Global vs. Local LSTM Models for Forecasting Daily Business Sales

Madison Humphries
University of Texas at Dallas
Richardson, TX, USA
mrh210006@utdallas.edu

***Abstract*—Long Short-Term Memory (LSTM) neural networks are widely used in business for time-series forecasting, which involves predicting the next event based on sequential data. While prior studies often compare LSTMs to other machine learning models, few analyze the LSTM architecture itself. This study examines the effect of locality on the predictive accuracy of univariate LSTMs by comparing a Global LSTM (trained across all stores) and Local LSTMs (trained separately for each store type). Experiments systematically vary key hyperparameters (hidden dimension, batch size, learning rate, and epochs) and evaluate performance using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Percentage Error (% Error), Mean Absolute Error (MAE), and R^2. Results indicate that, for this dataset, the choice of hyperparameters, particularly batch size and epoch count has a larger effect on accuracy than the choice between global and local architectures. We conclude that when time series exhibit similar dynamics, a global model can achieve comparable performance to local models, highlighting the critical importance of hyperparameter tuning in LSTM forecasting.**

***Index Terms*—Long Short-Term Memory (LSTM), Sales forecasting, Global model, Local model**

I. Introduction

Accurate time-series forecasting of sales is crucial for businesses to make data-driven actionable insights and identify potential opportunities or risks. Modern forecasting approaches extend beyond traditional statistical methods, including Artificial Neural Networks (ANNs), Gaussian Process Regression (GPR), Support Vector Machines (SVMs), Recurrent Neural Networks (RNNs), Transformers, and even hybrid models [2]. Each method has specific strengths and limitations, motivating focused studies on individual models to fully understand their capabilities and potential improvements.

This study focuses on Long Short-Term Memory (LSTM) networks, a type of RNN designed to handle long-term dependencies in sequential data. LSTMs have shown promise in forecasting applications, but research often emphasizes comparisons between different model types rather than how the LSTM is applied to the data. Here, we examine how the choice of modeling locality (global versus local) affects predictive accuracy in sales forecasting.

We use the Rossmann Store Sales dataset [5] from the Rossmann Kaggle Competition, which contains over 1 million sales records across multiple store types. This dataset provides a natural partitioning for local models, while global models are trained across all stores. By focusing on univariate sales predictions, we isolate the effect of model architecture and hyperparameter tuning without conflicting additional influences. The dataset was chosen because it contains modern and current sales data that illustrate the need for modeling and predicting, allowing for a more realistic experimentation and study.

Two modeling strategies are being compared (1) Global LSTM, trained on all stores jointly to capture overarching patterns across the data, and (2) Local LSTMs, trained separately for each store type to capture store-specific sales dynamics. The comparison highlights trade-offs between capturing local patterns versus general trends and informs resource allocation decisions when deploying forecasting models.

In this study, the LSTM models were created from scratch using NumPy and systematically evaluated with varying hyperparameters and a dataset of moderate complexity to assess whether one model consistently outperformed the other. The specific hyperparameters tuned are hidden dimensions, batch size, learning rate, and epochs, while the evaluate metrics are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Percentage Error, Absolute Error, and R^2. The results illustrate that performance was comparable in a majority of situations, with specific parameters favoring specific model architectures. These findings suggest that after hyperparameter tuning, the choice between global and local architectures contributes less to performance than expected.

II. Background

Traditional machine learning methods often struggle to model sequential data because they lack the ability to retain information over time. Recurrent Neural Networks (RNNs) were introduced to address this limitation by incorporating recurrent connections that allow information to persist across time steps. However, RNNs can suffer from vanishing or exploding gradients during backpropagation through time, which hinders their ability to learn long-term dependencies [4]. In the vanishing gradient problem, gradients shrink exponentially as they propagate backward, leading to minimal weight updates. Conversely, exploding gradients occur when gradients grow exponentially, potentially destabilizing training.

To overcome these issues, Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM) network [1]. LSTMs extend RNNs by adding memory cells and gated mechanisms, enabling the network to maintain stable gradients and retain information over longer sequences [3], [4]. An

LSTM unit consists of a cell state C_t, which acts as long-term memory, and a hidden state h_t, which represents short-term memory used for predictions at each time step. Information flow is controlled by three gates: (1) forget gate, (2) input gate, (3) output gate. This system allowed for more control and longer persistence of memory over standard RNNs.

In time-series applications, model architecture can also vary by scope. Local models train a separate predictive model for each individual time series (e.g., each store type in this dataset), allowing them to capture unique seasonal or trend patterns. In contrast, global models are trained across all available time series, capturing overarching trends and benefiting from larger training datasets. The choice between local and global modeling remains context-dependent: local models excel when series differ significantly, whereas global models generalize better when patterns are shared across series or when the data is noisy.

This study investigates how these architectural choices, local versus global LSTM models, given interactions with hyperparameter tuning to affect forecasting accuracy in a business sales context.

III. Theoretical & Conceptual

Recurrent Neural Networks (RNNs) are widely used for modeling sequential data because of their ability to incorporate temporal dependencies, however they struggle with long-term dependency learning due to the vanishing and exploding gradient problems that arise during the backpropagation step [4]. Hochreiter and Schmidhuber's Long Short-Term Memory (LSTM) architecture expands on RNN by adding a gated memory mechanism to enable longer memory while avoiding the vanishing or exploding gradient issue [1].

A. LSTM Architecture

An LSTM unit consists of two primary components:

1) The cell state (C_t): acts as a long-term memory, carrying information across time steps
2) The hidden state (h_t): represents the short-term working memory used for prediction at each time step t

LSTMs regulate information flow through three multiplicative gates, which allow the network to add, remove, or give information, helping with long-range dependencies:

1) Forget Gate (f_t): controls what previous information should be forgotten
2) Input Gate (i_t): controls what new information to be learned
3) Output Gate (o_t): controls what information to be revealed

B. LSTM Gate Mechanisms

At time step t, given input x_t and previous hidden state $h_t - 1$ the following occurs:

1) The forget gate f_t determines which information from the previous state should be forgotten: $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$
2) The input Gate i_t controls how much new information is added to the cell: $i_t = \sigma(W_i x_t + U_i h_t - 1 + b_i)$
3) And a candidate memory state $\tilde{C}$ is proposed: $\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$
4) The cell state is updated by combining the retained information with the new candidate memory state content: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
5) The output gate o_t determines which parts of the cell state should affect the hidden state and constructs the final hidden state: $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$

Here, σ denotes the sigmoid activation function, tanh denotes the hyperbolic tangent function, and $\odot$ represents Hadamard Product (element-wise multiplication. The weights W, U, and biases b are learned during training.

C. LSTM Training

LSTMs are trained using backpropagation through time (BPTT), similar to standard RNNs, but with the inclusion of their gate mechanisms enable them to maintain long-term memory. Model performance is influenced by several key hyperparameters, in this study, we systematically varried the following in Table I:

TABLE I
KEY LSTM HYPERPARAMETERS AND THEIR EFFECTS

Hyper parameter	Description	Effect of Increasing	Effect of Decreasing
Hidden Dimensionality	Size of the hidden state (model capacity)	Can capture more complex patterns, but may overfit if too large	May underfit if too small, limiting pattern capture
Batch Size	Number of samples processed per update	Faster training, but may overfit on small datasets	More noisy updates and better generalization on diverse data
Learning Rate	Step size for weight updates	Too high can cause unstable training or divergence	Too low can cause slow convergence or get stuck in local minima
Epochs	Number of passes over the training data	Longer training and risk of overfitting	Shorter training and risk of underfitting

Optimizing these hyperparameters is critical, as model performance is highly sensitive to their values. For instance, local LSTMs, being more specialized for each store type, are prone to overfitting if trained for too many epochs, while global LSTMs, trained on a larger combined dataset, may tolerate longer training without overfitting.

IV. Results & Analysis

This section evaluates the predictive performance of a Global LSTM (trained on all stores jointly) versus Local LSTMs (individual models trained separately based on store

type) under a systematic series of hyperparameter experiments. All models were trained on the Rossmann Store Sales dataset, with the task of forecasting next-day sales using seven sequential days of historical data. The models are univariate LSTMs, implemented entirely from scratch. The objective is to assess whether locality impacts predictive accuracy and to identify hyperparameter configurations that minimize error.

Both model types were evaluated using standard regression metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Percentage Error, Absolute Error, and R^2, which is treated as the primary measurement of predictive performance. This evaluation framework allows for a consistent comparison across different architectures and hyperparameter settings.

A. *Effect of Epochs on Performance*

TABLE II
EFFECT OF EPOCHS ON LSTM PERFORMANCE

Epoch	Global Model	Local Model
1	R^2 = 0.906, MSE = 38,587,236,403, RMSE = 196,436, MAE = 115,526, %Error = 9.469	R^2 = 0.906, MSE = 8,564,679,319, RMSE = 196,379, MAE = 121,760, %Error = 8.675
3	R^2 = 0.926, MSE = 30,336,358,536, RMSE = 174,173, MAE = 111,006, %Error = 4.364	R^2 = 0.914, MSE = 35,492,780,697, RMSE = 188,395, MAE = 114,602, %Error = 7.799
5	R^2 = 0.913, MSE = 36,012,263,468, RMSE = 189,769, MAE = 116,896, %Error = 7.853	R^2 = 0.926, MSE = 30,497,803,116, RMSE = 174,636, MAE = 111,300, %Error = 4.360

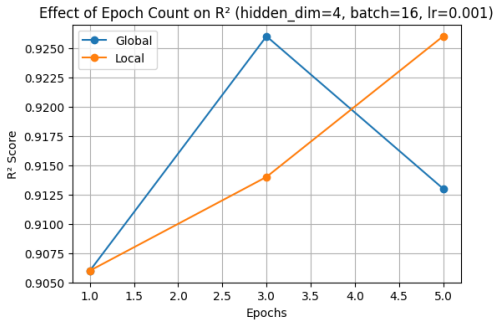

Fig. 1. Effect of Epochs on LSTM Models

Table II and Figure 1 illustrate the effect of training epochs on both Global and Local LSTM models. For the Global model, increasing the number of epochs from 1 to 3 improves R^2 from 0.906 to 0.926, accompanied by reductions in MSE, RMSE, MAE, and percentage error, indicating better predictive accuracy. However, increasing epochs further to 5 results in a slight decline in R^2 to 0.913 and higher error metrics, suggesting overfitting. In contrast, Local LSTMs achieve their highest R^2 at 5 epochs (0.926), although the improvement from 3 epochs (0.914) is modest. This pattern indicates that Local models efficiently capture store-specific patterns with additional epochs, whereas Global models require careful tuning to avoid overfitting. The percentage error follows a similar trend, highlighting that proper epoch selection is critical for both model types.

TABLE III
EFFECT OF BATCH SIZE ON LSTM PERFORMANCE (HIDDEN DIM = 4, LR = 0.001, EPOCHS = 3)

Batch Size	Global Model	Local Model
8	R^2 = 0.912, MSE = 36,372,397,371, RMSE = 190,715, MAE = 116,267, %Error = 8.158	R^2 = 0.927, MSE = 30,257,582,728, RMSE = 173,947, MAE = 112,104, %Error = 4.103
16	R^2 = 0.926, MSE = 30,336,358,536, RMSE = 174,173, MAE = 111,006, %Error = 4.364	R^2 = 0.914, MSE = 35,492,780,697, RMSE = 188,395, MAE = 114,602, %Error = 7.799
32	R^2 = 0.933, MSE = 27,616,021,894, RMSE = 166,181, MAE = 108,627, %Error = 1.863	R^2 = 0.914, MSE = 35,516,778,366, RMSE = 188,459, MAE = 114,767, %Error = 7.892
64	R^2 = 0.903, MSE = 39,764,330,064, RMSE = 199,410, MAE = 119,001, %Error = 9.836	R^2 = 0.903, MSE = 40,013,624,826, RMSE = 200,034, MAE = 121,190, %Error = 9.629

B. *Effect of Batch Size on Performance*

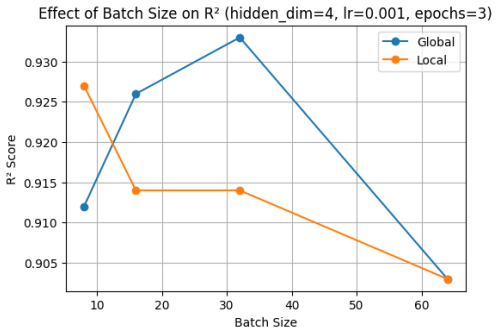

Fig. 2. Effect of Batch Size

Table III and Figure 2 examines the impact of batch size on LSTM performance, with other hyperparameters held constant (Hidden Dim = 4, Learning Rate = 0.001, Epochs = 3). For the Global model, increasing batch size from 8 to 32 improves R^2 from 0.912 to 0.933 and reduces MSE, RMSE, and MAE, indicating better predictive performance. However, further increasing the batch size to 64 leads to decreased accuracy ($R^2 = 0.903$), likely due to fewer gradient updates per epoch. For the Local models, the highest R^2 (0.927) is achieved at a batch size of 8, while larger batches (16–64) result in modest reductions in performance. These results suggest that the optimal batch size balances training stability

and the granularity of weight updates, with Local models being somewhat less sensitive to very large batch sizes compared to Global models.

C. Effect of Hidden Dimension on Performance

TABLE IV
EFFECT OF HIDDEN DIMENSION ON LSTM PERFORMANCE (BATCH SIZE = 32, LR = 0.001, EPOCHS = 3)

Hidden Dim	Global Model	Local Model
4	R^2 = 0.933, MSE = 27,616,021,894, RMSE = 166,181, MAE = 108,627, %Error = 1.863	R^2 = 0.914, MSE = 35,516,778,366, RMSE = 188,459, MAE = 114,767, %Error = 7.892
8	R^2 = 0.868, MSE = 54,433,867,117, RMSE = 233,311, MAE = 129,366, %Error = 15.625	R^2 = 0.925, MSE = 30,833,066,768, RMSE = 175,593, MAE = 110,152, %Error = 5.066

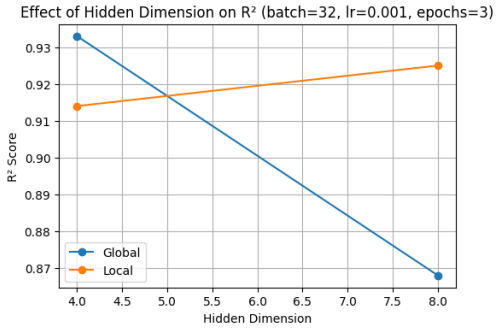

Fig. 3. Effect of Hidden Dimensions

Table IV and Figure 3 shows the effect of hidden layer size on LSTM performance, with other hyperparameters held constant (Batch Size = 32, Learning Rate = 0.001, Epochs = 3). For the Global model, increasing the hidden dimension from 4 to 8 substantially decreases R^2 from 0.933 to 0.868 and increases MSE, RMSE, and MAE, suggesting over-parameterization and potential overfitting. In contrast, the Local models experience a modest improvement in R^2 from 0.914 to 0.925, with reductions in error metrics, indicating that a larger hidden dimension can slightly enhance the model's ability to capture store-specific patterns. These results suggest that smaller hidden dimensions are sufficient for Global models, which learn aggregate patterns, whereas Local models may benefit from additional capacity to model local nuances.

D. Effect of Learning Rate on Performance

Table V and Figure 4 summarizes the impact of learning rate on LSTM performance (Hidden Dim = 4, Batch Size = 32, Epochs = 3). For the Global model, increasing the learning rate from 0.001 to 0.003 slightly improves R^2 from 0.933 to 0.934 and reduces error metrics, while further increasing the rate to 0.005 drastically reduces R^2 to 0.869, indicating training instability. For the Local models, optimal performance occurs

TABLE V
EFFECT OF LEARNING RATE ON LSTM PERFORMANCE (HIDDEN DIM = 4, BATCH SIZE = 32, EPOCHS = 3)

Learning Rate	Global Model	Local Model
0.001	R^2 = 0.933, MSE = 27,616,021,894, RMSE = 166,181, MAE = 108,627, %Error = 1.863	R^2 = 0.914, MSE = 35,516,778,366, RMSE = 188,459, MAE = 114,767, %Error = 7.892
0.003	R^2 = 0.934, MSE = 27,129,802,886, RMSE = 164,711, MAE = 108,890, %Error = 1.146	R^2 = 0.907, MSE = 38,203,622,595, RMSE = 195,457, MAE = 116,583, %Error = 9.692
0.005	R^2 = 0.869, MSE = 54,024,066,174, RMSE = 232,431, MAE = 130,354, %Error = 15.462	R^2 = 0.914, MSE = 35,465,401,398, RMSE = 188,323, MAE = 115,835, %Error = 7.699

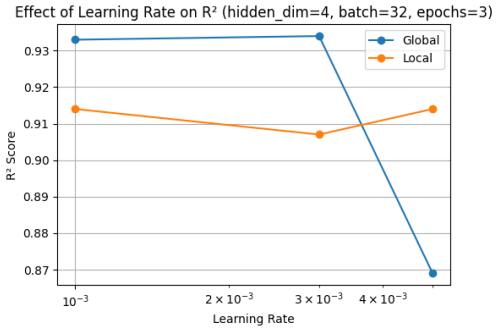

Fig. 4. Effect of Learning Rate

at a learning rate of 0.001 ($R^2 = 0.914$); increasing the rate to 0.003 slightly decreases performance ($R^2 = 0.907$), whereas 0.005 returns R^2 to the baseline of 0.914. These results highlight that both Global and Local LSTMs are sensitive to learning rate selection, and moderate rates are preferable to ensure stable convergence.

E. Overview

The best models discovered:

1) Global Model: $R^2 = 0.934$
   a) Hidden Dimensions: 4
   b) Learning Rate: 0.003
   c) Batch Size: 32
   d) Epochs: 3
2) Local Models: $R^2 = 0.929$
   a) Hidden Dimensions: 4
   b) Learning Rate: 0.005
   c) Batch Size: 32
   d) Epochs: 5

Overall, there is no distinctive pattern in the performance of Global versus Local models across different hyperparameter configurations. While there are instances where the Global model performs worse when the Local model performs better, this behavior is not consistent. Furthermore, the overall performance is comparable, with the majority of models achieving

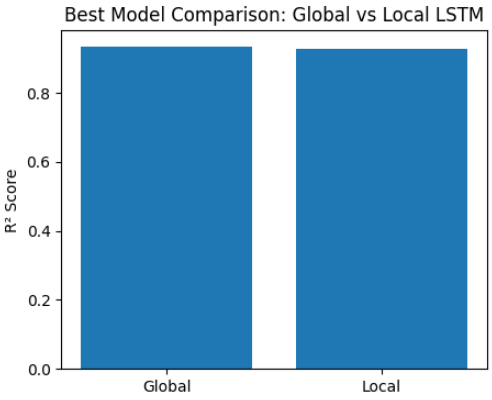

Fig. 5. Comparison of Model Architectures

$R^2 > 0.90$. As shown in Figure 5, the difference between the best R^2 values across all training iterations is minimal, with the Global model reaching $R^2 = 0.934$, slightly edging out the Local LSTMs at $R^2 = 0.929$. These results indicate that locality does not have a meaningful impact on performance for time series forecasting of sales data.

Conclusion & Future Work

The results of this study indicate that altering the model architecture between Global and Local LSTMs does not significantly impact predictive accuracy. Instead, careful hyperparameter tuning, particularly of batch size, learning rate, hidden dimension, and epochs in this scenario, plays a more influential role in optimizing performance.

For future work, similar experiments could be conducted on datasets with different characteristics or distributions to assess how general these findings are. Additionally, the LSTM models can be extended from univariate to multivariate forecasting by incorporating additional inputs such as promotional events, competitor activity, or other relevant features in this dataset or another. Another avenue for exploration is to evaluate the impact of forecasting longer time horizons, for example predicting weekly sales instead of daily sales, to determine whether locality or hyperparameter choices has more influence on the performance under extended prediction periods.

References

[1] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[2] Z. Liu, Z. Zhu, J. Gao and C. Xu, "Forecast Methods for Time Series Data: A Survey," in IEEE Access, vol. 9, pp. 91896-91912, 2021

[3] Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., and Ragab, M. G. (2024). RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. Journal of King Saud University-Computer and Information Sciences, 36(5), 102068.

[4] Grosse, R. (2017). Lecture 15: Exploding and vanishing gradients. University of Toronto Computer Science.

[5] FlorianKnauer and Will Cukierski. Rossmann Store Sales. https://kaggle.com/competitions/rossmann-store-sales, 2015. Kaggle.