Developing Machine Learning Algorithms to Price American-Style Stock Options

Nathaniel James Pacey[1]

[1]Queen's University, Department of Physics, Engineering Physics, and Astronomy

Introduction and Motivation

Stock options enable investors to trade significant company shares at fixed prices within a set period, offering **substantial capital leverage with minimal investment** [1]. In today's competitive trading market **computer algorithms account for 50 to 60%** of stock and options trading on a typical day [2]. This project aims to predict call and put option prices; call options are for anticipated share price increases, allowing the purchase of shares at a set strike price before expiry, while put options are for expected price drops, permitting the sale of shares within the contract's term [1]. Accurately forecasting stock option prices with models like Black-Scholes formula remains a challenge, particularly for American options which unlike European style options can be exercised at anytime before expiration [3].

This research is focused on overcoming the predictive challenges of stock option pricing, specifically by developing and evaluating **many-to-many Recurrent Neural Networks (RNNs)** to accurately **forecast bid and ask prices for large-cap American-style options.** Insights from this study not only advance time series machine learning research but also aids in identifying potential arbitrage opportunities, thereby enhancing traders' ability to make well-informed and potentially lucrative decisions.

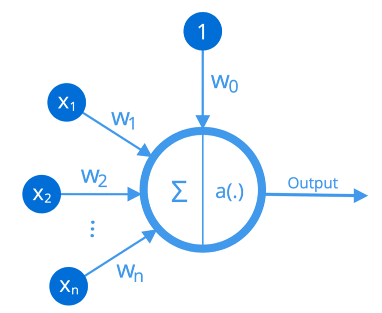

Figure 1. Single Neuron with weights [4].

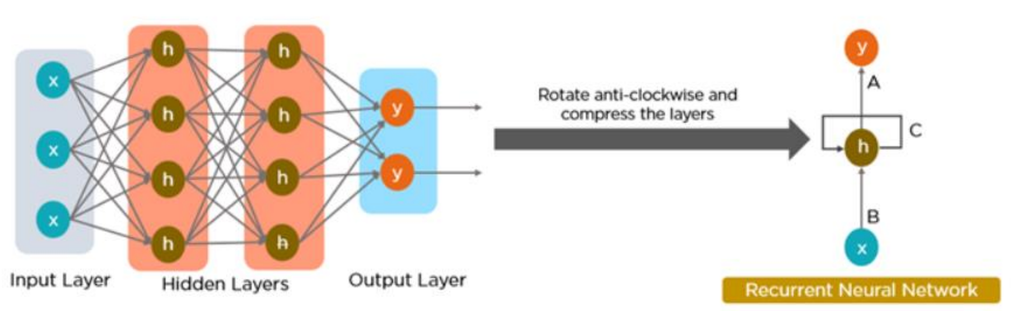

Figure 2. Many-to-many RNN including the essential layers [5].

Introduction and Background

At the core of machine learning models are artificial neurons. In a neuron model, input values x_i are multiplied by corresponding weights w_i, summed with a bias b to produce z, and passed through a nonlinear activation function a to output $a(z)$ [4].

$$z = \left(\sum_{i=1}^{n} x_i \times w_i\right) + b \qquad \rightarrow \qquad a(z) = a\left(\sum_{i=0}^{n} x_i \times w_i\right) \qquad (1)$$

This nonlinearity enables complex pattern handling in data. Complex data necessitates a network of interconnected neurons using backpropagation, gradient descent, and input-output cycles, all integral to RNNs. However, **classical RNNs struggle with short-term memory** due to vanishing gradients that diminish their capacity to maintain information across extended sequences, an issue mitigated by advancements such as Long Short-Term Memory **(LSTM)** networks and Gated Recurrent Units (**GRU**) [6] [7].

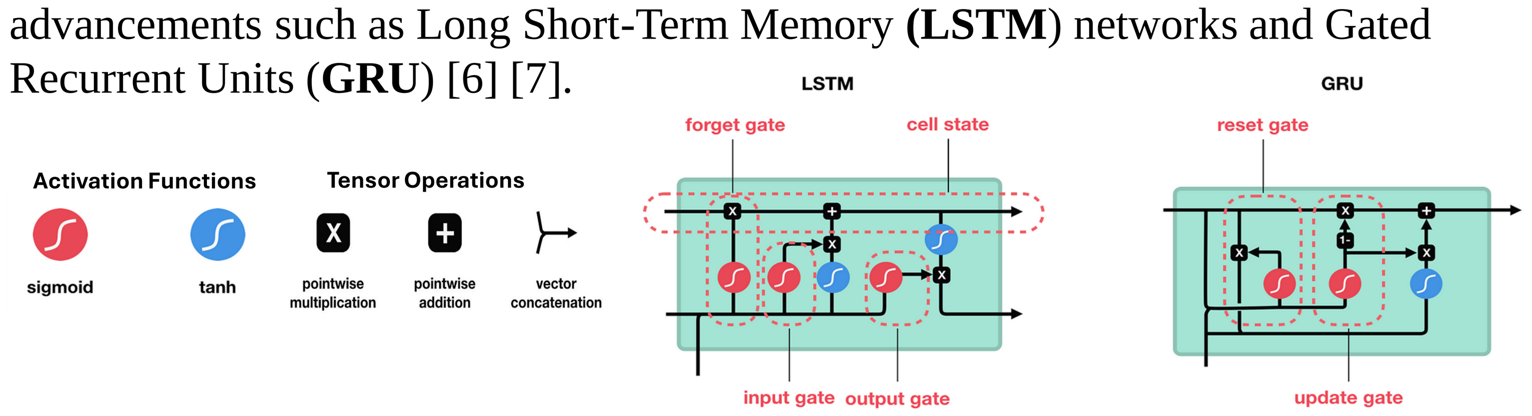

Figure 3. LSTM and GRU internal mechanisms as taken from an Illustrated Guide to LSTM's and GRU's [10].

LSTM networks employ Input, Forget, and Output Gates with a central Cell State to preserve and manage data across long sequences for intricate temporal processing. GRUs simplify this architecture by combining these gates into an Update Gate and introducing a Reset Gate, thereby enhancing computational efficiency without sacrificing sequence modeling effectiveness [7] [8].

Methods and Materials

Data Extraction: Options data for 56 companies from February to December of 2022, were compressed and stored on the Neutrino server filtering by strike price and adding temporal features for model inputs and outputs.

Data Encoding: Applied min-max normalization to the prepared datasets, partitioning them into an 80/20 training/testing split, readying them for RNN input.

Model Development and Evaluation: Using TensorFlow and Keras, LSTM, GRU, and Hybrid GRU models with standardized hyperparameters were developed. Initially each model was trained and tested on Apple and AMD contracts, employing MSE and MAE for accuracy evaluation and utilizing detailed visual analyses to compare model performance.

Model Iteration and Evaluations

Following established frameworks, initial RNNs, LSTM and GRU, had two 50-unit layers, 0.1 dropout, ReLU and linear activations, using Adam optimization at a 0.001 learning rate with MSE evaluation. The Hybrid GRU separated data with a dense layer for static data and a bidirectional 32-unit GRU for temporal data, with dropout and normalization, converging at the final prediction layer.

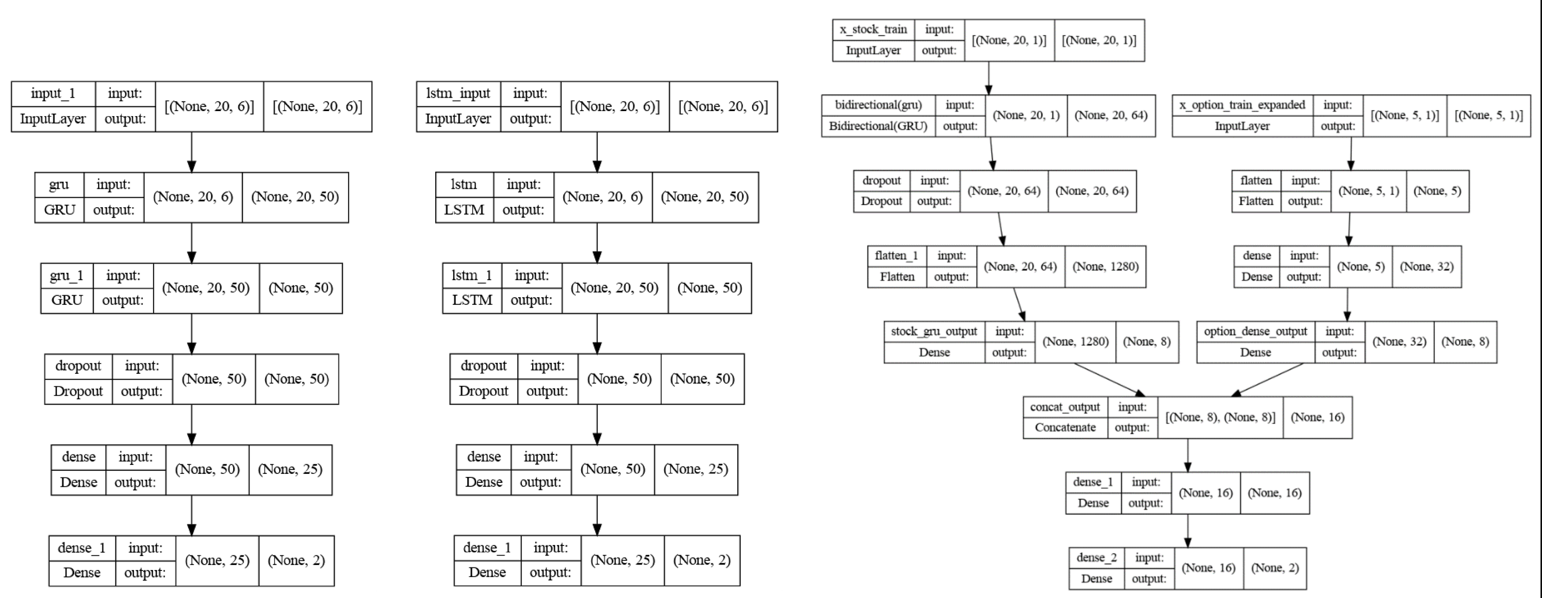

Figure 4. LSTM, GRU-only and Hybrid GRU and FC model architecture diagrams as created by TensorFlow plot_model [9].

On the Apple and AMD testing dataset, the LSTM model showed an MSE of 97.39 and MAE of 6.82, with bid and ask errors at 46.68% and 26.50%. The GRU model outperformed with an MSE of 46.59, MAE of 4.48, and reduced errors of 7.86% and 12.95%. The Hybrid GRU lagged, with a MSE of 73.47 and MAE of 6.00, indicating lower accuracy and correlation compared to the GRU and LSTM models.

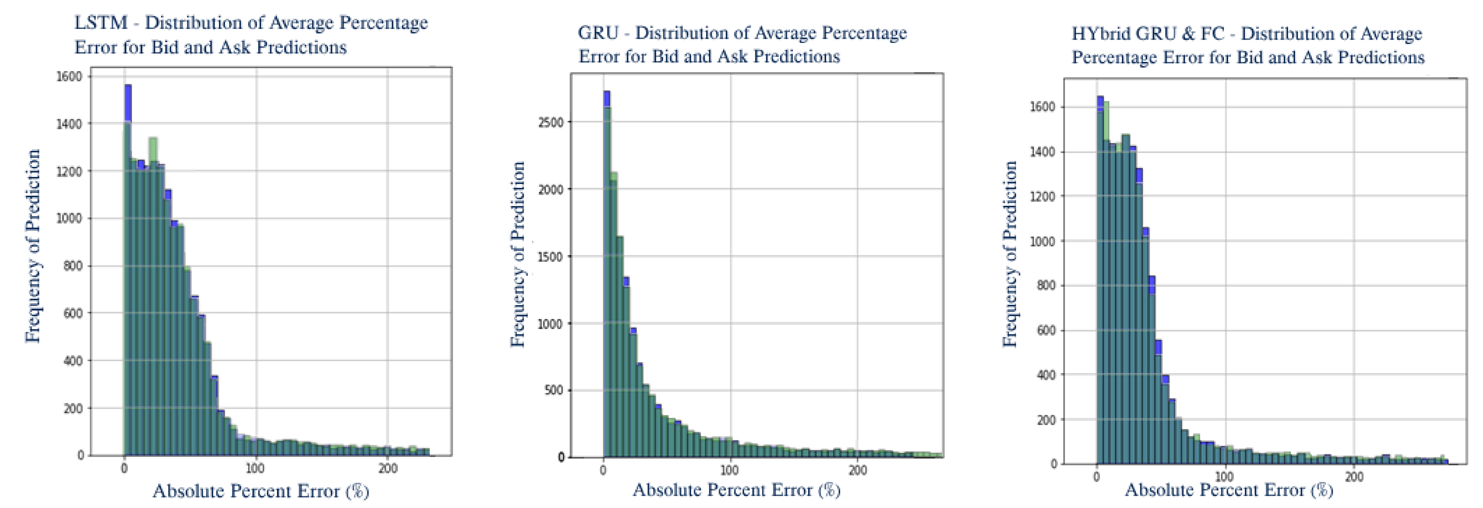

Figure 5. LSTM, GRU-only and Hybrid GRU and FC models average bid and ask percent error distribution on Apple and AMD.

To assess performance in practical scenarios, each model was tested on Amazon contracts. The LSTM model recorded a MAE of 3.56, with errors mostly below 25%. The GRU model, however, despite a lower MAE of 3.17, had significantly higher average prediction bid/ask errors, of 84.4% and 114% respectively, most likely due to outliers. The Hybrid GRU model performed worse, with errors exceeding 170% and an MAE of 4.66. Iterative adjustments to epochs and hyperparameters aimed to enhance each model's forecasting accuracy.

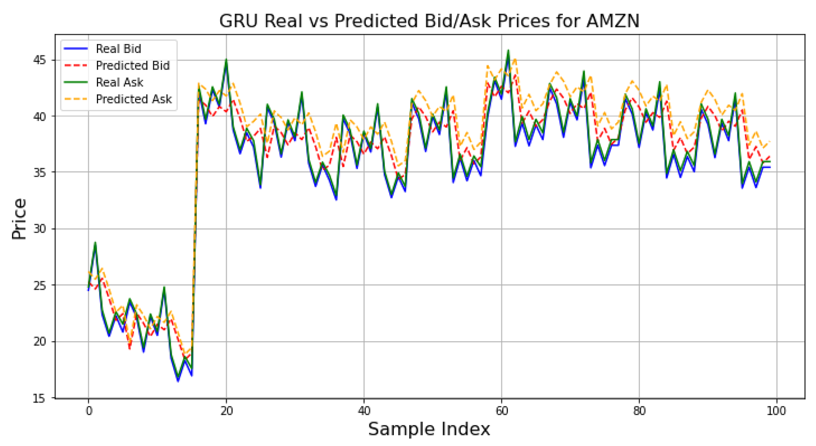

Figure 5. GRU-only predictions on the first 100 Amazon option contracts.

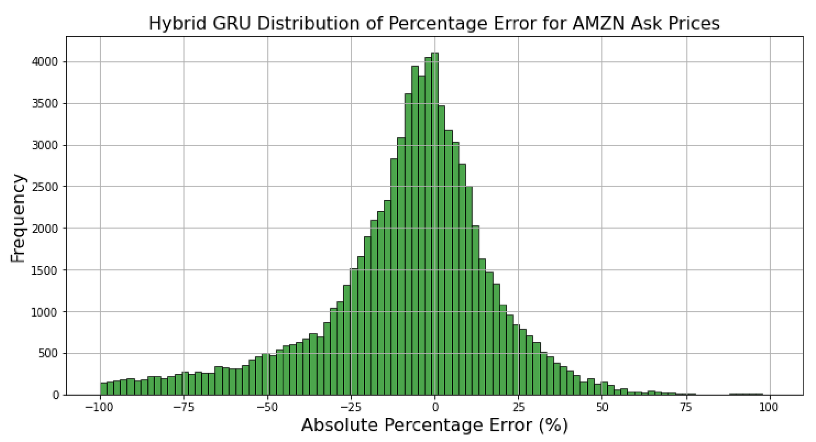

Figure 6. GRU-only percent error distribution on ask price predictions.

Results and Discussion

To enhance accuracy, the training and testing datasets were **expanded to include Google and Microsoft contracts** and underwent 100 epochs of training, with checkpoints **saving the iteration with the lowest error.** Additionally, for comparative analysis, performances were benchmarked against an adapted Hybrid GRU and Convolutional model provided by Professor Ryan Martin.

The expanded dataset led to improvements across all models, with the **GRU-only** model showing notable gains by reducing MSE to 21.21 and MAE to 3.22, and refining bid and ask errors to 2.80% and 1.81% on the testing data. This model maintained strong performance on Amazon contracts, with errors recorded at **12.24% and 7.23%,** with **~75% of predictions falling within 25% of true prices**. This consistency, particularly in underestimating peaks and overvaluing troughs, indicates the GRU model's potential in unveiling arbitrage opportunities, thus validating its simplicity and effectiveness in financial time series forecasting compared to LSTM and hybrid models.

Table 1. Key results of each RNN on the expanded dataset colored based on forecasting performance.

Models Trained on AMD, Apple, Microsoft and Google	Testing Data				Amazon Data			
Model Type	Bid Error	Ask Error	MSE	MAE	Bid Error	Ask Error	MSE	MAE
LSTM	6.34%	4.41%	108.75	6.63	1.50%	62.31%	43.39	3.74
GRU	2.80%	1.81%	21.21	3.22	12.23%	7.23%	22.64	2.87
GRU and FC Hybrid	0.53%	1.58%	27.33	3.94	10.21%	3.93%	25.62	3.12
GRU and Convolutional Hybrid	31.39%	30.50%	54.18	6.36	75.73%	77.97%	28.60	3.63

Conclusions

This research assessed various RNN architecture's ability to forecast American-style stock options, utilizing consistent hyperparameters and datasets. Key findings revealed the **GRU model's superior accuracy**. Notably, expanding the training set with additional tickers like Google and Microsoft enhanced predictive precision, particularly for the GRU model. **These results challenge the notion that more complex architectures yield better outcomes**.

The study extended model analysis to encompass options with higher prices, encountering predictive challenges due to stock splits and diverse pricing. This emphasized the necessity for improved normalization methods. Advancing these techniques and widening the dataset could heighten the models' alignment with market trends and enhance predictive precision, thus providing traders with more accurate insights for identifying arbitrage opportunities. This contribution enriches the discourse on RNN utility in financial analytics, spotlighting the GRU model's potential for advancing stock option pricing strategies.

Queen's University
April 3rd, 2023

18njp@queensu.ca

References

[1] J. Chen, "What are Options? Types, Spreads, Example, and Risk Metrics," Investopedia, 24 April 2023. [Online]. Available: https://www.investopedia.com/terms/o/option.asp. [Accessed 13 October 2023].
[2] C. Isidore, "Machines are driving Wall Street's wild ride, not humans," CNN, 6 February 2018. [Online]. Available: https://money.cnn.com/2018/02/06/investing/wall-street-computers-program-trading/index.html. [Accessed 11 March 2024].
[3] A. Hayes, "Black-Scholes Model: What It Is, How It Works, Options Formula," Adam Hayes, 5 May 2023. [Online]. Available: https://www.investopedia.com/terms/b/blackscholes.asp. [Accessed 12 October 2023].
[4] S. Cristina, "Calculus in Action: Neural Networks," Machine Learning Mastery, 16 March 2022. [Online]. Available: https://machinelearningmastery.com/calculus-in-action-neural-networks/. [Accessed 22 November 2023].
[5] D. Kalita, "A Brief Overview of Recurrent Neural Networks (RNN)," Analytics Vidhya, 3 August 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/. [Accessed 13 October 2023].
[6] M. Phi, "Illustrated Guide to LSTM's and GRU's: A step by step explanation," Medium, 24 September 2018. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.
[7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[8] I. Goodfellow, Y. Bengio and A. Courville, "1.2.1 The Many Names and Changing Fortunes of Neural Networks," in Deep Learning, Boston, MIT Press, 2016, p. 18.
[9] I. Jahan, "Stock Price Prediction Using Recurrent Neural Networks," North Dakota State University, Fargo, 2018.