UNIVERSIDAD EAFIT

# Cotton Price Long-Term Time Series Forecasting: A look at Transformers Suitability

Carlos Enrique Salazar Escobar

Tesis de Grado

Asesor

Tomás Olarte

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA

MEDELLÍN

2024

**Abstract**

Recent years have witnessed a surge of Transformer-based models for long-term time series forecasting (LTSF). These models boast impressive results in Natural Language Processing (NLP) and Computer Vision (CV), but their effectiveness in capturing the crucial temporal order inherent in time series data remains a question. This work investigates the suitability of Transformer-based models for long-term commodity price prediction, by replicating the work presented in "Are Transformers Effective for Time Series Forecasting?" by Zeng et al. (2022). We aim to evaluate their effectiveness compared to simpler baselines and analyze their limitations in capturing long-range dependencies. By delving deeper into these limitations, this research seeks to contribute to the development of more effective forecasting models for commodity price prediction.

# 1. Introduction

## *1.1 Problem Statement*

Despite the Transformer-based model success in Natural Language Processing and Computer Vision (Li et al., 2021; He et al., 2021), their effectiveness for long-term time series forecasting (LTSF) remains an open question. Specifically, can these models capture the crucial temporal dependencies inherent in time series data, which is critical for accurate forecasting?

This research investigates this very question by applying Transformer-based models to long-term commodity price prediction using multivariate time series data. Our objectives are twofold:

- **Comparative Evaluation:** We will compare the effectiveness of Transformer models against simpler baselines that explicitly model temporal relationships.
- **Limitation Analysis:** We will analyze the limitations of Transformers in capturing long-range dependencies within the time series data.

By addressing these objectives, this research aims to contribute to the development of more effective forecasting models for commodity price prediction.

### *1.2 Considerations*

Standard Transformers (Vaswani et al., 2017) face challenges when applied to Long Short-Term Forecasting. While their core component, the self-attention mechanism, excels at capturing semantic relationships between words (tokens), so valuable in tasks like NLP, its inherent permutation-invariance can be detrimental for LTSF (Zhou et al., 2019). This property treats elements in any order as equivalent, disregarding the crucial temporal order inherent in time series data. While techniques like positional encoding partially address this, the core mechanism might still struggle to fully capture these essential relationships.

Additionally, existing comparisons of Transformer-based LTSF solutions often use baselines with known limitations, such as iterated multi-step (IMS) forecasting. IMS forecasting predicts future values by iteratively making one-step forecasts and using them as input for the next one, but this can accumulate errors over longer horizons (Granger, 1993; Schnabel et al., 2017). This can potentially inflate the perceived effectiveness of Transformer models by making them seem more impressive in comparison.

Direct Multi-Step (DMS) forecasting offers a more suitable alternative. DMS models predict all future steps in a single go, avoiding the error accumulation issue of IMS. However, DMS models also have their own drawbacks, such as potentially requiring more complex architectures or training data compared to simpler models.

Building on the work of Zeng et al. (2022), this work explores the effectiveness of Transformer-based models for long-term (over two months) commodity price prediction, using multivariate time series data. We will leverage a similar methodological approach to evaluate the true advantage of Transformers in this context, comparing their performance against simpler methods that explicitly model temporal order.

Furthermore, we delve deeper into the capabilities of Transformers for LTSF. We specifically address whether existing Transformer-based models can effectively capture long-range dependencies within cotton price time series data.

By exploring these questions, this paper aims to shed light on the strengths and limitations of Transformers for commodity price LTSF applications.

## 2. Standard Transformers

Vaswani et al. (2017) introduced Transformers, a deep learning architecture that significantly impacted various sequence modeling tasks. Their core strength lies in the self-attention mechanism, which allows the model to focus on relevant parts of the input sequence when processing each element. This capability is particularly useful for tasks like machine translation and sentiment analysis, where understanding the relationships between words is critical.

A typical Transformer architecture follows an encoder-decoder structure. The encoder processes the input sequence through multiple self-attention layers, capturing the relationships between elements. The decoder then leverages this encoded representation to generate the output sequence, one element at a time.

## 2.1 Self-*Attention Mechanism in Standard Transformers*

The self-attention mechanism lies at the core of Transformers. This mechanism empowers each element within a sequence to attend to, or focus on, other elements in the sequence. Through this process, the model learns the relative importance of each element to provide more context to the fed data. The self-attention mechanism operates in a series of well-defined steps.

First, each element in the input sequence goes through a linear transformation. This process essentially creates three new, distinct representations of the original element. These representations are captured in vectors designated as query (Q), key (K), and value (V). Each of these vectors captures a different facet of the element.

Following the initial transformation, the similarity between each query vector and all key vectors is calculated by a dot product process. This comparison results in an attention matrix, where each element reflects the relevance (attention score) of a specific element in the sequence to the current element being processed.

Those attention scores are later transformed into a probability distribution using a softmax function. This distribution plays a crucial role in determining how much weight to assign to each value vector by their relevance based on their initial attention scores.

Finally, the weighted value vectors are combined through summation, resulting in a context vector. This context vector encapsulates the most pertinent information extracted from other elements within the sequence. By analyzing this comprehensive representation, the model gains a better understanding of the element’s role within the broader context.

To further enhance the model's ability to capture diverse relationships within the sequence, multiple self-attention layers are often stacked on top of each other. Each layer utilizes a distinct set of weight matrices for the linear transformations. This approach effectively creates multiple "attention heads," each specializing in focusing on different types of relationships between elements within the sequence. This multi-headed structure significantly improves the Transformer's capacity to comprehend complex relationships and dependencies within a sequence.

## 2.2 *Other Components in Standard Transformers*

In addition to using the self-attention mechanism, Transformers include other important components to improve their effectiveness.

One such component is positional encoding. Recurrent neural networks inherently capture the order of elements in a sequence, and in Transformers, its lack of this capability is normally compensated by adding a positional encoding to the input embeddings, that incorporate temporal information. In standard Transformer, for example, a sine and cosine strategy is used to assign a unique positional encoding to each time step, but in general, different proposals for this task propend for provide relative positional information to help the model understand the order of elements in the sequence and how this order might influence their relationships.

Another aspect is the implementation of residual connections and layer normalization in both the encoder and decoder of the Transformer architecture. Residual connections create a direct path from the input to the output of each layer, helping address the vanishing gradient problem, an issue often seen in deep neural networks due its tendence of having increasingly weaker information as it propagates trough the network. This additional pathway facilitates smoother gradient flow, enabling the model to learn more complex relationships within the data. On the other hand, layer normalization ensures that the activations of each layer are normalized independently, enhancing training stability and ultimately contributing to the model's overall performance.

## 3. New Developments in Transformer-Base Models

The effectiveness of Transformers for long-term time series forecasting (LTSF) relies on their ability to process the natural order of the data. However, there are concerns regarding their suitability for this task.

One key challenge is that Transformer are inherent focused on capturing semantic correlations between words, quite important in Natural Language Processing (NLP) tasks (Vaswani et al., 2017). However, time series data typically consists of raw numerical values like stock prices or energy consumption, which lack inherent semantic meaning. This mismatch can limit the effectiveness of the self-attention mechanism in Transformers for extracting patterns from time series data (Wu et al., 2022).

Additionally, the permutation-invariant nature of the self-attention mechanism, which means it treats elements in any order as equivalent, may hinder its ability to capture the essential temporal dependencies present in time series data. Despite attempts to address this issue through positional encoding, there may still be limitations in capturing these dependencies accurately (Li et al., 2023).

Finally, the effectiveness of Transformer-based LTSF solutions can be skewed by the choice of baseline models used for comparison. Existing studies often compare Transformers against models that use iterated multi-step (IMS) forecasting. A key weakness of IMS is the inherent accumulation of errors with each prediction (Zhang et al., 2020). This can make Transformers appear more effective than they truly are, simply because they might be better at learning from the accumulated errors present in the baseline forecasts.

Research efforts have been addressing former issues through several key approaches. A compilation of principal work lines is the following:

### *3.1. Efficiency Enhancements:*

The core self-attention mechanism in standard Transformers suffers from quadratic complexity (Parmar et al., 2020**)** in relation to sequence length. This translates to high computational costs for lengthy time series data, so several techniques have been proposed to address this problem.

Sparse Transformers offer one solution. These methods introduce sparsity in the attention weights, drastically reducing the number of computations required. This can be achieved through various approaches, such as induced sparsity or locality-sensitive hashing (LSH) as implemented by Reformer (Parmar et al., 2020). Additionally, LogSparse Transformer (Shao et al., 2021) tackles this issue with a different approach, utilizing logarithmically scaled attention weights.

Low-Rank Techniques provide another avenue for improvement. These approaches focus on reducing the dimensionality of either the data itself or the attention matrices. Linformer (Tay et al., 2020) exemplifies this strategy by employing low-rank factorization of attention matrices to accelerate computations, particularly beneficial for high-dimensional data.

Finally, Informer (Zhou et al., 2021) leverages dilated causal convolutional layers. While primarily designed to capture long-range dependencies, this technique also offers the advantage of reducing the number of parameters compared to standard attention mechanisms.

### *3.2. Time Encoding Strategies:*

Effective capture of temporal information is crucial for time series forecasting (TSF). While standard Transformers use positional encoding, more advanced techniques have been explored to achieve a more effective understanding of temporal order.

One such technique is Relative Positional Encoding, employed in the Transformer-XL model (Vaswani et al., 2019). This approach focuses on the relationships between elements in the sequence rather than their absolute position. This can potentially lead to a more accurate understanding of the underlying temporal order.

Another approach is Time-to-Vector (Time2Vec) Encoding. This method learns a continuous vector representation of time steps, potentially capturing richer temporal features compared to basic positional encoding. These learned time embeddings can then be incorporated into the Transformer architecture (Bai et al., 2020).

Finally, Learned Periodicity Encoding addresses the limitations of standard positional encoding in capturing seasonal patterns in time series data. Research explores learning

these periodicities directly, as demonstrated in the Seasonal-Self-Attention model (Zhou et al., 2020).

### *3.3. Data Preprocessing (Decomposition):*

Decomposing the time series data can simplify the forecasting task for Transformer models. This technique involves using transformations like Fourier or Wavelet transforms (employed by Fedformer, Zhou et al., 2021) to explicitly extract trend and seasonal components from the data before feeding it into the Transformer. This allows the model to focus on learning relationships within the decomposed components.

Another approach involves statistical decomposition methods. Autoformer (Wu et al., 2021) demonstrates this by employing statistical methods to decompose the time series into trend, seasonal, and residual components. These components are then fed into separate branches of the Transformer architecture.

### *3.4. Capturing Long-Range Dependencies:*

The standard self-attention mechanism may struggle to capture long-range dependencies inherent in time series data. Researchers have developed techniques to address this limitation:

Informer (Haoyi et al., 2020) utilizes dilated causal convolution layers alongside self-attention. These convolutions have exponentially increasing receptive fields, allowing them to capture long-range dependencies while maintaining causality.

Autoformer (Wu et al., 2021) employs a hierarchical attention structure with local and global attention layers. Local attention layers focus on capturing short-term relationships, while global attention layers allow the model to attend to distant elements in the sequence, effectively capturing long-range dependencies.

Some approaches integrate recurrent units, such as Long Short-Term Memory (LSTM) units, into the Transformer architecture. Recurrent units excel at capturing long-term dependencies by processing information sequentially and maintaining a state across time steps. This integration can leverage the strengths of both architectures, as seen in the ConvLSTM model (Shi et al., 2015).

By incorporating these advancements, researchers are enabling Transformer models to become more efficient, capture temporal information more effectively, and handle long-range dependencies within time series data. This enhances their overall performance for TSF tasks.

# 4. Methodology

This chapter outlines the methodology employed to investigate the effectiveness of Transformer-based models for long-term cotton price forecasting compared to simpler baselines.

## *4.1 Data Acquisition and Preprocessing*

Two groups of tests were performed: the first was executed over a synthetic multivariable dataset, and the latter was executed over a multivariable cotton price-related dataset.

### *4.1.1. Synthetic Dataset*

One extremely predictable dataset was used to perform the first group of tests, composed by three timeseries polynomial equations with sinusoidal affectations in the form:

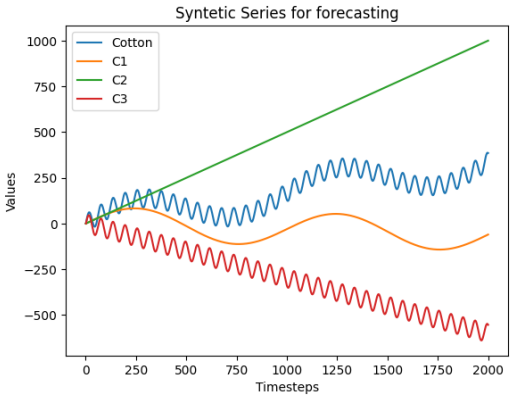
$$y = \beta_3 x^3 + \beta_2 x^2 + \beta_1 x + \beta_0 + sin\,(\Phi \pi x)\mu$$

These three series were then added to conform a fourth target series denominated 'Cotton'. Parameters of the generator series and a plot their beheaivour are shown in the following figures:

*Table 1. Parameters of the generator series*

| Serie | $\beta_3$ | $\beta_2$ | $\beta_1$ | $\beta_0$ | $\phi$ | $\mu$ |
|---|---|---|---|---|---|---|
| C1 | 0 | 0 | -0.03 | 0 | 500 | 90 |
| C2 | 0 | 0 | 0.5 | 0 | 20 | 0 |
| C3 | 0 | 0 | -0.3 | 0 | 30 | 50 |

*Figure 1. Synthetic Generated series vs. generators*




### *4.1.2 Real Cotton-Related Dataset*

The second dataset used for the experiments is a time series, multivariate collection of 34 features comprised of financial, economic, and environmental data. This data covers

a period spanning from January 3rd, 2011, to August 22nd, 2018, and allows us to capture long-term trends by including:

- **Commodity and Stock Prices:** Futures prices for several commodities like cotton, along with stock market indexes like the S&P 500. This data is provided by the Intercontinental Exchange (ICE) and retrieved through Nasdaq.
- **Economic Indexes:** This includes data on the Dollar Index, a measure of the value of the US dollar relative to a basket of foreign currencies. This data is also sourced from the ICE via Nasdaq.
- **Supply and Demand Estimates:** This data refers to estimates of future supply and demand for Cotton. The source of this data is the United States Department of Agriculture through The World Agricultural Supply and Demand Estimates (WASDE).
- **Speculative Positions:** This data likely refers to the holdings of speculators in Cotton futures contracts. The source of this data is also the Commodity Futures Trading Commission (CFTC).
- **Environmental Indexes:** This data includes the Southern Oscillation Index (SOI), an indicator of El Niño and La Niña events. This data is provided by the National Oceanic and Atmospheric Administration (NOAA).
- **Time References:** Days to nearby cotton future expirations and days from the beginning of the Cotton Season (August 1) are also provided.
- **Additional Metrics:** The data includes additional metrics such as moving averages, open interest, and unfixed purchases/sales.

The Characterization of the Time Series is like following:

- **Data Type:** The data is continuous series from 2011-01-03 to 2018-08-22.
- **Frequency:** The dataset is composed of daily data. The data will undergo preprocessing steps to ensure its suitability for modeling. In the first stage, missing values are imputed using forward fill for weekdays or periodicity deviated from daily.
- **Seasonality, Trend and Stationarity:** Augmented Dickey Fuller test was performed on the response variable, and results indicate non-stationarity. While variables like commodity and stock prices, supply and demand estimates, and some environmental indexes can exhibit trends or seasonality, it's important to consider that some Transformer models used in this work employ decomposition methods to handle these characteristics. These methods, which include techniques like Fourier transforms or statistical methods, can simplify the forecasting task.
- **Presence of outliers:** Outliers in futures data are very rare due to Stock Exchanges intraday variation restrictions. In fundamental forecasting data, outliers are also uncommon, except for unexpected adjustments or error corrections. Overall, the presence of outliers in the data is not a major concern.

In reference to the dataset selected time horizon, it's important to note that we focused on long-term forecasting, specifically two to six months. This is because cotton is a short-

cycle crop, and this timeframe is a valuable window for production activity in terms of hedging or forward sales practices.

The former array can be used to study how commodity prices are affected by economic conditions, speculative activity, and environmental factors. A table description of the features is the following:

*Table 2: Cotton-Related dataset fields description.*

| FIELD | TICKER BASE | DESCRIPTION | SOURCE | PROVIDER | DTYPE |
|---|---|---|---|---|---|
| Date | | Date | | | datetime |
| Cotton | ICE_CT1 | Cotton nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton 2nd Near | ICE_CT2 | Second nearby future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton 3rd Near | ICE_CT3 | Third nearby future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton 4th Near | ICE_CT4 | Fourth nearby future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Sugar | ICE_SB1 | Sugar nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Corn | CME_C1 | Sugar nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Soybeans | CME_S1 | Soybeans nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Wheat | CME_W1 | Wheat nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Rice | CME_RR1 | Rough Rice nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Dollar Index | ICE_DX1 | Sugar nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| S&P_500 Index | CME_SP1 | Sugar nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Brent | CME_CL1 | Brent nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Gold USD | LBMA/GOLD | Gold nearest future price close | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton OpenInt | CFTC/033661_F_L_ALL.1 | Cotton Open Interest | CFCT Commodity Futures Trading Commission | Nasdaq | float32 |
| Cotton Position Indx | _F_L_ALL_OI.7/ _F_L_ALL_( | Cotton OpenInterest Long/ Cotton OpenInterest Short | CFCT Commodity Futures Trading Commission | Nasdaq | float32 |
| days2 futexpire | | days to nearest future expiration | ICE Intercontinental Exchange | Nasdaq | float32 |
| market days | | days from Cotton Year beginning | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton 200MA | ICE_CT1 | 200 days Cotton future moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| Cotton 50MA | ICE_CT1 | 50 days Cotton future moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| Dollar Index 200MA | DXY | 200 days Dollar moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| Dollar Index 50MA | DXY | 50 days Dollar moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| S&P_500 Index 200MA | S&P 500 | 200 days S&P 500 moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| S&P_500 Index 50MA | S&P 500 | 50 days S&P 500 moving average | ICE Intercontinental Exchange | Nasdaq | float32 |
| Tot_Unfxd Sls | | Unfixed Purchases | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Tot_Unfxd Purchs | | Unfixed Sales | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Tot_Unfxd 3Near_Sls | | Three nearby futures unfixed sales | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Tot_Unfxd 3Near_Purchs | | Three nearby futures unfixed sales | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Tot_Unfx SlsVsPurch | | ICE | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Tot_3Near_Unfx SlsVsPurch | | ICE | CFCT Commodity Futures Trading Commission | CFCT | float32 |
| Value World | | Cottton Ending StocksWorld | WASDE World Agricultural Supply and Demand Es | USDA | float32 |
| weeklyExports | | ICE | Us Cotton Weekly Export Sales Report | USDA | float32 |
| accumulatedExports | | ICE | Us Cotton Weekly Export Sales Report | USDA | float32 |
| Cert_Stocks | | ICE | Us Cotton Weekly Export Sales Report | USDA | float32 |
| SOI_3m | SOI | Southern Oscillation Index (SOI) | National Centers for Environmental Information | NOAA | float32 |

## *4.2 Baseline Models*

A set of baseline models was established that explicitly model temporal relationships in the data. These models serve as a benchmark to compare the performance of Transformer-based models.

The chosen baselines include the Linear Regression, which captures the overall trend in the data and can be effective for long-term forecasting of trends. The second one is Deep Neural Linear Regression (Dlinear). A brief description is the following:

- **LTSF-Linear:** The simplest baseline model is the Long-Short Term Forecast Linear model (LTSF-Linear). It essentially performs a linear regression on the historical time series data to predict future values. The model uses a single linear layer (equivalent to a weighted sum operation) across the temporal dimension of the input data. This implies that the model assigns weights to past observations and combines them to generate a single predicted value for the future. It assumes a linear relationship and doesn't capture spatial correlations.

- **DLinear (Decomposition Linear):** DLinear builds upon the LTSF-Linear model by incorporating a decomposition step inspired by Autoformer and FEDformer. It addresses the potential presence of trends in the data using a moving average kernelI. It then applies separate linear models to each component and combines them for the final prediction. This approach improves handling of trends in the data.

### *4.3 Transformer-based Tested Models*

Transformer-based architecture models specifically designed for time series forecasting tasks were selected to address the specific challenges, including those using dilated causal convolutions for capturing long-range dependencies, learned periodicity encoding to identify seasonal patterns, and even integration with recurrent neural networks (LSTMs) to leverage their strength in long-term dependency modeling. A brief description of their architecture is like following:

- **Standard Transformer (Vaswani et al., 2017):** The standard Transformer serves as a benchmark for comparison. While powerful, its self-attention mechanism, which computes relationships between all elements in a sequence, becomes computationally expensive for long time series like cotton prices. This inefficiency restricts its suitability for the task at hand.
- **Informer (Haoyi et al., 2020):** The Informer model specifically addresses the long-range dependency challenge inherent in time series data. It incorporates dilated causal convolutions within its architecture. These convolutions allow the model to capture long-range dependencies more efficiently compared to the standard Transformer's full self-attention mechanism. Additionally, Informer utilizes learnable filters within the convolutions, enabling it to incorporate exogenous factors like seasonality or trends into the forecasting process.
- **Fedformer (Fourier & Wavelets) (Zhou et al., 2021):** Fedformer tackles the challenge of efficiently capturing periodic and trend patterns in time series data. It leverages a unique combination of Transformers with Fourier or wavelet transforms. These external transforms decompose the data by extracting the seasonal and trend components before feeding the features into the Transformer itself. This decomposition allows Fedformer to focus its attention mechanism on the remaining, more complex aspects of the data, leading to efficient and accurate forecasting, particularly for data with strong seasonal or trend patterns.

The specific list with their technical characteristics, is detailed in the following table:

*Table 3: Transformer-based Tested Models Description*

| *Model* | *Key Characteristic* | *Focus* | *Specific Techniques* |
|---|---|---|---|
| *Vanilla Transformer (Vaswani et al., 2017)* | *Standard transformer architecture* | *Benchmark, not ideal for time series due to inefficiency* | *Self-attention mechanism: Computes relationships between all elements in the sequence, leading to high computational cost for long-time series* |
| *Informer (Haoyi et al., 2020)* | *Self-attention with dilated causal convolution* | *Capturing long-range dependencies* | *Convolution with learnable filters for incorporating exogenous factors (e.g., seasonality, trends)* |
| *Fedformer (Fourier & Wavelets) (Zhou et al., 2021)* | *Combines transformers with Fourier or wavelet transforms* | *Efficiently capturing periodic and trend patterns* | *Incorporates seasonal and trend decomposition: Utilizes Fourier or wavelet transforms to extract seasonal and trend components before feeding features into the transformer* |

### *4.4. Model Selection and Hyperparameter Tuning*

The chosen models will also be subjected to hyperparameter tuning using a validation set to optimize their performance for cotton price forecasting.

For each model, an Early Stopping Optimization technique is employed to identify the hyperparameter configuration that yields the best performance on a validation set, tunning the learning rate, epochs, batch size and dropout rate. Unfortunately, due to machine restrictions, high Transformers system resources consumption forced to run the models with no more than 2 layers, which private of having more intricated relationships forecasted.

### *4.5 Evaluation Metrics*

The performance of all models is evaluated using standard metrics for time series forecasting. Squared Error (MSE), that measures the average squared errors, and the Mean Absolute Error (MAE), which measures the average absolute errors, were used.

Those metrics are non-normalized metrics which provide absolute values that represent the performance of the model in terms of the original unit of measurement of the data, but they can be difficult to compare between different models or datasets, as their scale can vary depending on the unit of measurement of the data.

To facilitate comparison of performance across different models, the percentual metric SMAPE (Symmetric Mean Absolute Percentage Error) is added as an accuracy measure based on relative errors.

All metrics were calculated on a hold-out test set not used for training or hyperparameter tuning. This ensures an unbiased assessment of the models' generalizability.

### *4.6 Data Split*

The preprocessed data is then split into three sets with a specific proportion of data allocated to each set:

- Training set (70%): Used to train the models.
- Validation set (10%): Used to fine-tune hyperparameters for each model.
- Test set (20%): Used for final evaluation of model performance, unseen by the models during training or hyperparameter tuning.

### *4.7 Software*

All the experiments were run in Python using Colab web platform and taking advantage of a PyTorch implementation of the paper replicated, provided on GitHub https://github.com/cure-lab/LTSF-Linear.

## 5. Results and Discussion

### *5.1 Results*

*Tests were performed for three types of feature type forecasting: Univariate predict Univariate (ftS), Multivariate predict Multivariate (ftM), and Multivariate predict Univariate (ftMS). Transformer models (Transformer, FEDformer, Informer) were tested in every feature type forecasting; Base Line models (Linear, DLinear) just in one feature type forecasting each. All these combinations were tested with label lengths (label_len) = 45 timesteps, sequence_lengths (seq_len) of 60, 120 and 180 timesteps, and prediction lengths of 60, 120 and 180 timesteps. MSE metric results are shown in figure.1.1 and table 1.1.*

*Figure 2. Metrics Performance Small Multiplies for Synthetic Series*

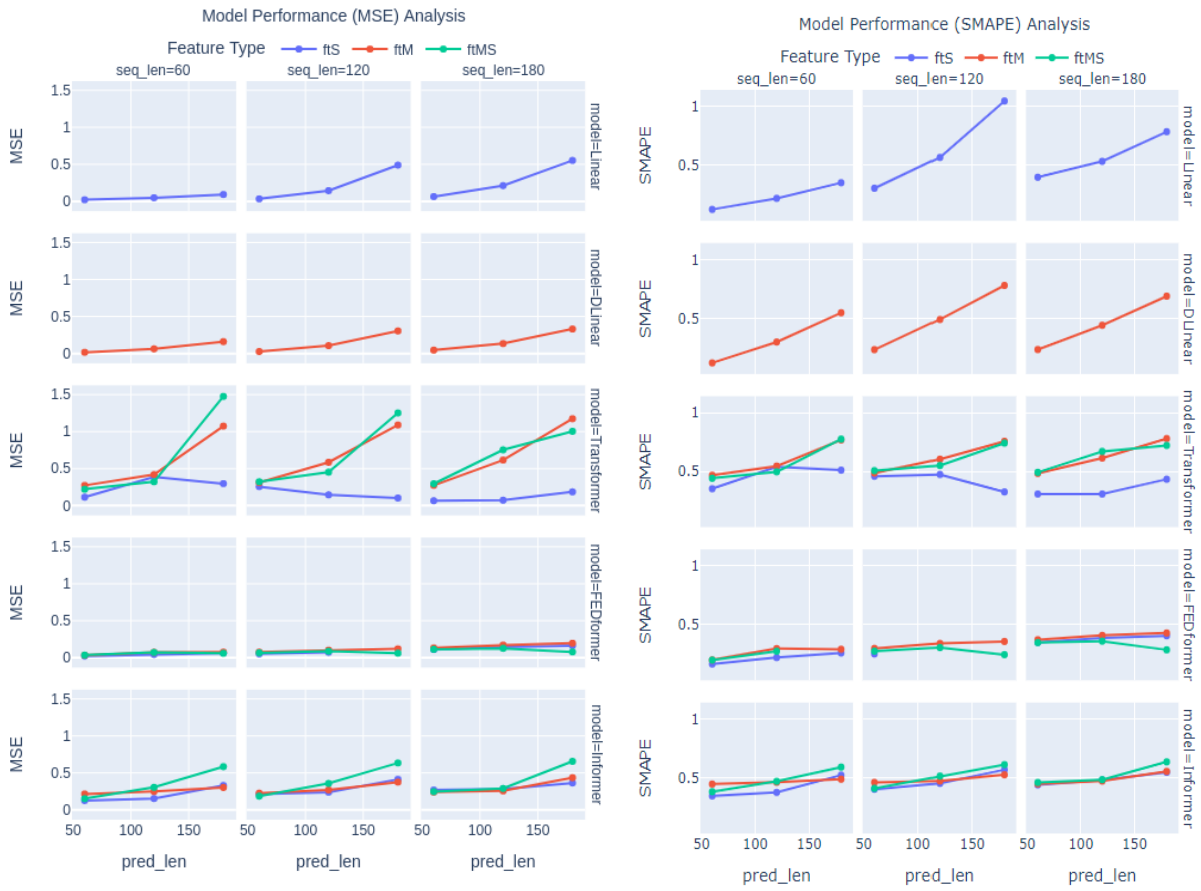



*Table 4. Metrics Performance Pivot Table for Synthetic Series*

MSE Model Performance Summary

| features | | ftS | | | ftM | | | ftMS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| pred_len | | 60 | 120 | 180 | 60 | 120 | 180 | 60 | 120 | 180 |
| seq_len | model | | | | | | | | | |
| 60 | Linear | 0.02 | 0.05 | 0.09 | | | | | | |
| | DLinear | | | | 0.02 | 0.06 | 0.16 | | | |
| | Transformer | 0.11 | 0.39 | 0.3 | 0.27 | 0.42 | 1.07 | 0.22 | 0.32 | 1.48 |
| | FEDformer | 0.02 | 0.05 | 0.06 | 0.03 | 0.07 | 0.08 | 0.04 | 0.07 | 0.06 |
| | Informer | 0.13 | 0.15 | 0.33 | 0.22 | 0.25 | 0.3 | 0.15 | 0.31 | 0.58 |
| 120 | Linear | 0.03 | 0.14 | 0.49 | | | | | | |
| | DLinear | | | | 0.03 | 0.11 | 0.3 | | | |
| | Transformer | 0.26 | 0.15 | 0.1 | 0.31 | 0.59 | 1.09 | 0.32 | 0.45 | 1.25 |
| | FEDformer | 0.05 | 0.07 | | 0.08 | 0.1 | 0.12 | 0.07 | 0.09 | 0.06 |
| | Informer | 0.22 | 0.24 | 0.41 | 0.23 | 0.27 | 0.38 | 0.19 | 0.36 | 0.63 |
| 180 | Linear | 0.06 | 0.21 | 0.55 | | | | | | |
| | DLinear | | | | 0.05 | 0.13 | 0.33 | | | |
| | Transformer | 0.07 | 0.07 | 0.19 | 0.27 | 0.62 | 1.17 | 0.3 | 0.75 | 1.0 |
| | FEDformer | 0.11 | 0.15 | 0.16 | 0.13 | 0.17 | 0.2 | 0.11 | 0.13 | 0.08 |
| | Informer | 0.27 | 0.28 | 0.36 | 0.24 | 0.26 | 0.44 | 0.25 | 0.29 | 0.66 |

SMAPE Model Performance Summary

| features | | ftS | | | ftM | | | ftMS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| pred_len | | 60 | 120 | 180 | 60 | 120 | 180 | 60 | 120 | 180 |
| seq_len | model | | | | | | | | | |
| 60 | Linear | 0.13 | 0.22 | 0.35 | | | | | | |
| | DLinear | | | | 0.12 | 0.3 | 0.55 | | | |
| | Transformer | 0.35 | 0.54 | 0.51 | 0.47 | 0.55 | 0.77 | 0.44 | 0.5 | 0.78 |
| | FEDformer | 0.17 | 0.22 | 0.26 | 0.2 | 0.3 | 0.29 | 0.2 | 0.28 | |
| | Informer | 0.35 | 0.38 | 0.52 | 0.45 | 0.46 | 0.49 | 0.38 | 0.47 | 0.59 |
| 120 | Linear | 0.31 | 0.57 | 1.05 | | | | | | |
| | DLinear | | | | 0.24 | 0.49 | 0.78 | | | |
| | Transformer | 0.46 | 0.48 | 0.33 | 0.49 | 0.61 | 0.76 | 0.51 | 0.55 | 0.74 |
| | FEDformer | 0.25 | | | 0.3 | 0.34 | 0.36 | 0.28 | 0.31 | 0.25 |
| | Informer | 0.4 | 0.45 | 0.57 | 0.46 | 0.47 | 0.53 | 0.41 | 0.52 | 0.61 |
| 180 | Linear | 0.4 | 0.53 | 0.78 | | | | | | |
| | DLinear | | | | 0.24 | 0.44 | 0.69 | | | |
| | Transformer | 0.31 | 0.31 | 0.44 | 0.48 | 0.61 | 0.78 | 0.49 | 0.67 | 0.72 |
| | FEDformer | 0.35 | 0.39 | 0.41 | 0.37 | 0.41 | 0.43 | 0.35 | 0.36 | 0.29 |
| | Informer | 0.44 | 0.48 | 0.55 | 0.45 | 0.47 | 0.56 | 0.46 | 0.49 | 0.64 |

*Figure 3 Metrics Performance Small Multiplies for Real Cotton-Related Series*

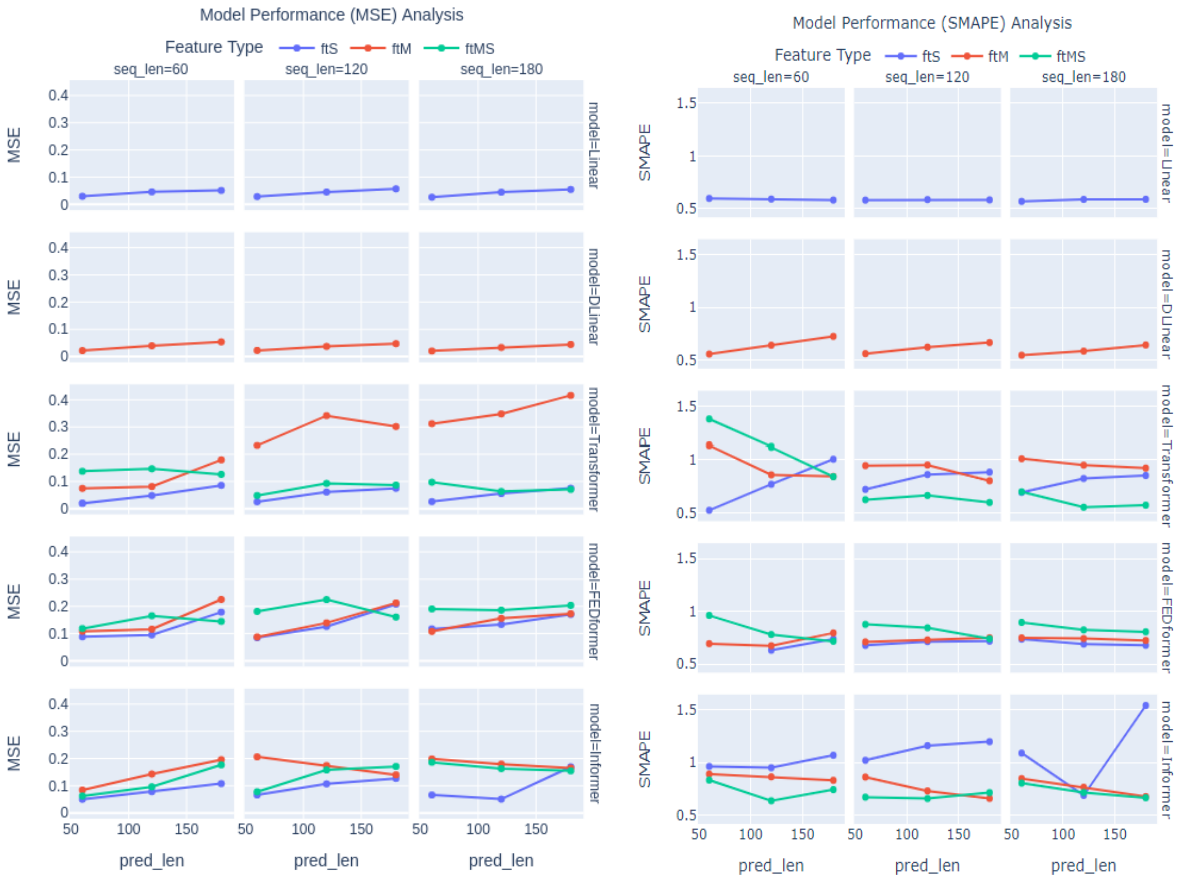



*Table 5: Metrics Performance Pivot Table for Real Cotton-Related Series*

MSE Model Performance Summary

| features | | ftS | | | ftM | | | ftMS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| pred_len | | 60 | 120 | 180 | 60 | 120 | 180 | 60 | 120 | 180 |
| seq_len | model | | | | | | | | | |
| 60 | Linear | 0.03 | 0.05 | 0.05 | | | | | | |
| | DLinear | | | | 0.02 | 0.04 | 0.05 | | | |
| | Transformer | 0.02 | 0.05 | 0.09 | 0.07 | 0.08 | 0.18 | 0.14 | 0.15 | 0.13 |
| | FEDformer | 0.09 | 0.1 | 0.18 | 0.11 | 0.12 | 0.23 | 0.12 | 0.17 | 0.14 |
| | Informer | 0.05 | 0.08 | 0.11 | 0.08 | 0.14 | 0.2 | 0.06 | 0.1 | 0.18 |
| 120 | Linear | 0.03 | 0.05 | 0.06 | | | | | | |
| | DLinear | | | | 0.02 | 0.04 | 0.05 | | | |
| | Transformer | 0.03 | 0.06 | 0.07 | 0.23 | 0.34 | 0.3 | 0.05 | 0.09 | 0.09 |
| | FEDformer | 0.09 | 0.13 | 0.21 | 0.09 | 0.14 | 0.21 | 0.18 | 0.23 | 0.16 |
| | Informer | 0.07 | 0.11 | 0.13 | 0.21 | 0.17 | 0.14 | 0.08 | 0.16 | 0.17 |
| 180 | Linear | 0.03 | 0.04 | 0.05 | | | | | | |
| | DLinear | | | | 0.02 | 0.03 | 0.04 | | | |
| | Transformer | 0.03 | 0.06 | 0.08 | 0.31 | 0.35 | 0.42 | 0.1 | 0.06 | 0.07 |
| | FEDformer | 0.12 | 0.13 | 0.17 | 0.11 | 0.16 | 0.17 | 0.19 | 0.19 | 0.2 |
| | Informer | 0.07 | 0.05 | 0.17 | 0.2 | 0.18 | 0.17 | 0.19 | 0.16 | 0.15 |

SMAPE Model Performance Summary

| features | | ftS | | | ftM | | | ftMS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| pred_len | | 60 | 120 | 180 | 60 | 120 | 180 | 60 | 120 | 180 |
| seq_len | model | | | | | | | | | |
| 60 | Linear | 0.6 | 0.59 | 0.58 | | | | | | |
| | DLinear | | | | 0.56 | 0.64 | 0.72 | | | |
| | Transformer | 0.52 | 0.77 | 1.01 | 1.13 | 0.86 | 0.84 | 1.38 | 1.12 | 0.84 |
| | FEDformer | | 0.63 | 0.74 | 0.69 | 0.67 | 0.79 | 0.96 | 0.78 | 0.72 |
| | Informer | 0.96 | 0.95 | 1.07 | 0.89 | 0.87 | 0.83 | 0.83 | 0.64 | 0.74 |
| 120 | Linear | 0.58 | 0.59 | 0.58 | | | | | | |
| | DLinear | | | | 0.56 | 0.62 | 0.67 | | | |
| | Transformer | 0.72 | 0.86 | 0.88 | 0.95 | 0.95 | 0.8 | 0.62 | 0.66 | 0.6 |
| | FEDformer | 0.68 | 0.71 | 0.72 | 0.71 | 0.73 | 0.75 | 0.88 | 0.84 | 0.74 |
| | Informer | 1.02 | 1.16 | 1.2 | 0.86 | 0.73 | 0.66 | 0.67 | 0.66 | 0.72 |
| 180 | Linear | 0.57 | 0.59 | 0.59 | | | | | | |
| | DLinear | | | | 0.55 | 0.58 | 0.64 | | | |
| | Transformer | 0.69 | 0.82 | 0.85 | 1.01 | 0.95 | 0.92 | 0.7 | 0.55 | 0.57 |
| | FEDformer | 0.74 | 0.69 | 0.68 | 0.75 | 0.74 | 0.72 | 0.89 | 0.82 | 0.81 |
| | Informer | 1.09 | 0.69 | 1.54 | 0.85 | 0.77 | 0.68 | 0.81 | 0.72 | 0.67 |

### *5.2 Results Analysis*

Key trends and observations from the provided results are:

- **Baseline vs. Transformer models performance:** In line with the findings of the replicated paper (Zeng et al., 2022), baseline models (DLinear and Linear) outperform more complex models for longer prediction horizons. It is essential to investigate the reasons behind this phenomenon, but it seems Transformer-based models had problems capturing long-range dependencies. Since the results are similar for both synthetic and real-based time series experiments, the lack of clear trend and periodicity in the real data cannot be attributed to these findings.
- **Univariate vs. Multivariate predictions:** Additionally, a particular finding emerged: univariate Transformer models achieved lower Mean Squared Errors (MSEs) compared to multivariate models. This suggests that Transformer-based models may struggle to learn or exploit the relationships between a high number of variables.
- **MSE Trends with Prediction Length**: As expected, MSE tends to increase as the prediction length becomes longer. This trend is observed across several models and feature type techniques. However, there is a notable exception with the Informer model, where MSE tends to decrease for sequence lengths of 120 and 180 when using multivariable predictions. This suggests that the Informer model could be better at handling longer prediction horizons compared to other models in certain configurations.
- **Transformer Performance with Long Sequence Lengths:** The Transformer model exhibits significantly high MSE values, particularly with long sequence lengths for the Multivariable predict Multivariable technique. This observation indicates a challenge for canonical Transformers in effectively capturing and processing information from longer sequences with the mentioned technique. Notably, this issue appears to be addressed or improved in newer models such as FEDformer and Informer, which exhibit better performance under similar conditions.

While interpreting the results, it's important to consider that the experiments used a limited number of layers in both the encoder and decoder due to hardware constraints. This limitation might have affected the models' ability to capture intricate relationships within the data. Further studies with higher hardware capacity could explore the impact of using more layers and potentially improve the results.

## 6. Conclusion

In summary, this study sheds light on the performance of Transformer-based models (Transformer, FEDformer, and Informer) compared to linear baseline models (DLinear and Linear) for LTSF applied to a multivariate commodity price time series dataset. While Transformer models are known for their complexity and ability to capture intricate temporal dependencies, our findings reveal unexpected results.

Surprisingly, in scenarios with longer prediction horizons (120 and 180 timesteps), the simpler linear baseline models, namely DLinear and Linear, outperform the more complex Transformer models. This unexpected outcome challenges conventional wisdom and highlights the importance of empirical validation and careful selection of models for specific forecasting tasks.

These results underscore the need for a nuanced approach in model selection, considering factors such as model complexity, dataset characteristics, and task requirements. Further research is warranted to delve deeper into the reasons behind the superior performance of linear baseline models in certain scenarios and to explore avenues for enhancing the performance of Transformer models in handling longer sequences.

Overall, our study contributes to a deeper understanding of the relative strengths and weaknesses of different modeling approaches in time series forecasting, providing valuable insights for practitioners and researchers in the field.

## 7. Future Research Directions

This research opens doors for further exploration in several directions:

- Investigating the impact of incorporating supplemental features that might influence cotton prices, such as government policies, news, severe weather alerts, between others.
- Exploring the potential of additional Transformer-based models (like Autoformer, Pyraformer etc.) or even hybrid models that combine Transformer-based architectures with other forecasting techniques to leverage their complementary strengths.
- Identifying novel mechanisms for self-attention and positional encoder.

By delving deeper into these areas, researchers can continue to improve the accuracy and effectiveness of long-term time series forecasting models, benefiting various stakeholders in the agricultural commodity industry and beyond.

## References


- *Bai, S., Kolen, J. T., & Ozbay, D. (2020, July). Time series forecasting with attention based lstm networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 2645-2654).*
- *Granger, C. W. J. (1993). Testing for cointegration and causality in econometric models. Greenwood Publishing Group.*
- *He, K., Chen, H., Gou, J., Xu, Y., & Liu, T. (2021). Transformers revisit: Deeper, broader and longer. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 1, pp. 15277-15287).*
- *Li, Y., Song, Z., Zhang, L., Gao, X., & Li, H. (2021). Exploring a crawler-based news reading comprehension dataset and a transformer-based model. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Vol. 1, pp. 3775-3784).*

- *Parmar, N., Vaswani, A., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I., ... & Polosukhin, I. (2020). Efficient transformers: A survey. arXiv preprint arXiv:2009.0673*
- *Schnabel, S., Rinne, J., & Hooker, B. (2017). On the appropriateness of correlation metrics for performance evaluation of time series forecasts. arXiv preprint arXiv:1704.06749.*
- *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 599-609).*
- *Zeng, A., Chen, M.-H., Zhang, L., & Xu, Q. (2022). Are Transformers Effective for Time Series Forecasting? (pp. 11121-11128). https://doi.org/10.48550/arXiv.2205.13504*
- *Zhou, F., Bao, Y., & You, Z. (2019). A comprehensive survey on time series forecasting with deep learning. Artificial Neural Networks and Machine Learning - ICANN 2019 (pp. 801-817). Springer, Cham.*
- *Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2020). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting (pp. 11106-11115). https://doi.org/10.1609/aaai.v35i12.17325*
- *Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In ArXiv: Vol. abs/2201.12740. ArXiv. https://arxiv.org/abs/2201.12740*