Journal of
Risk and Financial Management

MDPI

Article

In-Season Price Forecasting in Cotton Futures Markets Using ARIMA, Neural Network, and LSTM Machine Learning Models

Jeffrey Vitale [1,*] and John Robinson [2]

[1] Department of Agricultural Economics, Oklahoma State University, Stillwater, OK 74078, USA
[2] Department of Agricultural Economics, Texas A&M University, College Station, TX 77843, USA; john.robinson@ag.tamu.edu
* Correspondence: jeffrey.vitale@okstate.edu

Abstract: This study explores the efficacy of advanced machine learning models, including various Long Short-Term Memory (LSTM) architectures and traditional time series approaches, for forecasting cotton futures prices. This analysis is motivated by the importance of accurate price forecasting to aid U.S. cotton producers in hedging and marketing decisions, particularly in the Texas Gulf region. The models evaluated included ARIMA, basic feedforward neural networks, basic LSTM, bidirectional LSTM, stacked LSTM, CNN LSTM, and 2D convolutional LSTM models. The forecasts were generated for five-, ten-, and fifteen-day periods using historical data spanning 2009 to 2023. The results demonstrated that advanced LSTM architectures outperformed other models across all forecast horizons, particularly during periods of significant price volatility, due to their enhanced ability to capture complex temporal and spatial dependencies. The findings suggest that incorporating advanced LSTM architectures can significantly improve forecasting accuracy, providing a robust tool for producers and market analysts to better navigate price risks. Future research could explore integrating additional contextual variables to enhance model performance further.

Keywords: cotton; LSTM; machine learning; futures

check for updates

Academic Editors: Thanasis Stengos and Leo H. Chan

Received: 12 November 2024
Revised: 20 January 2025
Accepted: 27 January 2025
Published: 10 February 2025

Citation: Vitale, J., & Robinson, J. (2025). In-Season Price Forecasting in Cotton Futures Markets Using ARIMA, Neural Network, and LSTM Machine Learning Models. *Journal of Risk and Financial Management*, *18*(2), 93. https://doi.org/10.3390/jrfm18020093

1. Introduction

Cotton futures markets have been an important marketing component of the U.S. cotton market since their inception in 1870 by the New York Cotton Exchange (Lipartito, 1983). Today the exchange is part of the Intercontinental Commodity Exchange, or ICE (Malliaris & Ziemba, 2015). This electronic cotton futures market still provides the important functions of global price discovery and price risk transfer (Gairola & Dey, 2023).

Repeated surveys of U.S. cotton producers indicate that only a very small percentage hedge their price risk directly (Pace & Robinson, 2012). Most U.S. cotton growers use a variety of cash marketing outlets (Vergara et al., 2004). For example, consider the earliest seasonal production region for U.S. cotton in South Texas. Growers in this region engage in cash forward contracting prior to planting (approximately 45% of the typical crop), enroll acreage in seasonal pools (approximately 30%), and/or contract post-harvest in the spot market (approximately 25%). The managers of seasonal pools may or may not hedge their expected inventory. Forward cash contracting and post-harvest spot purchases are typically conducted through merchant buyers who routinely hedge their expected inventory of physical cotton using ICE cotton futures (Adams, 2015).

Growers who contract with merchants still have pricing choices (Bailey & Richardson, 1985). The cash forward contracts in South Texas initially fix the grower's cash basis, but it is up to the grower to fix the futures price on which the contract price will be finalized (Xie et al., 2016). This arrangement gives growers the responsibility of forecasting futures prices and acting on the higher prices of the season (Darekar & Reddy, 2017; Hoffman & Meyer, 2018; Isengildina-Massa & MacDonald, 2013; Tomek, 1997). Cotton futures prices display significant annual seasonality, which is visually apparent in most years, as illustrated in Figure 1 (Hudson & Coble, 1999; Janzen et al., 2018; Keef & Zhu, 2009; Schneider & Tavin, 2024; Sørensen, 2002; West, 2012). Seasonally adjusted ICE December futures prices are, on average, the highest in the second quarter of the calendar year (Figure 2). This average is, of course, subject to conditioning variables. On average, the December post-harvest spot market has the seasonally lowest values for the December contract (Figure 1). The upward variation around these average prices can, however, provide desirable pricing opportunities for producers making accurate forecasts.

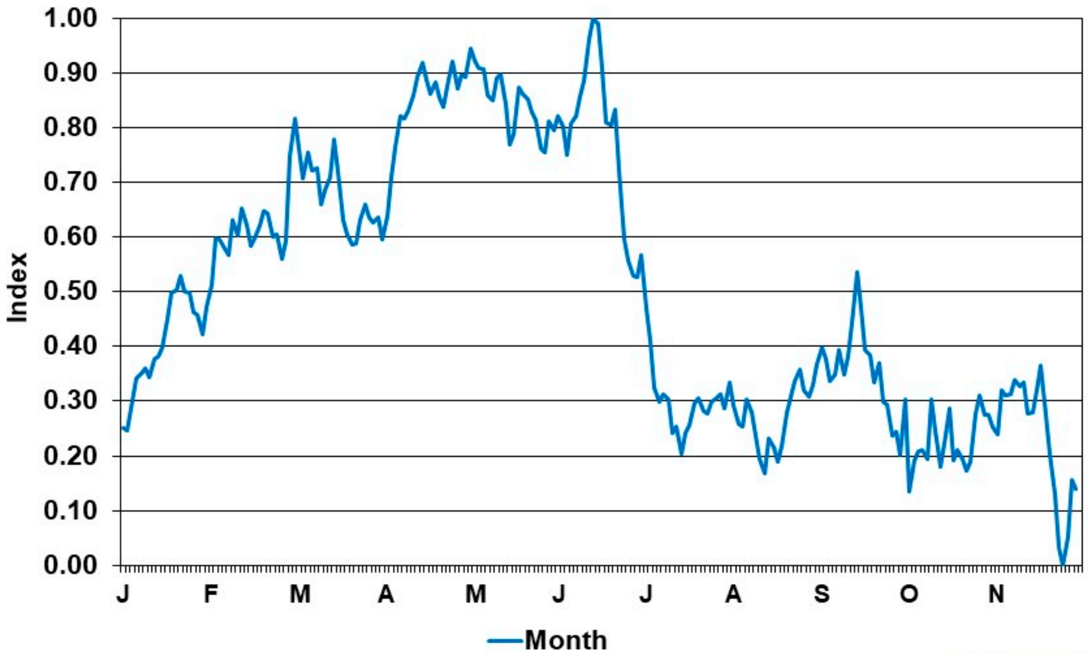


Figure 1. Seasonal index of ICE seasonal December prices, averaging daily settlements (1987–2023).

There have been a variety of methods utilized in the commodity marketing literature to forecast prices (Brandt & Bessler, 1983; L. Wang et al., 2020). Time series econometric regressions, modeling current prices through one or lagged prices and other exogenous variables, can significantly explain price movements (Darekar & Reddy, 2017; Shivakumar & Kotreshwar, 2017; D. Wang et al., 2017). Other domains, such as finance, weather prediction, and energy consumption forecasting, have also applied time series models to explain significant movements in stock prices, rainfall, and oil demand (Ariyo et al., 2014; Elsaraiti et al., 2021; Geetha & Nasira, 2016; Mondal et al., 2014; Pai & Lin, 2005). Forecasting models based on their results have also been found to be useful forecasting tools, including ARIMA and exponential smoothing. Although traditionally they have been widely used for time series forecasting, these traditional methods often struggle with capturing complex patterns and dependencies in the data (Makridakis et al., 1998; Smyl, 2020; Zhang, 2003).

Beginning in the early 1990s, neural network models have advanced as an alternative to time series econometric regressions (G. Zhang et al., 1998). Prior to 2010, feedforward neural networks (FNNs) significantly advanced the field of commodity price analysis, demonstrating superior performance over the statistically based time series methods in various applications (Zhang, 2003). Studies consistently showed that FNN models outperformed traditional statistical methods like ARIMA and GARCH in terms of forecasting accuracy (G. Zhang et al., 1998). This is attributed to the ability of FNNs to capture nonlinear relationships and interactions among variables (Kaastra & Boyd, 1996).

Figure 2. ICE December futures prices: 2009–2023.

For example, research by Zhang and Hu (G. Zhang & Hu, 1998) demonstrated that FNNs provided more accurate and reliable forecasts for exchange rates compared to ARIMA models. The neural network literature often criticizes FNNs as opaque, black-box models that lack interpretability, making it challenging to understand the patterns and relationships learned during training. This lack of interpretability was a significant drawback for stakeholders seeking to understand the underlying drivers of commodity prices. Despite their challenges, including being prone to overfitting, especially when training data was small, FNNs provided valuable insights and more accurate forecasts by effectively capturing nonlinear relationships and complex dynamics in commodity markets.

Recent advances and interest in machine learning and artificial intelligence have brought a renewed interest in the use of neural networks in forecasting commodity price movements. One of the more promising approaches to improve forecasting accuracy is Long Short-Term Memory (LSTM) networks, an extension of ordinary "vanilla" recurrent neural networks (RNNs). Although not specifically designed for time series, LSTM models have often outperformed both traditional time series models, e.g., ARIMA, as well as RNNs and their variants. LSTM architecture was developed to identify long-term temporal dependencies and nonlinear spatial patterns in data. Initially conceived by Hochreiter and Schmidhuber (Hochreiter & Schmidhuber, 1997), LSTM models consist of a series of gates (input, forget, and output gates) that regulate the flow of information in a parsimonious manner, enabling the network to retain prior neural information identified as having significant explanatory value while parsing less valuable neural information. Through improved neural information flow, LSTMs reduce problems associated with the backward propagation's "vanishing gradient". This can occur in traditional RNNs when the gradient's weight vector quickly diminishes as updating proceeds, making it difficult to identify long-term patterns and trends in time series data (Hochreiter, 1998).

LSTMs have empirically outperformed traditional time series models, such as ARIMA and exponential smoothing, in forecasting prediction accuracy in major agricultural commodities, including cotton (Z. Jiang et al., 2021), maize (Gu et al., 2022; Jaiswal et al., 2022), and wheat (Murugesan et al., 2022). In other commodity applications, LSTMs have also outperformed traditional time series approaches, including energy (Memarzadeh & Keynia, 2021; Zhou et al., 2019), metals (Nguyen et al., 2024; Ozdemir et al., 2022), stock prices (Lu et al., 2020; Sunny et al., 2020; Y. Wang et al., 2018), and exchange rates (Cao et al., 2020; Wijesinghe, 2020).

There are several common LSTM architectures that are used for different types of tasks, including sequence prediction, sequence classification, and many-to-many sequence processing. The most basic, or vanilla, LSTM is a network with a single hidden LSTM layer and is acceptable for simple sequence prediction problems (Bukhari et al., 2023; Wu et al., 2018). A stacked LSTM is a network with multiple LSTM layers stacked on top of each other, allowing the network to learn more complex representations (Du et al., 2017; Ma et al., 2022; Y. Wang et al., 2018). A bidirectional LSTM is a network where each layer has two parallel LSTMs, one processing the input sequence from start to end, and the other from end to start (He et al., 2021; Ullah et al., 2017; Zhao et al., 2017). This helps in capturing information from both past and future states. An encoder–decoder LSTM is used for variable length sequence-to-sequence prediction problems, such as language translation, image captioning, machine translation, etc. (Z. Wang et al., 2020; Bappy et al., 2019). The duo work in consecutive tandem: the encoder transforms the input sequence into a context vector that the decoder transforms into the output sequence. An advanced LSTM architecture is one that includes an attention mechanism, designed to enhance the encoder–decoder functioning when handling long input sequences through selective attention, i.e., focusing only on the most relevant portion of the sequence when generating output sequences (Abbasimehr & Paki, 2022; B. Jiang et al., 2022). A CNN LSTM combines convolutional layers for feature extraction from spatial data (like images) with LSTM layers for sequence processing, often in combination with other architecture (Behera et al., 2021; Karim et al., 2017; Yu et al., 2018; Zhang et al., 2018).

This paper contributes to the AI-ML literature by comparing a suite of LSTM models to traditional NNs and time series models. Our paper extends previous work by including new LSTM architecture and addressing price forecasts in cotton futures markets not previously considered. Hence, the purpose of this paper is to explore how recent advances in machine learning can generate more powerful price forecasting tools to assist producers in hedging and subsequent marketing of cotton on spot markets in the Texas Gulf region. Forecasts of the basis used by producers in pre-planting hedging are developed and assessed by presenting a likely range of dates on which futures contracts could be sold prior to planting and a similar range on which harvested ginned cotton could be shipped to spot markets.

2. LSTM Architecture

LSTM models operate by carefully controlling the flow of model information through cell states using gates. Through a complete range of open to closed, gates enable the network to either retain or purge information over long modeling periods, making LSTMs particularly effective for tasks involving sequences and time series data, such as natural language processing, image processing, commodity prices, speech recognition, and more. The LSTM architecture has three main primary components: cell state, hidden state, and gates (Figure 3). The cell state C_t is the main memory of the network. It flows along the entire sequence and maintains information over extended modeling periods, enabling it to capture long-term dependencies. Functionally, it is defined primarily in a linear manner to avoid large changes that can create "vanishing gradient" problems, also allowing information to be selectively managed through hidden states and memory gates. The hidden state h_t carries information across time from one step to the next, i.e., the output of the LSTM cell for a particular time step. Gates are key to LSTM's ability to pass information throughout the cell state by either adding new or removing old information to and from the cell state. Mathematically, gates employ the sigmoid function and weight matrices that mimic gate operation numerically (0 = open and 1 = closed), governing how much of each cell state's memory component should be allowed through the LSTM gated network structure.

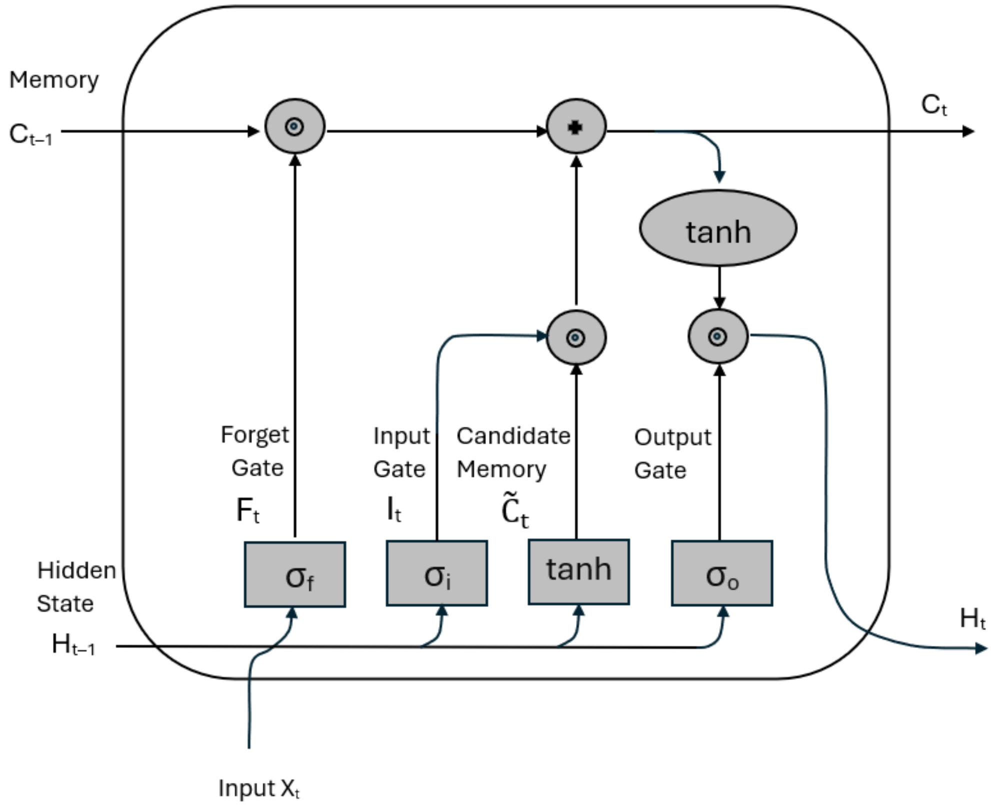


Figure 3. LSTM architecture illustrating the flow of information across the forget input, candidate, and output gates.

2.1. Forget, Input, and Output Gates

Cells are structures within the LSTM network that parsimoniously control information flow through a handful of gates with specified functions (Figure 3). The forget gate functions to determine what information to remove from the cell state. It processes information from previous states and takes the previous hidden state, h_{t-1}, along with the current input, x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. An f_t value of 0 removes all previous cell information while f_t = 1 retains all previous cell information. Values between 0 and 1 retain partial information based on the following sigmoid function:

$$f_t = \sigma_f\left(W_f \bullet [h_{t-1}, x_t] + b_f\right)$$

where σ is the sigmoid function, which outputs values between 0 and 1, W_f is the weight matrix for the forget gate, b_f is the bias term for the forget gate, and the other terms have been previously defined. The sigmoid function outputs a value between 0 and 1 based on the information flowing into the gate. The bias term is a linear intercept term that shifts the output of the sigmoid and assists the model in recognizing more flexible mappings, i.e., enabling the forget gate to adjust its output to be more (or less) biased towards retaining (or forgetting) information. Positive b_f values will tend to retain more information from the previous cell state while the reverse is true when b_f is negative, helping the LSTM balance the retention of long-term dependencies.

The input gate governs which values flowing into it will be updated in the current cell state. The input gate layer decides which values will be updated using the following sigmoid function, similar in structure to the forget gate:

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)$$

The candidate value, $\widetilde{C}_t$, creates a vector of new candidate values in each time step that could be added to the current cell state and operates in conjunction with the input gate. A tanh layer is typically used to create the vector of new candidate values, as given by

$$\widetilde{C}_t = tanh(W_C \bullet [h_{t-1}, x_t] + b_C)$$

The update cell state finalizes the candidate value by replacing the old cell states from the previous time step, C_{t-1}, into the new cell states in the current time step C_t using the following:

$$C_t = f_t \bullet C_{t-1} + i_t \bullet \widetilde{C}_t$$

The update process is completed by multiplying the old state by ft, in essence repeating the process of forgetting information that was forgotten earlier in the solution process. The second term on the right-hand side of the equation, $i_t \bullet \widetilde{C}_t$, is added to the previous term to form the new candidate values, which are scaled by how much the cell decided to update each state value.

The output gate decides what part of the cell state should be used in the output and passed to the successive hidden state, h_t. The updating decisions are based on the following sigmoid function:

$$o_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)$$

and the corresponding updating function:

$$h_t = o_t \bullet tanh(C_t)$$

2.2. Data Analysis and Model Parameterization

Five alternative LSTM models and an ARIMA time series model were evaluated on their forecast accuracy for the December futures cotton price. The dataset used in this study comprises historical time series data of average daily December cotton futures prices from 6 December 2009 to 9 December 2023 (Figure 2). To ensure efficient training of the LSTM model, the data was normalized to a range between 0 and 1 using min–max normalization:

$$y_{norm} = \frac{(y - y_{min})}{(y_{max} - y_{min})}$$

where y is the original non-normalized value, and y_{min} and y_{max} are the respective minimum and maximum values of y. The LSTM and ARIMA model architectures utilized in the respective models were designed to capture the temporal dependencies in the time series data. The LSTM models were thus compiled using the optimizer's mean squared error (MSE) loss function to provide a comprehensive assessment of model forecasting accuracy, including the ARIMA model:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_{i+t} - y_{i+t})^2$$

where i is the daily price observations, t is the daily price, T is the number of time step forecasts, N is the total number of price observations, $\hat{y}_{i+t}$ is the forecasted price, and y is the actual price. To provide a more practical interpretation, RMSE, the square root of the previous equation, is used when presenting model performance results.

2.3. Model Training

In this study, hyperparameter selection for the LSTM models was conducted using a K-fold random search strategy. The validation dataset was selected to represent a portion of the time series that was independent of the training data, but contiguous with the test data, reflecting real-world forecasting scenarios. Importantly, no "cherry-picking" of the validation data was performed. The same validation set was consistently used across all model configurations to ensure fairness in comparing model performance.

To minimize overfitting and prevent data leakage, the test set, which remained completely unseen during training and hyperparameter optimization, was used solely for final evaluation. This methodology ensures that the reported test results accurately reflect model performance on new, unseen data, avoiding potential biases from overfitting to validation data. While unforeseen trends in the test data remain a common challenge in time series forecasting, this study adheres rigorously to established best practices to mitigate such risks.

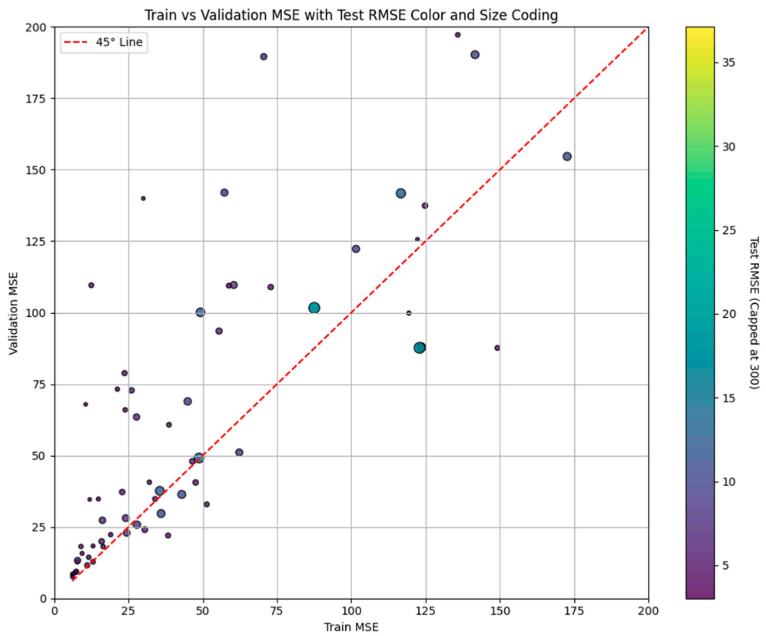
The dataset was split into training–validation and test sets with a ratio of 60:40 for the neural network models. The training set was used to fit the model, while the test set was reserved exclusively for evaluating forecasting performance. To achieve a balance between underfitting and overfitting, Keras's EarlyStopping function was employed during training, stopping the process when validation loss failed to improve for five consecutive epochs.

To optimize the models, RandomSearch was employed to fine-tune parameters, such as the number of LSTM units, batch size, learning rate, patience (early stopping), and number of epochs. Five folds were used in the hyperparameter search process to increase robustness. Figure 4 illustrates the results of the hyperparameter random search for the convolutional 2D model, highlighting the sensitivity of forecasting accuracy to hyperparameter selection. Suboptimal parameter choices resulted in substantially higher MSE values or overfitting, where a model performed well on training data but poorly on unseen data.

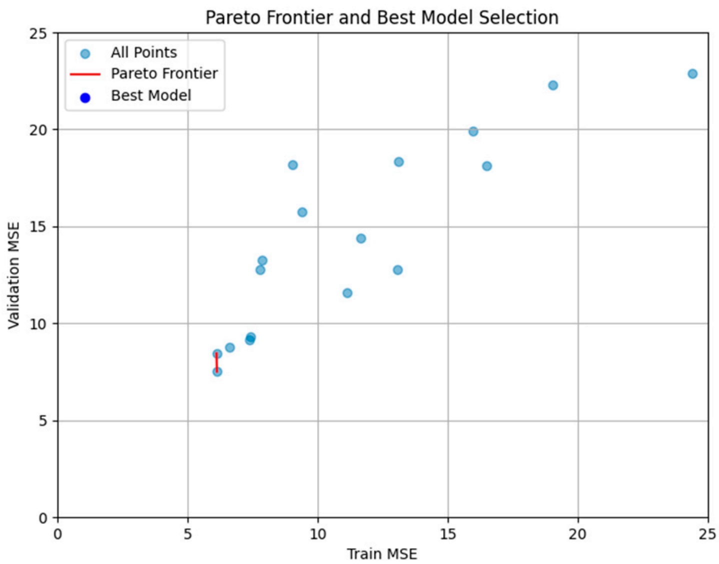
A Pareto frontier was developed, as shown in Figure 4, to identify models where any improvement in training MSE came at the expense of validation MSE, or vice versa. Since the Pareto frontier contains at least two points, as illustrated in Figure 4, a metric needs to be invoked to select a single best model. To mitigate overfitting, the best model was selected using a composite metric that equally weighed two factors: (1) the model's distance from the 45-degree line where training RMSE equals validation RMSE and (2) the training RMSE itself. This systematic selection process ensured the chosen models were robust and generalizable to unseen data, i.e., the test data was not included.

Figure 4 illustrates the training and validation losses during the F-fold random search. The observed difference between training and validation RMSE across models suggests that any overfitting is likely a result of inherent tendencies within the data. After achieving satisfactory performance, the trained LSTM model for each architecture was deployed for real-time forecasting on the test data. This systematic approach highlights the complexity of constructing neural network models, especially LSTM architectures. Even seemingly minor variations in hyperparameters, such as the number of epochs, learning rates, or dropout rates, can significantly impact model performance (Figure 4).

The ARIMA model was selected using the Auto ARIMA procedure in Python, which automates the process of identifying the best model configuration by minimizing the Akaike Information Criterion (AIC). Auto ARIMA evaluates multiple combinations of model parameters, including the autoregressive (AR) terms, differencing (I), and moving average (MA) terms, as well as the inclusion of intercepts. In this study, the model configurations ranged from simple ARIMA(0,1,0) to more complex models like ARIMA(2,1,2). Each configuration was assessed based on its AIC value, with lower values indicating better model fit while balancing complexity.



(**a**) Train versus validation



(**b**) Pareto frontier

Figure 4. K-fold (k = 5) model hyperparameter random search results for 2D convolution LSTM model (5-day forecast) and Pareto frontier.

The best-performing model, ARIMA(0,1,2), achieved the lowest AIC of 7546.257, demonstrating a superior balance between goodness-of-fit and model complexity compared to alternatives such as ARIMA(2,1,2) (AIC = 7550.699) and ARIMA(0,1,3) (AIC = 7547.468). The total fit time for the procedure was approximately 12 s, reflecting the efficiency of the Auto ARIMA algorithm in exploring the parameter space. Additionally, the diagnostic statistics for the selected model indicate its robustness: the Ljung-Box test for autocorrelation returned a non-significant p-value of 0.94, suggesting the residuals are uncorrelated, while the Jarque-Bera test confirmed non-normality, potentially due to the presence of heavy tails in the distribution.

The ARIMA(0,1,2) model implies a non-stationary process that was made stationary through first-order differencing (I = 1) and exhibits a dependence structure captured by the two moving average terms (MA). The parameter estimates for MA(L1) and MA(L2) were statistically significant ($p < 0.001$), with coefficients of 0.0531 and −0.0857, respectively, further validating the model. The small variance in the residuals ($\sigma^2 = 1.6294$) indicates that the model captures the underlying data trends effectively, making it a robust choice for forecasting.

3. Results

Forecasts developed for five, ten, and fifteen days serve as this paper's primary metric for comparison among the alternative models (Table 1). Figure 5 illustrates the time progression of the cumulative RMSE for five-, ten-, and fifteen-day predicted cotton futures prices using the 2019–2023 test data set. The following compares the overall performance of each model, beginning with the best-performing model.

Table 1. Comparison of the root mean square error (RMSE) model performance for 5-, 10-, and 15-day cotton futures price forecasts during the 1st and 2nd price shocks and the total over the test data set.

Model	5 Day			10 Day			15 Day		
	1st Shock [a]	2nd Shock	Total	1st Shock	2nd Shock	Total	1st Shock	2nd Shock	Total
ARIMA	0.75	0.59	2.79	1.40	1.27	5.46	1.29	1.22	0.75
Simple Neural Network	0.67	0.61	3.55	1.08	0.81	6.00	1.86	1.37	0.67
Sequential (Univ) LSTM	0.69	0.65	3.69	1.03	0.70	5.68	1.22	0.99	0.69
Stacked LSTM	0.88	0.93	4.96	1.45	1.97	9.37	2.05	1.36	0.88
Bidirectional LSTM	0.50	0.33	2.98	1.01	0.65	5.92	1.39	0.82	0.50
CNN LSTM	0.54	0.39	3.75	1.28	1.00	6.98	1.59	1.06	0.54
2D Convolutional LSTM	0.70	0.19	1.70	0.64	0.19	1.67	0.85	0.17	0.70

[a] RMSE values reported for the 1st and 2nd shocks are the differences in cumulative RMSEs between the beginning and the end of the shock.

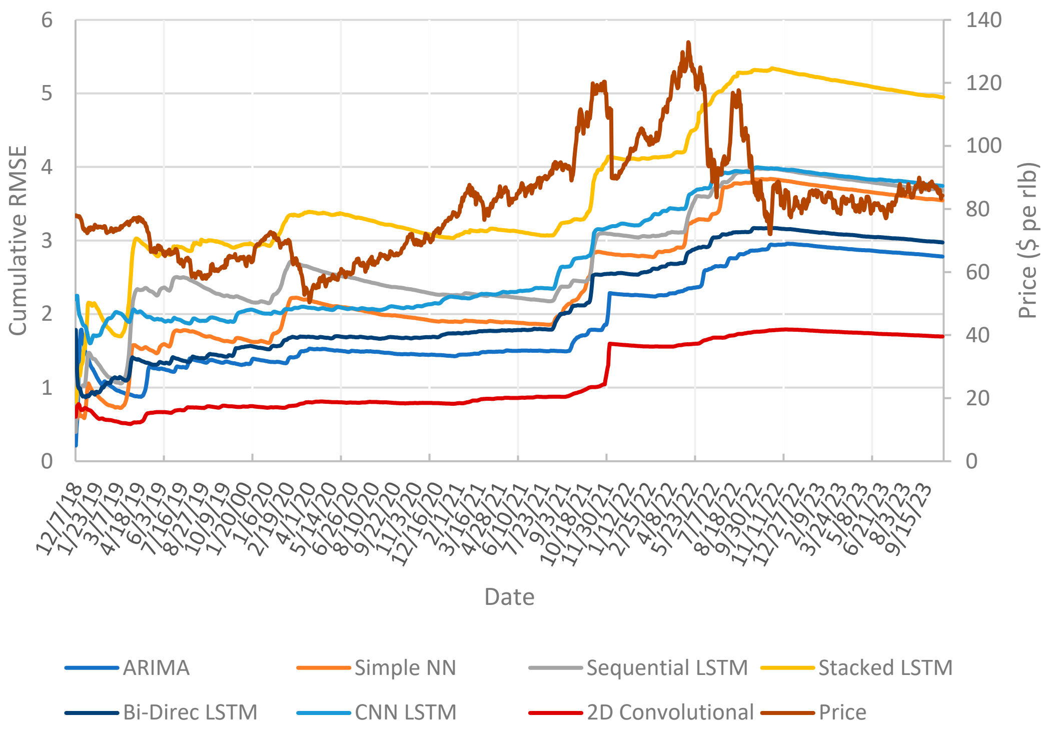


Figure 5. Five-day forecast cumulative RMSE performance results.

3.1. Model Performance

The 2D LSTM model yielded the best RMSE performance across all three forecast periods (Table 1). The results suggest that the 2D LSTM is better at capturing complex, non-linear, and multi-dimensional relationships in time series data, making it more versatile and powerful than a linear model like ARIMA or any of the LSTM variants. This capability is especially useful for data with intricate spatial-temporal patterns, where their ability to stack multiple layers and apply pooling operations allows the convolutional models to learn and retain long-term dependencies by focusing on important parts of the input data across longer time periods. The attention mechanism enabled the model to focus on important time steps in the input sequence, thereby enhancing its predictive capability.

The ARIMA model provided the second-best performance, with RMSE forecast errors that were 66, 227, and 177 percent higher than the 2D convolution LSTM model across the five-, ten-, and fifteen-day forecasts (Table 1). ARIMA has been shown from prior studies to perform well for simple, stationary, and linear time series data, but often struggles with more complex, non-linear relationships compared to other models, such as LSTM. This suggests that the cotton futures price dataset contained enough non-linearities, and other complexities, that were better handled by the 2D convolution LSTM model. ARIMA can, however, still be effective and computationally efficient as it outperformed all the LSTM variants except the 2D convolutional.

The basic NN model provided a modest degree of RMSE performance in all three of the five-, ten-, and fifteen-day forecasts, outperforming at least one of the LSTM models in each forecast (Table 1). In the five- and ten-day forecasts, it outpredicted the Stacked and CNN LSTM models but had higher errors compared to the remaining LSTM and ARIMA models (Table 1). Its RMSE performance compared to other models remained essentially the same in the 15-day forecast, outperforming the stacked LSTM model while underperforming the remaining LSTM and ARIMA models. Hence, the model results from this study suggest that the LSTM architecture did not provide any meaningful increase in predictive power except for the bidirectional and 2D convolutional models, which both outperformed the basic NN in all three forecasts (Table 1). This finding could be explained by the time period's lack of complex non-linearities and time dependencies that LSTM architecture would better explain than simple NN models.

Compared to the 2D convolutional LSTM model, none of the other LSTM variants performed as well. While there was some variation across the different forecast periods, the 2D Convolutional LSTM outperformed the next-best LSTM model's MSE by an average of 143% (Table 1). Among the LSTM variants, the bidirectional and sequential models performed, generally speaking, as "second best" to the 2D convolutional model. Both of their architectures captured more of the complex temporal dependencies due to their deeper architecture compared to the stacked, CNN, and vanilla LSTM models. The importance of the LSTM architecture is most apparent when considering the substantial improvement in forecasting performance obtained through including the 2D features of the 2D convolutional model. The model performance improved by an average of 192% with the 2D features included in the convolution model (Table 1). The results suggest using 2D tensor transformations to convert time series data from its original one-dimensional format into a 2D format before being processed by the CNN, which leverages the strengths of 2D convolutions in capturing intra- and inter-period price variations. The presence of the 2D filter prior to the CNN enhances its capability to recognize local patterns that repeat, such as time series data with patterns that occur in weekly, monthly, and seasonal trends that other LSTM and ARIMA models are less able to recognize. The improved performance utilizing the 2D filter with a CNN LSTM has also been found in other studies using time series data (Lahboub & Benali, 2024; Ampountolas, 2024).

3.2. Forecasting During Price Volatility

Forecasting performance experienced two challenging periods of price volatility, one beginning circa September 2021 and the other in June 2022 (Figure 2). The first shock began in mid-September 2021. Market prices took an unexpected, sharp increase from \$0.89 to \$1.11 in the 15-day market period from 20 September 2021 to 8 October 2021, which persisted until early December (9 December 2021) when prices reverted back to their long-term trend (Figures 5–7). Price forecasting prior to the first shock showed only modest variation among all models except the stacked LSTM, which had the highest MSE values across all three forecasting lengths. The 2D convolution model performed best prior to the first price shock for all three forecasts. The forecasted cumulative RMSE values for the 5-day period prior to the first shock ranged from a low of 0.89 for the 2D convolutional model to a high of 3.25 for the stacked LSTM, corresponding to a 265% increase, with the other models performing within the range of 69% to 196% above the best performing 2D convolutional model (Figure 5). Similar MSE forecasting performance differences were found in the 10 and 15-day forecasts.

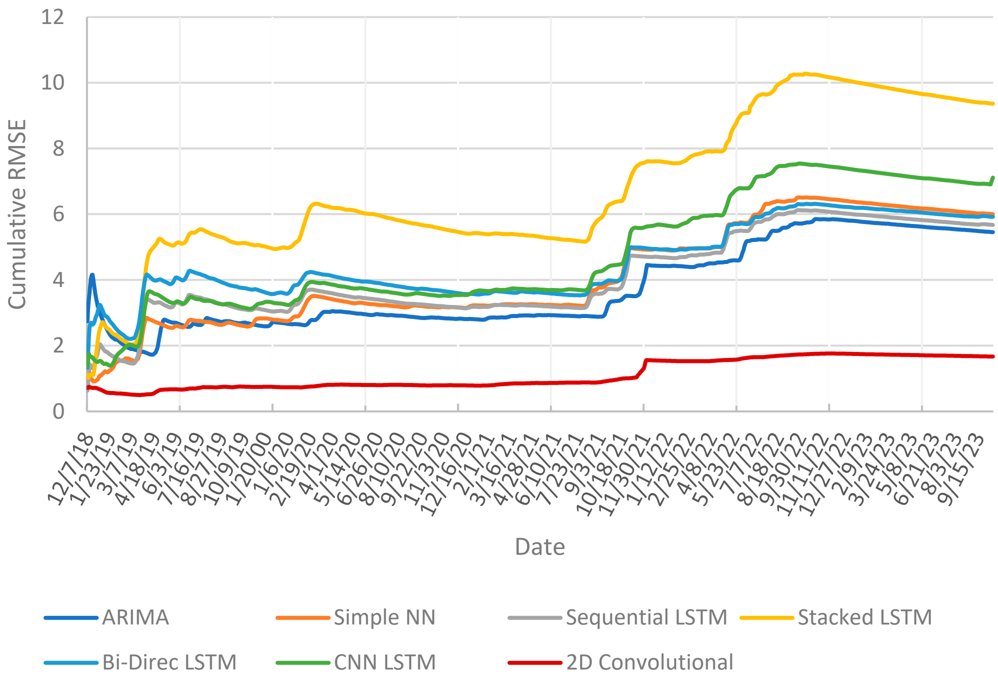


Figure 6. Ten-day forecast cumulative RMSE performance results.

Price volatility significantly impaired market forecasting performance during the price shock as the unexpected price movement increased forecasting accuracy, which is visually apparent for all three forecasting lengths (Figures 5–7). Throughout the September–December 2021 price shock period, the average cumulative RMSE across all models for the 5-day forecast increased by 0.67, with corresponding increases in daily MSE forecasts for the 10- and 15-day lengths of 1.40 and 1.47 (Table 1). The differences in model performance became more apparent during the period of volatility, and the 2D convolutional LSTM performed best overall across all three forecasting periods, with performance strengthening from the 5- to 15-day forecasting lengths (Table 1). The second-best performing model, the bidirectional LSTM, outperformed the convolutional model in the 5-day forecast during shock 1 by 40% but had RMSE values of 57% and 373% higher for 10- and 15-day forecasts compared to the 2D convolutional LSTM (Table 1). The ARIMA model's performance, however, experienced a significant loss in predictive power when confronted with the price volatility, as the bidirectional LSTM model outperformed it during the price shock period, a reversal from the prior pre-shock period (Figures 3–5). The ARIMA forecast RMSE is higher than the bidirectional LSTM model across the 5-, 10-, and 15-day forecasts during the first price shock (Table 1). Moreover, in the 10-day forecasts, the ARIMA model performed worse than both the simple NN and the sequential LSTM models, and in the 15-day forecast, it performed worse than the CNN LSTM model (Table 1). This finding suggests that the ARIMA model was less able to predict the price volatility compared to the LSTM models, which contain architecture that can often better identify patterns of sudden price movements.

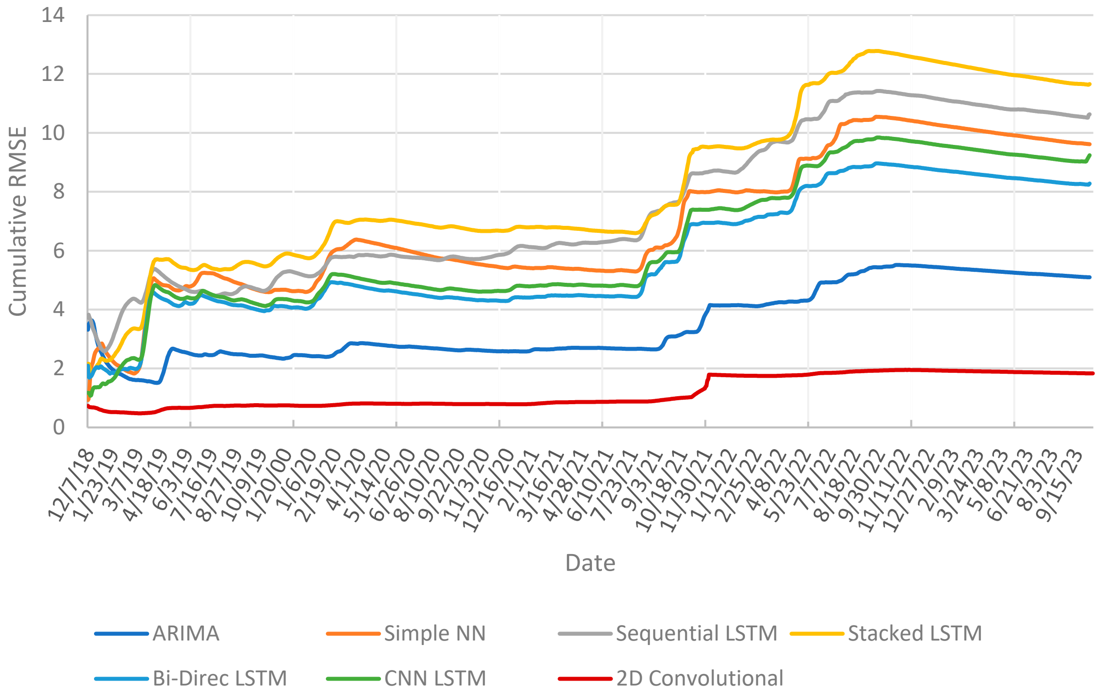


Figure 7. Fifteen-day forecast cumulative RMSE performance results.

A second period of price volatility ensued the following spring and lasted several months from 16 May 2022 to 31 October 2022. This was a period of a dramatic price collapse from $1.33 on 16 May 2022 to $0.94 on 1 August 2022, followed by a temporary price recovery that lasted until 26 August 2022 when it reached $1.18 and subsequently collapsed again to $0.72 on 31 October 2022 (Figure 2). Average cumulative RMSE values across all seven models were similar across both price volatility periods, with slightly higher cumulative RMSE values for the second shock in the 5- and 10-day forecasts that averaged 0.53 and 0.94. Average cumulative RMSE values across the three forecasting periods were 27.6%, 19.6%, and 46.7% higher in the first shock compared to the second (Table 1).

A noticeable difference in this second price shock was, however, the superior performance of the 2D convolutional model compared to the others, particularly for the 15-day forecast, in which it was the only model that experienced no substantial MSE increase (Figure 7). As in the first price shock period, the 2D convolutional LSTM's performance grew stronger as the forecasting period increased. Across the 5- to 15-day forecasts, it substantially outperformed all other models. The second-best-performing model had cumulative RMSE values 76%, 274%, and 373% higher than the 2D convolutional LSTM model across the three forecasting periods (Table 1). The second price shock reinforced the finding from the prior shock that the ARIMA model had more difficulty forecasting following an unexpected price movement compared to many of the LSTM models. ARIMA was outperformed again by the bidirectional LSTM across all three forecast periods, and ARIMA performance grew worse as the number of forecasting days increased (Table 1). In the 10 forecasts, the simple NN, sequential, and bidirectional models outperformed the ARIMA model, whose cumulative RMSE values were 58.1%,82.9%, and 94.9% higher than the three models, respectively (Table 1). In the 15-day forecast, the ARIMA model was outperformed during shock 2 by all models except the simple NN and stacked LSTM models.

3.3. Discussion

Forecasting issues during periods of price volatility have been addressed in the time series literature. Research findings suggest that CNN-based architectures typically outperform other models at handling noisy and volatile data because they emphasize localized feature extraction without undue overfitting to past trends. Convolutional models, especially when fortified with 2D convolutional layers, are structured to identify localized patterns and short-term fluctuations in price movements. Hence, during a period of sudden price increase, convolutional models enhanced feature extraction can effectively capture short-term, high-frequency variations in time series data due to their pattern recognition capability, enabling timely response to recent, unexpected price fluctuations. Specifically, convolutional layers act like pattern detectors, making them more resilient to shocks or abrupt changes in data through the 2D layers' filtering algorithm that is less affected by dynamic changes in temporal price movements.

The ARIMA model performed generally well during the price volatility periods, though weaker than the 2D convolutional LSTM model. ARIMA models are inherently built to capture linear dependencies and temporal trends, and they do well when the data is mostly linear with occasional abrupt price changes. ARIMA models rely heavily on recent observations, which helps them adjust predictions quickly when there is a sudden shift in the trend, as it can quickly factor in the most recent data, adjusting to the trends that occur abruptly. Large price changes though can still increase its MSE since it lacks the ability to dynamically adapt as quickly as convolution models, particularly layered 2D convolutional models, as found in this research.

The other LSTM models also had weaker forecasting performance compared to the 2D convolutional LSTM model, as their architecture was less able to adapt to sudden price movements. Basic feedforward neural networks are limited in capturing sequential dependencies, especially in the face of sudden, significant changes in data trends. These models may suffer because they do not inherently understand underlying patterns in the temporal nature of time series data without additional structural layers, such as LSTMs and/or CNNs, which are explicitly designed for sequence learning. This limitation can cause MSE to spike when there is sudden volatility. In the forecasting literature, basic neural networks often underperform in cases where sequential dependencies or trend shifts are critical, as they lack the temporal memory necessary to account for such sudden changes.

The other LSTM models, such as stacked or bidirectional LSTM models, are designed to learn long-term dependencies. While this makes them appropriate and often powerful for stable, complex time series data, they can struggle with overfitting to past patterns when a sudden and unexpected change occurs, which appeared to be less of an issue with the 2D convolutional LSTM and ARIMA models, whose design was better suited to handle unexpected price movements. Price volatility disrupts the temporal patterns that non-convolutional LSTM models rely on, causing errors to propagate through the model. Bidirectional models, in particular, process data in both forward and backward directions, which can be problematic during periods of rapid change, as they may attempt to "smooth" the data inappropriately during training. Additionally, stacked models increase complexity, and they may suffer during volatile periods because they try to reconcile abrupt shifts with learned patterns from stable periods. This aligns with findings from the forecasting literature where LSTM models, though powerful, sometimes face difficulties when dealing with highly volatile or noisy data due to their reliance on prioritizing a consistent long-term memory. Hence, they are sensitive to noise and unexpected changes, and reliance on past sequences can cause predictions to lag the actual data, leading to higher errors.

Other studies report similar findings on convolutional LSTM models' superior performance compared to ARIMA and other LSTM architecture. In forecasting stock prices of Moroccan Credit Companies, Lahboub and Benali (2024) reported that convolutional LSTM models outperformed ARIMA models in time series containing frequent price fluctuations and short-term volatility (Lahboub & Benali, 2024). Their results suggest that convolution LSTM layers help capture localized temporal features, making them well-suited to volatile data. In forecasting futures prices in orange markets, Ampountolas (2024) reported that convolution LSTM models consistently outperformed ARIMA and LSTM across multiple marketing horizons, resulting in superior MSE compared to ARIMA (Ampountolas, 2024). Their convolutional component effectively captures temporal price patterns in volatile markets, resulting in lower prediction errors compared to ARIMA. Similar findings from Dong and Zhou (2024) documented that convolution LSTM provided lower error rates, and more accurately captured temporal and spatial patterns in complex, non-stationary time series data (Dong & Zhou, 2024).

3.4. Model Interpretability Versus Accuracy

While the advanced LSTM architectures demonstrated superior forecasting accuracy compared to ARIMA models, interpretability remains a key concern. Traditional ARIMA models offer well-established statistical inference capabilities, enabling users to derive clear insights into the relationships between variables and their contributions to the forecast. For example, ARIMA models have successfully identified statistically significant effects of exogenous variables, such as interest rates on exchange rates (Pai & Lin, 2005) or rainfall levels on crop yields (Geetha & Nasira, 2016), providing meaningful real-world implications. Furthermore, ARIMA models can estimate structural equations grounded in economic and financial theory, allowing researchers to test hypotheses about causal relationships that machine learning models, including LSTMs, cannot equally replicate. This transparency and theoretical congruence aid in trust and understanding trust, making ARIMA more accessible to decision-makers and academics seeking to analyze the theoretical underpinnings of observed market phenomena.

In contrast, ML models, particularly neural networks, are often criticized as "black-box" approaches due to their reliance on complex, non-linear transformations and numerous parameters. This opacity can hinder the ability of practitioners to understand the drivers behind specific predictions, which is critical in high-stakes domains like agricultural commodity forecasting. While this study did not include LSTM architectures with integrated attention mechanisms, such models could provide pathways to enhance interpretability by explicitly focusing on the most influential temporal features in the structured, lagged price data. However, these tools are still evolving and require further refinement to achieve the level of transparency provided by traditional statistical models. Bridging the gap between accuracy and interpretability will be a crucial area for future research, potentially through hybrid models or frameworks explicitly designed to explain ML-based forecasts.

3.5. Summary

The results demonstrate that more sophisticated LSTM architectures, such as 2D convolutional, significantly improve the accuracy of cotton futures price forecasts compared to basic NN and LSTM models, as well as the ARIMA model. These advanced models effectively capture complex temporal dependencies and enhance the model's ability to generalize from the training data. The incorporation of dropout regularization, and the stacking of multiple LSTM layers, contribute to better performance by mitigating overfitting and capturing deeper patterns in the futures price data. One of the primary reasons for the superior performance of some of the LSTM models is their ability to handle long-term dependencies in time series data. Unlike traditional neural networks, which often struggle with vanishing or exploding gradient problems, LSTMs have a unique architecture with memory cells that can store information over long time periods. This memory capability allows LSTMs to effectively learn temporal patterns and dependencies that are crucial for accurate time series forecasting, particularly those that are not explained by typical seasonal price trends.

Time series data often exhibit complex patterns, including seasonality, trends, and noise. Traditional neural networks, with their feedforward architecture, lack the inherent ability to model these temporal dependencies effectively. While ARIMA models are designed to handle linear patterns and can capture some aspects of seasonality and trend, they fall short when dealing with non-linear and more complex temporal patterns. LSTMs, on the other hand, are designed to capture both linear and non-linear dependencies, making them more adaptable to a wider range and more complex time series data.

LSTM models also offer scalability and flexibility, making them suitable for various types of time series data. Time series data often contain noise and irregular fluctuations that can negatively impact forecasting accuracy. LSTMs have demonstrated robustness to such noise, as their recurrent nature allows them to smooth out short-term fluctuations and focus on the underlying trend. Traditional neural networks and ARIMA models, however, are more susceptible to being influenced by noise, leading to less reliable forecasts.

While the results demonstrate the superior performance of advanced LSTM architectures, particularly the 2D convolutional LSTM, this improvement comes at the cost of increased computational complexity. The optimization process required extensive hyperparameter tuning using random search and K-fold cross-validation, which demanded significant computational time and resources. Additionally, the architecture's complexity, including multiple layers and dropout mechanisms, adds to the challenges of implementation. These trade-offs highlight a critical consideration for practitioners: while LSTM models offer improved forecasting accuracy, their deployment may require specialized expertise, greater computational resources, and careful balancing of model complexity against practical constraints. Future research should explore strategies to streamline this complexity, such as leveraging automated machine learning (AutoML) techniques or hybrid models that combine the strengths of simpler approaches with advanced neural networks.

In addition to the improved forecast accuracy demonstrated by 2D convolutional LSTMs, our models provide tangible benefits to U.S. cotton producers, particularly those in South Texas. Producers facing significant price volatility can leverage these models to make more informed hedging and marketing decisions, reducing financial risk and enhancing profitability. This practical utility underscores the value of deep learning in agricultural commodity forecasting and offers a foundation for broader applications across other markets and commodities. By addressing the challenges of forecasting in volatile commodity markets, our findings provide actionable tools for producers and traders. The methodologies can also serve as a template for other sectors, such as energy or finance, where accurate time series forecasting is critical.

4. Conclusions

This research underscores the potential of advanced machine learning models, specifically the 2D convolutional LSTM, in forecasting cotton futures prices with higher accuracy than traditional ARIMA and other neural network architectures. The comparative analysis revealed that while simpler models like ARIMA and feedforward neural networks provided reasonable predictions, they struggled during periods of significant price volatility. In contrast, the 2D convolutional LSTM consistently outperformed other models due to its ability to capture both short-term fluctuations and long-term dependencies through its unique layered architecture.

This study's results suggest that cotton producers and market analysts could leverage such advanced forecasting models to make more informed hedging and marketing decisions, ultimately improving profitability and reducing financial risk. The superior performance of the 2D convolutional LSTM highlights the importance of adopting modern machine learning techniques that can adapt to complex and volatile market environments.

While machine learning and AI algorithms can, in certain applications, outperform traditional econometric and statistical methods, their benefits should be weighed alongside the increased computational time and complexity. This study was able to be conducted using a typical personal computer, but real-time and real-world implementation would likely require much larger computer facilities. Businesses and institutions could face human capital limitations given the complexity of constructing and implementing advanced neural network models that have yet to become standard curricula in most business and economic universities.

Neural network models, particularly deep learning architectures, are often considered "black boxes" because they rely on complex, non-linear transformations and large numbers of parameters to make predictions. This makes it difficult to interpret the relationships captured by the model or understand how specific inputs influence outputs. In contrast, traditional time series models, such as ARIMA, offer the advantage of statistical inference, allowing researchers to test structural variables and directly evaluate the significance of model components. This transparency enables a deeper understanding of the underlying data-generating process, making these models more accessible and interpretable for decision-making.

Future work should consider incorporating external variables, such as weather data or other agricultural commodity prices, to potentially enhance the predictive power of these models. For instance, Thaker et al. (2024) utilized convolutional neural networks to analyze aerial images of wheat fields and cloud coverage, effectively forecasting crop yields and futures prices. Their approach demonstrates that integrating weather-related data improves trading performance, highlighting the potential benefits of such methodologies. Building upon this, future research could explore the integration of various weather parameters, including precipitation, temperature, and cloud cover, into LSTM models to capture their influence on agricultural commodity prices more accurately.

In addition to weather data, incorporating data from related commodity markets or leveraging sentiment analysis could further enhance forecasting accuracy. For example, An et al. (2022) developed a two-stage deep learning model that integrated text-based sentiment analysis of agricultural market reports with traditional LSTM forecasting, significantly improving soybean futures price predictions. Their work underscores the importance of considering both structured and unstructured data to capture market dynamics effectively. Such an approach could be applied to cotton futures markets by integrating sentiment indices derived from agricultural reports, news, or social media alongside fundamental data.

Additionally, developing hybrid models that combine traditional statistical methods with deep learning approaches may offer more comprehensive forecasting tools for market stakeholders. Such models could leverage the interpretability of statistical methods and the pattern recognition capabilities of deep learning, providing both accuracy and transparency in forecasts. Exploring these avenues could contribute to more robust and reliable forecasting models in the agricultural sector.

Author Contributions: Conceptualization, J.V. and J.R.; methodology, J.V.; software, J.V.; validation, J.V. and J.R.; formal analysis, J.V.; investigation, J.V. and J.R.; resources, J.V.; data curation, J.V. and J.R.; writing—original draft preparation, J.V. and J.R.; writing—review and editing, J.V.; visualization, J.V.; funding acquisition, J.V. All authors have read and agreed to the published version of the manuscript.

Funding: USDA-ERS Grant: 58-3000-2-0099.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

Abbasimehr, H., & Paki, R. (2022). Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 673–691. [CrossRef]

Adams, G. M. (2015). Marketing and economics. *Cotton*, *57*, 783–796.

Ampountolas, A. (2024). Forecasting orange juice futures: LSTM, ConvLSTM, and traditional models across trading horizons. *Journal of Risk and Financial Management*, *17*(11), 475. [CrossRef]

An, W., Wang, L., & Zeng, Y. R. (2022). Text-based soybean futures price forecasting: A two-stage deep learning approach. *Journal of Forecasting*, *42*(2), 312–330. [CrossRef]

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March 26–28). *Stock price prediction using the ARIMA model*. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (pp. 106–112), Cambridge, UK.

Bailey, D., & Richardson, J. W. (1985). Analysis of selected marketing strategies: A whole-farm simulation approach. *American Journal of Agricultural Economics*, *67*(4), 813–820. [CrossRef]

Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B. S., & Roy-Chowdhury, A. K. (2019). Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, *28*(7), 3286–3300. [CrossRef] [PubMed]

Behera, R. K., Jena, M., Rath, S. K., & Misra, S. (2021). Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Information Processing & Management*, *58*(1), 102435.

Brandt, J. A., & Bessler, D. A. (1983). Price forecasting and evaluation: An application in agriculture. *Journal of Forecasting*, *2*(3), 237–248. [CrossRef]

Bukhari, M., Yasmin, S., Naz, S., Durrani, M. Y., Javaid, M., Moon, J., & Rho, S. (2023). A smart heart disease diagnostic system using deep vanilla LSTM. *Computers, Materials & Continua*, *77*(1), 1251–1279.

Cao, W., Zhu, W., Wang, W., Demazeau, Y., & Zhang, C. (2020). A deep coupled LSTM approach for USD/CNY exchange rate forecasting. *IEEE Intelligent Systems*, *35*(2), 43–53. [CrossRef]

Darekar, A., & Reddy, A. A. (2017). Cotton price forecasting in major producing states. *Economic Affairs*, *62*(3), 373–378. [CrossRef]

Dong, Z., & Zhou, Y. (2024). A novel hybrid model for financial forecasting based on CEEMDAN-SE and ARIMA-CNN-LSTM. *Mathematics*, *12*(16), 2434. [CrossRef]

Du, X., Zhang, H., Van Nguyen, H., & Han, Z. (2017, September 24–27). *Stacked LSTM deep learning model for traffic prediction in vehicle-to-vehicle communication*. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall) (pp. 1–5), Toronto, ON, Canada.

Elsaraiti, M., Ali, G., Musbah, H., Merabet, A., & Little, T. (2021, April 7–9). *Time series analysis of electricity consumption forecasting using ARIMA model*. 2021 IEEE Green Technologies Conference (GreenTech) (pp. 259–262), Denver, CO, USA.

Gairola, G., & Dey, K. (2023). Price discovery and risk management in asset class: A bibliometric analysis and research agenda. *Applied Economics Letters*, *30*(17), 2320–2331. [CrossRef]

Geetha, A., & Nasira, G. M. (2016). Time-series modelling and forecasting: Modelling of rainfall prediction using ARIMA model. *International Journal of Society Systems Science*, *8*(4), 361–372. [CrossRef]

Gu, Y. H., Jin, D., Yin, H., Zheng, R., Piao, X., & Yoo, S. J. (2022). Forecasting agricultural commodity prices using dual input attention LSTM. *Agriculture*, *12*(2), 256. [CrossRef]

He, J.-Y., Wu, X., Cheng, Z.-Q., Yuan, Z., & Jiang, Y.-G. (2021). DB-LSTM: Densely-connected Bi-directional LSTM for human action recognition. *Neurocomputing*, *444*, 319–331. [CrossRef]

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116. [CrossRef]

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. [CrossRef]

Hoffman, L., & Meyer, L. (2018). Forecasting the US season-average farm price of upland cotton: Derivation of a futures price forecasting model. *Electr. Outlook Rep. Econ. Res. Serv.* Available online: https://ers.usda.gov/sites/default/files/_laserfiche/outlooks/90245/CWS-18I-01.pdf (accessed on 4 February 2025).

Hudson, D., & Coble, K. (1999). Harvest contract price volatility for cotton. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, *19*(6), 717–733. [CrossRef]

Isengildina-Massa, O., & MacDonald, S. (2013). Forecasting US cotton prices in a changing market. In *Advances in business and management forecasting* (Vol. 9, pp. 91–113). Emerald Group Publishing Limited.

Jaiswal, R., Jha, G. K., Kumar, R. R., & Choudhary, K. (2022). Deep long short-term memory based model for agricultural price forecasting. *Neural Computing and Applications*, *34*(6), 4661–4676. [CrossRef]

Janzen, J. P., Smith, A., & Carter, C. A. (2018). Commodity price comovement and financial speculation: The case of cotton. *American Journal of Agricultural Economics*, *100*(1), 264–285. [CrossRef]

Jiang, B., Gong, H., Qin, H., & Zhu, M. (2022). Attention-LSTM architecture combined with Bayesian hyperparameter optimization for indoor temperature prediction. *Building and Environment*, *224*, 109536. [CrossRef]

Jiang, Z., Wang, Y., Yan, J., & Zhou, T. (2021). Modeling and analysis of cotton price forecast based on bilstm. *Journal of Chinese Agricultural Mechanization*, *42*(8), 151.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, *10*(3), 215–236. [CrossRef]

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. *IEEE Access*, *6*, 1662–1669. [CrossRef]

Keef, S. P., & Zhu, H. (2009). The monday effect in US cotton prices. *Agribusiness: An International Journal*, *25*(3), 427–448. [CrossRef]

Lahboub, K., & Benali, M. (2024). Assessing the predictive power of transformers, ARIMA, and LSTM in forecasting stock prices of moroccan credit companies. *Journal of Risk and Financial Management*, *17*(7), 293. [CrossRef]

Lipartito, K. J. (1983). The New York cotton exchange and the development of the cotton futures market. *Business History Review*, *57*(1), 50–72. [CrossRef]

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, *2020*(1), 6622927. [CrossRef]

Ma, M., Liu, C., Wei, R., Liang, B., & Dai, J. (2022). Predicting machine's performance record using the stacked long short-term memory (LSTM) neural networks. *Journal of Applied Clinical Medical Physics*, *23*(3), e13558. [CrossRef] [PubMed]

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting methods and applications*. John Wiley& Sons. Inc.

Malliaris, A. G., & Ziemba, W. T. (2015). Futures markets: An overview. In *The world scientific handbook of futures markets* (pp. 3–22). Available online: https://www.worldscientific.com/doi/abs/10.1142/9789814566926_0001 (accessed on 4 February 2025).

Memarzadeh, G., & Keynia, F. (2021). Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm. *Electric Power Systems Research*, *192*, 106995. [CrossRef]

Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, *4*(2), 13. [CrossRef]

Murugesan, R., Mishra, E., & Krishnan, A. H. (2022). Forecasting agricultural commodities prices using deep learning-based models: Basic LSTM, bi-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM. *International Journal of Sustainable Agricultural Management and Informatics*, *8*(3), 242–277. [CrossRef]

Nguyen, H., Kim, Y. E., & Topal, E. (2024). Application of deep learning techniques for forecasting iron ore prices: A comparative study of long short-term memory neural network and convolutional neural network. In *Applications of artificial intelligence in mining, geotechnical and geoengineering* (pp. 149–163). Elsevier.

Ozdemir, A. C., Buluş, K., & Zor, K. (2022). Medium-to long-term nickel price forecasting using LSTM and GRU networks. *Resources Policy*, *78*, 102906. [CrossRef]

Pace, J. D., & Robinson, J. R. (2012). Marketing choices by Texas cotton growers. *Journal of Agribusiness*, *30*(2), 173–184.

Pai, P.-F., & Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, *33*(6), 497–505. [CrossRef]

Schneider, L., & Tavin, B. (2024). Seasonal volatility in agricultural markets: Modelling and empirical investigations. *Annals of Operations Research*, *334*(1), 7–58. [CrossRef]

Shivakumar, N. M., & Kotreshwar, G. (2017). "Modelling maize futures price"—An empirical analysis based on ARIMA approach. *Asian Journal of Management*, *8*(3), 734–740. [CrossRef]

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75–85. [CrossRef]

Sørensen, C. (2002). Modeling seasonality in agricultural commodity futures. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, *22*(5), 393–426. [CrossRef]

Sunny, M. A. I., Maswood, M. M. S., & Alharbi, A. G. (2020, October 24–26). *Deep learning-based stock price prediction using LSTM and bi-directional LSTM model*. 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES) (pp. 87–92), Giza, Egypt.

Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., & Smith, V. (2024). Guardrail baselines for unlearning in llms. *arXiv*, arXiv:2403.03329. Available online: https://arxiv.org/abs/2403.03329 (accessed on 4 February 2025). [CrossRef]

Tomek, W. G. (1997). Commodity futures prices as forecasts. *Applied Economic Perspectives and Policy*, *19*(1), 23–44. [CrossRef]

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2017). Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access*, *6*, 1155–1166. [CrossRef]

Vergara, O., Coble, K. H., Knight, T. O., Patrick, G. F., & Baquet, A. E. (2004). Cotton producers' choice of marketing techniques. *Agribusiness: An International Journal*, *20*(4), 465–479. [CrossRef]

Wang, D., Yue, C., Wei, S., & Lv, J. (2017). Performance analysis of four decomposition-ensemble models for one-day-ahead agricultural commodity futures price forecasting. *Algorithms*, *10*(3), 108. [CrossRef]

Wang, L., Feng, J., Sui, X., Chu, X., & Mu, W. (2020). Agricultural product price forecasting methods: Research advances and trend. *British Food Journal*, *122*(7), 2121–2138. [CrossRef]

Wang, Y., Liu, Y., Wang, M., & Liu, R. (2018, October 19–23). *LSTM model optimization on stock price forecasting*. 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (Dcabes) (pp. 173–177), Wuxi, China.

Wang, Z., Su, X., & Ding, Z. (2020). Long-term traffic prediction based on lstm encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems*, *22*(10), 6561–6571. [CrossRef]

West, J. (2012). Long-dated agricultural futures price estimates using the seasonal Nelson-Siegel model. *International Journal of Business and Management*, *7*(3), 78–93. [CrossRef]

Wijesinghe, S. (2020). Time series forecasting: Analysis of LSTM neural networks to predict exchange rates of currencies. *Instrumentation*, *7*(4), 25.

Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, *275*, 167–179. [CrossRef]

Xie, R., Isengildina-Massa, O., Dwyer, G., & Sharp, J. (2016). The impact of public and semi-public information on cotton futures market. *Applied Economics*, *48*(36), 3416–3431. [CrossRef]

Yu, Z., Liu, G., Liu, Q., & Deng, J. (2018). Spatio-temporal convolutional features with nested LSTM for facial expression recognition. *Neurocomputing*, *317*, 50–57. [CrossRef]

Zhang, G., & Hu, M. Y. (1998). Neural network forecasting of the British pound/US dollar exchange rate. *Omega*, *26*(4), 495–506. [CrossRef]

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175. [CrossRef]

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, *14*(1), 35–62. [CrossRef]

Zhang, L., Zhu, G., Mei, L., Shen, P., Shah, S. A. A., & Bennamoun, M. (2018). Attention in convolutional LSTM for gesture recognition. *Advances in Neural Information Processing Systems*, *31*, 1957–1966.

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017). Learning to monitor machine health with convolutional bi-directional LSTM networks. *Sensors*, *17*(2), 273. [CrossRef] [PubMed]

Zhou, S., Zhou, L., Mao, M., Tai, H.-M., & Wan, Y. (2019). An optimized heterogeneous structure LSTM network for electricity price forecasting. *IEEE Access*, *7*, 108161–108173. [CrossRef]