Comparison Between Elementary Network Structure and Prophet Model for Time-Series Forecasting: Merchandise Sales

Xu Zhang [#, *], Lixian Zhang [#], Ziang Yang [#]

School of Computer Science and Engineering, Sun Yat-sen University, Guangdong, China, 510275

* Corresponding Author Email: zhangx787@mail2.sysu.edu.cn

[#]These authors contributed equally.

Abstract. In order to maximize the profits of supermarkets, this article discusses different models in relation to time series prediction, based on previous time points, to help with setting the merchandise price. In detailed, we compared 2 deep learning methods(BiLSTM and Transformer) and Prophet model for predicting the future sales information. The overall dataset comes from certain information in the cost unit price of each category in a certain market within continuous 61 days. We evaluated these models based on the gaps bewteen their predictions and actual values, as well as the model's Interpretability. Our results indicate that the BiLSTM model could predict smoothly, while the Transformer excels in short-term prediction accuracy. The Prophet model, though less precise, offers valuable insights into seasonal patterns. We find that no single model is universally superior; instead, their efficacy varies with the nature of the sales data. A combined approach, utilizing the strengths of each model, may provide the most comprehensive forecasting strategy. This study underscores the importance of model selection in enhancing retail pricing strategies through accurate sales forecasting.

Keywords: BiLSTM, Transformer, Prophet, RMSE, MAPE.

1. Introduction

In the contemporary landscape of retail, optimizing merchandise pricing strategies is paramount for supermarkets aiming to maximize profits. There have been numerous studies in the time-series prediction, as the comprehensive study and analysis carried out by Fatoumata Dama and Christine Sinoquet in 2021[1], or the all-compassing book which is of times-series forecasting and control [2]. This study delves into the realm of time-series prediction models, specifically focusing on leveraging historical data to enhance merchandise pricing decisions. The investigation centers around the comparison of two advanced deep learning models, Bidirectional Long Short-Term Memory (BiLSTM) and Transformer, with the traditional Prophet model. The primary objective is to forecast future sales information based on a comprehensive dataset spanning 61 consecutive days, encompassing cost unit prices for various product categories in a specific market. The datasets originate from the last 2 months sales conditions at CUMCM's C problem in 2023. The innovation lies in adopting a multi-model comparison approach, thoroughly examining the performance of different models in time series prediction. Additionally, the study emphasizes the importance of the evaluation metrics MAPE and RMSE [3] in the introduction, highlighting their crucial role in revealing model prediction errors, providing interpretability, and guiding pricing decisions. Through a comprehensive consideration of these metrics, the research offers a new perspective and approach for supermarket merchandise pricing.

2. Method

2.1. Bi-directional LSTM

2.1.1 Overview

BiLSTM[3] is an expansion of LSTM[5], which records not only backward contents structure but forward contents as well. It helps with obtaining contexts from both directions. Besides, it could learn

information from farther size of a sequence of time-series information and reduce the potential vanishing gradient problem as in the RNNs. Here, the implementation uses the pytorch's inner package which contains the torch.nn.LSTM, which is largely used in many other articles.

2.1.2 Model

(1) LSTM-unit

BiLSTM could be divided into single LSTM unit, as in Fig.1. And the LSTM unit is comprised of 4 sub-unit: input gate, memory-cell, forget-gate and output-gate. And we have one hidden layer accompanied by too. At any given time, the inputs to it are of input vector x_t, previously hidden layer h_{t-1}, previously memory cell c_{t-1}, current memory cell c_t. And outputs are current hidden state h_t.

Detailed introduction of inner components of a single LSTM unit:

Here W represents the corrresponding weight in gate or cell unit, b denotes the bias, and σ signals the activation function as sigmoid and etc. The symbol . represents element-wise vectors multiplication.

InputGate:

$$I_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{1}$$

ForgetGate:

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \tag{2}$$

MemoryCell:

$$c_t = c_{t-1}.f_t + I_t.tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3}$$

OutputGate:

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{4}$$

HiddenLayer:

$$h_t = o_t\tanh(c_t) \tag{5}$$

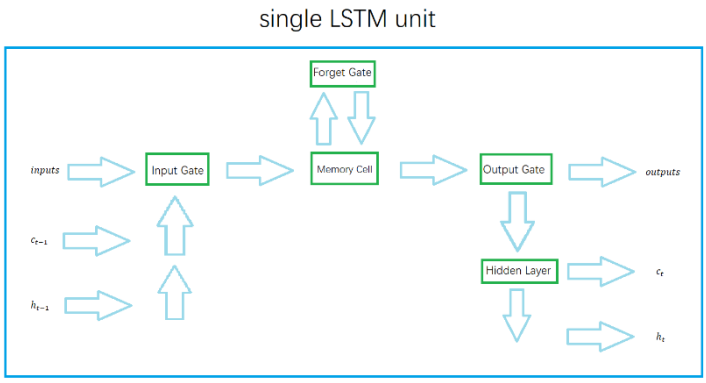


Fig. 1 LSTM unit

(2) BiLSTM structure

The bidirectional nature of BiLSTM, as shown in Fig.2, employs two separate LSTM layers – one processing the input sequence in the forward direction, and the other in the backward direction.

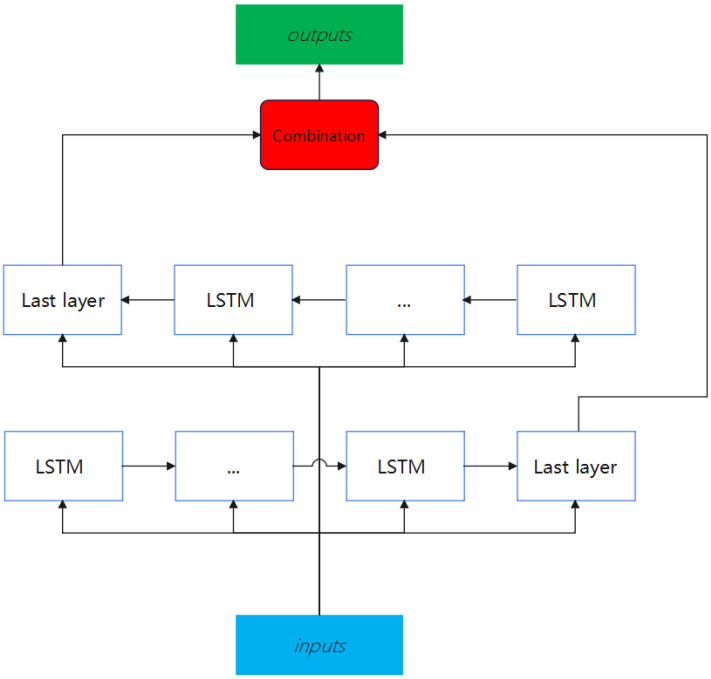


Fig. 2 BiLSTM network

2.1.3 Significance

The Bi-directional LSTM could reach a higher accuracy and lower loss compared with LSTM in prediction[6]. Apart from time-series prediction[7], it is even helpful for text classification[8] and recognition[9].

2.2. Transformer

2.2.1 Overview

The Transformer architecture, a pivotal development in the field of natural language processing, represents a departure from traditional sequential data processing methods. Introduced in the seminal work by Vaswani and colleagues in 2017[10], this model is distinguished by its utilization of an 'attention mechanism'. This unique feature enables the simultaneous processing of entire data sequences, markedly enhancing efficiency and processing speed. As a foundational element in numerous cutting-edge NLP applications, ranging from language translation to sentiment analysis, the Transformer has significantly advanced the capabilities of artificial intelligence, demonstrating a remarkable proficiency in understanding and generating at human language.

2.2.2 Model

(1) Data preprocessing

For a given data set, the time series T in the data set is divided into $t_1, t_2, \ldots\ldots, t_i$ with equal time steps according to the historical window, which contain seq items and label items, and are spliced into a time series matrix.

(2) Embedding

When we apply the Transformer model to time series prediction, there are two key embeddings to consider: feature embedding and location embedding.

(3) Feature embedding

For each feature, we use an embedding matrix, where each row represents the embedding vector of a feature. Given the index of a feature (usually an integer), its embedding vector can be represented by the following mathematical formula:

$$\text{Feature Embedding}(\text{Character\_index}) = \text{Embedding Matrix}[\text{character\_index}] \quad (6)$$

(4) Location embedding

In order to introduce the position information of the sequence in the Transformer, we use Positional Encoding. A common method of position encoding is to use trigonometric functions (usually sine and cosine functions) to generate position embeddings. The mathematical formula for positional encoding is usually expressed as:

$$\text{Positional Encoding}(pos, 2i) = sin\left(\frac{pos}{10000^{\frac{2i}{\text{Embed\_dim}}}}\right) \quad (7)$$

$$\text{Positional Encoding}(pos, 2i+1) = cos\left(\frac{pos}{10000^{\frac{2i+1}{\text{Embed\_dim}}}}\right) \quad (8)$$

Here, posit $2i, 2i+1$, and pos be the indexes of the embedding dimension in the input sequence. By adding feature embedding and position embedding, we provide the model with an input embedding that contains both feature and position information.

(5) Q, K, V matrix calculation

In the Transformer model, the QKV matrix (Query-Key-Value matrix) is obtained by linearly transforming the input matrix X with three weight matrices. As the input obtained above, assume that the dimensions of the input matrix X are (batch size, sequence length, feature dimension), where the sequence length represents the length of the time series, and the feature dimension represents the number of features at each time step.

First, we obtain the Query (Q), Key (K), and Value (V) matrices by multiplying the input matrix X and the three weight matrices (W_Q, W_K, W_V) respectively, where $(\cdot)$ represents matrix multiplication Fig.3:

$$Q = X \cdot W_Q \quad (9)$$

$$K = X \cdot W_K \quad (10)$$

$$V = X \cdot W_V \quad (11)$$

(6) Calculate the output of Self-Attention

In the self-attention mechanism, we need to calculate the attention score and use it for weighted summation to get the output. The attention score is calculated as follows:

$$\text{Attention Scores} = \frac{QK^T}{\sqrt{d_k}} \quad (12)$$

Among them, d_k is the dimension of each Query and Key (usually equal to the feature dimension), QK^T which represents the multiplication of the Q matrix and the transpose matrix of the K matrix. Then, the attention score is normalized by the Softmax function to obtain the attention weight.

Finally, use the attention weight to perform a weighted summation of the Value matrix to obtain the output of Self Attention Output:

$$\text{Self-Attention Output} = \text{Softmax}(\text{Attention Scores}) \cdot V \quad (13)$$

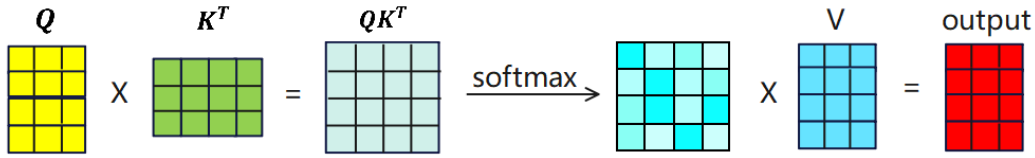


Fig. 3 softmax function

Among them, the Softmax color block matrix represents the correlation between each time step and other time steps, and each row of the output matrix represents the sum of the proportion coefficients of all time steps multiplied by the **V** matrix after Softmax normalization, thereby extracting Key information and features.

(7) Multi-Head-Attention

Multi-Head-Attention is connected based on the original single Self-Attention, which contains multiple Self-Attention layers. After obtaining the Q, K, and V matrices, the matrix needs to be split into multiple heads to obtain:

$$Q_i = head_i(Q) \tag{14}$$

$$K_i = head_i(K) \tag{15}$$

$$V_i = head_i(V) \tag{16}$$

For each header, calculate the Attention-Score value:

$$Attention_score_i = \frac{Q_i {K_i}^T}{\sqrt{d_k}} \tag{17}$$

For the obtained $Attention - Score\_i$, the calculated output is:

$$Output_i = SoftMax(Attention - Score_i) \cdot V_i \tag{18}$$

Finally, concatenate all outputs:

$$Multi - Head - Output = Concact(Output_1, Output_2, \dots\dots, Output_i) \tag{19}$$

(8) Add & Norm, Feed Forward

The function of Add is equivalent to the residual connection to prevent network degradation. Layer Normalization is a normalization technology used in neural networks. It normalizes each feature dimension of each sample so that the mean value of each feature Close to 0, variance close to 1.

$$Layer\_Norm\big(X + Multi - Head(X)\big) \tag{20}$$

$$Layer\_Norm\big(X + FeedForward(X)\big) \tag{21}$$

The Feed Forward layer corresponds to two fully connected layers. The first layer uses the Relu function for activation, and the second layer does not use the activation function.

$$max(0, W_1X + b_1)W_2 + b_2 \tag{22}$$

2.3. Prophet

2.3.1 Overview

Prophet[11] is a Facebook's open source time series prediction algorithm, which can effectively process holiday information and fit the changing trends of time series data by week, month, and year. According to the official website, Prophet has a good fitting effect on historical data with strong periodic characteristics. It can not only handle the situation where there are some outliers in the time

series, but also handle the situation of some missing values. The algorithm provides two implementations based on Python and R.

Judging from the description in the paper, the prophet algorithm is based on time series decomposition and machine learning fitting. The open source tool pyStan is used when fitting the model, so it can get the required results swiftly.

2.3.2 Model

The Prophet model uses a decomposable time series model, which is mainly composed of trend term, seasonal term (seasonality) and holiday factors (holidays).

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{23}$$

$g(t)$ is a trend function, representing non-periodic changing values, $s(t)$ represents cyclical changes (such as weekly and yearly seasonality), $h(t)$ indicates the impact of holidays that occur on a potentially irregular schedule.

(1) Trend term

The trend term is the core component of Prophet, which is used to analyze and fit non-periodic changes in time series. We chose the saturated growth model among them. The saturated growth model does not have an infinite upward trend. When the trend reaches a certain level, it will become saturated, and the saturation value changes dynamically with time.

$$g(t) = \frac{C(t)}{1+exp\left(-(k+\alpha(t)^t\delta)\cdot\left(t-(m+\alpha(t)^T\gamma)\right)\right)} \tag{24}$$

$C(t)$ is the carrying capacity, k is the growth rate, m is the offset parameter.

(2) Seasonal term

$s(t)$ represents the periodic change of the time series, which can be used to simulate various periodic change trends such as weekly, monthly, and yearly. It is expressed by Fourier series.

$$s(t) = \sum_{n=1}^{N}\left(a_n cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{P}\right)\right) \tag{25}$$

P represents the period, and the parameters can be expressed as $\beta = [\alpha_1, b_1, \ldots, a_N, b_N]^T$.The adjustment of N acts as a low-pass filter. We found that for the annual cycle and weekly cycle, N is better when selected as 10 and 3 respectively.

(3) Holiday term

Holidays and important events will have a greater impact on time series forecasting, and these effects are usually predictable. Incorporating these influencing factors into the model as prior knowledge is of major significance to improving the accuracy of the model. $h(t)$ represents non-periodic irregular holiday effects. The model implements predictions under holiday or emergency scenarios by customizing the holiday list.

$$h(t) = Z(t)k = \sum_{i=1}^{L} k_i \cdot 1_{t\in D_i} \tag{26}$$

Note that $Z(t) = [1(t \in D_1), \ldots, 1(t \in D_L)]$

(4) Errors term

$\epsilon(t)$ represents the part of noise that is not reflected in the model and assumes that the noise factor follows a normal distribution.

2.3.3 Significance

The Prophet model exhibits commendable scalability, demonstrating compatibility across various dimensions of influence and a closer alignment with real-life scenarios. The inherent periodic nature of its functions renders it capable of offering more effective guidance for market replenishment

strategies. Moreover, the model exhibits the capability to handle missing values, thereby reducing the stringent demands on data integrity. Its versatility positions it as a robust analysis method, making it a valuable asset in diverse applications.

2.4. Model accuracy criterion

2.4.1 RMSE and MAPE method

The experiment adopted the the root mean square error (RMSE) and mean absolute percentage error (MAPE) to evaluate the model results, which are shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\widehat{y_k} - y_k)^2} \tag{27}$$

$$MAPE = \frac{100\%}{n}\sum_{k=1}^{n}\left|\frac{\widehat{y_k} - y_k}{\widehat{y_k}}\right| \tag{28}$$

Where $\widehat{y_k}$ is a true value, y_k is an estimated value, n is the number of index variables.
m categories are listed, so we calculate the average of $RMSE$ and $MAPE$ given 3 models.

$$\overline{RMSE} = \frac{1}{m}\sum_{l=1}^{m} RMSE_l \tag{29}$$

$$\overline{MAPE} = \frac{1}{m}\sum_{l=1}^{m} MAPE_l \tag{30}$$

2.4.2 Significance

The MAPE method provides a measure of the average magnitudes of errors in percentage terms, which makes data more interpretable and communicable; while RMSE could evaluate the magnitude of errors by considering the squared values.

3. Experiments

3.1. Dataset proprocessing and synthesis

3.1.1 Cost Calculation

$$Cost = wholesaleprice(yuan/kg) * salesvolume(kg)/(1 - lossrate(\%) * 0.01) \tag{31}$$

3.1.2 Prophet

In this section, we detail the crucial steps involved in preparing the dataset for analysis with the Prophet model. The primary objective is to load the CSV file from the initial dataset into a DataFrame, denoted as data_df. Subsequently, the data undergoes an initial classification based on the assigned classification name. This process extracts two key elements from the dataset: the date and purchase quantity, which are identified as specified attribute columns. To ensure compatibility with the Prophet model's data format, a renaming procedure is executed, where the date column is renamed to 'ds' (timestamp), and the purchase quantity column is renamed to 'y' (target value).

3.1.3 BiLSTM

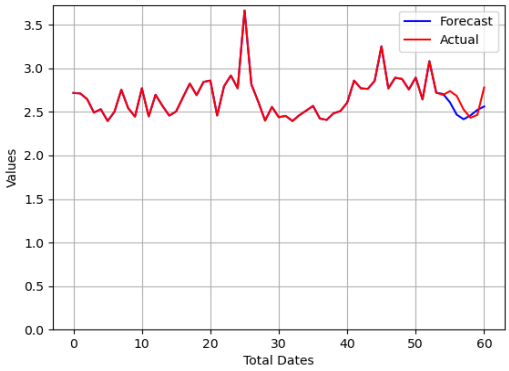
The process begins by loading a CSV file containing raw data. Specific vegetable categories and attributes are then selected to filter the relevant data. Subsequently, the date and specified attribute columns are extracted, the index is reset, and the values of the attribute columns are normalized. The data is further organized into a format suitable for LSTM model training through the creation of a sliding window dataset. This dataset encompasses both input (X_train, X_test) and output (Y_train, Y_test) components for the training and test sets. This meticulous preparation ensures that the data adheres to the input requirements of the LSTM model, setting the stage for subsequent training.

3.1.4 Transformer

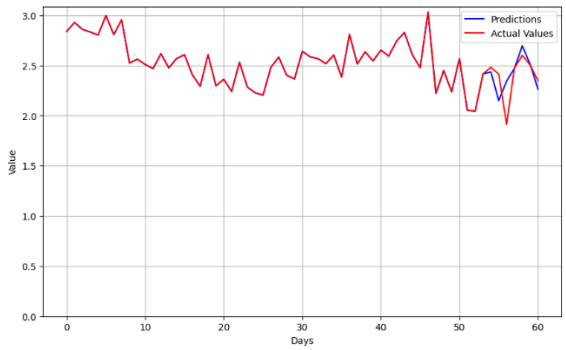
This section details the code implementation for preparing data for time series forecasting using an LSTM model. The process involves reloading the CSV file of the original data, selecting specific vegetable categories and attributes, and extracting values from the 'Costs(yuan)' column, which are stored in the variable data_values. Subsequently, input and output sequences for training and test data are created based on the specified history window size. The training data encompasses the first 54 days, while the test data covers the last seven days. Finally, the data is organized into batches, and the training data is shuffled to ensure that the model encounters a different order of samples at each time step. This systematic approach organizes the entire dataset into sequential samples based on sliding windows, with each sample incorporating sales data from a historical window size (7 days) as input and the sales of the next day as the output label. This organization is crucial for creating a dataset suitable for training time series forecasting models.

3.2. Parallel comparison between 3 models

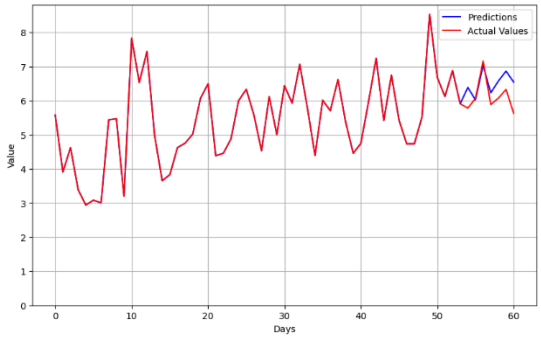
3.2.1 trend and deviation analysis from figures



(a)BiLSTM:Edible-fungi-genus



(b)Transformer:Florifolias



(c)Prophet:Aquatic-rhizomes

Fig. 4 Examples of forecasting, using different models.

In order to diversify the results, we adopted 3 models to different categories as in Fig.4.

3.2.2 Independent analysis

(1) For BiLSTM model Fig.4(a), it seems to capture the general trend very smoothly, but sometimes less responsive to rapid turns in the actual values.

(2) The Transformer model's Fig.4(b) prediction well approximates the actual values in respect of both trend and values, though a little deviation still occurs.

(3) The Prophet model Fig.4(c) could occasionally secure the changes but contains some overshooting and undershooting of peaks and troughs.

3.2.3 Comparative analysis

(1)All of the three models could generally capture the trends, but Transformer's sensibility is higher than both BiLSTM and Prophet, which might be due to its self-attention mechanism that Transformer utilizes.

(2)Transformer model fits better than BiLSTM, as from major turns and predicted values' closeness to actual values.

3.2.4 Data analysis

Both MAPE and RMSE methods are implemented for calculation, which are of last 7 days' predicted values and actual values.

Table.1. Evaluations for 6 genera based on 3 models.

(a)MAPE

Cost(yuan)	BiLSTM	Transformer	Prophet
aquatic rhizomes	14.24	13.02	7.39
mosaic	6.99	6.85	7.43
Florifolias vegetables	31.06	16.37	34.33
eggplant	11.90	13.68	6.98
pepper	6.25	5.82	6.50
edible fungus	4.26	6.12	6.96

(b)RMSE

Cost(yuan)	BiLSTM	Transformer	Prophet
aquatic rhizomes	1.05	1.023	0.518
mosaic	0.22	0.20	0.23
Florifolias vegetables	1.28	0.70	1.36
eggplant	0.36	0.38	0.21
pepper	0.19	0.16	0.21
edible fungus	0.13	0.20	0.22

We adopt the model evaluation criteria for MAPE as shown below [12][13]

Table.2. One MAPE classification standard

MAPE	Prediction classes
$\leq \mathbf{10}\%$	High-accuracy
$\mathbf{10}\% < \boldsymbol{MAPE} \leq \mathbf{20}\%$	Good
$\mathbf{20}\% < \boldsymbol{MAPE} \leq \mathbf{50}\%$	Reasonable
$> \mathbf{50}\%$	Inaccurate

As from the Table.1(a) and Table.2, mosaic, pepper and edible-fungus are of high-accuracy prediction class. Aquatic-rhizomes and eggplant are between the high-accuracy prediction class and good class. Florifolias vegetables are between the good class and Reasonable class.

When the deviation from the standard value is high, which means more noice affecting the prediction, Transformer could make better forecastings given its Good predicion for Florifolias

vegetables, since its MAPE is 16.37%; while the other two models reach the MAPE of 31.06% *and* 34.33%.

And from Table.3, we could learn that Transformer performs better than other two models, since its lower MAPE, though all of them exhibit comparable outcomes.

Table.3. Average of MAPE for Cost(yuan) in 3 genera

	BiLSTM	Transformer	Prophet
Average MAPE	12.45	10.31	11.59

Since direct calculation of average RMSE values for 6 genera could wrongly amplify distinctions in one genus for overall effects, and affect impact carried by other genera, certain adaptaions have to be made. The passage, devises a new mechanism for evaluating RMSE, as shown below.

For each genus, divide all models' RMSE by the minimal RMSE. Then, calculate the average of RMSE. This mechanism could only be used with comparison between models. If all the average values are close to each other, then the models' prediction accuracy are analogous. And the closer to the 1, the better the accuracy.

Table.4. RMSE-revised evaluation method

Cost(yuan)	BiLSTM	Transformer	Prophet
aquatic rhizomes	2.027	1.975	1
mosaic	1.1	1	1.15
Florifolias vegetables	1.83	1	1.94
eggplant	1.71	1.81	1
pepper	1.19	1	1.31
edible fungus	1	1.54	1.69
average	1.48	1.39	1.35

From Table.4, all the models' forecasting accuracy are comparable to each other. Both Prophet and Transformer seems to reach higher prediction precision, compared with BiLSTM, though Prophet is a little better than Transformer.

3.3. Further results of the Prophet model and of its special functions

3.3.1 Overview

The Prophet model stands out for its exceptional mathematical interpretability and robust capabilities in fitting time series data. Its application extends to forecasting specific attributes within various vegetable categories. This section focuses on the forecasting process and results for the cost attribute of aquatic rhizome vegetables, with the understanding that similar methodologies can be applied to other vegetable categories.

3.3.2 Model reset-up

To initiate the analysis, the initial data is trimmed to include information only from the most recent May and June. Subsequently, a search is conducted by category name to extract the cost attribute of aquatic rhizome vegetables. Given the diverse types within this category, averaging the costs of individual items is necessary for a comprehensive understanding. The data is then aggregated by date, resulting in a standardized format suitable for input into the Prophet model.

The data is fitted using the Prophet model, and the fitted model is employed for forecasting the upcoming week. Acknowledging the significant cost fluctuations among individual items within aquatic rhizome vegetables, a more lenient confidence interval is set to enhance the model's robustness. This adjustment maintains the accuracy of the central prediction while expanding the forecast interval, facilitating a clearer observation of the forecast trend and improving overall robustness.

3.3.3 Results

In Figure 5, the deep blue line depicts precise forecast values generated by the model, illustrating a discernible upward trend. The curve exhibits a distinct periodicity, aligning with a seven-day cycle, precisely a week. Considering the regular patterns in people's daily lives, weekly forecasting aligns well with practical expectations, showcasing commendable results by the model. The light blue area and black dots in the figure represent the credible forecast region within the pre-set confidence interval and the actual label points, respectively. Notably, a majority of the actual label points fall within the range of the confidence interval, reinforcing the model's reliability and accuracy in capturing the underlying trends.

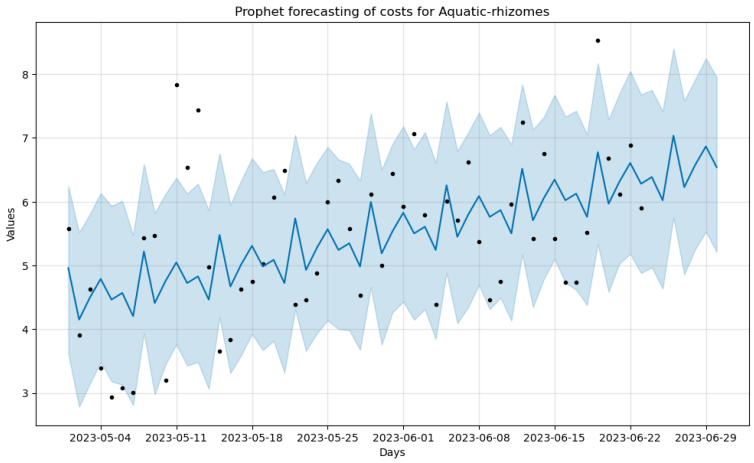
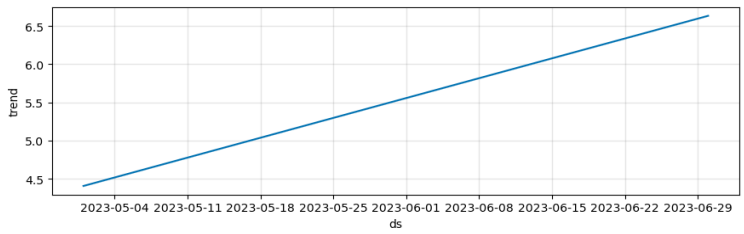
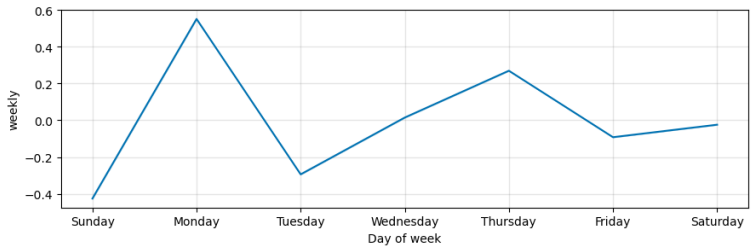


Fig.5 Prediction with upper and lower bounds

Fig.6(a) shows the overall trend, revealing that the cost of aquatic rhizome vegetables is consistently rising at a very stable rate. Fig.6(b) displays the situation within a specific period, and we have mapped it to the days of the week from Monday to Sunday based on actual dates. We notice a significant increase in the cost of aquatic rhizome vegetables on Mondays. Our conjecture, supported by surveys of nearby vegetable markets, is that Monday sees the highest volume of vegetable purchases and demand. Additionally, the quality of vegetables is generally best at this time, which reasonably explains the cost increase on Mondays.



(a)Overall trend



(b)Weekly trend

Fig.6 Prophet-trends

4. Conclusions

In this study, we compared the performance and accuracy of BiLSTM, Transformer, and Prophet models in trend prediction. Our findings revealed that, while all three models effectively captured data trends, the Transformer model demonstrated superior accuracy and noise resistance. This enhanced performance was likely due to its attention mechanism and a larger number of parameters. The BiLSTM model, although slightly less accurate, excelled in producing smoother predictions. The Prophet model, employing mathematical methods, slightly outperformed elementary BiLSTM in terms of MAPE and RMSE-revised metrics. Notably, Prophet's capacity for weekly trend prediction and boundary-setting for forecasts offered valuable insights for data analysis. These results highlighted the distinct strengths of each model, providing guidance for their application in predictive tasks within the field of Computer Science.

References

[1] Dama F, Sinoquet C. Time series analysis and modeling to forecast: A survey[J]. arXiv preprint arXiv:2104.00164, 2021.

[2] Box G E P, Jenkins G M, Reinsel G C, et al. Time series analysis: forecasting and control[M]. John Wiley & Sons, 2015.

[3] Rai N K, Saravanan D, Kumar L, et al. RMSE and MAPE analysis for short-term solar irradiance, solar energy, and load forecasting using a Recurrent Artificial Neural Network[M]//Applications of AI and IOT in Renewable Energy. Academic Press, 2022: 181-192.

[4] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE transactions on Signal Processing, 1997, 45(11): 2673-2681.

[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735_1780, 1997.

[6] Siami-Namini S, Tavakoli N, Namin A S. The performance of LSTM and BiLSTM in forecasting time series[C]//2019 IEEE International conference on big data (Big Data). IEEE, 2019: 3285-3292.

[7] Zeroual A, Harrou F, Dairi A, et al. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study[J]. Chaos, solitons & fractals, 2020, 140: 110121.

[8] Liu Z-x , Zhang D-g , Luo G-z , Lian M , Liu B . A new method of emotional analysis based on CNN–BiLSTM hybrid neural network. Cluster Comput 2020:1–13 .

[9] Graves A , Jaitly N , Mohamed A-r . Hybrid speech recognition with deep bidi-rectional LSTM. In: 2013 IEEE workshop on automatic speech recognition and understanding. IEEE; 2013. p. 273–8 .

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems (pp. 5998-6008)

[11] Taylor S J, Letham B. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45.

[12] Zeng Y-R, Zeng Y, Choi B, Wang L (2017b) Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network. Energy 127:381–396.

[13] Liu B, Song C, Wang Q, et al. Forecasting of China’s solar PV industry installed capacity and analyzing of employment effect: based on GRA-BiLSTM model[J]. Environmental Science and Pollution Research, 2022, 29(3): 4557-4573.