Comparison between elementary network structure and Prophet model for time-series forecasting: Merchandise sales

Xu Zhang 1, \*, Lixian Zhang1, Ziang Yang1

1School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong, China

\*Corresponding author: Xu Zhang(Elanander2000@gmail.com)

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

**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.

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[1][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 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.

1. Method

**2.1 Bi-directional LSTM**

1. **Overview**

BiLSTM[4] 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. Bseides, 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.

1. **Model**
   1. **LSTM-unit**

BiLSTM could be divided into single LSTM unit. 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 , previously hidden layer previously memory cell , and outputs are current hidden state , and current memory cell .

Detailed introduction of inner components of a single LSTM unit:

Here represents the corrresponding weight in gate or cell unit, denotes the bias, and signals the activation function as sigmoid and etc.

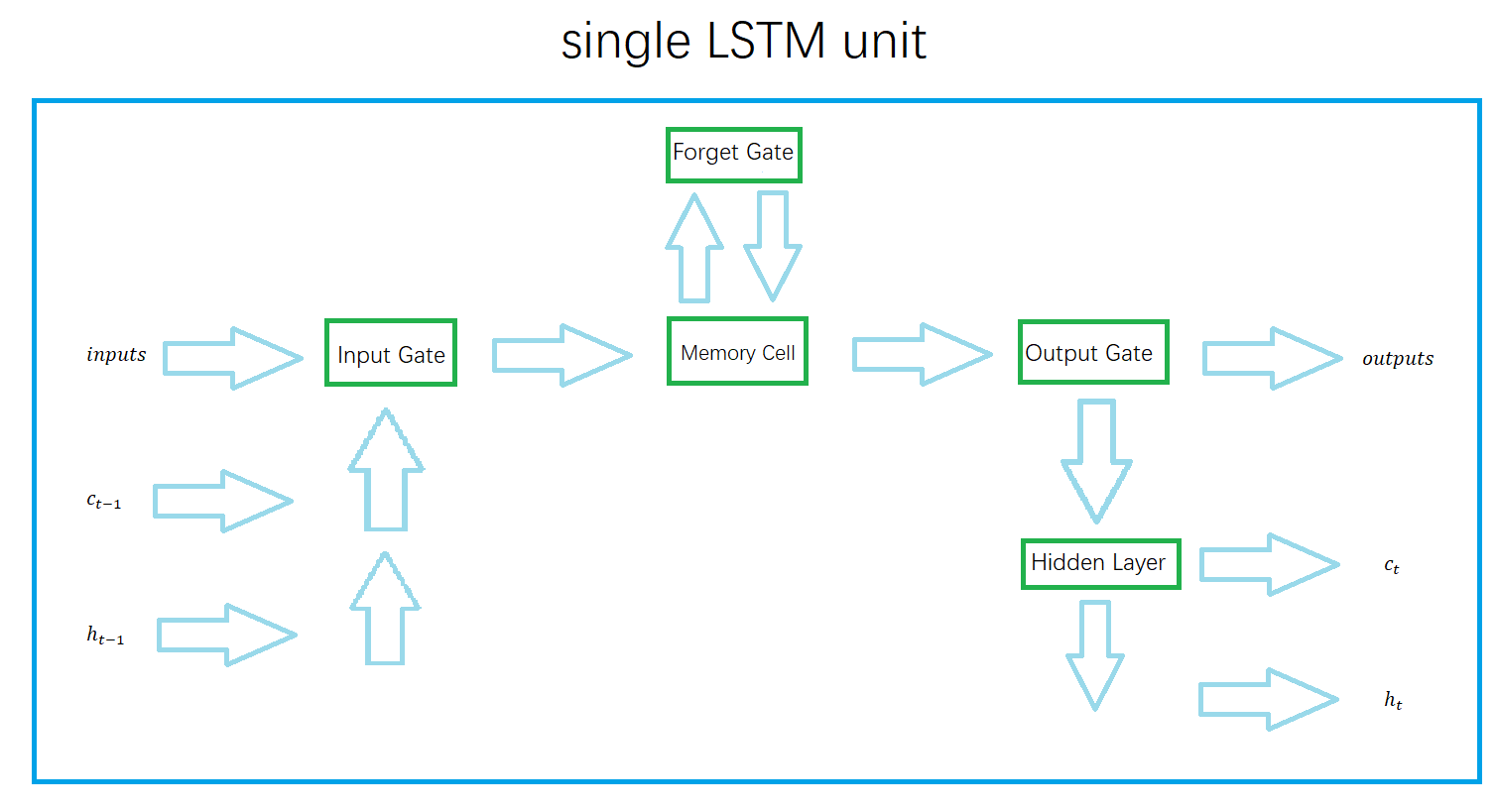
Input Gate:

Forget Gate:

Memory Cell:

Output Gate:

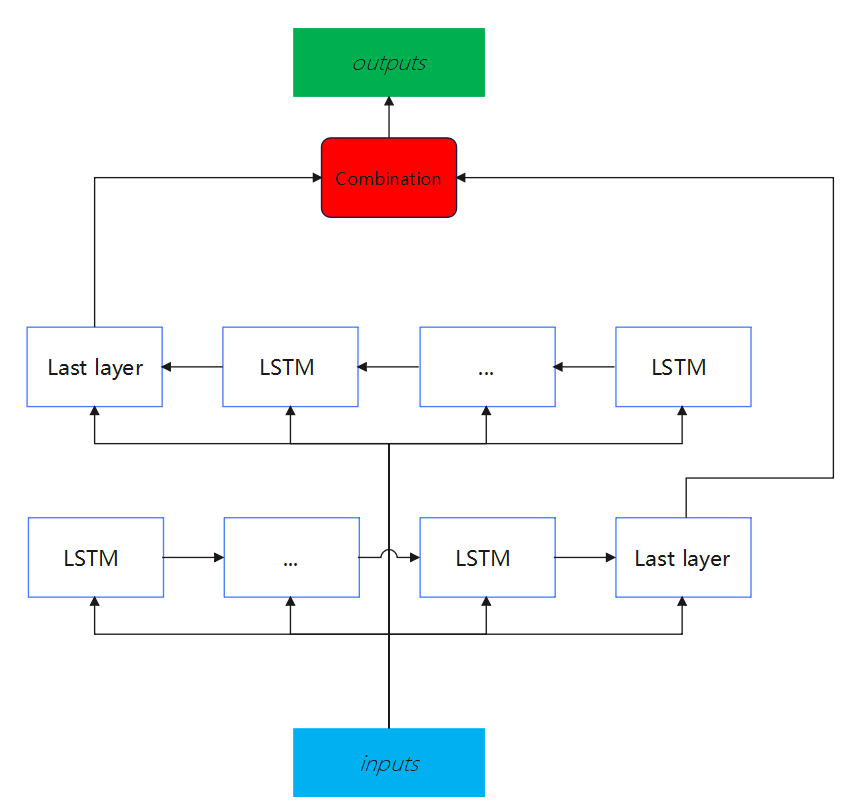
Hidden layer:



**Fig.1** LSTM unit

* 1. **BiLSTM structure**

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



**Fig.2** BiLSTM unit

* 1. **Meaning**

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].

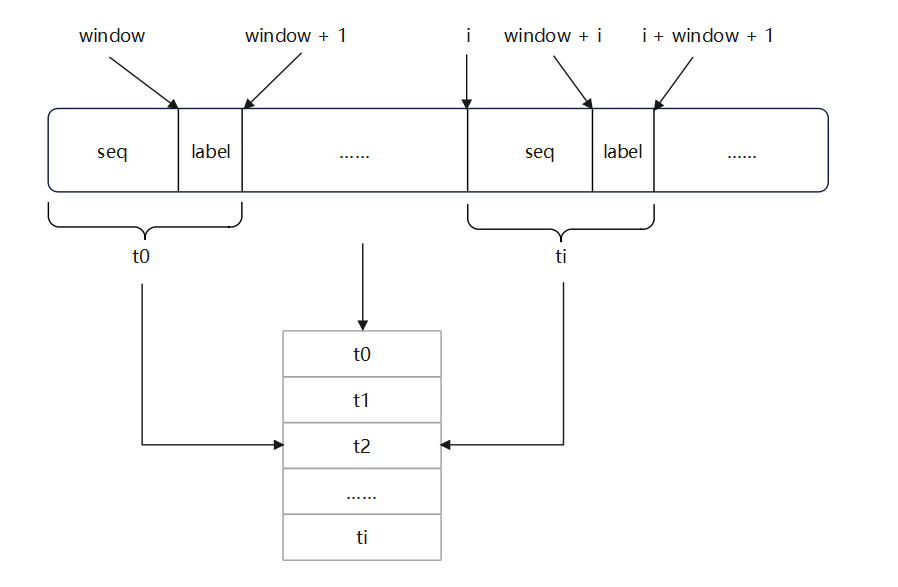
**2.2 Transformer**

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, Transformers have significantly advanced the capabilities of artificial intelligence, demonstrating a remarkable proficiency in understanding and generating human language.

1. **Model**
   1. **Data preprocessing**

For a given data set, the time series T in the data set is divided into equal time steps according to the historical window, which contains seq items and label items, and is spliced into a time series matrix. The specific diagram is as in **Fig.3**:



**Fig.3** Datasetcreation

* 1. **Embedding**

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

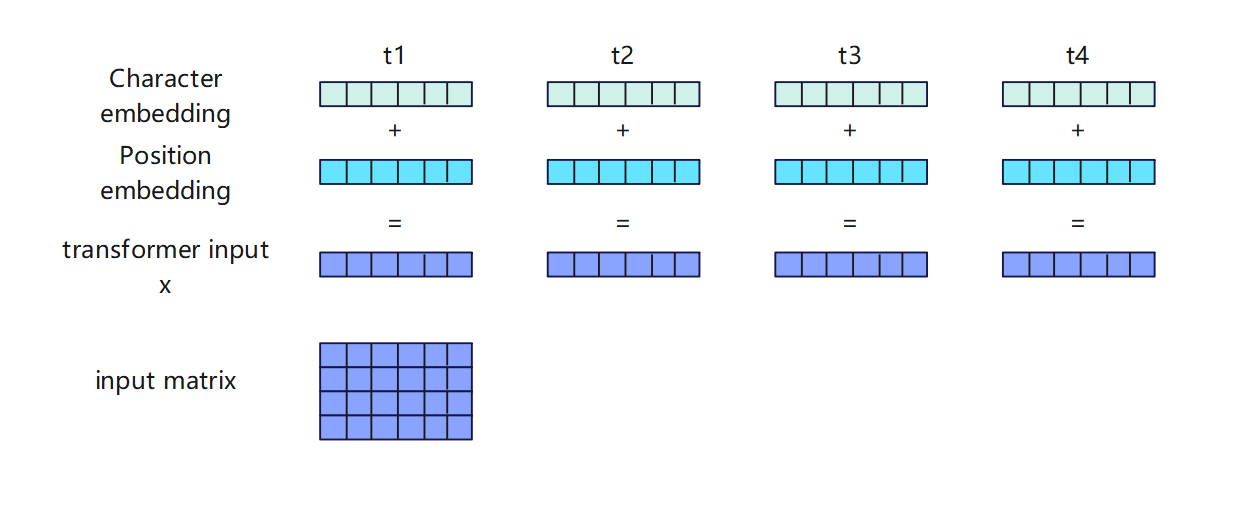
* + 1. **Feature embedding**

The purpose of feature embedding is to map each feature into a continuous vector representation. 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:

* + 1. **Location embedding**

In order to introduce the position information of the sequence in the Transformer, we use Positional Encoding. The purpose of positional encoding is to provide the model with information about the position of elements in the input sequence. 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:

Here, is the position of the element in the input sequence, and pos are the indexes of the embedding dimension. By adding feature embedding and position embedding, we provide the model with an input embedding that contains both feature and position information **Fig.4**.



**Fig.4** Embedding function

* 1. **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. These three weight matrices are used to calculate linear mappings of queries, keys, and values respectively.

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.

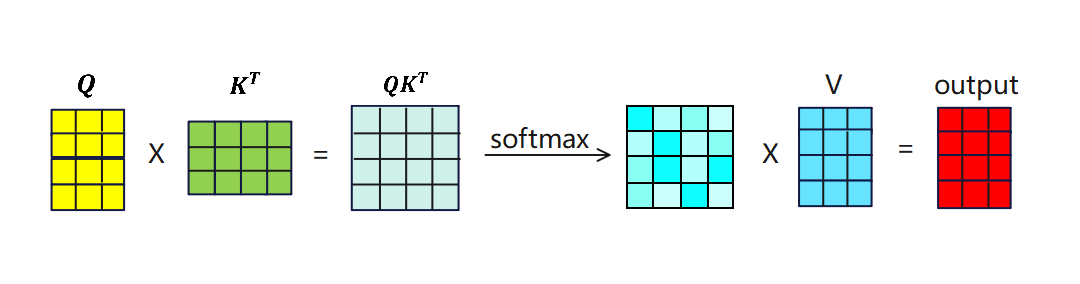
First, we obtain the Query (Q), Key (K), and Value (V) matrices by multiplying the input matrix X and the three weight matrices respectively, where represents matrix multiplication **Fig.5**:

* 1. **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:

Among them, is the dimension of each Query and Key (usually equal to the feature dimension), which represents the multiplication of the transpose matrix of the Q matrix and 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:



**Fig.5** 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.

* 1. **Multi-Head-Attention**

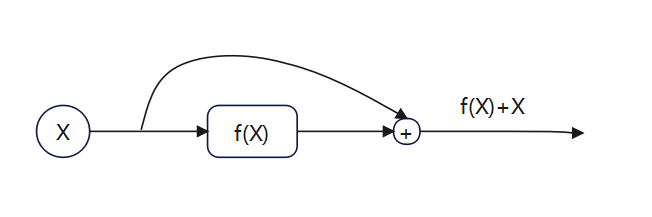
Multi-Head-Attention is connected based on the original single Self-Attention, which contains multiple Self-Attention layers. After obtaining the Q, K, and V matrices, the matrix needs to be split into multiple heads to obtain :

For each header, calculate the Attention-Score value:

For the obtained , the calculated output is:

Finally, concatenate all outputs:

* 1. **Add & Norm, Feed Forward**

****

**Fig.6** Feed forward function

The function of Add is equivalent to the residual connection to prevent network degradation, as shown in **Fig.6**. 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.

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.

* 1. **Prophet**

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, this 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 in a faster time. Predicted results.

1. **Model**

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

is a trend function, representing non-periodic changing values, represents cyclical changes (such as weekly and yearly seasonality), 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.

is the carrying capacity, is the growth rate, is the offset parameter.

* 1. **seasonal term**

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.

P represents the period, and the parameters can be expressed as.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.

* 1. **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. represents non-periodic irregular holiday effects. The model implements predictions under holiday or emergency scenarios by customizing the holiday list.

Note that

* 1. **errors term**

represents the part of noise that is not reflected in the model and assumes that the noise factor follows a normal distribution.

* 1. **Advantages**

The Prophet model has good scalability, can be compatible with many aspects of influence, and is closer to life scenes. The obvious nature of the periodic function can provide more effective guidance for the market's replenishment strategy. In addition, this model can also complete missing values, reduce the requirements for data integrity, and is more versatile, making it a relatively good analysis method.

* 1. **Model accuracy criterion**

1. 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:

Where is a true value, is an estimated value, is the number of index variables.

m categories are listed, so we calculate the average of MAPE given 3 models.

1. 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.
2. Experiments
   1. **Dataset proprocessing and synthesis**

**1）Cost Calculation**

**2）Prophet**

First, load the csv file contained in the initial data set into dataframe format for storage, which is counted as data\_df. Then the data is preliminarily classified according to the classification name, and the two elements of date and purchase quantity are extracted as specified attribute columns. In order to match the data format of the prophet, you need to rename the date column to ds and the purchase quantity column to y, which represent the timestamp and target value respectively.

**3）BiLSTM**

First load a CSV file of raw data , then select specific vegetable categories and attributes to filter the corresponding data. Next, extract the date and specified attribute columns, reset the index, and normalize the values of the attribute columns. The data is organized into a form suitable for LSTM model training by creating a sliding window data set, which includes the input (X\_train, X\_test) and output (Y\_train, Y\_test) of the training and test sets. This ensures that the data format meets the input requirements of the model and provides preparation for subsequent training.

**4）Transformer**

The above code first reloads the CSV file of the original data, then selects specific vegetable categories and attributes, extracts the value of the 'Costs(yuan)' column, and stores it in data\_values .

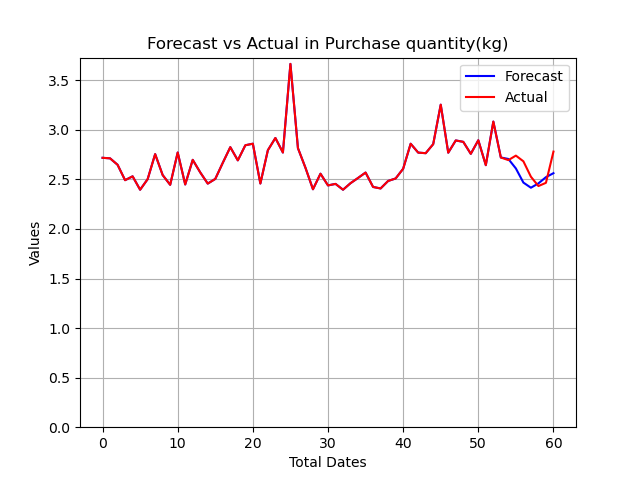
Next, input and output sequences of training and test data were created based on the size of the history window. The training data uses the data of the first 54 days, and the test data uses the data of the last seven days .

Finally, the training and test data are provided in batches , and the training data is shuffled to ensure that the model sees a different order of samples at each time step .

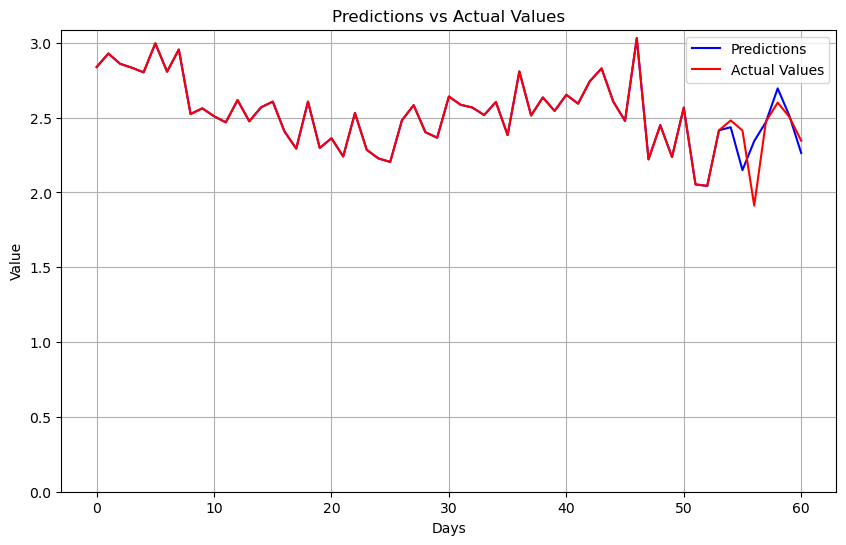
Therefore, the entire data set is organized into sequential samples based on sliding windows, where each sample includes the sales data of the historical window size (7 days) as input, and the sales of the next day as the output label. This organization helps organize the data into a form suitable for time series forecasting models.

* 1. **Parallel comparison between 3 models**

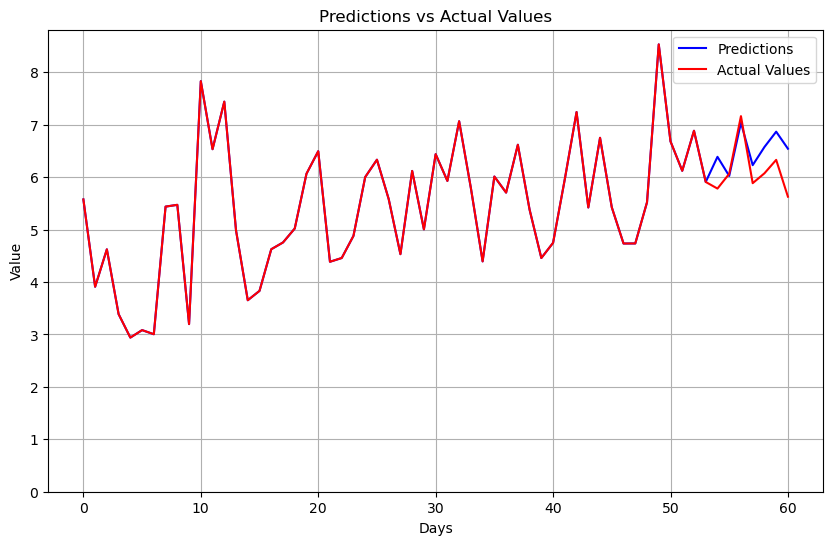
1. **trend and deviation analysis from figures**



**(a)** BiLSTM:Edible-fungi-genus



**(b)**Transformer:Florifolias



**(c)**Prophet:Aquatic-rhizomes

**Fig.7**:examples of forecasting, using different models

In order to diversify the results, we adopted 3 models to different categories.

1. **Independent analysis**
   1. For BiLSTM model(**Fig.7(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.7(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.7(c)**) could occasionally secure the changes, but contains some overshooting and undershooting of peaks and troughs.
2. **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 transformers utilize.
   2. Transformer model fits better than BiLSTM, as from major turns and predicted values’ closeness to actual values.
   3. The prediction ability of both BiLSTM and Prophet are similar.
3. **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 |  |  |  |
| mosaic |  |  |  |
| Florifolias vegetables |  |  |  |
| eggplant |  |  |  |
| pepper |  |  |  |
| edible fungus |  |  |  |

**(b)**RMSE

|  |  |  |  |
| --- | --- | --- | --- |
| Cost(yuan) | BiLSTM | Transformer | Prophet |
| aquatic rhizomes |  |  |  |
| mosaic |  |  |  |
| Florifolias vegetables |  |  |  |
| eggplant |  |  |  |
| pepper |  |  |  |
| edible fungus |  |  |  |

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

**Table.2** One MAPE classification standard

|  |  |
| --- | --- |
| MAPE | Prediction classes |
|  | High-accuracy |
|  | Good |
|  | Reasonable |
|  | 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 better make better forecastings given its Good predicion for Florifolias vegetables, since its MAPE is ; while the other two models reach the MAPE of .

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 |  |  |  |

* 1. 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 |  |  |  |
| mosaic |  |  |  |
| Florifolias vegetables |  |  |  |
| eggplant |  |  |  |
| pepper |  |  |  |
| edible fungus |  |  |  |
| average |  |  |  |

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.

* 1. **Advantage of Prophet model and its special functions**

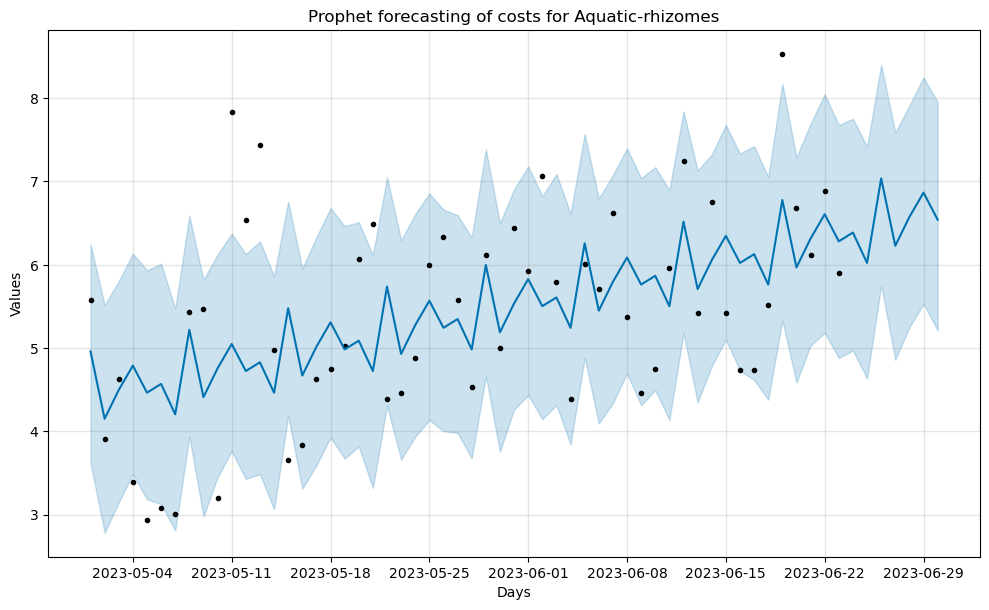
1. **Overview**

The Prophet model boasts excellent mathematical interpretability and robust capabilities in fitting time series data. We can apply it to forecast a specific attribute of a particular vegetable category. In this context, we primarily discuss the forecasting process and results for the cost attribute of aquatic rhizome vegetables. Similar methodologies can be applied to other vegetable categories.

1. **Model reset-up**

First, we trim the initial data to retain only the information from the most recent May and June. Then, we search by category name, extracting the cost attribute of aquatic rhizome vegetables. Given the numerous specific types of aquatic rhizome vegetables, studying the cost of the entire category requires averaging the cost of individual items. We then aggregate the data by date, which provides us with a standard format suitable for input into the Prophet model.

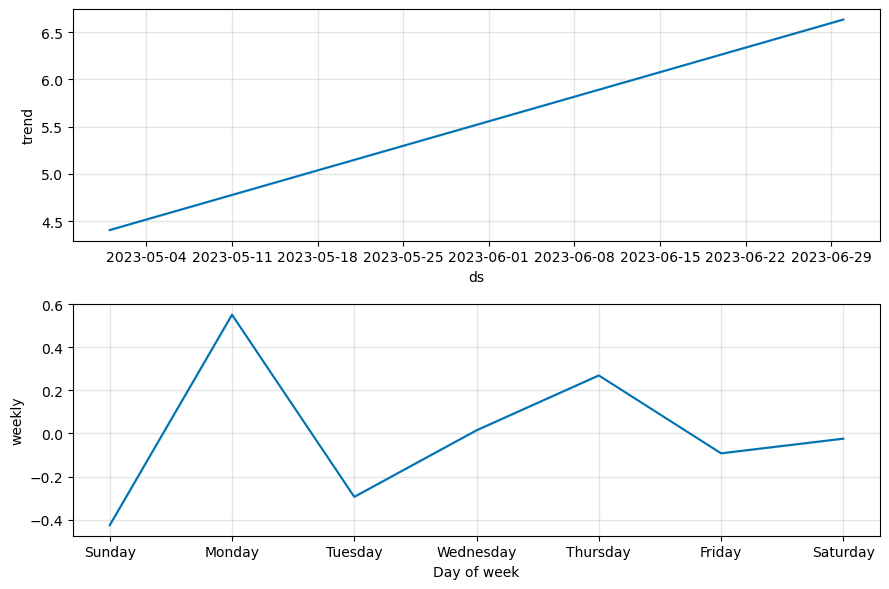
Next, we fit these data using the Prophet model and use the fitted model for the forecast of the upcoming week. Considering the significant cost fluctuations of individual items within aquatic rhizome vegetables, we set a more lenient confidence interval for the model to enhance its robustness. By doing this, while ensuring the accuracy of the central prediction remains unchanged, we expand the forecast interval. This approach allows for a clearer observation of the forecast trend and achieves better robustness.



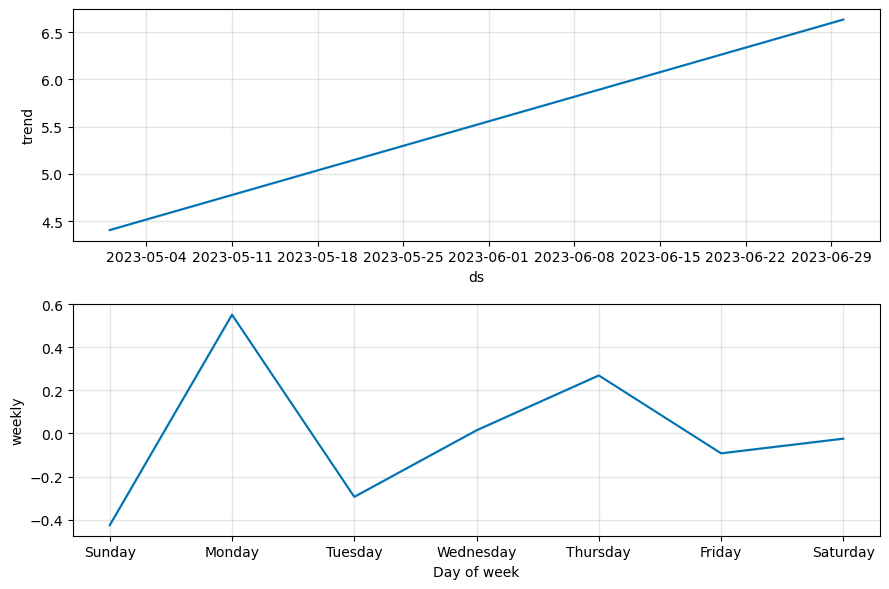
**Fig.8** prediction with upper and lower bounds

1. **Results**

In the **Fig.8**, the deep blue line represents the precise forecast values output by the model, showing a clear upward trend. On this curve, we can also observe a distinct periodicity. Upon comparing with the time axis, we find that this periodicity corresponds to seven days, exactly a week. Considering the regular patterns of people's daily lives, forecasting on a weekly basis aligns well with practical expectations. In this regard, we can say that the model has achieved commendable results.The light blue area and black dots represent the credible forecast region within the pre-set confidence interval and the actual label points, respectively. It can be observed that most of the actual label points still fall within the range of the confidence interval.



**(a)** Overall trend



**(b)** weekly trend

**Fig.9** Prophet-trends

**Fig.9(a)** shows the overall trend, revealing that the cost of aquatic rhizome vegetables is consistently rising at a very stable rate. **Fig.9(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.

1. Conclusions

This paper reported results, through which the performance and accuracy of BiLSTM, Transformer and Prophet models. The research question targeted by the experiment is focusing on comparaing results by BiLSTM model, Transformer model and Prophet model.

The results indicate that all of the three models exhibit relatively high capability with capturing the trend of data development, though the Transformer is with higher accuracy for prediction in approaching the original data and could better resist the noise affecting the prediction. This might be attributed to the focusing mechanism and number of parameters of the Transformer model. The BiLSTM could predict data more smoothly compared with the others, but a little lack of accuracy. The Prophet model which uses mathmatic method, though sometimes could not reach the desired effects as the Transformer, based on standards of MAPE and RMSE-revised method, have better performance, compared with the BiLSTM model. And, it contains mechanisms to predict data periodically in range of weekly-trend and could set the boundaries of the predictions, which provide further insights for our prediction.

References

1. Dama F, Sinoquet C. Time series analysis and modeling to forecast: A survey[J]. arXiv preprint arXiv:2104.00164, 2021.
2. Lara-Benítez P, Carranza-García M, Riquelme J C. An experimental review on deep learning architectures for time series forecasting[J]. International journal of neural systems, 2021, 31(03): 2130001.
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.
14. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International journal of forecasting, 22(4), 679-688.