Using stacked-LSTM networks in US Treasury Securities yield forecasting and quantitative trading

Sven Martien van Holten Charria - 1155210281[1]

[1]*Exchange student from the Erasmus School of Economics, Rotterdam*

***Abstract*—This paper investigates the practical application of Long Short-Term Memory (LSTM) networks for forecasting bond yields, specifically within the US Treasury securities debt market amid Federal Reserve interest rate decisions and the potential threat of rising inflation. Recognizing the complexity of financial markets, the study employs LSTM to predict bond yields across various maturities, leveraging big data on macroeconomic covariates. Addressing a gap in existing research, which predominantly focuses on stock markets, this study contributes to the limited body of knowledge on LSTM applications in fixed-income securities. The methodology involves developing single- and stacked-LSTM architectures, utilizing a comprehensive dataset from the world's largest debt market. The structured approach covers model architectures, dataset characteristics, methodological applications, and results, providing insights into LSTM's potential for forecasting bond yields and informing trading decisions in a volatile financial landscape.**

I. Introduction

The Federal Reserve Board has actively combated the potential threat of national hyperinflation by gradually raising its benchmark interest rate to slow down economic growth. As inflationary pressures ease in the US economy, there is speculation regarding the timing and extent to which the Federal Reserve will decrease interest rates. Notably, big institutional investors like GS, MS, and UBS have widely varying predictions on the size of these rate cuts in the near future, ranging from small 25 basis points cuts by one, to substantial 250-275 basis points cuts by the others (Smith and Atkins, 2023). This disparity in predictions highlights the difficulty in reaching a consensus, even among major financial players armed with cutting-edge economic research. It's likely that these institutional investors will have to heavily invest in developing and using more advanced algorithms to accurately identify and predict complex underlying patterns in the market.

The US Treasury securities debt market is the largest in the world, reaching over $5 trillion in US dollars by December 2022 (Neufeld, 2023). In comparison, the S&P500 has a 25% lower market cap at $38 trillion as of November 2023. Given its size, the Treasury debt securities market plays a crucial role in funding government projects and serves as a risk management tool for investors. The US Treasury provides various securities, ranging from short-term bills to medium-term notes and long-term bonds. Yields on these securities are typically influenced by market expectations: as expectations rise, yields increase, and vice versa. Therefore, macroeconomic factors that impact market expectations, such as the benchmark interest rate, inflation, global competitiveness, and market volatility, also have an effect on yields.

Quantitative finance firms leverage sophisticated mathematical models in conjunction with big data to accurately price securities. Traditionally, predictions heavily relied on linear models that could capture and foresee (lagged) linearity in a dataset. Two widely used models, ARIMA and GARCH, combined autoregressive and moving average components while incorporating differencing to establish stationarity in time series data and model the variance of a time series, respectively. However, with data set dimensionality and computing power experiencing exponential growth, advances in AI are ushering in a new era of more intricate pattern recognition. 'Deep learning' has become the norm, given its superior ability to recognize both linear and non-linear patterns, extending beyond the capabilities of traditional mathematical models.

Recurrent Neural Networks (RNNs) play a pivotal role in deep learning, specifically designed for processing sequential data. RNNs utilize recurrent connections, forming a cyclical structure that facilitates information transmission over different time steps. However, a drawback of traditional RNNs is the application of constant weights to 'data transmission,' leading to the 'vanishing' of memory from earlier entries after numerous iterations. The Long Short-Term Memory (LSTM) adaptation addresses this issue by selectively deciding which features to retain. With its more interconnected structure, the network has the ability to control how much of the long-term memory is incorporated into the next output and the extent to which the current input influences the long-term memory intake of the next iteration.

The robust sequential capabilities of LSTM make it a powerful tool for analyzing and forecasting extended periods of financial data. Its structure is adept at capturing prolonged patterns and cycles while remaining sensitive to rapid price changes. Consequently, substantial research has been conducted on LSTM applications in the stock market, including Fischer and Krauss (2018) work on modeling volatility based on macroeconomic trends, Kraus and Feuerriegel (2017) exploration of sentiment analysis in parallel with stock pricing, and Ray et al. (2021) expansion of the standard LSTM model with a Bayesian structural time series adaptation. However, research on LSTM applications in fixed income securities is currently limited and incomplete. To the best of my knowledge, Nunes et al. (2020) are the only researchers who have univariately modeled 10-year Euro government bonds using a single-layered LSTM network.

Given this information gap, this paper aims to contribute to a more comprehensive understanding of using LSTM to forecast bond yields across a broader range of data and network complexities. Specifically, the study will consider a wider range of maturities, a more comprehensive set of macroeconomic covariates, and five different (stacked-) LSTM networks. Additionally, once bond yields have been forecasted, they will be used as indicators for trading strategies in an attempt to outperform the market.

The remainder of the paper will be structured as follows: Section II introduces the single- and stacked-LSTM architectures. Section III outlines the data set and preprocessing operations, and Section IV details the application of the dataset to the models. In Section V, the findings are presented and discussed. Finally, Section VI offers conclusions and suggestions for further research.

II. Models

This section delves into the structural framework and the relevant mathematical background of single-layered and stacked LSTM models. It begins by exploring the theory and architecture of the single-layered LSTM model. Following this, it expands on the required adaptations to transition the model into a stacked-LSTM configuration.

A. Single-LSTM

As previously highlighted, the LSTM memory cell architecture empowers the neural network to capture and propagate important long-term (LT) information through a sequence of interconnected cells. Differing from traditional neural network nodes, LSTM's cells incorporate a carefully calibrated set of '*forget*,' '*input*,' and '*output*' gates, that define the information throughput within the recurring network. The set of equations defining the architecture of the LSTM cell, which is understood in parallel with Figure 1, is as follows:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}^f * \boldsymbol{h}_{t-1} + \boldsymbol{V}^f * \boldsymbol{x}_t + b_f) \tag{1}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}^i * \boldsymbol{h}_{t-1} + \boldsymbol{V}^i * \boldsymbol{x}_t + b_i) \tag{2}$$

$$\boldsymbol{g}_t = \tau(\boldsymbol{W}^g * \boldsymbol{h}_{t-1} + \boldsymbol{V}^g * \boldsymbol{x}_t + b_g) \tag{3}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}^o * \boldsymbol{h}_{t-1} + \boldsymbol{V}^o * \boldsymbol{x}_t + b_o) \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \boldsymbol{g}_t \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tau(\boldsymbol{c}_t) \tag{6}$$

Here, for time t, f_t, i_t and o_t represent the vectors governing the forget, input and output gates; g_t represents the input node, c_t denotes the cell state; x_t and h_t denote the input and output vector; W_γ signify the weight matrices that regulate information flow between gates for $\gamma \in \{f, i, o, c\}$; b_γ signify the bias vectors in the gates for $\gamma \in \{f, i, o, c\}$. τ represents the hyperbolic tangent (tanh) activation function; σ represents the logistic sigmoid activation function; $\circ$ signifies the element-wise matrix (Hadamard) product.

The model employs the sigmoid and tanh functions in memory selection and throughput. The $\sigma \in [0, 1]$ activation function serves as a gating mechanism. Due to its bounds, $\sigma = 1$ can allow full memory throughput, and $\sigma = 0$ can allow no information throughput. The activation function $\tau \in [-1, 1]$ mitigates the issue of gradient vanishing over multiple iterations, and the range of the function facilitates both addition and subtraction within the cell state, representing an improvement over conventional neural networks.

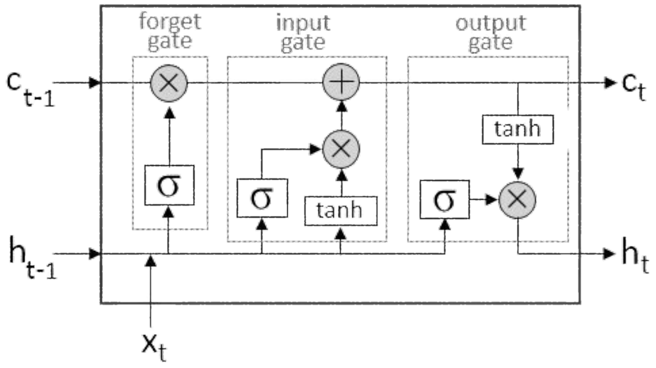

Fig. 1. Single-LSTM memory cell architecture (Rainardi, 2021)

The forget gate, ironically, determines which elements from the LT memory c_{t-1} are passed forward into c_t, essentially governing the recurrent LT memory component of the LSTM model. At first, h_{t-1} and x_t are multiplied by the predetermined W_f and increased by b_f. The resulting vector is fed into σ to yield $f_t \in [0, 1]$.

The input gate, similarly, determines which elements of x_t traverse into c_t: h_{t-1} and x_t are multiplied by W_i and augmented by b_i. The resulting vector is fed into σ to produce $i_t \in [0, 1]$.

Subsequently, still within the input gate, a secondary function creates a new initial cell state $\tilde{c}_t$ based on c_{t-1}. The τ evaluates $\tilde{c}_t$ and determines the effect it should have on c_t, yielding g_t. The updated cell state c_t results from the linear addition of the remembered LT memory $f_t \circ c_{t-1}$ and the proposed ST memory $i_t \circ gt$.

The output gate determines the extent to which o_t is transmitted to h_{t+1}. Similar to the input gate, h_{t-1} and x_t are multiplied by W_o and augmented by b_o. The resulting vector is fed into σ to yield $o_t \in [0, 1]$. τ evaluates c_t and determines the effect it should have on h_t and updates accordingly. In the single-LSTM, h_t is then passed to the dense layer, where a weighted transformation $y_t = W * h_t$ yields the desired output.

B. Stacked-LSTM

Reiterating the introductory paragraph, financial time-series present a formidable challenge to mathematical models owing to the complexity of underlying stochastic patterns. The 'depth' of a recurrent neural network characterizes the internal architecture: a 'deeper' network typically incorporates more hidden layers to enhance its ability to recognize patterns. An LSTM network with multiple hidden layers is termed a 'stacked-LSTM' network, where successive layers transmit their outputs as inputs to the subsequent layer. The following set of equations outlines how to *update* the l-th hidden layer in the stacked-LSTM network, to be understood in parallel with Figure 2:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_l^f * \boldsymbol{h}_{l,t-1} + \boldsymbol{V}_l^f * \boldsymbol{h}_{l-1,t} + b_f) \tag{7}$$
$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_l^i * \boldsymbol{h}_{l,t-1} + \boldsymbol{V}_l^i * \boldsymbol{h}_{l-1,t} + b_i) \tag{8}$$

$$\boldsymbol{g}_t = \tau(\boldsymbol{W}_l^g * \boldsymbol{h}_{l,t-1} + \boldsymbol{V}_l^g * \boldsymbol{h}_{l-1,t} + b_g) \tag{9}$$
$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_l^o * \boldsymbol{h}_{l,t-1} + \boldsymbol{V}_l^o * \boldsymbol{h}_{l-1,t} + b_o) \tag{10}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \boldsymbol{g}_t \tag{11}$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tau(\boldsymbol{c}_t) \tag{12}$$

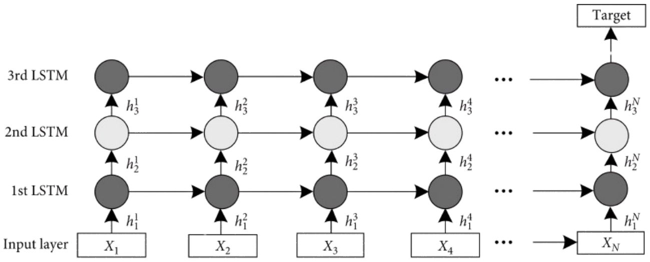
Target
3rd LSTM
2nd LSTM
1st LSTM
Input layer
X_1
X_2
X_3
X_4
X_N
h_3^1 h_3^2 h_3^3 h_3^4 h_3^N
h_2^1 h_2^2 h_2^3 h_2^4 h_2^N
h_1^1 h_1^2 h_1^3 h_1^4 h_1^N
...

Fig. 2. Stacked-LSTM network (Rainardi, 2021)

Here, the subscript *"l"* denotes the layer to which the cell belongs. The initialization of the first layer is identical to the single-LSTM, taking in the input vector x_t. From the second layer onwards, the input vector is instead supplied by $h_{l-1,t}$: the output of the previous layer at the same index in the time-series. The output of the previous cell in the same layer, $h_{l,t-1}$, is passed through similar to the single-LSTM. The output of the last hidden layer is connected to the dense layer.

C. Advantages of a stacked-LSTM network

The stacked-LSTM offers advantages through its enhanced feature extraction capabilities within the hierarchical structure. Each successive layer possesses the capacity to extract varied complexities in the data at every time step, contributing to a comprehensive understanding of distinct underlying patterns. Empirical findings from Lu et al. (2023) indicate that each additional layer can result in improved time-series predictions and better generalization to new data. Consistent with these observations, this paper anticipates that a higher number of layers will yield more accurate forecasting.

III. Data

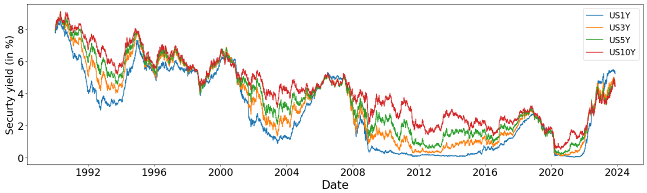

Fig. 3. US Treasury securities yield per maturity

This paper concentrates on four US Treasury bond maturity yields as labels: 1 year, 3 years, 5 years, and 10 years. Due to the tendency of longer maturities to exhibit a flatter yield curve, it is hypothesized that short-term macroeconomic trends will have a less pronounced impact on those yields (Livingston and Jain, 1982). A more comprehensive LSTM with additional hidden layers could prove beneficial in uncovering less obvious patterns for the longer maturity bonds.

It is important to note that Fig. 3 displays different phases of non-stationarity, characterized by varying gradients. The general trend can be described as a convex parabola, with a rapid increase in gradient near the end. Additionally, Fig. 3 displays numerous inversions of the yield curve. Inversions are strong predictors of economic recessions. Deeper LSTM models should better capture the patterns of the parabolic shape and yield curve inversion.

The following macroeconomic proxies are considered as features:

1) The daily return on the S&P500 daily adjusted closing price (collected from Yahoo Finance)
2) The daily return on the ICE US$ index adjusted closing price; measures the value of the US$ vis-à-vis a basket of foreign currencies (collected from Yahoo Finance).
3) The daily volatility index: VIX (collected from CBOE)
4) The monthly consumer price index: CPI (collected from FRED)

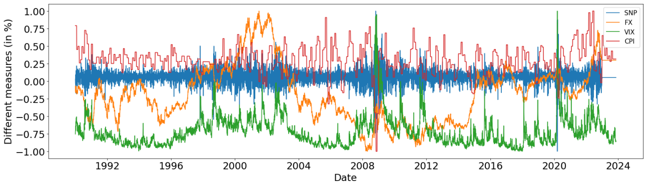

Fig. 4. Macroeconomic trends of the features

The features are merged into the labeled data with forward filling as needed. The resulting merged dataset comprises complete daily entries (trading days) from 01-01-1990 until 15-11-2022, totaling 8476 data points. Both the labels and features undergo normalization on the min-max scale, using -1 and 1 as bounds. The data is split into approximately 80% training and 20% testing, with the cut-off date set at 01-01-2017. Notably, the training data encompasses the anomaly of the 2008 financial crisis, while the testing data includes the anomaly of the Covid-19 pandemic. Both crises are evident in Fig. 3 through the yield curve inversion and in Fig. 4 through the atypical VIX and S&P500 returns.

To consider recurring effects, the following lags are introduced to both the label and all features: day-on-day, weekly, biweekly, monthly, and yearly. The inclusion of these lagged variables results in an additional 20 features, increasing the total from 4 to 24.

IV. Methodology

A. Supervised learning requirements

The data set is prepared for supervised learning through a sliding window adjustment, in which each label is trained on a pre-specified number of days of data. This paper follows Rodikov and Antulov-Fantulin (2022) in considering 28 days as an appropriate sliding window length for a financial-predicting LSTM network. A 28-day window signifies that the n_{th} label will be trained on the characteristics of the $n - 1_{th}$ to $n - 28_{th}$ days.

The research utilizes the PyTorch machine learning framework for all LSTM-related operations. The data is segmented into training and validation sets and then converted into PyTorch Tensors. The model is seeded to ensure the production of robust and comparable results.

B. Model specification

The LSTM models follow the specifications outlined in II and are configured as follows:

- *number of layers:* $\{1, 2, 3, 4, 5\}$
- *batch size:* 128
- *max epochs:* 1000
- *dropout rate:* 0.2
- *optimizer:* ADAM
- *loss fn:* MSE
- *early stopping criteria:*
  a) *patience:* 150
  b) *min delta:* 0.1

The models iterate over the number of epochs, stopping when it reaches the *max epochs* condition or the *early stopping criteria* is met. The RMSE and MAE of the validation results are gathered for each number of layers and for each bond maturity (2x5x4). Additionally, the mean and standard deviation of the loss values are similarly recorded, depending on the number of epoch iterations.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y_t})^2} \tag{13}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y_t}| \tag{14}$$

The use of both RMSE and MAE is common in ML analyses to obtain two distinct and meaningful metrics. The inclusion of squares in RMSE allows for the estimation of the distance from the actual values to the line, giving more weight to larger errors. On the other hand, the absence of squares in MAE ensures that outliers are not heavily penalized, providing a more robust measure of central tendency. The combination of both metrics is employed in the model evaluation to capture different aspects of the model's performance.

C. Macroeconomic extension

Furthermore, all combinations of hyperparameters will be executed in an identical manner for a parallel dataset that excludes the macroeconomic covariates. This extension is carried out to analyze the added significance of including the macroeconomic variables.

The adjusted dataset exclusively consists of the security yield rate data and its lags. Additionally, in response to the significantly superior results of the single-LSTM observed in the primary research, the forecasting is exclusively conducted using the single-LSTM model.

D. Quantitative Trading

After forecasting the US security yields, the predicted values are used as trade indicators. The trades are executed on the S&P500 index, employing two distinct strategies:

1) **Long-Only:** A positive trade indicator arises when the yield on the next day is higher than that of the current day. On such occasions, the index is bought at the opening price and sold at the closing price, implementing a ”going long” strategy. If the yield is lower the next day than the current day, no action is taken.
2) **Long-Short:** The strategy remains the same as in *Long-only*, with the addition of a short strategy: A negative trade indicator arises when the yield on the next day is lower than the yield on the current day. In this case, the index is sold at the opening price and bought at the closing price, implementing a ”going short” strategy.

The daily profits in both strategies are influenced by stock market actions, and these returns are aggregated over the validation period to compute the average return. Both strategies involve going all-in on every trade, although this approach exposes the system to potential vulnerabilities from significant events at the beginning.

The significance of integrating the trading component lies in the potential trade-off between additional layers introducing more noise and yielding a less metrically accurate result. Despite this possibility, deep learning holds the capability to more effectively capture market movements. As such, the effectiveness of the trading strategy serves as a robust indicator of the accurate depiction of underlying patterns, regardless of RMSE and MAE metrics.

V. Results

The objective of this paper is to enhance existing knowledge regarding LSTM models in the context of security yield forecasting. It aims to address key questions, including whether a deeper LSTM model demonstrates a better understanding of the data, and whether the inclusion of macroeconomic covariates improves forecasting accuracy. The availability of high-dimensional raw data provides ample opportunity to delve into each of these inquiries.

A. Forecast metrics

The detailed set of US Treasury security yield metrics, as outlined in Section IV, can be found in Appendix A. The results show that, across four bond maturities, the single-LSTM model outperformed in three instances, with the exception where it ranked second. Additionally, the 2-layered LSTM models consistently outperformed their deeper-layered counterparts. Therefore, for discussing the results, we'll refer to the three distinct groups as: *single-LSTM, double-LSTM, and multi-LSTM.*

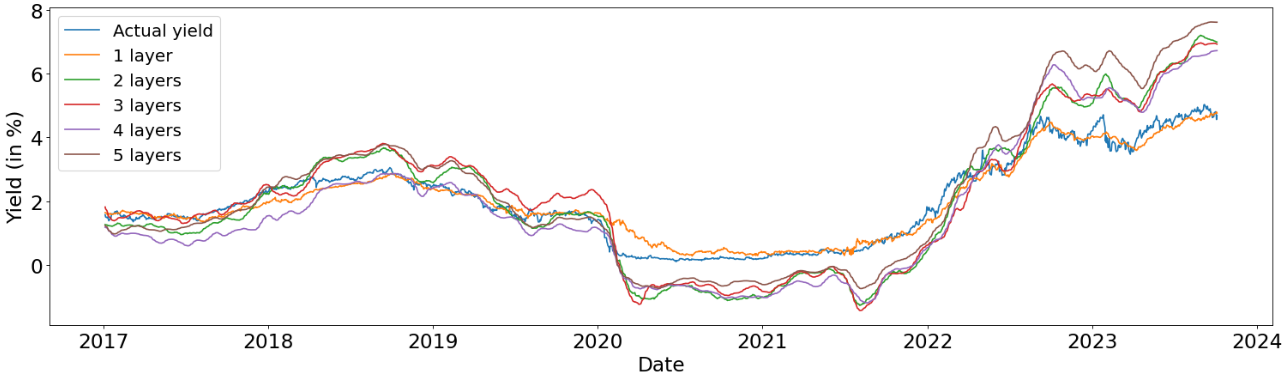

Fig. 5. Forecasted US1Y yield for different layered LSTM-models

TABLE I
SINGLE- VS DOUBLE- LSTM MODEL METRICS

	1 YEAR		3 YEARS		5 YEARS		10 YEARS	
Layers	1	2	1	2	1	2	1	2
RMSE ($*10^{-2}$)	41.597	111.509	27.707	91.955	27.307	61.017	29.697	43.077
MAE ($*10^{-2}$)	32.098	90.293	21.047	74.399	21.190	48.822	23.017	33.873
Mean Loss ($*10^{-2}$)	0.259	2.971	0.092	2.376	0.050	1.465	0.127	1.106
SD Loss ($*10^{-2}$)	0.673	6.562	0.269	4.985	0.150	4.898	0.356	3.756

A notable finding is the correlation between RMSE and MAE scores and the mean and standard deviation of the loss function. A well-performing LSTM model tends to exhibit considerably lower mean and standard deviation values of the loss functions throughout the epoch iterations. As shown in Table I, the lower RMSE and MAE values of the single-LSTM model align with significantly reduced mean and standard deviation values of the loss function, typically falling within the range of 10^{-2}. Further examination, as depicted in Fig. 5, reveals that this observation is attributed to the heightened volatility of multi-LSTM models compared to the single-LSTM model. While the single-LSTM model closely follows the actual yield, the double- and multi-LSTM models exhibit a notably wider spread. The single-LSTM model also demonstrates superior pattern recognition during the COVID-crisis, while the double- and multi-LSTM models exhibit negative yield rates. This outcome could be attributed to a deeper emphasis on the macroeconomic covariates in the model.

This discovery, combined with the earlier established result that the single-LSTM consistently outperformed the double- and multi-LSTM models, raises the question of why deeper LSTM models perform worse, considering that theoretically, more layers should lead to better recognition of underlying patterns. Two potential reasons underlie this finding:

1) The time-series data may lack complex underlying patterns, causing deeper LSTM models to attempt to learn non-existent trends. This phenomenon is evident in Fig. 5, where deeper-layered LSTM models exhibit much higher volatility, potentially tracking non-existent trends.
2) Conversely, the more complex neural network architecture of deeper LSTM models may have resulted in overfitting the time-series data. The higher number of neurons and parameters might not generalize effectively to out-of-sample forecasting scenarios. The LSTM models were trained with a dropout rate of 20% to prevent overfitting, but there's a possibility that this value could have been incorrectly calibrated.

Another observation is that as maturity lengthens, accuracy also improves. Table I shows a notable increase in accuracy across all metrics from one to three years, with a steady improvement seen in longer maturities. This aligns with findings from Livingston and Jain (1982), indicating that longer maturities often yield flatter yield curves, reducing the impact of short-term macroeconomic trends on yields. The higher accuracy in longer maturities supports this notion, suggesting they provide a safer but less lucrative trading approach

B. *Macroeconomic exclusion*

TABLE II
SINGLE-LSTM MODEL METRICS INCLUDING / EXCLUDING MACROECONOMIC COVARIATES

	1 YEAR		3 YEARS		5 YEARS		10 YEARS	
Layers	Macro	w/o Macro	Macro	w/o Macro	Macro	w/o Macro	Macro	w/o Macro
RMSE ($*10^{-2}$)	41.597	39.490	27.707	30.601	27.307	30.889	29.697	19.886
MAE ($*10^{-2}$)	32.098	32.191	21.047	24.400	21.190	24.795	23.017	15.179
Mean Loss ($*10^{-2}$)	0.259	0.027	0.092	0.019	0.050	0.015	0.127	0.029
SD Loss ($*10^{-2}$)	0.673	0.099	0.269	0.089	0.150	0.074	0.356	0.191

The RMAE and MAE scores show no significant difference between forecasts with and without macroeconomic covariates, both for short and long-term predictions. This contradicts expectations that including such covariates would improve the LSTM model's predictive accuracy, particularly in short-term forecasts. Surprisingly, when comparing mean and standard deviation metrics of the loss functions, the dataset without covariates consistently outperforms, sometimes by an order of magnitude.

This finding supports the earlier argument that the data lacks complex underlying trends, and adding macroeconomic covariates may not improve results but could instead lead to overfitting. It's worth noting that the dataset without covariates doesn't trigger early stopping limits and reaches the maximum iterations threshold. This suggests that the loss function doesn't converge, allowing the model to continue improving its performance solely based on the loss value. Once again, this supports the idea that the LSTM model might perform better when trained on a simpler dataset without covariates.

C. Quantitative trading results

In the quantitative strategy returns analysis, we will compare the performance of the leading LSTM model that includes macroeconomic covariates with that of a single LSTM model excluding these covariates. It's important to highlight that, in contrast to the results shown in Table II, the findings in Table III present signficant results.

TABLE III
RETURNS OF THE QUANTITATIVE TRADING STRATEGIES

	1 YEAR		3 YEARS		5 YEARS		10 YEARS	
	Macro	w/o Macro	Macro	w/o Macro	Macro	w/o Macro	Macro	w/o Macro
Long-Only	22.82%	-2.48%	35.59%	-7.44%	16.19%	-10.79%	20.02%	-13.34%
Long-Short	37.08%	-13.58%	67.06%	-22.14%	22.68%	-27.68%	30.89%	-31.75%

The dataset incorporating macroeconomic covariates displays substantial outperformance across all maturities for both Long-Only and Long-Short strategies. Despite the loss function's means and standard deviations suggesting a more volatile forecast, the model effectively captures more precise daily patterns. This is crucial, as these daily yield movements serve as the trading indicator, providing the trading strategy with a clearer signal on whether to buy, abstain from buying, or short the index.

This observation directly challenges the earlier conclusion that a simpler dataset leads to better forecasts. A balanced perspective emerging from these two conclusions is that a more complex dataset may yield an overall less accurate forecast but excels in capturing daily movements to a higher extent. Moreover, the uniformity observed in the data contradicts the hypothesis that strategies applied to longer-term maturities are less profitable but more stable.

Lastly, it's important to note that only one of the models outperforms the adjusted S&P 500 benchmark of 65.49% over the validation period (6 years). However, given the significantly lower results in other cases, this outcome is likely an anomaly.

VI. CONCLUSION

This paper aimed to advance our understanding of Long Short-Term Memory (LSTM) models in yield forecasting by investigating whether a deeper LSTM model enhances data comprehension, the impact of including macroeconomic covariates on forecasting precision, and the effectiveness of the trained model in simple quantitative finance trading strategies. Toy-model single- and multi-layered LSTM models were created and trained on forecast US Treasury Security yields across various maturities (1, 3, 5, and 10 years), considering both the inclusion and exclusion of macroeconomic covariate data.

Surprisingly, the single-LSTM model significantly outperformed over all double- and multi-LSTM models across all maturities, as indicated by RMSE and MAE metrics, generating a less volatile and unbiased forecast. Notably, using the dataset without macroeconomic covariates yielded similar RMSE and RME metrics but surpassed the original dataset when considering loss functions. However, in both Long-Only and Long-Short trading strategies, the original dataset consistently and significantly outperformed the dataset without macroeconomic covariates.

These findings suggest that, despite the seemingly less complex architecture of the single-LSTM model, it performs better on the dataset, possibly due to deeper models overfitting or misinterpreting the data's complexity. Additionally, the study highlights that a less accurate LSTM model has the potential to identify and mimic daily movements to a higher extent, leading to better adaptations in quantitative trading strategies.

The results of this paper could be strengthened by leveraging higher computational power and conducting more complete hyperparameter tuning. Additionally, extending the domain of macroeconomic covariates by including a broader range of exogenous factors, such as business-specific and exogenous global metrics, could better exploit the capabilities of LSTM and RNN models.

An extension of this study could involve utilizing all yields simultaneously to forecast the yield curve and predict conditions under which yield curve inversion may occur. This expanded approach would likely provide deeper insights into the dynamics of yield forecasting and offer valuable information for financial decision-making and macroeconomic policy implementation.

References

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.

Kraus, M. and Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, 104:38–48.

Livingston, M. and Jain, S. (1982). Flattening of Bond Yield Curves for Long Maturities. *The Journal of Finance*, 37(1):157–167. Publisher: [American Finance Association, Wiley].

Lu, H. Y., Lu, P., Hirst, J. E., Mackillop, L., and Clifton, D. A. (2023). A Stacked Long Short-Term Memory Approach for Predictive Blood Glucose Monitoring in Women with Gestational Diabetes Mellitus. *Sensors (Basel, Switzerland)*, 23(18):7990.

Neufeld, D. (2023). Ranked: The largest bond markets in the world.

Nunes, M., Gerding, E., McGroarty, F., and Niranjan, M. (2020). Long short-term memory networks and laglasso for bond yield forecasting: Peeping inside the black box. arXiv:2005.02217 [cs, q-fin].

Rainardi, V. (2021). Recurrent Neural Network (RNN) and LSTM.

Ray, P., Ganguli, B., and Chakraborty, A. (2021). A Hybrid Approach of Bayesian Structural Time Series With LSTM to Identify the Influence of News Sentiment on Short-Term Forecasting of Stock Price. *IEEE Transactions on Computational Social Systems*, 8(5):1153–1162.

Rodikov, G. and Antulov-Fantulin, N. (2022). Can LSTM outperform volatility-econometric models? arXiv:2202.11581 [q-fin].

Smith, M. and Atkins, A. (2023). Goldman Sachs, Morgan Stanley, UBS Diverge on Fed Rate-Cut Forecasts - Bloomberg.

APPENDIX A
US TREASURY SECURITY YIELD FORECASTS USING SINGLE AND STACKED-LSTM MODELS, INCLUDING / EXLCUDING MACROECONOMIC COVARIATES

Layers	1	2	3	4	5
RMSE $(*10^{-2})$	41.597	111.509	132.152	221.502	174.213
MAE $(*10^{-2})$	32.098	90.293	105.584	207.953	161.284
Mean Loss $(*10^{-2})$	0.259	2.971	3.722	5.318	4.732
SD Loss $(*10^{-2})$	0.673	6.562	9.093	14.455	11.177
Long-Only	717.271	-709.135	-614.184	86.898	134.827
Long-Short	1148.108	-1704.704	-1514.802	-112.639	-16.782
Epochs	560	360	440	380	180

TABLE IV
US 1Y MATURITY YIELDS - INCL. MACRO COVARIATES

Layers	1	2	3	4	5
RMSE $(*10^{-2})$	27.707	91.955	92.170	94.150	109.392
MAE $(*10^{-2})$	21.047	74.399	76.394	80.009	83.222
Mean Loss $(*10^{-2})$	0.092	2.376	3.927	5.876	4.567
SD Loss $(*10^{-2})$	0.269	4.985	9.509	16.411	9.592
Long-Only	717.271	-709.135	-614.184	86.898	134.827
Long-Short	1148.108	-1704.704	-1514.802	-112.639	-16.782
Epochs	560	240	280	260	280

TABLE V
US 3Y MATURITY YIELDS - INCL. MACRO COVARIATES

Layers	1	2	3	4	5
RMSE $(*10^{-2})$	27.307	61.017	66.984	60.598	38.846
MAE $(*10^{-2})$	21.190	48.822	52.519	44.530	31.565
Mean Loss $(*10^{-2})$	0.050	1.465	2.509	2.548	5.746
SD Loss $(*10^{-2})$	0.150	4.898	6.202	9.810	15.525
Long-Only	717.271	-709.135	-614.184	86.898	134.827
Long-Short	1148.108	-1704.704	-1514.802	-112.639	-16.782
Epochs	860	380	360	520	180

TABLE VI
US 5Y MATURITY YIELDS - INCL. MACRO COVARIATES

Layers	1	2	3	4	5
RMSE $(*10^{-2})$	29.697	43.077	67.165	26.703	52.902
MAE $(*10^{-2})$	23.017	33.873	54.474	20.866	37.197
Mean Loss $(*10^{-2})$	0.127	1.106	1.503	2.560	5.657
SD Loss $(*10^{-2})$	0.356	3.756	3.592	8.941	14.497
Long-Only	717.271	-709.135	-614.184	86.898	134.827
Long-Short	1148.108	-1704.704	-1514.802	-112.639	-16.782
Epochs	560	360	440	380	180

TABLE VII
US 10Y MATURITY YIELDS - INCL. MACRO COVARIATES

	1 YEAR	3 YEAR	5 YEAR	10 YEAR
Layers	1	1	1	1
RMSE $(*10^{-2})$	39.490	30.601	30.889	19.886
MAE $(*10^{-2})$	32.191	24.400	24.795	15.179
Mean Loss $(*10^{-2})$	0.027	0.019	0.015	0.029
SD Loss $(*10^{-2})$	0.099	0.089	0.074	0.191
Long-Only	19.026	226.397	4.524	467.606
Long-Short	-248.382	166.360	-277.387	648.776
Epochs	1000	1000	1000	1000

TABLE VIII
US 1,3,5,10Y MATURITY YIELDS - EXCL. MACRO COVARIATES