TimSPred: Long-Term Time Series Prediction Based on Gaussian Process Regression

Wenjia Ye
The University of Hong Kong
Bizseer Technology
yewenjia@outlook.com

Christopher Zheng
McGill University
Bizseer Technology
christopher.zheng@mail.mcgill.ca

Zhenhe Yao
Tsinghua University
Bizseer Technology
yaozh16@mails.tsinghua.edu.cn

Cheng Huang
Bizseer Technology
huangcheng@bizseer.com

Wenchi Zhang
Bizseer Technology
zhangwenchi@bizseer.com

Kaixin Sui
Bizseer Technology
suikaixin@bizseer.com

***Abstract*—Time series typically exhibits patterns that are significant for analytic purposes. Because of its ability to capture various structures by kernel functions, Gaussian Process Regression (GPR) is a handy tool for time series forecasting. Precisely, linear and periodic kernels can be used to extract time series' linear trends and repetitive structures, respectively. Besides, by combining various kernels, GPR can handle more complex structures. Thus, we safely assume that intricate time series can be fitted if kernel combinations are well selected. We propose, TimSPred, a framework that employs GPR to provide accurate and fast long-term predictions for time series. It is inspired by the greedy kernel searching algorithm "ABCD", and we make the following optimizations. First, we estimate the parameters of all kernels of historical data before searching. This estimation narrows down the scope. Second, we reduce the negative impacts imposed by abnormal data patterns, *e.g.*, holidays. Third, we novelly employ the Mean Squared Error metric in place of BIC to evaluate each kernel composition during the greedy searching to achieve better performance. We evaluate TimSPred on real-world datasets of transactions and disk usages from a leading financial institution, and it outperforms common time series predictors in the industry in both the accuracy and time efficiency.**

***Index Terms*—Gauss Process Regression, Period Kernel, Linear Kernel, Fast Fourier Transformation, Linear Regression, Prediction, Additive, Multiplicative, Time Series, Greedy**

I. INTRODUCTION

Time series analysis and forecasting are under intense studies and are increasingly applicable to real-life scenarios. Analyzing historical time series to plan in advance for various affairs, such as the use of storage space [1], the amount of bank transaction data [2], market prices [3], is a common practice for decision makers. Take an example for illustration. Banks are running a huge number of machines every day which serve different purposes but are all considerably expensive. If a bank runs an excessive number of machines, a monetary deficit is highly likely to be incurred whereas, a bank possessing not enough machines also faces a number of problems, such as quality of service (QoS) violations and brand depreciation. Thus, accurate and fast predictions are of essence.

A time series has two key characteristics: the trend and the seasonality [4]. The trend indicates whether the overall data level rises or falls over time; the seasonality implies the periodic fluctuations of time series like weekly, monthly or annual cyclic patterns. Therein lie several common prediction methods: (1) Biexponential Smoothing. Exponential smoothing can predict the trend based on previous data while the cubic exponential smoothing can predict both the trend and seasonality of time series [5]. (2) Auto-Regressive Integrated Moving Average(ARIMA). It has better linear structure and has no effect on the non-linear prediction [6]. (3) Prediction with Deep Learning. Deep learning algorithms such as LSTM are also widely used for the prediction of sequential data. Nevertheless, it is difficult to employ deep learning if only a limited amount of data is provided during statistical modeling. Further, methods using deep learning suffer a strong prediction latency which renders them unqualified for long-term prediction. Plus, the aforementioned algorithms are all to some extent vulnerable to data turbulence and outlier impact during the training phase where they over-fit offline datasets and their results turn out to be quite off. In summary, the existing algorithms are not feasible enough in our case.

We aim for a time series predictor which can not only capture the trend and periodicity but also make reliable long-time predictions. There already exists a seemingly promising predictor ("ABCD") built on Gaussian Process Regression (GPR) and greedy kernel searching [28], but we experience some severe problems when applying it. First, the run time of ABCD is dynamically influenced by many factors and cannot be determined or guaranteed. Second, due to certain feature of the Gaussian Process, the noise or any other types of data aberrations, such as holiday pattern, negatively impact the prediction results. Last, the ABCD mostly stays on the level of research and lacks applicability to real datasets.

To tackle the aforementioned challenges, we present our framework, Trepred, which provides accurate and efficient long-term time-series forecasting based on Gaussian Process Regression. This framework, embedded in an easy-to-deploy and user-friendly tool, very quickly makes precise time-series predictions with the robustness against data turbulence.

II. Background

We carefully apply and integrate three state-of-the-art methods in our framework: Gaussian Process Regression, Fast Fourier Transform and Linear Regression. In this section, we briefly introduce each method individually and we present the motivational algorithm that inspires and facilitates our work.

Gaussian Process Regression (GPR): Gaussian Process (GP) is widely used in Time Series Analysis and Machine Learning. GP is a stochastic process in probability theory and mathematical statistics. GP is determined by the expectation and the co-variance function. Modeling and prediction of Gaussian Processes is an important field of machine learning and signal processing [10]. GPR makes predictions based on prior GP knowledge. GPR is usually utilized for regression problems with low and small samples because of high computational overheads but there are some extend algorithms to adapt to large samples and higher dimensions [7]. The most important of GPR is the covariance matrix which is decided by kernel functions [9] that we often use include Radial Basis Function (RBF) [10], Periodic kernel, Linear kernel.

Kernel: is the most important component to help calculate the covariance matrix which represents the similarity of data. Fundamental kernels we utilize in TimSPred are Periodic, Linear and Noise Kernels.

Fast Fourier Transform (FFT): can accurately and efficiently handle time series data [11]. It is an optimized variant of Discrete Fourier Transform (DFT). Given that N is the number of data entries, the basic process of FFT is to decompose the original N-point sequence into a series of short sequences [14]. The larger N is, the better result FFT optimizes. FFT can eliminate the issue of "duplicate counting" and can reduce time complexity.

Linear regression (LR): is to describe the linear relationship among data entries. Remarkably, the linear regression model can fairly quickly fit a large-scale dataset. Since LR is sensitive to outliers [15], they should be properly analyzed and handled. LR models are often fitted by least squares approximation [16] to achieve the best fitting of the data.

Our work is inspired by JR Lloyd et al ([28]) who propose an Automatic Bayesian Covariance Discovery (ABCD) system to process unknown regression functions non-parametrically using GP. After greedily searching for kernels and corresponding kernel parameters, ABCD evaluates the model using Marginal Likelihood and Bayesian Information Criteria (BIC), and nominates the kernel with the smallest BIC for the best kernel for each layer. Each layer in the kernel searching process is modified and extended accordingly. Next, it uses one kernel as a core kernel and the others as fixed parts to extend and modify the core's expression. The experimental results demonstrate that after searching for and fitting a certain number of layers, the prediction on the given time series data turns out to be excellent and its overall trend becomes stable.

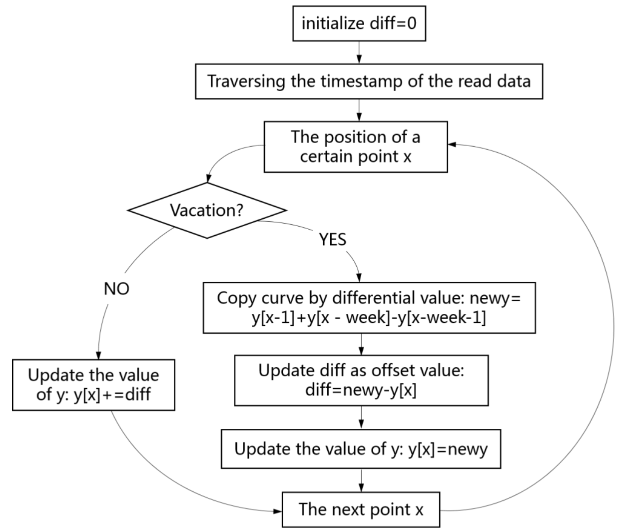


Fig. 1. Procedure to fix holiday data when the data has a trend

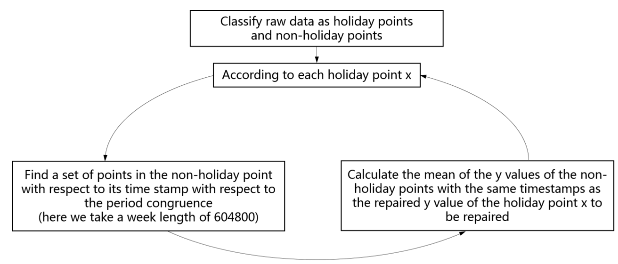


Fig. 2. Procedure to fix holiday data when the data is periodic

III. The TimSPred framework

We present our TimSPred framework designed for accurate and efficient long-term time-series forecasting based on Gaussian Process Regression. To start with, we introduce each step of our framework. First, we check if there is any unusual pattern in our data which may negatively hinter our predictions. We replace unusual patterns with the normal pattern in the training stage and perform an extra step of conversion in the prediction stage. Then, we use LR and FFT to extract the linear and periodic features [8] of time series, respectively. Based on these features, the parameters of the linear [9] and periodic kernels can be accurately estimated. Next, we search for the optimal kernel compositions of time series using a greedy algorithm. Finally, GPR comes to light to train the model and make the prediction.

The effect of long-term prediction can be negatively affected by any unusual patterns due to the property of Gaussian Process. These unusual patterns are commonly incurred by holidays that may cause mutations to time series. In TimSPred, we solve this problem by curve shape translation. In the preprocessing stage, we first identify the shape of a time series. If the time series has a long-term trend, we go through all the data and check whether the timestamp of an incoming data is a date of a holiday. If yes, we replace the data by a differential

Fig. 3. The frequency bands after FFT

value and use the differential value to renew the offset value. If it is not a holiday, we simply use the offset value to update that piece of data. The detailed process is shown in Fig 1. On the other hand, if the time series is periodic, we divide the data into holidays and ordinary values. We further divide the ordinary values into subsets corresponding to each day of a week, time of a day, and calculate the mean of each set. Then, we replace a holiday's value with one of the mean values by identifying which set it belongs to. The detailed process is shown in Figure 2. We also calculate the residual values to document the difference between the training and raw data so that, in the prediction stage, we can convert the training and prediction data back to the correct values.

To narrow down the search scope and improve the efficiency, we estimate the parameter of each kernel before searching. The *trend parameter* of the linear kernel is the intercept derived by fitting a linear equation to the preprocessed data. To extract the *periodicity parameter* of the periodic kernel, we remove the trend from the time-series, and transform the residual into the frequency domain by applying FFT. An example of the power of each frequency calculated by FFT is shown in Figure 3. The periodicity parameter is decided according to the following principle: 1) If the daily and weekly periodicity are both dominant frequencies, the weekly one should be chosen for higher fault tolerance; 2) Otherwise, yearly, monthly, weekly or daily periodicity with the highest power should be chosen according to the common sense. The *amplitude parameter* of the periodic kernel is properly extracted in a grid search strategy. We use the highest and the third highest amplitude of all waves in the frequency domain as the search boundary, and choose the value with least MSE when applied to the period kernel by grid search as the amplitude parameter.

To find the optimal composition of kernels, we use the greedy kernel searching algorithm defined in [28] with two major modifications. One is that we use MSE instead of Bayesian Information Criterion (BIC) to evaluate each composition. BIC is an effective weapon in combating overfitting. Nevertheless, the kernels in TimSPred is almost fixed after their parameters are estimated, the only uncertainty lies in the combinative strategy. This is quite different from [28] where more random search effort has to be made to find the best parameters alongside with the combination. The direct benefit from that is the number of layers to search and number of times to restart can be much smaller compared with ABCD algorithm. Therefore, the risk of overfitting has already been effectively suppressed by our parameter estimating procedure and we can replace BIC with some other criterion which focus more on the effectiveness of the model itself. Here we choose MSE as our loss function since it is the practical best criterion. The other modification is that we abandon the RBF kernel during the searching procedure since in practice, RBF kernel often leads to overestimation (or underestimation) of the long-term trend in predictions.

Algorithm 1 Gaussian Process Regression

Preprocess
  if raw data has trend **then**
    fix holiday data according to Fig. 1
  else if raw data is periodic **then**
    fix holiday data according to Fig. 2
  end if
End
Estimate Parameters
  Perform linear regression on preprocessed data.
  Use intercept as the parameter of linear kernel.
  Calculate residual by substracting linear trend from original data.
  Perform FFT on residual and sort the frequency bands.
  Obtain cyclic parameter of the periodic kernel according to the sorted frequency.
  Perform grid search to get the amplitude parameter.
End
Iteratively search for the kernels
  initialize search parameter m and restart parameter n
  initialize $MSE_g = \infty$
  while $n > 0$ **do**
    initialize $MSE_l = \infty$
    while $m > 0$ **do**
      create new kernel compositions
      perform GPR for each kernel composition
      cut search branches and update MSE_l
      $m \leftarrow m - 1$
    end while
    update MSE_g if $MSE_g > MSE_l$
    $n \leftarrow n - 1$
  end while
  output the kernel composition corresponding to MSE_g
End
Prediction
  Use this kernel to fit data and predict future pattern.
End

IV. Evaluation

Implementation Specifications: Our framework is implemented using Python3.6 and runs on a MacBook Pro 2015 single core.

Datasets: We collect real data from a leading financial institution and separate it into Data 1 - 5, respectively. They represent the disk usage, transaction amount, table space usage, response time and so on. We sample the data one point per hour for the following experiments.

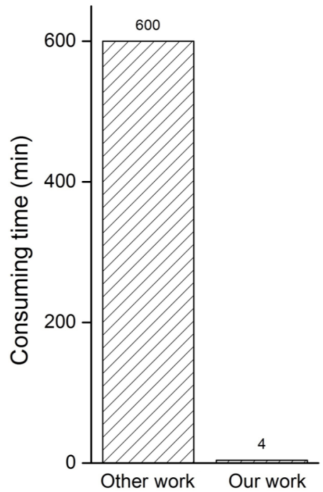


Fig. 4. The run time difference between our work and other work

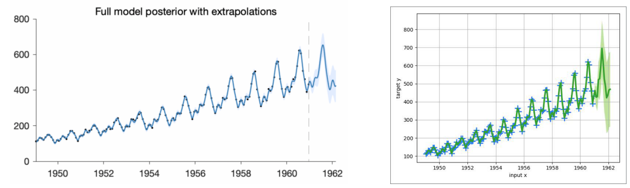
Experimental Results: We compare and contrast our work with three other baselines, *i.e.*, ABCD, Facebook's Prophet, and DeepAR.

In Figure 5, we present the prediction results of ABCD and our TimSPred on the dataset of airline passengers. Both algorithms obtain similarly excellent results, but our algorithm requires significantly shorter run time *i.e.*, TimSPred is way more efficient as shown in the bar plot in Figure 4 which addresses the run time difference of ABCD and TimSPred. TimSPred spends four minutes producing the prediction whereas ABCD spends almost ten hours. This huge reduction in time consumption results from the extra procedure in TimSPred to estimate the kernel parameters before searching, and this procedure, in turn, enables us to set our searching parameters in a reasonably small range. In addition, if the data contains unusual patterns including holidays or change points, ABCD chokes up. As illustrated in Figure 6, when predicting Data 1 intervened with public holidays, ABCD is misled by the unusual pattern while TimSPred still performs well.

Our following experiments focus on Facebook's Prophet and DeepAR. In Figure 8 and Figure 7, we demontsrate the experimental results of Facebook's Prophet and DeepAR separately on Data 1-5, respectively. Blue lines denote the original time series in both figures and red (Prophet) or green (DeepAR) lines represent their prediction results.

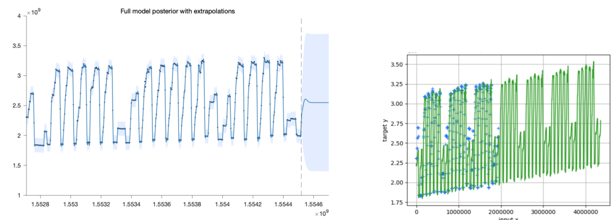
TABLE I
MSE COMPARISON OF OUR WORK WITH DEEPAR AND PROPHET

Method	Mean Squared Error (MSE)				
DeepAR	9.15e+16	8.29e+11	1.74e+6	219.96	2.3e-3
Prophet	2.79e+16	8.22e+10	1.70e+5	213.07	4e-4
Our Method	2.67e+16	1.70e+10	1.07e+5	172.12	2e-4



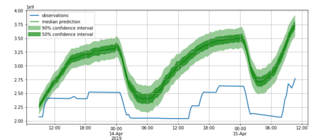
(a) ABCD's result (b) **TimSPred's result**

Fig. 5. The results of ABCD and TimSPred on Airline passenger data

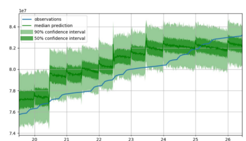


(a) ABCD's result (b) **TimSPred's result**

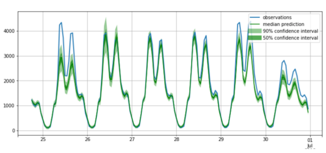
Fig. 6. The results of ABCD and TimSPred on Data1



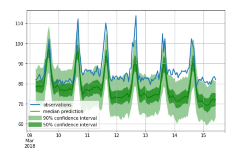
(a) Data1 result



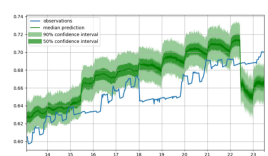
(b) Data2 result



(c) Data3 result



(d) Data4 result



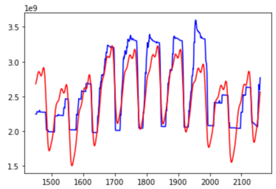
(e) Data5 result

Fig. 7. The results of DeepAR

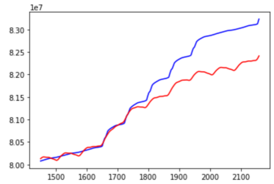
The results of our framework tested on these datasets are plotted in Figure 9. Again, the original time series data is in blue and TimSPred's predictions are in green. Clearly, in terms of long-term prediction accuracy, our algorithm outperforms Prophet and DeepAR on real life datasets across-the-board.

Aside of plotting the long-term predictions of the mentioned algorithms, we calculate and present every algorithm's mean squared error (MSE) on each dataset in Table I. Our TimSPred leads other algorithms by constantly obtaining the lowest MSEs on all datasets.

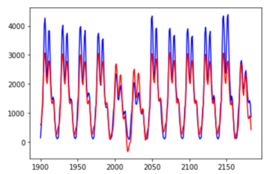
Deployment: Because of the accuracy and efficiency of our framework TimSPred, it is deployed in a leading banking system in China and responsible for the routine predictions



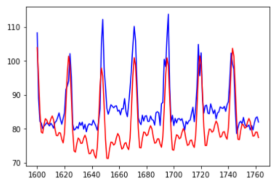
(a) Data1 result



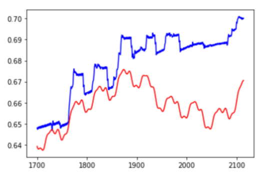
(b) Data2 result



(c) Data3 result



(d) Data4 result



(e) Data5 result

Fig. 8. The results of Prophet

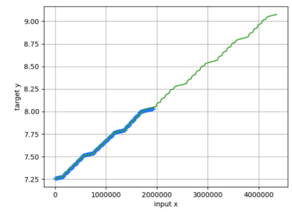
of various time series data such as the transaction amount, disk usage, and response time.

V. Discussion and Future Work

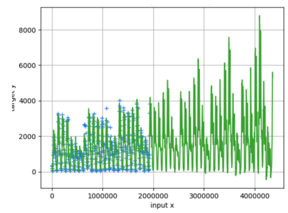
Although TimSPred can produce acceptably promising results, therein lie some future improvements to be followed. For instance, for the currently optimal performance, we use MSE as the standard metric in the searching procedure while we may try many other metrics to test and improve our model. Besides, the kernels we use are actually limited in population. As a future work, we may create more kernels to fit more patterns of time series. In addition, searching greedily is implemented to obtain the best kernel compositions, whereas we can try later other searching methods to make a comparison.

VI. Related work

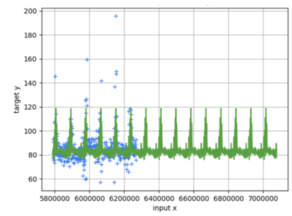
Many algorithms can be implemented to analyze and predict time series. Gradient Boosting Regression Tree (GBDT) can effective for prediction and all the trees that GBDT builds can extract many features for regression tasks [17] [18]. LGBM is an advanced algorithm in the field of GBDT. It is a histogram-based algorithm and does not require one-hot coding [19]. Holt-Winters is also a common predict method which can predict time series' trend and seasonality [20] [21]. AutoRegressive Integrated Moving Average (ARIMA) is widely used in stock forecasting [22], currency change [23] [24], sales [25] and other fields. ARIMA model, however, assume that the predicted value has a linear relationship with the addition of White Noise so that ARIMA is difficult



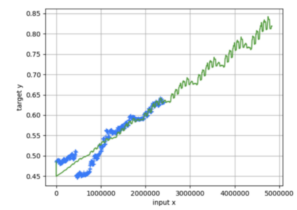
(a) Data2 result



(b) Data3 result



(c) Data4 result



(d) Data5 result

Fig. 9. The results of TimSPred. (TimSPred's result of Data1 is in Figure 6)

to fit given complex non-linearities [26]. [27] compare two automatic predictors based on ES and ARIMA. Prophet can handle outliers and obtain desired predictions within a short time. Nonetheless, the result of time series with tendency in our own scene is unacceptable. A lot of data, especially time series data, forecasts based on LSTM such as some intelligent transportation system [29] [30] and stock price [31]. SHI Xingjian et al. [33] propose a Convolutional LSTM Network (ConvLSTM) for precipitation forecasting which can predict not only time series data, but also other space-time series data. DeepAR designs an RNN to generate different probability predictions in various scenarios to achieve a higher prediction accuracy. In addition, it can also predict items with little or no datasets at all [34].

VII. Conclusion

A lot of time series' patterns house precious information that can be utilized for great analytic purposes and commercial affairs, if properly processed and forecasted. Hence the reason why we propose, TimSPred, a comprehensive framework that employs Gaussian Process Regression to provide accurate and rapid long-term predictions for time series. Firmly based on a GPR-related previous study, TimSPred is greatly enhanced by optimal kernel parameter estimation, kernel searching, and novel kernel composition. Tested on real-world datasets from a leading financial institution, TimSPred is proved to outperform commonly used time series predictors in the industry in terms of both the accuracy and, equally importantly, time efficiency.

References

[1] Benirschka M, Binkley J K. Optimal storage and marketing over space and time[J]. American Journal of Agricultural Economics, 1995, 77(3): 512-524.

[2] Ravi V, Kurniawan H, Thai P N K, et al. Soft computing system for bank performance prediction[J]. Applied soft computing, 2008, 8(1): 305-315.

[3] Zareipour H, Caizares C A, Bhattacharya K, et al. Application of public-domain market information to forecast Ontario's wholesale electricity prices[J]. IEEE Transactions on Power Systems, 2006, 21(4): 1707-1717.

[4] Cryer J D, Kellet N. Time series analysis[M]. Royal Victorian Institute for the Blind. Tertiary Resource Service, 1991.
[5] Brown R G. Smoothing, forecasting and prediction of discrete time series[M]. Courier Corporation, 2004.
[6] Box G E P, Pierce D A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models[J]. Journal of the American statistical Association, 1970, 65(332): 1509-1526.
[7] Duvenaud D. Automatic model construction with Gaussian processes[D]. University of Cambridge, 2014.
[8] Ebden M. Gaussian processes for regression: A quick introduction[J]. The website of robotics research group in department on engineering science, University of Oxford, 2008, 91: 424-436.
[9] Fine S, Scheinberg K. Efficient SVM training using low-rank kernel representations[J]. Journal of Machine Learning Research, 2001, 2(Dec): 243-264.
[10] Park J, Sandberg I W. Universal approximation using radial-basis-function networks[J]. Neural computation, 1991, 3(2): 246-257.
[11] Cochran W T, Cooley J W, Favin D L, et al. What is the fast Fourier transform?[J]. Proceedings of the IEEE, 1967, 55(10): 1664-1674.
[12] Cooley J W, Tukey J W. An algorithm for the machine calculation of complex Fourier series[J]. Mathematics of computation, 1965, 19(90): 297-301.
[13] Weisstein E W. Fast fourier transform[J]. 2015.
[14] Van Loan C. Computational frameworks for the fast Fourier transform[M]. Siam, 1992.
[15] Redden D T, Woodall W H. Properties of certain fuzzy linear regression methods[J]. Fuzzy sets and Systems, 1994, 64(3): 361-375.
[16] Draper N R, Smith H. Applied regression analysis[M]. John Wiley & Sons, 1998.
[17] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. Annals of statistics, 2001: 1189-1232.
[18] Si S, Zhang H, Keerthi S S, et al. Gradient boosted decision trees for high dimensional sparse output[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 3182-3190.
[19] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems. 2017: 3146-3154.
[20] Brown R G. Smoothing, forecasting and prediction of discrete time series[M]. Courier Corporation, 2004.
[21] Gardner Jr E S. Exponential smoothing: The state of the art[J]. Journal of forecasting, 1985, 4(1): 1-28.
[22] Ariyo A A, Adewumi A O, Ayo C K. Stock price prediction using the ARIMA model[C]//2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. IEEE, 2014: 106-112.
[23] Meyler A, Kenny G, Quinn T. Forecasting Irish inflation using ARIMA models[J]. 1998.
[24] Weisang G, Awazu Y. Vagaries of the Euro: an Introduction to ARIMA Modeling[J]. Case Studies in Business, Industry, and Government Statistics, 2008, 2(1): 45-55.
[25] Permatasari C I, Sutopo W, Hisjam M. Sales forecasting newspaper with ARIMA: A case study[C]//AIP Conference Proceedings. AIP Publishing, 2018, 1931(1): 030017.
[26] Khashei M, Bijari M, Ardali G A R. Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs)[J]. Neurocomputing, 2009, 72(4-6): 956-967.
[27] Hyndman R J, Khandakar Y. Automatic time series for forecasting: the forecast package for R[M]. Clayton VIC, Australia: Monash University, Department of Econometrics and Business Statistics, 2007.
[28] Lloyd J R, Duvenaud D, Grosse R, et al. Automatic construction and natural-language description of nonparametric regression models[C]//Twenty-eighth AAAI conference on artificial intelligence. 2014.
[29] Zhao Z, Chen W, Wu X, et al. LSTM network: a deep learning approach for short-term traffic forecast[J]. IET Intelligent Transport Systems, 2017, 11(2): 68-75.
[30] Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction[J]. Transportation Research Part C: Emerging Technologies, 2017, 79: 1-17.
[31] Kim H Y, Won C H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models[J]. Expert Systems with Applications, 2018, 103: 25-37.
[32] Hansson M. On stock return prediction with LSTM networks[J]. 2017.
[33] Xingjian S H I, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[C]//Advances in neural information processing systems. 2015: 802-810.
[34] Salinas D, Flunkert V, Gasthaus J. DeepAR: Probabilistic forecasting with autoregressive recurrent networks[J]. arXiv preprint arXiv:1704.04110, 2017.