Time-Series Prediction Research Based on Combined Prophet-LSTM Models

Gao Jun-Gang

March 7, 2023

Time-series prediction research based on combined Prophet-LSTM models

Gao Jungang[1]
1 Southwest Jiaotong University Chengdu,
Sichuan ,China
e-mail:gaowestlake@163.com

***Abstract*—Time series forecasting models are an important model for practical applications, and are some relatively complex class of modelling and forecasting problems due to their reliance on the sequence of events. The advent of Facebook's open source framework Prophet is a breakthrough in traditional time series forecasting models, which is simple, efficient, flexible and highly robust. However, any single prediction model for non-linear time series prediction still suffers from low accuracy and inability to extract the composite features of time series well. To this end, we propose an innovative approach to time series prediction based on the Prophet model and adding the long-short memory network model LSTM to form a combined Prophet-LSTM model. Firstly, the origin of time series forecasting is introduced, and several classical time series forecasting models are listed and their shortcomings are analyzed; secondly, the principles and advantages of the combined Prophet-LSTM model are elaborated; finally, the trend change of temperature is predicted using the Shanghai temperature data set as a sample, and the good prediction results confirm that the combined model is an excellent forecasting tool, which is worth studying and promoting application.**

Keywords-Prophet;Regressionmodels;Gradient disappearance; Gradient explosion; Overfitting; Robustness

I. INTRODUCTION

All Time series generation models have a very wide range of applications in the field of deep learning and are therefore very important research directions. It is well known that in the application of machine learning, many data exist in the form of sequences, such as sound, language, temperature, railway operation or other time-series data, etc. If these data are to be used as samples for predictive modelling, then what needs attention is the issue of the sequence of events, which is different from the prediction of regression analysis models [1].

Time series prediction has been developed to date, there are two main categories of traditional statistical algorithms and deep learning algorithms that have emerged in recent years. The main models of traditional statistical learning methods are ARIMA, ETS, GARCH, etc. These models first have to do smoothness testing on the observed series, and then convert them by differencing if they are not smooth, and then perform white noise detection, and finally build linear regression models for prediction, which The advantage of these models is that they do not require a large number of data samples to model [2], but the disadvantage is that they require a deep understanding of statistics and are difficult to use on a large scale in practical applications; deep learning algorithms mainly include LSTM, RNN, Transformer, etc. The advantage of these models is that they do not involve too much statistical analysis, and the algorithms can automatically find patterns in the data set. The disadvantage is that one must have a large sample for the trained model to generalise better, but for today's major commercial and industrial environments, this is undoubtedly available. In fact, the most successful sequence prediction models are currently self-attentive models, such as the Transformer model.

However, the above single forecasting model is still unable to meet the accuracy and generalisation required for forecasting in today's complex time-series forecasting environment. Therefore, this paper proposes to explore and improve this problem by using a combined Prophet-LSTM forecasting model to model and forecast historical temperature data in Shanghai as an example, and finally evaluate the forecasting effect to verify the accuracy and feasibility of the model.

II. RELATED WORK

A. The Prophet forecasting model

While traditional time series forecasting is based on statistical principles that transform the forecasting problem into a regression problem to deal with, the Prophet model offers an alternative solution that transforms the forecasting problem into a fitting problem. The solution first appeared in the Facebook Data Science team's paper Forecasting at scale in 2017 and was developed to be implemented in Python and R. It requires only a simple configuration of parameters to achieve highly accurate time series forecasting [3].

The Prophet model is a self-additive regression model consisting of four components.

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \qquad (1)$$

Where g(t) is the trend function, which represents the trend of the time series over the non-period; s(t) represents the periodic variation, called the period term or seasonal term, which is generally measured in weeks or years; h(t) represents the holiday term, which represents the effect of a chance day or days such as a holiday; and Ɛ represents the error term or the residual term, which represents the effect of the error not taken into account by the model. The Prophet algorithm is based on fitting these terms and adding them up to obtain the predicted value of the time series.

The Prophet model has the advantages of higher accuracy, faster fitting, generation of adjustable forecasts and robustness to outliers and missing data. The three most important ones are described separately below.

a. **Trend term**: Within the Prophet algorithm, there are two important functions for the trend term, one based on a logistic function and the other on a piecewise linear function.

Non-linear saturation growth is usually similar to the growth in the number of races in a natural ecosystem, reaching saturation after a non-linear growth, e.g. the carrying capacity of the number of WeChat users in a given area may be the number of people with access to the internet. This growth is usually modelled using a logistic regression model, which in its basic form is:

$$g(t) = \frac{c(t)}{1 + \exp(-(k + a(t)^T \sigma)(t - (m + a(t)\gamma)))z} \quad (2)$$

where C(t) is the saturation value (carrying capacity) over time, $[k+a(t)^T\delta]$ is the growth rate over time, $[(m+a(t)^T)\gamma]$ is the corresponding bias parameter, and [δ] is the amount of change in the growth rate at the turning point.

A segmented linear function is one that is linear in each subinterval but is not exactly linear when viewed over the overall interval and is often a polygonal-shape . Thus, the model shape based on segmented linear functions is as follows.

$$g(t) = (k + a(t)^T \sigma)t + (m + a(t)^T \gamma) \quad (3)$$

where K is the growth rate, δ is the change in growth rate and m is the offset parameter.

b. **Periodic term**: relies on the Fourier series to model a flexible periodic model, assuming that P denotes the period of the time series, with P = 365.25 denoting a period in years and P = 7 denoting a period in weeks. It has a Fourier series of all the forms of :

$$s(t) = \sum_{n=1}^{N} (a_n \cos(\frac{2\pi nt}{p}) + b_n \sin(\frac{2\pi nt}{p})) \quad (4)$$

For a sequence with a yearly period (P=365.25), N=10; for a sequence with a weekly period (P=7), N=3 . The parameters here can be formed as column vectors.

$$\beta = [a_1, b_1, \ldots\ldots, a_N, b_N]^T \quad (5)$$

This is achieved by constructing a matrix of seasonal vectors for each matrix. The seasonal term function for the time series is therefore s(t) = X(t)β.

c. **Holiday entries**: Many countries around the world have important holidays and the impact of a particular holiday on the time series is often recurring, so it is necessary to include it in the forecast. Users can also design holidays specific to their own situation, such as "Double 11". For each holiday i, D_i represents the period before and after the holiday. Assuming that there are L holidays, and using K_i to represent the range of holiday effects, the indicator function for the holiday effect is as follows.

$$h(t) = Z(t)K = \sum_{i=1}^{L} K_i \cdot 1_{\{t \in D_i\}}, \quad (6)$$

As with the seasonal trend, the previous normal distribution K~Normal (0, υ^2) is used, which is related to the indicator υ=holidays_prior_scale (distribution of holiday growth rates) and defaults to 10, with larger values indicating a greater impact of holidays on the model, and vice versa, a smaller impact, which can be set by the user as appropriate.

B. LSTM prediction model

LSTM (Long Short-Term Memory) was proposed by Hochreiter and Schmidhuber in 1997, with the addition of the forgetting gate by Graves. The cell state, shown as a line at the top in Figure 1, is equivalent to a memory chain, which only performs some simple linear operations with the rest of the chain, so that it can "remember" information. [4]

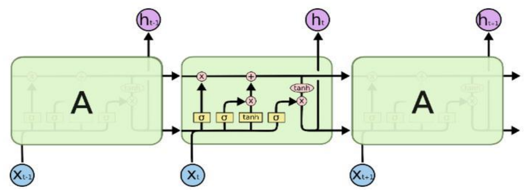

Fig. 1 LSTM structure

In Figure 1, h, x, tanh and σ represent the output, input, hyperbolic tangent and sigmoid functions respectively, and A represents a hidden layer node.

The LSTM protects and regulates the condition of the "cell" by means of "forgetting gates", "input gates" and "output gates". The LSTM protects and regulates the state of the "cell" by means of "forgetting gates", "input gates" and "output gates", and can also "add" or "remove" the state of the cell. Input gates are used to control the strength of the input signal and output gates are used to control the strength of the output signal.

The function of each status gate is as follows.

a. **Forgetting gate**: This gate determines what is to be discarded in the condition of the unit and what is to be saved, the method has a selective processing of historical information. $[h_{t-1}]$ and $[x_t]$ are input items in the range 0 to 1. The Sigmoid function is used to derive this threshold $[f_t]$, with the following formula.

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \quad (7)$$

In the above equation, W_f , U_f and b_f are the weight and bias terms of the forgetting gate.

b. **Input gate:** This gate determines the information stored in the cell cell and allows the status of the cell cell to be updated. Specifically, the sigmoid is used to determine which values will be updated and then a candidate value vector is created by the tanh layer, which eventually updates the old cell state C_{t-1} to the new cell state C_t. The calculation equation is as follows.

$$i_t = \sigma(w_t \bullet [h_{t-1}, x_t] + b_t) \quad (8)$$

$$C_t = f_t C_{t-1} + i_t * \tanh(W_c[h_{t-1}, x_t]) + b_c \quad (9)$$

c. **Output gate**: This gate determines what information is required to be output, depending on the state of the cell. The steps are as follows: first use the sigmoid function to decide what to input (1 for output, 0 for no output required), then operate it with the state of the cell information passing through the tanh layer to finally obtain the required data h(t).

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = O_t + \tanh(C_t) \quad (11)$$

In the above equation, W , U and b are the weight vector and bias vector. It is clear from the information processing flow of the LSTM that it has a good memory of previous information and can effectively solve the problem of gradient disappearance in traditional RNNs, because the new cell state can be seen as some form of accumulation of the previous cell state, and this accumulation form leads to its derivatives being also in accumulation form, thus avoiding the problem of gradient disappearance [5].

C. Prophet-LSTM combined model

The advantages of Prophet model are convenient analysis of various time series features such as periodicity, trend, holiday effect, etc., high operation efficiency, strong interpretation ability, and high accuracy for time series prediction with strong trend[6]. The LSTM model is an improvement of RNN, which solves the problem of gradient disappearance and gradient explosion in the training process of long series, has a long time memory function and is simple to implement. the Prophet-LSTM model is precisely a model that optimally combines the advantages of both, firstly, the historical data set is pre-processed and divided into three parts: trend term, periodic term and random term, and the trend term and periodic P(t) is predicted by Prophet and L(t) is predicted by LSTM, and the result L(t) is obtained by linear weighting of the two.

$$Y(t) = \omega_1 P(t) + \omega_2 L(t), \omega_1 + \omega_2 = 1, t = 1,2,..., N \quad (12)$$

Where t is the time at which the predicted value occurs and the weighting coefficients ω1 and ω2 are determined by least squares, the resulting ω1 and ω2 are the values that bring the combined prediction model Y(t) closest to the true value. Equation (12) above can be written in the following form.

$$y = \omega x \quad (13)$$

Therefore :

$$y(x) = \omega_1 x_1 + \omega_2 x_2 = \sum_{i=1}^{2} \omega_i x_i \quad (14)$$

For the predicted value of the model to be closest to the true value, the following function is derived.

$$f(\omega_1, \omega_2) = \sum_{i=1}^{2} \delta_i = \min \sum_{i=1}^{2} [y(x_i) - y_i]^2 \quad (15)$$

To find the extreme value of the function, we have below :

$$\frac{\partial f(\omega_1, \omega_2)}{\partial \omega_j} = 0, \; j = 1,2 \quad (16)$$

The ω1 and ω2 obtained in equation (16) are the optimal weights.

It is important to note that the model fusion here uses a weighted average method, but voting, stacking and blending can also be used. The algorithmic flow of the combined model is as follows.

- Obtain the time-series data and pre-process the time-series data.
- Divide the time series data into training set, testing set and validation set for learning, testing and validation.
- Decompose the time series into trend, periodic and random terms, and train and test with Prophet model and LSTM model respectively.
- Calculate the optimal weights of the two models after integration.
- Integrate the prediction results of the 2 models, LSTM and Prophet, to obtain the final prediction results.
- Finally, the prediction results are compared with the true values.

III. EXPERIMENTATION AND ANALYSIS

A. Experimental data sources and processing

The training dataset used in this experiment was the daily maximum and minimum temperatures in Shanghai over a period of 5 years, and the task was to predict the daily maximum temperature and the trend of the historical cycle of temperature in the Shanghai area for the next 60 days. The temperature data were crawled from the website Temperature Hindcast (www.tianqihoubao.com). The data structure is shown in Table 1.

date	maxT	minT	weather
2022/06/20	30	23	cloudy
2022/06/19	31	23	light rain

Table 1.Experimental data structures

There are some format transformations to be made for the data set. The only two columns required for the Prophet model are ds and y. The ds is a timestamp column and must be time information; the y column must be a numerical value representing the information to be predicted. The starting point for the temperature time is: 2018/05/1 and the ending point is: 2022/06/20.

B. Experimental environment

- Python 3.8 development environment with Pycharm 2020 IDE tool for debugging and running code.
- Internet Explorer or Chrome browser.

- Jupyter Notebook interactive web application development tool, using Jupyter notebook as the interactive testing platform; based on Python language, using pandas and numpy for data collation and statistical analysis, using matplotlib for visualisation; using the most commonly used Tensorflow framework to build predictive models.

C. *Build and fit weather prediction portfolio model*

In the Prophet model, the scale parameter changepoint_range for finding mutation points is set to 0.8; the fit-following parameter changepoint_prior_scale is set to 0.5, the confidence interval interval_width is set to 0.8, and the model learning is set to multiplicative by. seasonality_mode=multiplicative; the flexibility of seasonality_prior_scale for quarterly periodic mutations is set to 11; the weekly fit pattern is turned off: weekly_seasonality=False. then the forecast length periods is set to 60 and the minimum time granularity freq is "d", i.e. days, specifying the annual learning rule, and the model is trained to fit.

In the LSTM model, the number of neurons is set to 64, the number of training rounds epochs is 50, 100, 200, 500. the learning rate is 0.0003. relu is chosen as the activation function; 'Adam' is used as the optimiser, 'rmse' as the loss function, and the window length batch size is 72. the model is compiled and trained using the same temperature time series data as input. The model was compiled and trained using the same temperature time series data as input. When training the model, an overfitting parameter (Dropout=0.2) can be set to improve this phenomenon in order to prevent overfitting. This parameter means that some features are randomly discarded at each input, thus improving the robustness of the model.

The Prophet-LSTM combined model is constructed based on equation (12), and the optimal values of the weight coefficients ω1 and ω2 are determined by the least squares method. The test set prediction results of the Prophet model and the LSTM model are fully utilised as input for training, and the optimal weights can be determined according to equation (16). results.

D. *Forecast daily maximum temperatures for the next 60 days in the Shanghai area*

This paper examines the predictive capability of the combined Prophet-LSTM model using temperature data from the Shanghai region as an example. In order to achieve the task of predicting the temperature of Shanghai in the next two months, the original data set as a whole was first divided into a training set, a test set and a validation set in the ratio of 7:2:1, and then the training data set was input to the Prophet model and the LSTM model respectively, and the prediction results obtained from the training set were P(t) and L(t) respectively. Then the combined model Y(t) is constructed and trained with the obtained P(t) and L(t) test sets, and the optimal weight coefficients of the combined model are obtained; after the training of the combined model is completed, the validation data set is input to the Prophet-LSTM model for prediction, and the final prediction results are obtained.

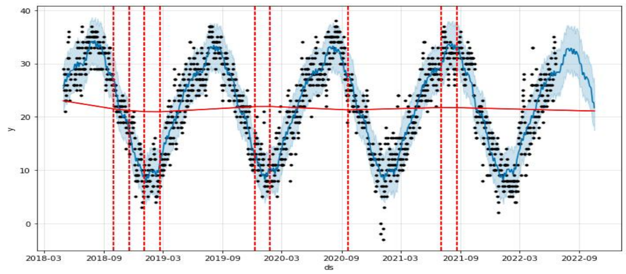

Fig.2 Temperature forecast trend graph

Figure 2 shows the results of an overall forecast, showing the change in trend from the start of the data collection to the end of the expected future time. In this graph, the ds coordinates represent the time, while the y coordinates represent the predicted values; the red dashed line is automatically detected by the model, while the red solid line is the overall trend obtained from the model. In the graphs, the black dots represent the historical information from which the change points can be easily found, while the blue lines represent the predicted trend of the model.

From the above graph, we know that the temperature in Shanghai in the next 60 days will remain above 30° with a floating range between 30° and 38° . Therefore, during the period from mid-June to mid-August, it will always be hot weather in the Shanghai area and measures should be taken to prevent heat; and it can be seen that the temperature will only drop significantly after September.

It can be seen that the practicality and accuracy of the combined model is excellent, which is due to the fact that it combines the respective advantages of the Prophet model and the neural network model LSTM.

E. *Comparative analysis of experimental results*

After training the above proposed Prophet-LSTM combined model, the temperature data from January to June 2022 was selected for prediction with iterations of 50, 100, 200 and 500, and compared with the predicted values of the LSTM and Prophet single model, and the results are shown in Figures 4-7.

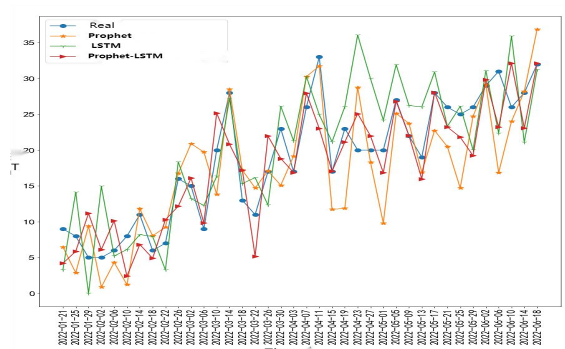

Fig. 4 Comparison of prediction results at 50 iterations

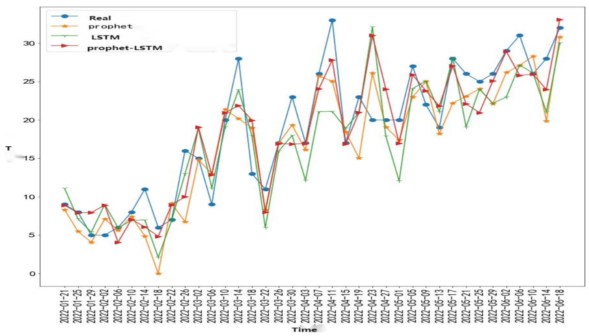

Fig.5 Comparison of prediction results at 100 iterations

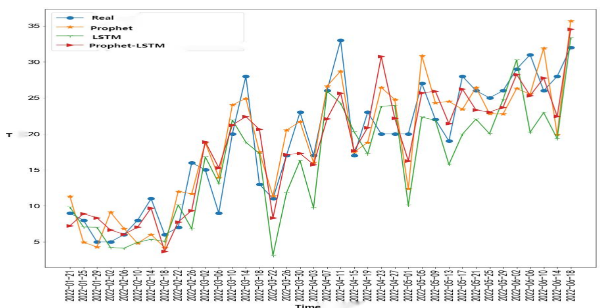

Fig. 6 Comparison of prediction results for 200 iterations

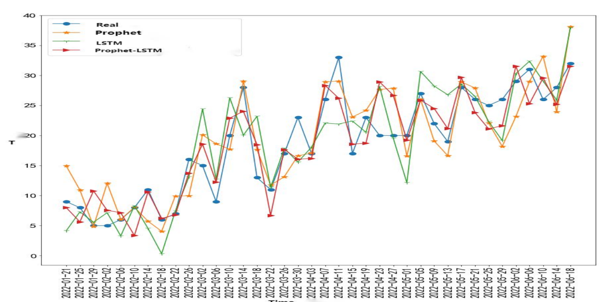

Fig.7 Comparison of prediction results at 500 iterations

As can be seen from Figures 4-7, the accuracy of the combined Prophet-LSTM model is higher than that of the separate LSTM prediction model and the Prophet model, and the prediction results are significantly better than those of the separate model, and the difference between the separate and combined models looks more pronounced as the number of iterations increases. When the number of iterations increases to 500, the prediction curve of the combined model is very close to the curve of the true value. The prediction results of the combined model are experimentally proven to be more accurate than those of the individual models.

F. Assessment models

In order to measure and test the effectiveness of model fitting and prediction, the following two common evaluation metrics are used: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The evaluation metrics of the data sets used in the single and combined models are shown in Table 2.

Data set Name	Models	Normalised evaluation indicators	
		RMSE	*MAPE*
Historical Shanghai Temperature	*LSTM*	0.13657	0.10605
	Prophet	0.12130	0.09463
	Prophet-LSTM	**0.11546**	**0.08586**

Table 2 Comparison of RMSE and MAPE for each prediction model

The root mean square error and the mean absolute percentage error are typical indicators of a regression model and are used to indicate how much error the model will produce in its predictions, with smaller values representing closer to the true value. According to Table 2, the root mean square error and the mean absolute percentage error of the combined model Prophet-LSTM are smaller than those of the single model, so the combined model has better forecasting effect, indicating that the combined Prophet-LSTM model has higher forecasting accuracy and better generalisation ability, and is an excellent time series forecasting model.

IV. Conclusion

In summary, to address the problem of low accuracy of a single prediction model for complex non-linear time series prediction problems, this paper proposes the combined model Prophet-LSTM as an improved solution. By collecting five years of temperature data in the Shanghai area, the model is applied to make predictions for the next 60 days, and the results prove that its prediction results are relatively better and more accurate, and The results show that the prediction is relatively better, more accurate and has better generalization ability. In the next step of the study, more data sets with different business scenarios will be added to continue the exploration and optimisation of the model performance.

References

[1] Ian Goodfellow ,Yoshua Bengio ,Aaron Courville."Deep Learning"[M].2017,pages 228-229.

[2] Wang Wansen. "Principles of Artificial Intelligence and Applications "(4th ed.) [M]. Beijing: Electronic Industry Press, 2018: pages 158-159.

[3] Sean J. Taylor, Benjamin Letham (2017),"Forecasting at Scale"[J], Facebook, Menlo Park, California, *International journal of forecasting*, July 2017, pages 5-12.

[4] Qiu Xipeng. "Neural networks and deep learning" [M]. Beijing: Machinery Industry Press, 2020: pages 141-145.

[5] Tian Qichuan,Wang Manli. "Advances in deep learning algorithm research"[J]. *Computer Engineering and Applications*.2019,(22):25-33.

[6] He Zhengyi, Zeng Xianhua, Qu Shengwei, Wu Zhilong. "A time series forecasting model based on integrated deep learning"[J]. *Journal of Shandong University*. 2016 Dec,(6): 41-46