sharpe ratio for Base and *AI* + *Ep* long–short trading strategies for the linear baseline MLP and CNN-LSTM Inc models, accounting for multiples from 0 to 12 of our defined element of transaction cost. To be clear, the label 0 denotes the absence of transaction costs, and the upper label of 12 is the minimal integral multiple of transaction costs such that the Sharpe ratio for each presented strategy is negative. From this exhibit, it is immediately clear that the CNN-LSTM Inc model with uncertainty outperforms all other models at most levels of transaction cost and that the outperformance is greatest at the middle of the range. This analysis contributes somewhat to understanding the relative utility across alternative models and strongly vindicates the use of CNN-LSTM

EXHIBIT 8

Sharpe Ratio by Dropout Rate (left); Performance by Trading Threshold (right)



NOTES: **Left:** Five-month out-of-sample Sharpe ratio for trading strategy based on long–short positions without uncertainty (Base), with aleatoric uncertainty only (Alea), and using our full uncertainty estimate (*AI* + *Ep*), for varying dropout rates, given a diagonal covariance MLP model. **Right:** Five-month out-of-sample profit over volume (left-hand axis) for varying trading thresholds (*x*-axis) for the full covariance CNN-LSTM Inc model. The right-hand axis is the log average daily volume traded for each trading threshold.

Inc with uncertainty for trading. Of course, the careful reader may notice that the transaction costs analyzed here are significantly less than the median daily bid–ask spread of 0.5 bp quoted in "Data Commentary." With respect to this observation, it is important to understand that the approach presented may not necessarily be traded outright as a stand-alone strategy in practice. The utility of the approach may be derived from its addition to a broader market-making or execution strategy, in which trading is occurring anyway. Hence, a deeper understanding of the impact of transaction costs, especially relative to the bid–ask spread, may depend on a more realistic and sophisticated trading framework. We leave these considerations for future work.

Other points of interest around our backtest include observing how the choice of dropout rate or trading threshold affects out-of-sample performance. In the case of varying the dropout rate, for the diagonal MLP, we tested the cumulative five-month Sharpe ratios for varying dropout rates for Base, *AI* + *Ep*, and *AI*, and we chart the results in the left-hand panel of Exhibit 8. Interestingly, we see that for either of the two trading strategies using uncertainty, the Sharpe ratios are relatively stable, with a range of less than 1.8% each across our three tested nonzero dropout rates. However, similar stability is clearly not evident in the case of the Base strategy, which achieves peak Sharpe performance at a dropout rate of 0.1 but displays a sharp decrease in performance for increasing dropout rates and when dropout is not used at all. One may wonder about the extent to which this trading performance stability for varying dropout rates might be observed in increasingly sophisticated models such as CNN-LSTM Inc. If so, this could dominate the need for a large degree of tuning of the dropout rate when estimating the model—an increasingly resource-consuming task with such model complexity. We leave this as a consideration for future work.

Next, we consider the case of varying the trading threshold and refer to our results in the right-hand panel of Exhibit 8. Here, we see five-month out-of-sample profit over volume (left-hand axis) for varying trading thresholds (*x*-axis) for the full covariance CNN-LSTM Inc model. This statistic can be alternatively recognized as the breakeven net transaction cost for a strategy, with higher values corresponding to strategies

with greater average net profit per trade. It is interesting to note that we can optimize the threshold to maximize the Sharpe ratio, although we leave further consideration of this to the interested practitioner. For added interest, we also plot the log average daily volume traded for each trading threshold on the right-hand axis and see that it is approximately log linearly decreasing with an increasing threshold.

Finally, in the right-hand panel of Exhibit A1 in the appendix, we plot the estimated model reward–risk ratio versus the realized monthly Sharpe ratio on out-of-sample data for the diagonal MLP. In that this plot is sufficiently representative of each of the four models, we infer that, despite the obvious underperfomance in the tails (where sample sizes are relatively small anyway) the models appear somewhat sensibly calibrated, although we leave any other tuning in this respect to future work.

CONCLUSION

We present a model for price change prediction of the Eurodollar futures curve, as a function of a recent history of asset price data, within a high-frequency domain. In doing so, we extend the existing state-of-the-art model deep learning architecture of Zhang, Zohren, and Roberts (2018b) to multivariate, correlated price data and improve price prediction relative to benchmark models for small time horizons and in a regression setting. Importantly, we develop a preliminary analysis describing the estimation of deep learning model–generated uncertainty for financial prediction and further show how these uncertainties can be useful for investment size scaling within trading strategies. Future work could improve on this analysis, with much of it potentially important for applications at an industrial scale. With respect to data, a larger subset of available Eurodollar futures contracts could be used, as could alternative correlated asset prices or economic information. One could also include deeper order book data, such as Level 2 quotes. With respect to the models used, there is more to be done in optimally engineering model architecture for aleatoric uncertainty estimation. Alternative methods for estimating epistemic uncertainty could be explored. On one hand, concrete dropout could be attempted for automating the tuning of the dropout rate (Gal, Hron, and Kendall 2017). On the other, Hamiltonian Monte Carlo methods could be explored as an alternative to estimating epistemic uncertainty and potentially applied, for example, with hamiltorch (Cobb et al. 2019b). Finally, with respect to the realities of implementing a trading strategy in practice, we allude to three main avenues of further research: a deeper appreciation of the impact of transaction costs beyond our preliminary analysis, a return to fine tuning the dropout rate if performance metrics around trading are relatively stable, and further consideration of model calibration.

APPENDIX

CHOOSING α—SKETCH INTUITION

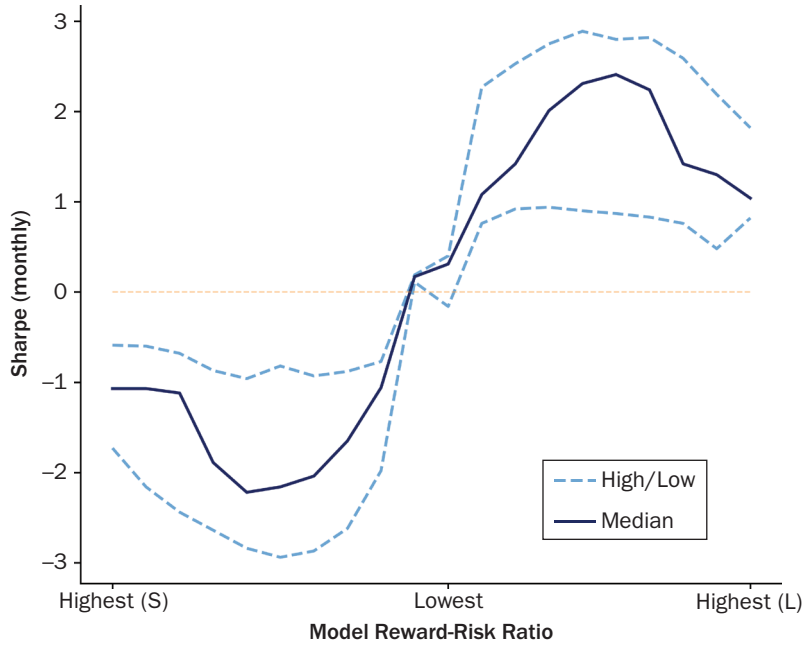
Assume we are given N trading opportunities $\{X_i\}$ with normally distributed and uncorrelated returns, parameterized by means and variances given by (μ_i, σ_i^2) for each choice i. Assume also that these parameters take strictly positive values and that we trade each opportunity an equal number of times. We seek to trade an optimal amount α_i of X_i to maximize the Sharpe ratio S of the return. We write

$$S = \frac{\sum \alpha_j \mu_j}{\sqrt{\sum \alpha_j^2 \sigma_j^2}}$$

EXHIBIT A1

Baseline Model Performance (left); Model Sharpe Ratio for Various Levels of Predicted Risk-Reward Ratio (right)

		Bayesian OLS	
		Base	*AI + EP*
	7	–	–
Full	8	0.90	0.85
Σ	9	2.23	1.46
	10	2.51	1.95
	11	1.32	0.90
	12	2.69	2.93
	Cuml	1.39	1.10



NOTES: **Left:** Monthly Sharpe ratio performance for the baseline Bayesian OLS model, in analogy with Exhibit 6, for estimating the full covariance matrix. Although the base strategy appears to show performance comparable to that of the more sophisticated models, it does not appear to generally improve when taking uncertainty into account. **Right:** Median out-of-sample model-predicted risk–reward ratio (for months 8 to 12 of out-of-sample data) versus monthly Sharpe ratio for a diagonal covariance MLP. The greatest long (short) ratio is shown to the right (left). Here, out-of-sample Sharpe performance for the short positions is multiplied by –1.

Taking the derivative of S with respect to α_i yields

$$\frac{dS}{d\alpha_i} = \frac{\mu_i \sum_{j\neq i} \alpha_j^2 \sigma_j^2 - \alpha_i \sigma_i^2 \sum_{j\neq i} \alpha_j \mu_j}{\left(\sum \alpha_j^2 \sigma_j^2\right)^{3/2}}$$

which equals 0 for

$$\alpha_i^* = \frac{\mu_i / \sigma_i^2}{\sum_{j\neq i} \alpha_j^* \mu_j / \sum_{j\neq i} \alpha_j^{2*} \sigma_j^2}$$

One can see that a solution for each i is given by

$$\alpha_i^* = \frac{\mu_i}{\sigma_i^2}$$

Taking the second derivative of S with respect to α_i yields

$$\frac{d^2S}{d\alpha_i^2} = \frac{-\sigma_i^2 \left(\sum_{j\neq i} \alpha_j \mu_j \sum \alpha_j^2 \sigma_j^2 + 3\alpha_i \mu_i \sum_{j\neq i} \alpha_j^2 \sigma_j^2 - 3\alpha_i^2 \sigma_i^2 \sum_{j\neq i} \alpha_j \mu_j\right)}{\left(\sum \alpha_j^2 \sigma_j^2\right)^{5/2}}$$

Substituting α_i^* we have that

$$\frac{d^2S}{d\alpha_i^2}\Big|_{\alpha_i=\alpha_i^*} = \frac{-\sigma_2^2\left(\sum_{j\neq i}\mu_j^2/\sigma_j^2 \sum \mu_j^2/\sigma_j^2 + 3\mu_i^2/\sigma_i^2 \sum_{j\neq i}\mu_j^2/\sigma_j^2 - 3\mu_i^2/\sigma_i^2 \sum_{j\neq i}\mu_j^2/\sigma_j^2\right)}{\left(\sum \mu_j^2/\sigma_j^2\right)^{5/2}}$$

$$= \frac{-\sigma_2^2 \cdot \sum_{j\neq i}\mu_j^2/\sigma_j^2}{\left(\sum \mu_j^2/\sigma_j^2\right)^{3/2}} < 0$$

Hence, for this optimal α_i^* the Sharpe ratio is indeed maximized.

ADDITIONAL DATA COMMENTARY

Futures Dataset

The set of 44 Eurodollar futures contracts consists of 40 quarterly contracts and four serial contracts; the former is based on a quarterly cycle (beginning in March), and the serial contracts are for the first four closest months (with respect to roll dates) outside of that cycle. Note that by this definition, the serial contracts must be contained in EDc1 to EDc6. Hence, depending on the month of the year, our focus on EDc7 to EDc15 represents a segment of an interest rate curve starting seven, eight, or nine months in the future, as well as the following two years.

Microprice

There are well-justified reasons to model using the microprice—including its utility in excess of using the simpler midprice for capturing an element of volume imbalance between each side of the order book. In our case, we also consider microprices from a practical point of view, given data quality.

Dataset Size

We collected 25 GB of raw data as .csv files for our target contracts and prepared just over 50 GB in .h5 files at the conclusion of preprocessing (and accounting for windows of input data).

MODELING NOTES

Initial Exploration

We compare a suite of models for predictive performance on out-of-sample data to develop intuition about suitable rates curve model architectures. This is independent of considering any uncertainty measures. Listing these models by increasing complexity, we explore simple and exponentially weighted averaging, ordinary least squares, principal component analysis, single- and multilayer perceptrons, CNN, LSTM, CNN-LSTM, and finally a CNN-LSTM Inc in the spirit of Zhang, Zohren, and Roberts (2018b). We evaluate model fit by comparing mean-squared error and Huber loss (Huber 1964) on our validation data, whereby the proceeding observations were broadly consistent. The first clear result is that, for the $t + 1$ prediction horizon, there is a clear trend of significantly decreasing prediction error as model complexity increases. Furthermore, the CNN-LSTM Inc always outperforms every other model for both error types, which complements the results of Zhang, Zohren, and Roberts (2018b). Second, each model broadly displays larger error for larger prediction horizons, which accords with intuition. Considering the $t + 100$ prediction

horizon, there is such little variation in error across rows that it is clear that not only do we have a poor ability to forecast price action relatively far into the future, but all models appear to be of similarly little utility. An inflection point for these observations seems apparent at the $t + 10$ prediction horizon.

CNN-LSTM Inc Model Details

The model architecture is as shown in Exhibit 4, and we follow our reference closely because we have no strict advantage in selecting this beyond intuition. A key difference for our setting, however, is that we do not include any information from the order book beyond Level 1, and the spatial component analogue for our model is equal to adjacent asset prices on the curve, rather than levels of limit order book data on the single asset. Another difference is that our reference work's initial three convolutional layers, which convolve price and volume levels at different depths of the limit order book, are not relevant in our case and so are excluded. To recapitulate some of the other details, we use six convolutional layers of 16 filters each and with kernel sizes, respectively, of (1,2), (4,1), (4,1), (1,8), (4,1), (4,1). All layers have a leaky ReLU output with parameter $\alpha = 0.01$. The same padding is applied to the latter two layers. The inception layer is a 2×3 tower whereby the input layer consists of a MaxPooling layer with a pool size of (3,1) and a stride of one and two convolutional layers with a kernel size of (1,1) and 32 filters. The output layer of the inception network is three more convolutional layers with kernel sizes of (1,1), (3,1), and (5,1), respectively, all with 32 filters. All inception modules have the same padding, and the convolutional layers have ReLU activation. The LSTM layers have 64 units, with a subsequent dense layer, and linear output activation function. Finally, for replicability's sake, we add that the MLP model uses ReLU activations at all dense layers but the last, in which a linear activation function is instead implemented.

Bayesian Linear Baseline Model Details

We follow the Bayesian framework for multioutput regression, outlined in "Additional Commentary," of Lebrun (2012) in estimating a suitable baseline model for comparing our results. Similar to the section "Model Specification," we model $\boldsymbol{Y}_t \sim \text{MVN}(\mu_t, \Sigma)$ where $\mu_t = \boldsymbol{D}_t\beta$ is an affine function. (Hence, we have adjusted our earlier definition of D_t by appending a constant term.) We make the natural assumption that the prior distribution for $\boldsymbol{\Sigma}$ is inverse-Wishart, that is, $\boldsymbol{\Sigma} \sim W^{-1}(\Omega, \nu_0)$. We further assume that the prior for $\beta|\boldsymbol{\Sigma}$ is matrix normal, that is, $\beta|\boldsymbol{\Sigma} \sim N_{(100c+1)\times c}(\beta_0, \Sigma, \Sigma_0)$. Hence, the posterior predictive of $\boldsymbol{Y}_t$ given n historical data observations $\mathcal{D} := (\boldsymbol{D},\boldsymbol{Y}) = \{(\boldsymbol{D}_i, \boldsymbol{Y}_i) : 1 \le i \le n\}$, and a single new observation $\boldsymbol{D}_t$ can be shown to be a multivariate Student distribution:

$$\boldsymbol{Y}_t \mid \boldsymbol{D}_t, \mathcal{D} \sim T((\Omega + \boldsymbol{A}^*).C^{-1}, n + \nu_0)$$

for matrixes $\boldsymbol{A}^* = \boldsymbol{Y}^T\boldsymbol{Y} + \beta_0{}^T\Sigma_0{}^{-1}\beta_0 - (\boldsymbol{D}^T\boldsymbol{D}\hat{\beta} + \Sigma_0{}^{-1}\beta_0)^T(\boldsymbol{D}^T\boldsymbol{D} + \Sigma_0{}^{-1})^{-1}(\boldsymbol{D}^T\boldsymbol{D}\hat{\beta} + \Sigma_0{}^{-1}\beta_0)$ and $\hat{\beta} = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\boldsymbol{Y}$ and scalars $C^{-1} = 1 + \boldsymbol{D}_t(\boldsymbol{D}^T\boldsymbol{D} + \Sigma_0{}^{-1})^{-1}\boldsymbol{D}_t^T$ and $\nu_0 = n_0 - (c + (100c + 1)) + 1$. Hence, the required output prediction and uncertainty estimates can be written

$$\mathbb{E}[\widehat{\boldsymbol{Y}_t \mid \boldsymbol{D}_t}, \mathcal{D}] = \boldsymbol{D}_t(\boldsymbol{D}^T\boldsymbol{D} + \Sigma_0{}^{-1})^{-1}(\boldsymbol{D}^T\boldsymbol{Y} + \Sigma_0{}^{-1}\beta_0), \qquad \text{and}$$

$$\text{Var}[\widehat{\boldsymbol{Y}_t \mid \boldsymbol{D}_t}, \mathcal{D}] = \frac{1}{n + \nu_0 - 2}(\Omega + \boldsymbol{A}^*).C^{-1}$$

Other Notes

We implement all models in Keras (Chollet 2015). We use the Adam optimization algorithm when training models (Kingma and Ba 2016). We batch-train our models whereby

each batch contains 1,024 samples, with the whole year of data consisting of about 7,450 batches. All results were generated using Tensorflow-GPU 1.9.0 on a six-core Xeon W-2133 3.2GHz/12GB NVIDIA GTX 1080 ti/64 GB RAM machine.

ACKNOWLEDGMENTS

The authors would like to thank the Oxford-Man Institute (OMI) of Quantitative Finance for its generous support, including data access. Trent Spears would further like to thank Dr. Jan-Peter Calliess and Prof. Nir Vulkan for their helpful insights and guidance, as well as the student members of the OMI, especially Bryan Lim, for their suggestions and encouragement.

REFERENCES

Borovkova, S., and I. Tsiamas. 2019. "An Ensemble of LSTM Neural Networks for High-Frequency Stock Market Classification." *Journal of Forecasting* 38 (6): 600–619.

Chen, J., W. Chen, C. Huang, S. Huang, and A. Chen. 2016. "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks." In *7th International Conference on Cloud Computing and Big Data (CCBD)*.

Choey, M., and A. S. Weigend. "Nonlinear Trading Models through Sharpe Ratio Maximization." In *International Journal of Neural Systems*, pp 417–431. Singapore: World Scientific, 1997.

Chollet, F. "Keras." 2015. https://github.com/fchollet/keras.

Cobb, A. D., A. G. Baydin, A. Markim, and S. J. Roberts. 2019b. "Introducing an Explicit Symplectic Integration Scheme for Riemannian Manifold Hamiltonian Monte Carlo." *arXiv* 1910.06243.

Cobb, A. D., M. D. Himes, F. Soboczenski, S. Zorzan, M. D. O'Beirne, A. G. Baydin, Y. Gal, S. D. Domagal-Goldman, G. N. Arney, and D. Angerhausen. 2019a. "An Ensemble of Bayesian Neural Networks for Exoplanetary Atmospheric Retrieval." https://arxiv.org/abs/1905.10659.

Donnelly, B. *The Art of Currency Trading: A Professional's Guide to the Foreign Exchange Market*. Hoboken: Wiley, 2019.

Dorta, G., S. Vicente, L. Agapito, N. Campbell, and I. Simpson. 2018. "Structured Uncertainty Prediction Networks." In *IEEE Conference on Computer Vision and Pattern Recognition 2018*. Piscataway, NJ: IEEE.

Gal, Y., and Z. Ghahramani. 2016a. "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks." In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 1027–1035.

——. "Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference." 2016b. https://arxiv.org/abs/1506.02158.

——. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." 2016c. https://arxiv.org/abs/1506.02142.

Gal, Y., J. Hron, and A. Kendall. 2017. Concrete Dropout. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3584–3593.

Gatheral, J., and R. Oomen. 2010. "Zero-Intelligence Realized Variance Estimation. *Finance and Stochastics* 14: 249–283.

Gonzalvez, J., E. Lezmi, T. Roncalli, and J. Xu. 2019. "Financial Applications of Gaussian Processes and Bayesian Optimization." https://arxiv.org/pdf/1903.04841.pdf.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. 2012. "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors." https://arxiv.org/abs/1207.0580.

Huber, P. J. 1964. "Robust Estimation of a Location Parameter." *Annals of Mathematical Statistics* 35: 73–101.

Kendall, A., and Y. Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In *Advances in Neural Information Processing Systems 30*, pp. 5574–5584. Curran Associates, Inc., 2017.

Kingma, D. P., and J. Ba. 2016. "A Method for Stochastic Optimization." https://arxiv.org/abs/1412.6980.

Kondratyev, A. 2018. "Learning Curve Dynamics with Artificial Neural Networks." https://ssrn.com/abstract=3041232.

Lebrun, P. "Bayesian Design Space Applied to Pharmaceutical Development." PhD thesis, Université de Liège, Belgique, 2012.

Leibig, C., V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. 2017. "Leveraging Uncertainty Information from Deep Neural Networks for Disease Detection." *Scientific Reports* 7: 17816.

Lim, B., S. Zohren, and S. Roberts. 2019. "Enhancing Time-Series Momentum Strategies Using Deep Neural Networks." *The Journal of Financial Data Science* 1 (4): 19–38.

Nix, D. A., and A. S. Weigend. 1994. "Estimating the Mean and Variance of the Target Probability Distribution." In *Proceedings of 1994 IEEE International Conference on Neural Networks*, pp. 55–60. Piscataway, NJ: IEEE.

Russell, R. L., and C. Reale. 2019. "Multivariate Uncertainty in Deep Learning." https://arxiv.org/abs/1910.14215.

Sharpe, W. F. 1994. "The Sharpe Ratio." *The Journal of Portfolio Management* 21 (1): 49–58.

Towers, N., and A. N. Burgess. 1999. "Implementing Trading Strategies for Forecasting Models." In *Proceedings of Computational Finance*. Cambridge, MA: The MIT Press.

Zhang, Z., S. Zohren, and S. Roberts. 2018a. "BDLOB: Bayesian Deep Convolutional Neural Networks for Limit Order Books." https://arxiv.org/abs/1811.10041.

——. 2018b. "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books." *IEEE Transactions on Signal Processing* 67: 3001–3012.