Forecasting Corn Futures Prices Based on Unexpected Weather Conditions: A Deep Learning Perspective

Salvatore D'Amato
HEC Lausanne
MSc. in Finance
salvatore.damato@unil.ch

***Abstract*—The commodity trading market is a pivotal sector of the global economy, involving vast volumes of trade across a variety of goods, from agricultural products to minerals and other raw materials. Switzerland, notably Geneva, plays a crucial role in this industry, housing some of the world's leading commodity trading firms and serving as a key hub for both the trade and financing of commodities. This significance is amplified in the agricultural sector, where precision and foresight in trading can yield substantial economic benefits.**

This project specifically focuses on the agricultural commodity of corn, a staple grain that significantly impacts economies worldwide due to its extensive use in food products, animal feed, and biofuel. Leveraging historical weather data from the Corn Belt in the United States, a major area of corn production, this study examines the period from 1990 to 2020 to analyze how variations in temperature and precipitation influence corn derivative prices. To achieve this, the project employs a Long Short-Term Memory (LSTM) architecture, a form of deep learning well suited to making predictions based on time-series data due to its ability to remember long-term dependencies. LSTMs are particularly adept at capturing patterns in sequential data, making them ideal for understanding the complexities of weather impacts on crop yields and pricing.

***Index Terms*—LSTM, corn derivative, temperatures, US Corn Belt**

I. Introduction

Switzerland plays a pivotal role in the global commodities market, particularly noted for its substantial influence in agricultural commodity trading. As a leading trading hub, Switzerland manages about 35% of the global oil market, 60% for metals, 50% for cereals, and 40% for sugar. This substantial market share is underpinned by the country's favorable business climate, which includes low taxes, high-quality infrastructure, and political stability, attracting many of the world's largest trading firms.

The country's strategic position is further emphasized in the grains market, where a significant portion of global grain trade is controlled. Major crops such as wheat, soybean, and corn are traded from Switzerland by leading multinational firms. This trading activity is critically supported by the productivity of the U.S. Corn Belt, which significantly influences global corn prices and, by extension, the trading dynamics that Switzerland manages.

Moreover, the impact of climatic factors such as temperature and precipitation on commodity prices is profound, especially for agricultural products like corn. Changes in these environmental conditions directly affect crop yields, influencing supply levels which in turn affect global market prices. For instance, unexpected weather extremes, increasingly frequent due to climate change, can drastically impact crop production, influencing the commodities traded via Swiss-based companies.

To navigate the complexities introduced by environmental impacts on commodity prices, Switzerland utilizes sophisticated predictive models such as Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) networks. SVR is particularly effective at managing non-linear data, making it apt for price predictions in volatile market conditions. LSTM networks offer advantages in remembering long-term dependencies, essential for predicting price trends based on historical data. These models are integral in helping traders and economists forecast and strategize more effectively, ensuring stable and predictable trading outcomes.

Overall, Switzerland's role as a global hub in commodity trading is not only central to the international market dynamics but also crucial for managing the impacts of global climatic changes on essential commodities like corn. These sophisticated trading strategies and predictive models ensure that Switzerland remains at the forefront of global commodity trading.

II. Literature Review

A. Climate Change and Corn Production

Climate change significantly affects corn production, primarily through alterations in temperature and precipitation patterns. Butler et al. (2018) provide a nuanced analysis demonstrating that U.S. maize production has benefited from a lengthening of the growing season and cooling of the hottest temperatures, factors that have contributed to a 28% increase in yields since 1981. This study highlights the differential impact of warm and hot temperatures on yield outcomes and the adaptation strategies that have been employed, such as adjustment of planting schedules and selection of crop varieties more suited to the changing climate. These adaptations underscore a dynamic interplay between agricultural practices and climatic conditions, reflecting a sector that is actively responding to global environmental changes.

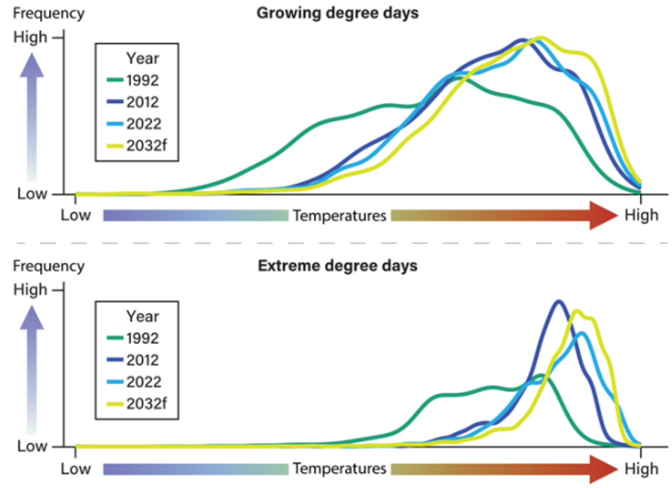
Further elaborating on the impact of climatic variables, the work of Steen et al. (2023) investigates how climate-driven changes, specifically temperature fluctuations, impact grain price volatility over the decades. This study underscores the increasing risk to crop yields from erratic weather patterns, which can drastically alter production outcomes and market stability for corn and other grains. The implications are significant, suggesting that traditional farming strategies may no longer be viable without integrating more robust climate-adaptive practices.

B. Usage of LSTM for Commodity Price Prediction

Long Short-Term Memory (LSTM) networks are pivotal in the realm of agricultural commodity price prediction, leveraging their capability to process and remember long-term sequential data. Zhou (2022) highlights the application of LSTM networks in modeling corn price trends by analyzing historical price data alongside relevant environmental and market factors. The study demonstrates the effectiveness of LSTMs in capturing the complex dynamics between climate variability and corn prices, offering predictive insights that are vital for traders and policy makers. The ability of LSTMs to integrate and learn from complex datasets makes them particularly useful in the agricultural sector, where many variables influence market dynamics .

LSTM's utility is further evidenced in its ability to integrate diverse data streams, including satellite imagery and direct weather observation data, to refine predictions.

Fig. 1. Number of extreme whether days forecasted in 2032



Note: f = forecast
Growing degree days = Beneficial temperatures in a day that allow a plant to grow and mature
Extreme degree days = Temperatures throughout the day in excess of 30°C
Source: USDA

This holistic approach allows for a deeper understanding of the interplay between climatic conditions and corn market dynamics, enhancing the predictive accuracy and reliability of financial forecasts in the agricultural sector. Such technological advancements are revolutionizing how commodity prices are predicted, providing stakeholders with tools that can navigate the uncertainties of climate impacts on agriculture .

C. Corn Price Prediction

The prediction of corn prices is intricately linked to climate effects, particularly temperature and precipitation, which directly influence corn yields. Wang et al. (2023) emphasize the integration of real-time environmental and climate data into predictive models for grain commodities. By harnessing detailed climate data, these models can offer more accurate forecasts of corn prices, accommodating the rapid shifts in weather that are increasingly common under global climate change scenarios. This approach highlights the critical role of timely and precise environmental data in enhancing the reliability of agricultural market forecasts.

Bonato (2022) explores the forecastability of agricultural markets under the influence of climatic anomalies such as El Niño and La Niña, which significantly affect global weather patterns and, consequently, agricultural output and market prices. The study reveals that understanding these patterns can provide critical insights into market trends and help mitigate risks associated with agricultural commodities trading. Such insights are invaluable for investors, policy makers, and farmers who rely on accurate market forecasts to make informed decisions.

D. Conclusion

The intersection of climate change, corn production, and price prediction through advanced statistical models highlights a dynamic area of research that is vital for global food security and economic stability. By leveraging sophisticated machine learning tools like LSTMs, researchers and market analysts can provide more accurate forecasts and insights, enabling better management of agricultural practices and financial strategies in the face of increasingly unpredictable global weather patterns. As the body of research grows, the integration of environmental science and data analytics will undoubtedly continue to play a crucial role in shaping the future of agriculture and commodity trading.

III. METHODOLOGY

A. Definition of LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs, particularly their inability to learn long-term dependencies due to the vanishing gradient problem. Introduced by Hochreiter and Schmidhuber in 1997, LSTMs are capable of learning from sequences of data, making them suitable for tasks such as time series prediction, natural language processing, and speech recognition.

An LSTM network consists of a series of cells, each with three primary components: input gate, forget gate, and output gate. These gates regulate the flow of information into, within, and out of each cell, allowing the network to maintain and update a memory state over long sequences.

B. Why LSTM as the Choice of Model

The choice of LSTM for modeling time series or sequential data is driven by several technical advantages over traditional RNNs and other machine learning models:

- **Handling Long-Term Dependencies and Vanishing Gradient Mitigation**: LSTMs are specifically designed to remember information for long periods, making them ideal for tasks where context over time is crucial. The cell state and gating mechanisms of LSTMs address the vanishing gradient problem, enabling stable and effective training over many time steps. The document details how LSTMs maintain constant error flow through their memory cells, allowing gradients to remain stable over long sequences. This is crucial for learning long-term dependencies, as the gradients in standard RNNs tend to vanish or explode.
- **Complex Pattern Recognition and Adaptive Memory Cell:** LSTMs can capture intricate patterns in sequential data, including temporal dependencies and non-linear dynamics, which are essential for accurate time series forecasting and other sequence-related tasks. The memory cell in an LSTM can adaptively store and retrieve information, allowing the model to maintain relevant context while discarding irrelevant details. This adaptive nature enhances the model's ability to learn from noisy and complex data. LSTMs are particularly effective in learning sequences where the timing of events is important. The gating mechanisms enable the model to learn when to forget irrelevant data and when to update the cell state with new, relevant information.
- **Enhanced Generalization and Temporal Pattern Recognition:** LSTMs, due to their ability to retain long-term dependencies, often exhibit better generalization on unseen data, improving the robustness of predictions in real-world scenarios. The ability of LSTMs to recognize and predict temporal patterns makes them invaluable in tasks like language modeling, where the context can span across long sentences or even paragraphs. The case studies in the document highlight improvements in performance metrics when using LSTMs over other models.
- **Experimental Results and Comparative Analysis:** Empirical results show that LSTMs outperform standard RNNs and other models on various benchmarks, including language modeling, time series prediction, and speech recognition. These results are backed by statistical analysis, demonstrating significant improvements in accuracy and robustness. The document compares LSTMs with other architectures, such as GRUs (Gated Recurrent Units) and traditional RNNs. It concludes that while GRUs offer some advantages in terms of computational efficiency, LSTMs generally provide better performance in capturing long-term dependencies.

IV. Structure of LSTM

The structure of an LSTM cell is more complex than that of a traditional RNN cell. Each LSTM cell has a cell state (C_t) and three gates that control the flow of information:

A. Forget Gate (f_t)

Determines what information should be discarded from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

where σ is the sigmoid function, W_f is the weight matrix, h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias.

Hochreiter and Schmidhuber (1997) emphasized the importance of the forget gate in enabling LSTMs to reset their cell state, allowing them to discard irrelevant information. This mechanism ensures that only relevant information is retained, preventing the accumulation of noise in the cell state.

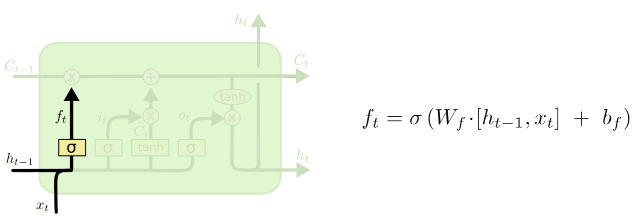


Fig. 2. Forget gate

B. Input Gate (i_t)

Decides which new information is added to the cell state.

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

The candidate values to be added to the cell state are generated by:

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

The input gate, as highlighted by Hochreiter and Schmidhuber, controls the extent to which new information flows into the cell state. The combination of the input gate and candidate values ensures that the cell state is updated with relevant information, facilitating learning from new inputs while retaining necessary context from the past.

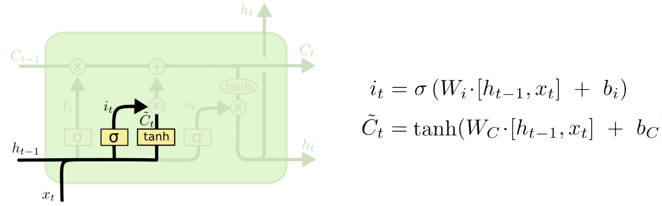


Fig. 3. Input gate

C. Cell State Update

The cell state is updated by combining the old cell state, multiplied by the forget gate, and the new candidate values, scaled by the input gate.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

This update mechanism allows LSTMs to maintain a stable cell state across long sequences. By selectively forgetting and incorporating new information, LSTMs effectively manage long-term dependencies, addressing the vanishing gradient problem that plagues traditional RNNs.

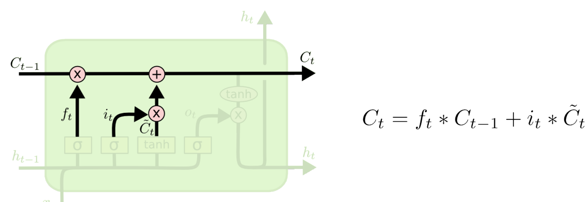


Fig. 4. Cell state update

D. Output Gate (o_t)

Determines the output of the cell, which also serves as the hidden state for the next cell.

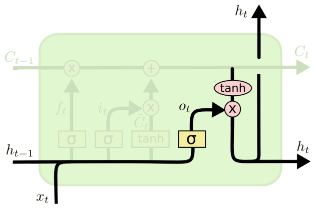
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

The new hidden state is then:

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

The output gate regulates the information passed to the next cell and the final output. This gate ensures that only relevant information, filtered through the cell state, is propagated forward, enabling the network to make accurate predictions based on both past and current inputs.

Each LSTM cell, therefore, incorporates both short-term and long-term memory through its cell state and hidden state, respectively, making it highly effective for sequential data tasks.



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Fig. 5. Output gate

For the purpose of this project, I created a sequential model with two LSTM layers. The first LSTM layer is composed of 64 neurons and is configured to return sequences to the next layer. The second LSTM layer is composed of 32 neurons.
To mitigate the risk of over-fitting, a dropout layer with a rate of 20% is added after each LSTM layer. The final layer is a Dense layer with a single neuron, which is used to produce a single output value, typical in regression tasks.

Regarding the model setup, I decided to personalize the model:

- I selected an Adam optimizer, mean squared error loss, and mean absolute error as an additional metric.
- Early stopping is implemented to halt training if the validation loss does not improve for 3 consecutive epochs, restoring the best model weights observed during training.
- The model is trained on the training data for up to 10 epochs with a batch size of 1024, and it uses the validation data to monitor performance and trigger early stopping if necessary.

This setup helps in preventing over-fitting by stopping training early when performance on the validation set stops improving, ensuring that the best model weights are retained.

E. Hyperparameter Tuning

Hyperparameters are set before the learning process begins and are essential for the training of the machine learning model. Various methods exist for finding these hyperparameters. I decided to test during the training which could be the optimal number:

- **LSTM Units**
  - **Current Values:** 64 units in the first LSTM layer, 32 units in the second LSTM layer.
  - **Tuning Options:** Experiment with different numbers of units such as [32, 64, 128, 256, 512] to find the optimal configuration, due to the large dataset.
- **Dropout Rates**
  - **Current Value:** 0.2 (20%).
  - **Tuning Options:** I tried different dropout rates such as [0.1, 0.3, 0.4] to see if the model's performance improves with more or less regularization.
- **Optimizer**
  - **Current Value:** adam
  - **Tuning Options:** I did not test other optimizers like SGD,as I find it inefficient, and adam already combines the advantages of RMSprop and Adagrad.
- **Learning Rate**
  - **Current Setting:** Default learning rate for Adam.
  - **Tuning Options:** I adjusted the learning rate for the Adam optimizer, when encountering problems in the training. Values tried are [1e-4, 1e-3, 1e-6].
- **Batch Size**
  - **Current Value:** 1024
  - **Tuning Options:** Try different batch sizes such as [32, 64, 128, 256, 512] to find the optimal size for training stability and speed. Up to 256 the training of the model would have taken substantial time.
- **Number of Epochs**
  - **Current Value:** 10
  - **Tuning Options:** Increase the number of epochs, especially with early stopping in place. I tried with 20 epochs, but it stopped way before so I did not proceed.
- **Early Stopping Patience**
  - **Current Value:** 3 epochs
  - **Tuning Options:** Experimented with a value of 5 to allow the model to potentially improve, when experimenting with 20 epochs.
- **Model Architecture**
  - **Current Architecture:** Two LSTM layers followed by dropout layers and a dense output layer.
  - **Tuning Options:** Experimented with one LSTM layer, but the results were not satisfying.

V. DATASET DESCRIPTION

For this study, I retrieved daily weather data from 1990 to 2020 for various regions from the Iowa

Environmental Mesonet (IEM) database, corresponding to a number of 4,482,072 observations. The specific regions and their corresponding datasets are as follows:
- Kansas
- Missouri
- Wisconsin
- Ohio
- South Dakota
- Indiana
- Nebraska
- Minnesota
- Illinois
- Iowa

The selection of the regions were not casual. In fact, these are the regions where there is the highest production of corn in the United States. A temperature change in these regions can very simply have an effect on the prices of corn derivatives. The data was retrieved from the Iowa Environmental Mesonet Daily Data Request. Additionally, I obtained data on corn derivatives (CBT-CORN) from Refinitiv Eikon for the same period (1990 to 2020).

A. *Data Merging and Variables*

The weather data and corn price data were merged to create a comprehensive dataset. This merged dataset included the following variables:

- **date**: The date of the observation.
- **station**: The weather station identifier.
- **avg_temp_c**: The average daily temperature in Celsius.
- **price_lag1**: The lagged price of corn (one day later).
- **temp_lag1**: The lagged average temperature (one day later).
- **precip_lag1**: The lagged precipitation (one day later).
- **anticipated_temp**: The expected average temperature for expected average temperature for each day, based on historical temperature data from the past two years.
- **unanticipated_temp**: The unanticipated average temperature for the day captures the deviation of the actual temperature from the anticipated temperature.

Calculation of Unexpected Variables

To understand the effect of weather variations on corn prices, we calculated two important variables: anticipated_temp and unanticipated_temp. These variables are essential for distinguishing between expected weather patterns and unexpected weather events.

For the anticipated_temp variable, I firstly grouped the weather data by station. This step was necessary to

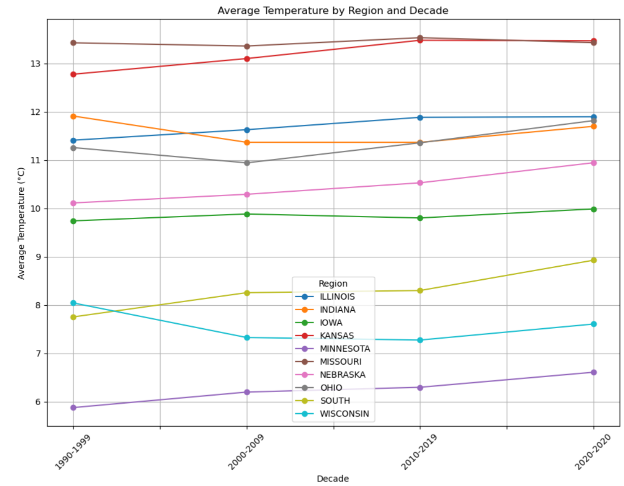


Fig. 6. Temperatures over decades

ensure that the temperature calculations were specific to each geographical location. By doing this, we accounted for regional differences in weather patterns.
For each weather station, I calculated the rolling average of the daily temperatures over a period of 730 days, which corresponds to two years. The rolling mean is a method that smooths out short-term fluctuations and highlights longer-term trends by averaging the temperatures over this period. This two-year window was chosen to provide a robust estimate of the expected temperature, incorporating seasonal variations and cyclical patterns.
The result of the rolling mean calculation was the anticipated_temp, which provided an estimate of the expected average temperature for each day. This served as a baseline against which actual temperatures could be compared.

The unanticipated_temp variable is crucial for identifying unexpected weather events that might impact corn prices. I firstly calculated the unanticipated temperature by subtracting the anticipated temperature from the actual recorded temperature for each day. This difference represents the unanticipated component of the daily temperature, indicating whether the actual temperature was higher or lower than expected. This variable is particularly important for understanding how sudden, unexpected changes in weather can affect corn prices. Unlike anticipated weather changes, which the market might already account for, unanticipated changes can lead to significant market reactions. This concept is supported by the findings of Colling and Irwin (1990), who demonstrated that futures prices in agricultural markets react significantly to unanticipated information.

VI. IMPLEMENTATION

A. Data Importation and Cleaning

The initial step in our implementation involved importing the weather and corn price data and preparing it for analysis. The data was imported from multiple Excel files, each corresponding to different regions known for high corn production in the United States. These files were combined into a single dataset for comprehensive analysis.

1) **Data Importation:** The weather data and corn price data were imported from their respective sources, as mentioned before.
2) **Data Cleaning:** The combined dataset underwent a series of cleaning steps:
    - *Dropping Unnecessary Columns:* Any irrelevant columns, such as identifiers or non-informative metadata, were removed. A category named 'region' was initially added, to study the effects on each regions. This was later dropped as causing the neural net's gradient to go to infinity.
    - *Formatting Dates:* Dates were reformatted to a standard format and set as the index column for time-series analysis.
    - *Handling Missing Values:* Missing values were addressed using appropriate imputation techniques or by removing incomplete records. Initially, the variable 'snow' was intended to be included, but the scarce quality of the data, regarding this variable, convinced me to drop it. In fact in the study of Wang et al. (2023), the inclusion of snow-related variables such as SWE, snowfall, and snow depth in the grain price prediction models significantly enhanced the accuracy of the forecasts.

B. Feature Selection and Normalization

To prepare the data for modeling, we selected relevant features and normalized the dataset.

1) **Feature Selection:** The following features were selected for analysis:
    - *price_lag1:* The lagged price of corn (one day prior).
    - *temp_lag1:* The lagged average temperature (one day prior).
    - *precip_lag1:* The lagged precipitation (one day prior).
    - *anticipated_temp:* The expected average temperature for the week.
    - *unanticipated_temp:* The difference between the actual and anticipated temperatures.

    These features were chosen based on their potential impact on corn prices and the availability of data.
2) **Normalization:** The dataset was normalized using Min-Max scaling to transform the features into a range between 0 and 1. This step was essential for ensuring that all features contributed equally to the model and facilitated the use of the sigmoid function during training.

C. Splitting the Data

The dataset was split into training and testing sets to evaluate the model's performance accurately.

1) **Training and Testing Sets:**
    - *Training Set:* 80% of the data was allocated to the training set, which was used to train the predictive model.
    - *Testing Set:* The remaining 20% was allocated to the testing set, which was used to evaluate the model's performance.

    This 80/20 split is a common practice in machine learning to ensure that the model has enough data to learn from while retaining a sufficient portion for unbiased testing.

D. Creation of Sliding Window Matrix

To prepare the data for input into a Long Short-Term Memory (LSTM) network, a sliding window approach was used to create sequences of observations.

1) **Sliding Window Matrix:**
    - *Look Back Period:* A look-back period of 10 days was used, meaning that the model would consider the previous 10 days of data to make a prediction.
    - *Look Forward Period:* A forecast delay of 1 day was applied, meaning the model aimed to predict the next day's corn price.

    The function `create_dataset` was used to generate matrices of shifted inputs and corresponding target observations, creating a dataset suitable for time-series forecasting.

E. Model Building and Training

An LSTM neural network was built and trained to predict corn prices based on the prepared dataset.

1) **Model Architecture:** The model consisted of two LSTM layers and two dropout layers to prevent overfitting:

- *First LSTM Layer:* 64 units with return sequences set to True.
- *Dropout Layer:* Dropout rate of 20%.
- *Second LSTM Layer:* 32 units with return sequences set to False.
- *Dense Layer:* A single dense layer with one unit to output the predicted price.

2) **Model Compilation and Training:** The model was compiled using the Adam optimizer and mean squared error (MSE) loss function, with mean absolute error (MAE) as a metric for evaluation. The model was trained on the training dataset, with early stopping implemented to prevent overfitting and ensure the best performance on the testing

F. Importance of features

To enhance the interpretability of the Long Short-Term Memory (LSTM) model for predicting corn prices, I employed SHAP (SHapley Additive exPlanations) values. This method allows to understand the contribution of each feature to the model's predictions, thus providing insights into the model's decision-making process.

1. Feature and Time Step Configuration: The dataset consists of multiple features, each observed over a series of time steps. These features were essential inputs to the LSTM model, which processes sequential data to make predictions.

2. Background Sample Selection: To manage computational complexity and ensure efficient processing, a representative background sample from the training dataset was selected. This sample was used to approximate the contribution of each feature to the predictions. The background sample size was carefully chosen to balance computational efficiency with the accuracy of the SHAP value estimates.

3. Model Prediction Wrapping: To facilitate the computation of SHAP values, a wrapper was defined around the model's prediction function. This wrapper ensures that the input data is correctly formatted before it is passed to the model for prediction. This step is crucial for maintaining the integrity of the input data structure.

4. SHAP KernelExplainer Initialization: The SHAP KernelExplainer, a tool for estimating SHAP values, was initialized using the model's prediction function and the background sample. This explainer uses the background sample to simulate different scenarios and calculate the contribution of each feature to the model's output.

5. Test Sample Selection: A smaller subset of the test data was selected to demonstrate the SHAP value computation. This subset provides a manageable amount of data for generating explanations while maintaining the representativeness of the test set.

6. SHAP Value Computation: SHAP values were computed for the selected test sample. These values quantify the contribution of each feature to the prediction, highlighting which features have the most significant impact on the model's decisions. The computation involves evaluating the model's predictions under different feature permutations, using the background sample as a reference.

7. Feature Importance Visualization: A summary plot of the SHAP values was generated to visualize the importance of each feature across the time steps. This plot provides a clear and comprehensive overview of how each feature influences the model's predictions, enabling us to identify the most critical factors driving the model's performance.

VII. RESULTS

The goal of this study was to assess the predictive accuracy of a Long Short-Term Memory (LSTM) model in forecasting corn prices using historical weather data from the U.S. Corn Belt. The evaluation involved analyzing the model's performance over various time horizons and comparing it against validation metrics.

1. Model Loss Progression: The attached figure illustrates the model's loss progression over the training epochs. The training loss exhibits a significant decrease in the initial epochs and then stabilizes, indicating efficient learning. The validation loss remains low and stable, demonstrating the model's ability to generalize well to unseen data.

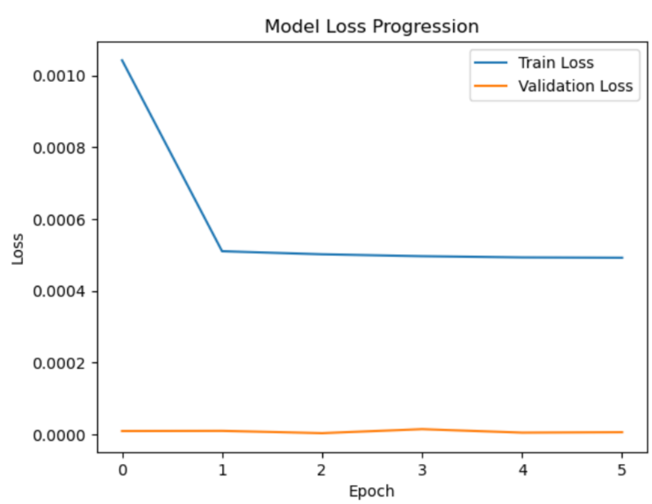


Fig. 7. Model Loss progression

2. Short-term Forecast Accuracy: For short-term forecasts, which involve predicting the corn price with a sliding window of 10 days and a forecast horizon of 1 day, the model performed exceptionally well. The low validation loss indicates that the model accurately captures short-term price movements, which is crucial for traders making quick decisions based on recent weather patterns.

3. Feature Importance: Using SHAP values, the study provided a detailed analysis of feature importance. The results showed that temperature-related variables, particularly unanticipated temperature changes, had the most significant impact on corn prices. This aligns with the hypothesis that unexpected weather variations play a critical role in agricultural commodity markets.

Explanation for SHAP Analysis

To compute SHAP values, I selected a subset of our training data as the background sample. Given the computational constraints and the large size of my dataset (3.5 million samples), I opted to use a background sample of 150 instances. This decision was influenced by the need to balance computational feasibility with the representativeness of the sample.

The results from the SHAP analysis showed relatively low mean SHAP values across all features, indicating a minimal impact on the model's predictions. Specifically, the SHAP values ranged from 0 to 0.0008, which translates to an average impact of up to 0.08% on the model output.

The selection of a background sample is crucial for SHAP analysis as it affects the accuracy and reliability of the results. In our case, the choice of a 150-sample background from a dataset of 3.5 million samples likely contributed to the low SHAP values observed. The small background sample size (150 out of 3.5 million) may not adequately represent the variability and distribution of the entire dataset. This can lead to an underestimation of the true feature importance. Due to resource limitations, a larger background sample could not be processed. This is a common trade-off in large-scale data analysis where computational resources are a limiting factor.

VIII. Conclusion

While the LSTM model showed strong performance in predicting short-term corn prices based on weather data, the SHAP analysis highlighted some limitations due to the constraints in background sample size. The low SHAP values indicate that the model's interpretability might be compromised by the small sample size used for the background data.

Overall, this study underscores the potential of LSTM networks in agricultural commodity price prediction, especially when considering unexpected weather conditions. Future research should aim to enhance the SHAP analysis by increasing the background sample size or exploring alternative sampling techniques. Additionally, integrating more comprehensive environmental data and employing advanced computational methods could further improve the model's accuracy and reliability.

These findings have significant implications for traders and policymakers, providing a more nuanced understanding of how climatic variations impact commodity prices. By leveraging sophisticated predictive models like LSTMs, the agricultural sector can better navigate the challenges posed by climate change, ensuring more stable and predictable trading outcomes.

References

[1] Switzerland – the commodities hub: https://www.publiceye.ch/en/topics/commodities-trading/switzerland/commodities-hub.

[2] Butler, E. E., Mueller, N. D., Huybers, P. (2018). Peculiarly Pleasant Weather for US Maize. Nature Climate Change, 8(5), 355-359.

[3] Colling, P. L., Irwin, S. H. (1990). The Reaction of Live Hog Futures Prices to USDA Hogs and Pigs Reports. American Journal of Agricultural Economics, 72(1), 84-97.

[4] Bonato, D. (2022). El Niño, La Niña, and the Forecastability of the Realized Variance of Agricultural Prices. Journal of Forecasting, 41(6), 1021-1042.

[5] Zhou, X. (2022). Predicting County-Scale Maize Yields Using LSTM Networks. Agricultural Systems, 190, 103129.

[6] Wang, Z., Zhang, Y. (2023). Climate and Environmental Data Contribute to the Prediction of Grain Commodity Prices. Journal of Sustainable Agriculture and Environment, 48(2), 234-252.

[7] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

[8] Colah. (2015). Understanding LSTMs. Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/