风险预警师

- Risk Prediction of Range Accrual Note Based on Long Short-Term Memory Model

GU Jingyi / Johns Hopkins University / Master / 2026
LIANG Linzhe / The Chinese university of Hongkong, Shenzhen / Bachelor / 2024
GAN Shulan / New York University / Master / 2026

Table of contents

01

Background

Background

The complex structure of range accrual note

No simple numerical analytical solution and usually use complex numericalx simulation methods (e.g. Monte Carlo simulation)

⬇

Slow calculation speed and insufficient sensitivity to changes in the external environment

⬇

May not be able to instantly and accurately reflect the product's risk exposure

Our Goal	Since the risk exposure of this type of product is particularly characterized by its sensitivity to interest rate fluctuations, we intend to build a neural network model to predict the product's Vega value for the purpose of efficiently and accurately predicting the risk exposure.

02

Modeling Assumptions

Modeling assumptions

Market Assumptions	<u>No Arbitrage</u>: there are no arbitrage opportunities for risk-free profits in the market. <u>Liquidity assumption</u>: assets are assumed to be readily tradable at fair value with no liquidity discounts or premiums. <u>Perfectly Liquid Market</u>: the market is assumed to be sufficiently liquid to ensure that the trading assumptions in the model can be executed instantly without liquidity constraints.
Volatility assumption	<u>Constant volatility</u>: the volatility of the asset is assumed to be static over the life of the model and does not change over time
Financial Product assumption	<u>No early redemption</u>: the model assumes that investors do not choose to redeem the notes before maturity, ignoring the effect of early termination.

03

Data preprocessing & Feature Engineering

Data preprocessing

Missing values:

Filling with 0.

Standardize formats and units:

Unify all time units such as Tenor and Expiry into months.

All Date into datetime64[ns].

Remove String and convert all features used for training to float64.

Merge Data:

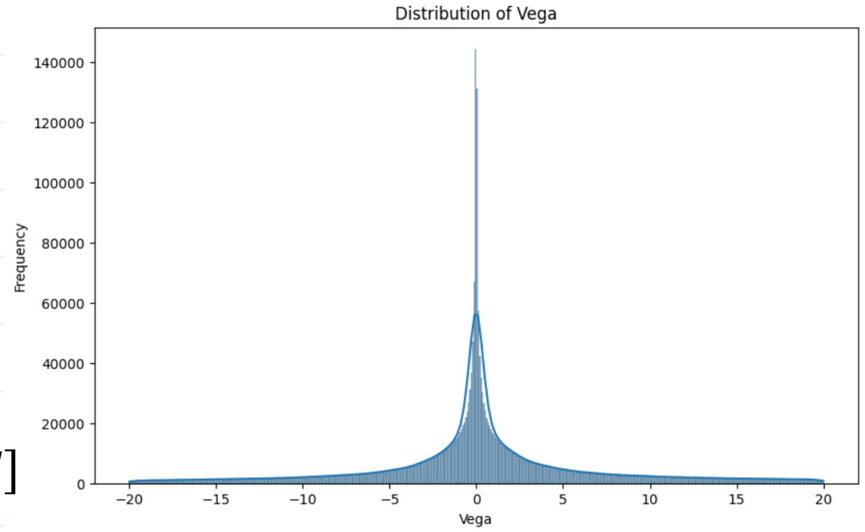
Merge all the dataset together based on ['Date', 'Expiry' , 'Tenor', and 'Trade Name'].

Data preprocessing

Outlier:
Perform outlier values on Vega columns.

We treat the Vega values that do not fall between the 0.01 quantile and the 0.95 quantile as outlier.

$$Vega = \begin{cases} 0.01\,QT. & Vega < 0.01QT \\ Vega. & Vega \in [0.01QT, 0.95QT] \\ 0.95QT. & Vega > 0.01QT \end{cases}$$



Feature Engineering

Here are the new FEATUREs we generated from the existing data:

TV Change Ratio	The rate of change of two quotes of a dummyTrade under a certain Zero Rate Shock
Check Swap Rate	Determine if the swap rate is between [lower_bound, upper_bound]
Vol Std	Calculate the standard deviation of vols under a [Date, Expiry, Tenor]

Data preprocessing

Training & Testing Data:

Split Dimension	Training	Testing
Predicting FUTURE Risks from Past Historical Data (Time-Based)	Before 2024	After 2024
Predicting NEW dummyTrade Risks from Other dummyTrade Data(Trade-Based)	dummyTrade 1-11	dummyTrade 12

Data preprocessing-normalization

- Selection of base values: For each input window, all feature values from the first point in time in the window are selected as base values. This is achieved in the code by `base = window_data[:, 0, :]`, where `base` has the shape `[num_windows, num_features]`.
- Compute the normalized data: Using the vectorization operation, each feature value at each time point in the window is normalized with respect to the base value. Handling cases where the base value is zero: To avoid divide-by-zero errors, the normalized value is set directly to zero when the base value is zero. The formula for calculating the normalization is as follows:

$$\text{normalized\_data} = \begin{cases} \frac{\text{window\_data}-\text{base}}{\text{base}} & \text{if base} \neq 0 \\ 0 & \text{if base} = 0 \end{cases}$$

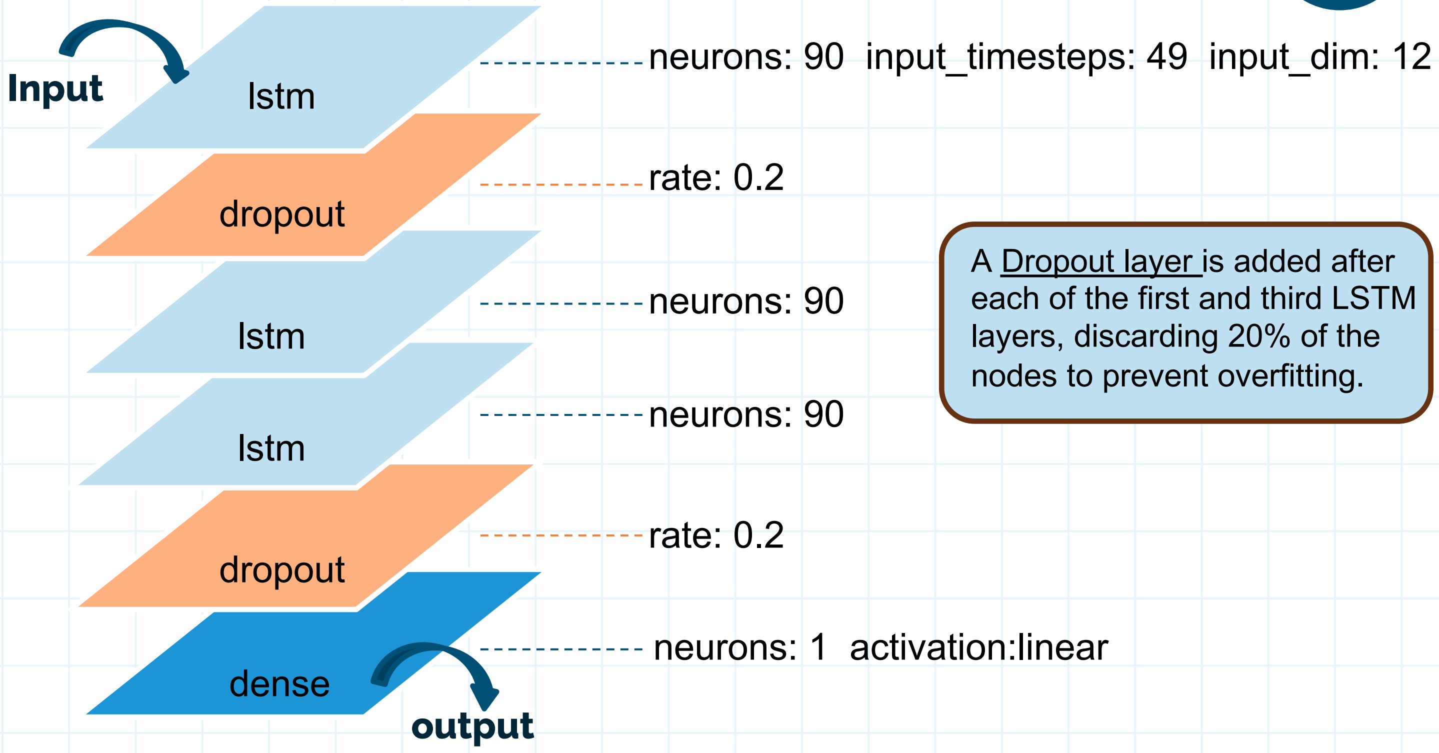
where `window_data` is the input data window and `base` is the base value extracted from each window.

04

Model Framework

Model Framework

Our model is designed with flexibility and efficiency, utilizing a deep learning framework to forecast data through various methods of sequence prediction. Their implementation using Python and Keras, a high-level neural networks API.



A Dropout layer is added after each of the first and third LSTM layers, discarding 20% of the nodes to prevent overfitting.

Model Framework

Model Compilation:

- Set the loss function to mean square error (`mse`).
- Set the optimizer to `adam`.

Model Training:

- Train the model, set the epoch number to 5 and the batch size to 32/64.
- Optional: If the data generator is enabled in the configuration, use the generator to train the model.

Model Prediction:

The model supports three types of predictions:

- Point-by-Point Prediction: Predict each point in the sequence independently.
- Multiple Sequence Prediction: Predict sequences using a sliding window approach.
- Full Sequence Prediction: Predict the full sequence iteratively using its previous outputs.

05

Model Evaluation & Results

Model Evaluation-Numerical

- Mean Square Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{true},i} - y_{\text{pred},i})^2$$

- Mean Absolute Error

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_{\text{true},i} - y_{\text{pred},i}\right|$$

- Root Mean Square Error

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\text{true},i} - y_{\text{pred},i})^2}$$

- Mean Absolute Percentage Error

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_{\text{true},i} - y_{\text{pred},i}}{y_{\text{true},i}}\right|$$

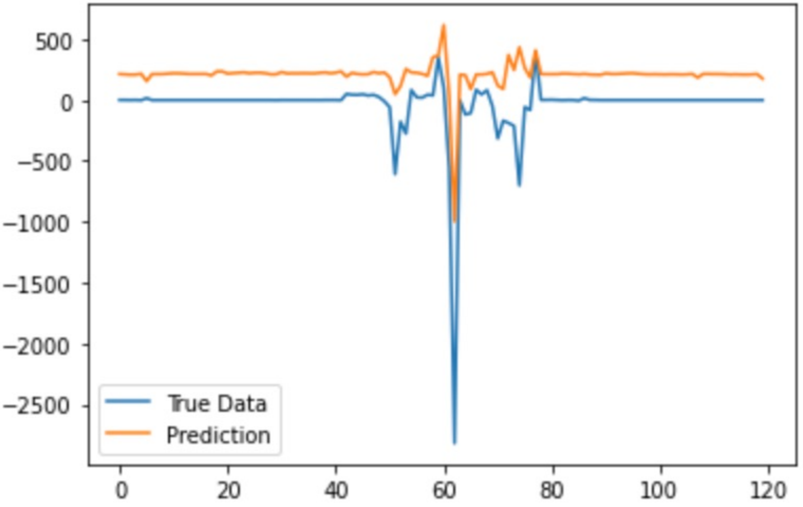
Model Evaluation-Numerical

	mse	mae	rmse	mape
Date Based	19208598.27	266.02	4382.76	85623.75
Trade Based	44391113.76	251.50	6662.67	27442.84

Date-based model
Better performance on MSE and RMSE, which indicates that its predictions are closer to the real data, albeit higher in terms of percentage error.

Trade-based model
Better performance on MAPE and MAE, but does not perform as well as the date-based model on the other metrics. This may indicate that although the transaction-based model predicts outcomes better in some cases, its predictions are more volatile and unstable.

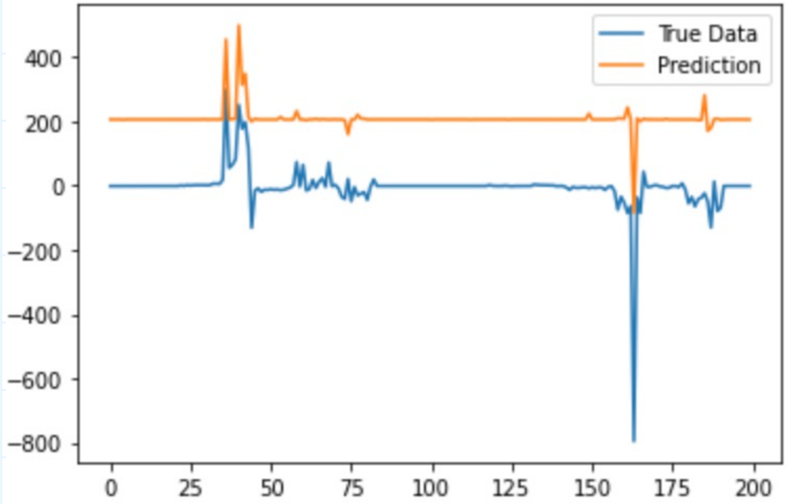
Model Evaluation-Visual (Time-Based)



Overall fit: the prediction line is high compared to the real data line for most of the time period, indicating that the model predicts large values in most cases, possibly due to the model fitting the training values better or the training values being large.

Anomalous fluctuations: At about 60 points in time, the real data fluctuates very sharply (the blue line plummets and then recovers quickly), and the prediction line captures this anomalous change, which may indicate that the model has some predictive power for extreme events. This may indicate that the model has some ability to predict extreme events, and that there can be a rise/decline at other inflection points that would like to correspond.

Model Evaluation-Visual (Trade-Based)



Overall fit: Overall, the trend of the forecast line is more consistent with the real data line, which indicates that the model is able to capture the general trend of the data better. But overall similar to the previous model, the predicted values are on the large side.

Anomalous fluctuations:At time points around 50 and 150, the real data shows sharp fluctuations, especially the sudden drop around 150, and the predictions capture some of this sharp movement. However, the model's response to less extreme events is inadequate.

Model Efficiency

Computer Hardware Configuration
GPU RTX 3090 (24GB)
CPU 10 vCPU Intel Xeon Processor (Skylake, IBRS)
Runtime memory usage: 6GiB
CPU Usage: 150%
GPU usage: 50%

Model Train & Predict
Training time: 30-35min
Prediction time: 1-2min

06

Limitations & Improvement

Current Limitations and possible solution

Limitations

<u>Strong Dependence on Data Quality</u>

The performance of the model heavily relies on the quality and quantity of the input data. Inadequate feature extraction or noisy and erroneous input data can lead to this issue. Inadequate features resulted in insufficient model fit.Vega's variation in this data is very large, and there are many outliers that are not easy to deal with.

solution

<u>Strong Dependence on Data Quality</u>

Improved feature extraction methods, such as using domain knowledge to construct more representative and relevant features, or using automatic feature selection techniques to reduce redundant features. Enhancements to the data, such as time window expansion or transformation of time series, to increase model robustness and data representation.

Current Limitations and possible solution

Limitations

<u>High Computational Requirements</u>

The model requires great computational power for training and predicting, limiting its deployment in resource-constrained environments. Adding some model complexity adds a very large arithmetic requirement.

solution

<u>High Computational Requirements</u>

Optimize the model structure to reduce the number of parameters or use a smaller network structure to reduce computational complexity and memory requirements. Apply model compression techniques such as weight pruning, quantization and knowledge distillation to reduce model size and runtime resource requirements.

Current Limitations and possible solution

Limitations

Limited Generalization

While the model performs better on a specific dataset, its performance degrades when applied to different data or different scenarios. The prediction for time series in this model is not accurate enough.

solution

Limited generalization capabilitie

Evaluate the stability and generalization ability of the model using cross-validation methods to ensure that the model performs well on different subsets of data. Use pre-trained models as a starting point for adapting to new datasets or tasks, and make the models better adapted to new scenarios through migration learning techniques.

07

Conclusion

Conclusion

We have designed and implemented two LSTM models to predict the risk of Range Accrual Note. The time-based model greatly enhances the assessment of future risk for the same product, enabling investors to adjust their strategies based on immediate risk. The trade-based model analyzes existing product data to predict the potential risk of new products, providing investors with insights into future performance and assisting them in making more informed decisions.

Both models provide investors with tools to make accurate decisions in uncertain markets. The application of this methodology demonstrates the great potential of technological innovation to improve the efficiency of risk assessment and management of financial products.

However, the deviation of the predicted values from the actual values shows that the trading models are limited in their ability to generalize across different datasets or market conditions. In the future, the accuracy and generalization ability of the model may be improved by increasing the diversity and quantity of data. We look forward to further research and development to refine and extend the model functionality to create greater value for financial market participants.

Thanks!