Machine Learning and Artificial Intelligence
J.-L. Kim (Ed.)

doi:10.3233/FAIA220419

Forecasting and XAI for Applications Usage in OS

Ruimin MA[a], Yuyi ZHANG[a], Jing LIU[a], Yin LI[a], Ovanes PETROSIAN[a, b, 1], and Kirill KRINKIN[b]
[a] *Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University, Universitetskii Prospekt 35, 198504 St. Petersburg, Russia*
[b] *Saint Petersburg Electrotechnical University "LETI", Professora Popova 5, 197376 St. Petersburg, Russia*

Abstract. In the context of digital informatization, the Internet is changing the way of human existence. The rapid development of the Internet has promoted the use of smartphones in people's daily lives, and at the same time, a large number of applications running on different operating system environments have appeared on the market. Predicting the duration of application usage is crucial for the management planning of related companies and the good life of users. In this work, a dataset containing time series of user application usage information is considered and the problem of "application usage" forecast is being addressed. The dataset used in this work is based on reliable and realistic user records of the usage of the application. Firstly, this paper investigates suitable forecast models for application development on the applied user usage time dataset, which includes neural network algorithms and ensemble algorithms, among others. Then, an Explainable Artificial Intelligence Approach (SHAP) is introduced to explain the selected optimal forecast models, thus enhancing user trust of the forecasting models. The forecast results show that the ensemble models perform better in the time series dataset of user application usage information, especially LightGBM has more obvious advantages. Explanation results show that the frequency of use of the target variables, category and lagged nature are important features in the forecast of the application dataset.

Keywords. Time-series forecasting, Ensemble model, Neural network, Explainable AI (XAI)

1. Introduction

In order to facilitate life, users have installed a large number of Apps on their smartphones. These installed Apps on mobile phones have a negative impact on the responsiveness of mobile phones, not only increasing the time it takes for users to find Apps, but also taking up mobile phone memory and causing phenomena such as mobile phone lag, which seriously affects the user experience.

The boosting algorithm (model) [1], bagging algorithm (model) [2], and neural network algorithm (model) [3] have been widely used by contestants in competitions concerning the forecast of time series datasets. The results generated using these

[1] Corresponding Author, Ovanes Petrosian, Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University, Universitetskii Prospekt 35, 198504 St. Petersburg, Russia; Saint Petersburg Electrotechnical University "LETI", Professora Popova 5, 197376 St. Petersburg, Russia; E-mail: petrosian.ovanes@yandex.ru.

algorithms [4] in different competitions have shown that their performance is unstable and therefore it is difficult for researchers to compare and measure them. Therefore, the task of choosing the optimal forecast model in the problem of forecasting time series data sets is of great importance for both the companies involved and the users.

The usage time of a certain application can be influenced by some applications (e.g., weather prediction applications, gaming applications) or whether users will access it during their commute to work or when they have free time in the evening [5]. Thus, by forecast the duration that a user will spend on an application, it is possible to forecast the usage patterns when using the application. Currently, there are few studies on modeling application usage time from large-scale datasets.

Faraki et al [6], Xu et al [7], Li et al [8], and Böhmer et al [9] have conducted studies based on application usage time. However, with basic data types such as application usage time aggregated by previous researchers, current researchers do not have insight into which factors (application category, application price, etc.) affect the duration of users' usage time in applications [10].

Therefore, on the one hand, we use popular prediction models to predict the usage time of applications and compare forecast models synthetically to explicitly identify the optimal forecast model at the real situation. On the other hand, we explain the selected optimal forecast model at the comprehensibility level to better understand the features learned by the forecast model. This work is appropriately supported by the user daily application usage dataset [11] and the LiveLab dataset [12]. These datasets include information about users' use of applications. Importantly, these data are reliable records of users' use of Google Play Store apps, which ensures that the forecast models used are meaningful.

Both ensemble models and neural networks are considered to define a relevant best prediction algorithm for the direction of use of the application. Among them, the ensemble model contains boosting algorithm and bagging algorithm. In this paper, we use LightGBM [12] algorithm in boosting algorithm, Random Forest algorithm [13] in bagging algorithm, Bi-RNN algorithm [14], Bi-LSTM algorithm [15] and Bi-GRU algorithm [16] in neural network model to forecast the duration of users' usage of the app.

In the currently existing work on the measurement of forecast models, XAI techniques take an alternative view of the results generated using XAI methods (importance of features) that can help users to intuitively understand the results generated by forecast models [17-19]. For example, XAI can be applied to medically assisted diagnosis, so that this has important implications for the doctor's diagnosis, prompting black-box models and doctors to make more beneficial decisions for patients [20-24], and XAI can also be applied to automated driving, when automated systems make decisions or recommendations, for practical factors and socio-legal reasons, to users, developers, and regulators is essential to provide explanations. In the face of the rapid development of information technology, there is an increasing interest in machine learning and deep learning, which are called "black box models" because most of the algorithms in machine learning and deep learning are not intuitively understandable [25-30]. Therefore, factors such as the inability of users to visually identify the correctness of the results produced by "black box models" have led to a greater interest in the field of XAI. When users use these "black box models", it is important for the trust and reliability of the forecast results.

Section 3 of this paper presents information about the dataset, such as the description of the data and the presentation of the data in the dataset. In Section 4, the forecast

algorithm used based on the application using the recorded data set is presented. In Section 5, the forecast results using classical measures and the results between forecast models are visualized. In Section 6, the XAI method chosen for use in this paper is presented and the optimal forecast model selected in Section 5 is explained using the Explainable AI algorithm. Conclusions and a brief description of future work are shown in Section 7.

2. Related work

Faraki et al [6] found that users spent less than 6 minutes on 90% of the apps. Xu et al [7] found that most of the sustained usage time of users on all apps was 10 seconds to 1 hour per user in a week. Li et al [8] analyzed the total usage time of different categories of apps and found that communication apps accounted for 49% of all apps. Böhmer [9] found that users had the longest average duration of use on Libraries & Demos applications (Default Updater, Google Services Framework, etc.).

In this work, firstly, five popular forecast models are used to predict the usage time of applications for Daily Phone Usage dataset [11] and Live Lab dataset [12] to find the optimal forecast model suitable for this type of dataset. Second, the selected optimal forecast models are explained and analyzed using the XAI method to identify the features that have the greatest impact on the forecast results using the optimal models.

3. Datasets

3.1. Descriptive information about the dataset

The Daily Phone Usage dataset [11] and LiveLab dataset [12] include application usage records and is important for a wide range of OS forecasting problems. Liu et al [31] mainly used the dataset to study the way users use various applications, e.g., how often users use the applications, and did not focus on forecasting issues concerning applications. While another paper [32] using this dataset is studying about the order in which various applications are launched. They both cover most of the unresolved issues in the forecasting of operating systems.

The description of the dataset containing records of users using the application used in this paper is as follows.

- The dataset of “Daily phone usage” includes usage information of various applications: applications (Settings, MTP application, Messages, Whatsapp, etc), the date, time, and duration of the user’s use of the application time [11]. (https://www.kaggle.com/johnwill225/daily-phone-usage)
- The dataset of “LiveLab Dataset” includes usage information of various applications: applications (Settings, MTP application, Messages, Whatsapp, etc), the date, time, and duration of the user’s use of the application time [12]. (http://yecl.org/livelab/traces.html)

3.2. Display chart about the dataset

In the Daily Phone Usage dataset, the time series ranges from approximately May 2019 through November 2019. In the LiveLab dataset, the time series ranged from approximately September 2010 until March 2011. Based on the order of the time series, approximately seventy percent of the data in the dataset were divided in the training set, and the remaining thirty percent was divided in the test set. Figures 1 and 2 show the change of "Duration" in the data set to the time series.

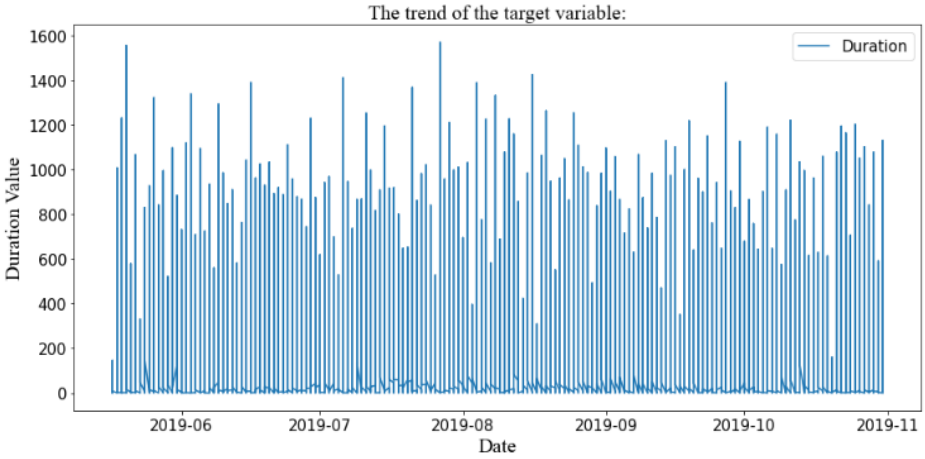

Figure 1. The target variable "Duration": the trend of them in the Daily Phone Usage.

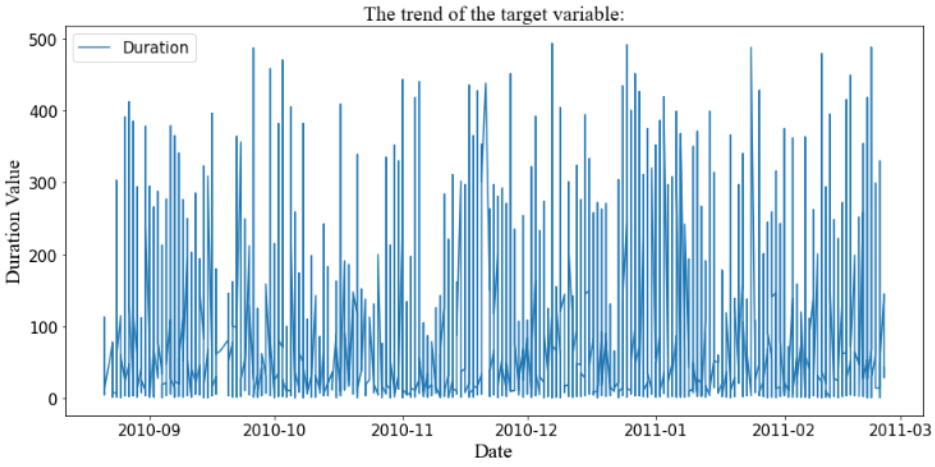

Figure 2. The target variable "Duration": the trend of them in the LiveLab Dataset.

4. Solution approach

At present, many users choose to use neural network algorithms and integrated models (boosting algorithm and bagging algorithm) as forecast models in competitions related to time series forecasting datasets. In this paper, we choose the LightGBM algorithm, which represents the ensemble model of boosting, the Random Forest algorithm, which represents the ensemble model of bagging, and the Bi-RNN, Bi-LSTM, and Bi-GRU algorithms, which are neural network algorithms with bi-directional accuracy, to forecast the duration of app usage.

4.1. LightGBM and random forest algorithms

The LightGBM algorithm is a gradient boosting algorithm based on decision trees. On the one hand, it adopts the histogram algorithm mutually Exclusive Feature Bundling algorithm (EFB) to reduce memory consumption, thereby improving the running speed of the algorithm. On the other hand, it uses the GOSS algorithm that chooses to leave only the instances with larger gradients and randomly samples the instances with smaller gradients, thus balancing the speed and performance of the algorithm [33]. The Random Forest algorithm [14] uses decision trees as a model in bagging, where random sampling results in a wide diversity, and the optimal result can be selected from the computation results of multiple independent decision trees through a "voting" mechanism.

4.2. Bidirectional RNN, bidirectional LSTM and bidirectional GRU algorithms (Bi-RNN, Bi-LSTM and Bi-GRU)

As we know, Bi-RNN, Bi-LSTM, Bi-GRU algorithm is actually a neural network with two layers. Starting from the left, the initial input value in the first layer of the time series dataset is the start time. Counting the first layer from the left and the second layer from the right is the input, we can understand it as the input value of the previous time series in the time series dataset, i.e. that is, the output of the forward state and the input of the backward state in the neural network model are not connected together [34]. Finally, the two results obtained from the two layers are processed separately.

5. Simulation results

The computer used in this paper, i.e. the code runtime environment, is as follows. (1) a computer with CPU—Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz; RAM—16.0GB; OS—Windows 10; (2) Python—Jupyter 3.8.3.

In Section 5, this paper only shows the graphs of the loss value results for the Daily Phone Usage dataset (see Figure 3) and the comparison graphs of the true and forecast values based on the LightGBM algorithm (see Figure 4). A comparison table of the quantification of five forecast models based on Daily Phone Usage and Live Lab datasets is also shown (see Table 1 and 2).

Observing Figure 3, the vertical coordinate of the learning curve of the LightGBM algorithm is RMSE, and the vertical coordinate of the learning curve of the neural network algorithm is MSE, we find that the learning curves of the prediction algorithms used in this paper both have a decreasing trend.

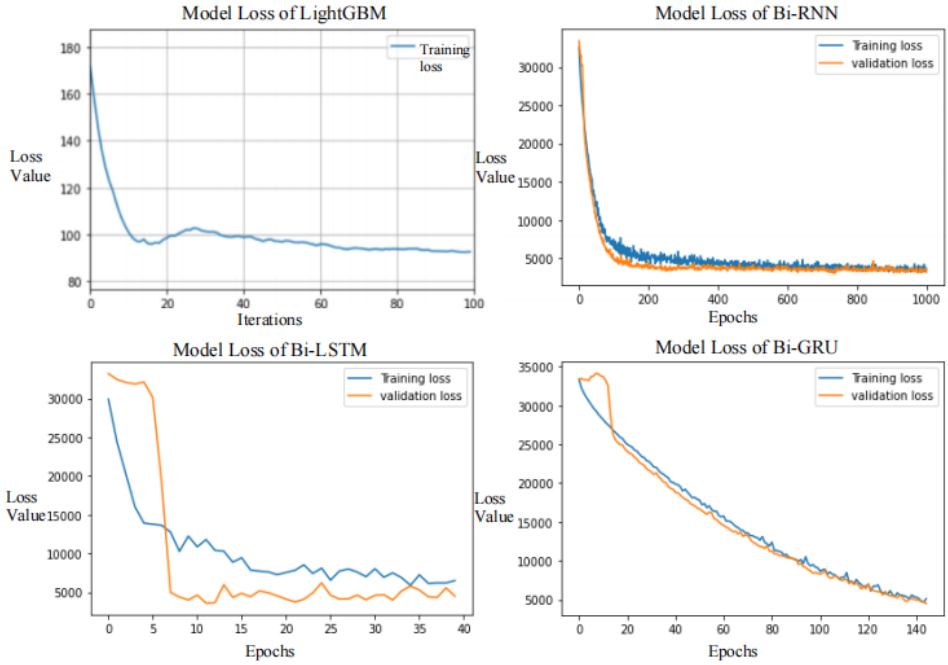

Figure 3. Daily Phone Usage: Learning curves of forecast models.

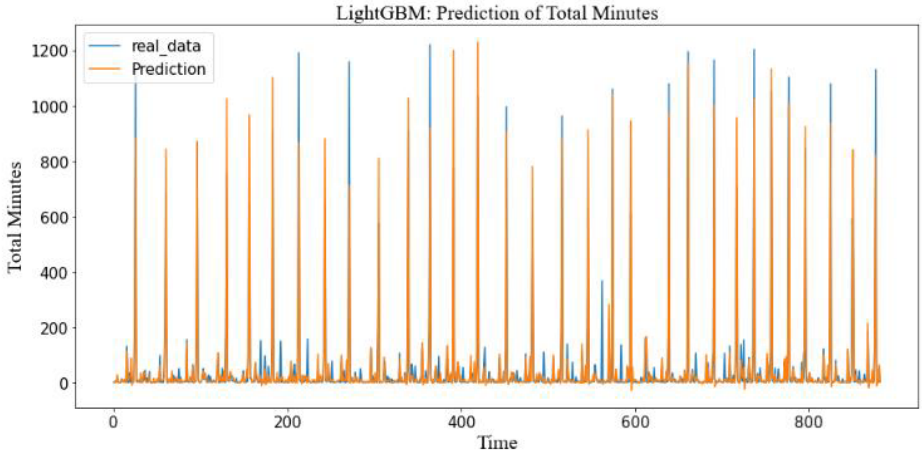

Figure 4. Daily Phone Usage: comparison of real and forecast values.

Table 1. Forecast Quality of Daily Phone Usage.

Daily Phone Usage	R^2	MSE	TimeSpent
LightGBM	0.9123	51.9678	0.18s
Random Forest	0.9133	51.6763	30.31s
Bi-RNN	0.8664	64.1259	34.34s
Bi-LSTM	0.8759	61.8016	32.60s
Bi-GRU	0.8406	70.083	48.86s

Tables 1 show the results of all forecast models on Daily Phone Usage. This paper mainly measures performance from the perspective of variance. It is easy to see that all forecast models produced good performance on Daily Phone Usage (see Figure 3 and Table 1), among which Random Forest had the best performance.

Next, we analyze the running time of the algorithm, and it is easy to find that the running time of the LightGBM algorithm is one hundredth of the running time of the other four forecast models, especially that it can achieve a fit close to that of the Random Forests algorithm without sacrificing the running time. Therefore, this paper finds that the LightGBM algorithm is the optimal forecast model for the Daily Phone Usage dataset.

Table 2 shows the results of all forecast models for the LiveLab dataset. The analysis in terms of variance and running time of the algorithms shows that the ensemble models perform well, with LightGBM performing the best and the neural network model performing poorly. Therefore, this paper finds that the LightGBM algorithm is the optimal forecast model for the LiveLab dataset.

Table 2. Forecast Quality of LiveLab Dataset.

LiveLab Dataset	R^2	MSE	TimeSpent
LightGBM	0.7785	3276.3813	0.18s
Random Forest	0.7629	3390.2444	5.75s
Bi-RNN	0.0475	6793.9188	30.63s
Bi-LSTM	0.228	6116.5953	23.13s
Bi-GRU	0.0461	6799.0204	69.25s

6. Explaining the optimal forecasting model

6.1. Explainable Artificial Intelligence (XAI)

XAI improves the transparency of black-box prediction models to humans through a variety of methods that allow humans to understand and trust the black-box models. The current widely used models require the use of feature importance values to be understood [35], so a popular approach is to observe the feature importance by calculating the contribution of each feature, where a larger calculated contribution indicates a greater impact on the prediction result and a smaller calculated contribution indicates a smaller impact on the prediction result. Also, this method by calculating the contribution of features can express the inter-influence relationship between features [18]. Classifying the XAI into local and global explanations due to the different explanatory objects, i.e., different target variables, BackProb and Perturbation according to different explanation algorithms, and Intrinsic and Post-hot according to different explanation principles. Local is an explanation of a single instance that builds an explanation of a particular outcome of a dataset or a decision made by an instance, while global is an explanation of all dataset, and global explanation is based on the conditional interactions between response variables and input features on the complete data set to understand and explain the decisions of the entire model at once. BackProb relies on the gradient back propagated from the output forecast layer to the input initial value layer, while Perturbation relies on features randomly selected from the input data instances or selected according to some rule. Intrinsic refers to the explanatory power of the forecast model itself, but Post-hot is the creation of explanatory algorithms that are not related to the forecast model itself [36]. Obviously, there is a complete code implementation of the

SHAP algorithm on github [37] (https://github.com/slundberg/shap) and SHAP also has completed and mature theoretical support for cooperative games, then this paper chooses the SHAP [35] technique to explain the existing forecast models.

6.2. SHapley Additive exPlanation (SHAP)

SHAP is an explanation algorithm based on the Shapley values from game theory. The predicted values can be interpreted by assuming that each feature of the instance is a "player" in the game, and what SHAP does is to calculate how much each feature ("player") pays for the forecast value obtained by the forecast model.

The Explainable AI algorithm SHapley Additive exPlanation (SHAP) [35] used in this paper has the following connections and differences with existing Explainable AI algorithms such as Local Interpretable Model-agnostic Explanations (LIME) [38].

(1) Connection: SHAP and LIME [38] essentially construct simpler explainable models, using these models as approximations to complex models, both independent of the internal construction of the forecasting model, and can be used to explain any forecasting model using both. They explain the individual forecasts of any classifier in an explainable and reliable manner by learning a locally explainable model (e.g., a linear model) for each forecasting, and they estimate the feature attributes of the instances to determine the contribution of each feature to the model forecasting.

(2) Differences: The effect of the Shapley values in SHAP ensures a fair distribution of forecasts among features, while LIME does not ensure a fair distribution of forecasts among features. SHAP allows comparative explanation and does not require a comparison of forecasts with the average forecast of the entire dataset; users can compare it with subsets or even individual data points. LIME does not allow comparative explainable. SHAP is the only explanation method with solid theory. Methods such as LIME assume linearity in machine learning models locally, but have no theory as to why they should work.

6.3. Explanation results

Figure 5 shows the explanation of the results based on the Daily Phone Usage dataset using the SHAP algorithm, on the left side of Figure 5, on the one hand, the transition of the color from blue to red in the plot from the horizontal direction indicates an increase in the value of the features, and on the other hand, the top-to-bottom arrangement in the plot from the vertical direction indicates a decrease in the impact of the features on the model forecast results. In this paper, this shows that the feature that has the most influence on the forecast results is "Count". In the right part of Figure 5, the results of the impact between the different features are shown. In this paper, this section shows that the feature that has the greatest impact on the "Count" feature is the "second-order lag of App".

Figure 6 shows the explanation of the results based on the LiveLab dataset using the SHAP algorithm. In this paper, this shows that the feature that has the most influence on the forecast results is "genre", and the feature that has the most influence on the "genre" feature is "App" itself.

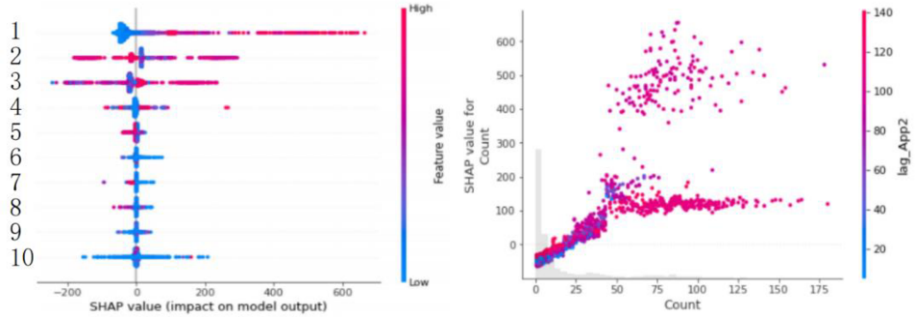

Figure 5. Daily Phone Usage. In the left section is global explanation; In the right section is inter-feature explanation. Left, 1: Count; 2: lag A2; 3: lag A1; 4: lag D1; 5: lag A4; 6: lag D2; 7: lag D3; 8: lag D8; 9: lag D4; 10: sum of 18 other features. The results in the left part of the image show that "Count" has the greatest impact on the forecast results. The results on the right side of the image show that "lag A2" has the most influence on "Count".

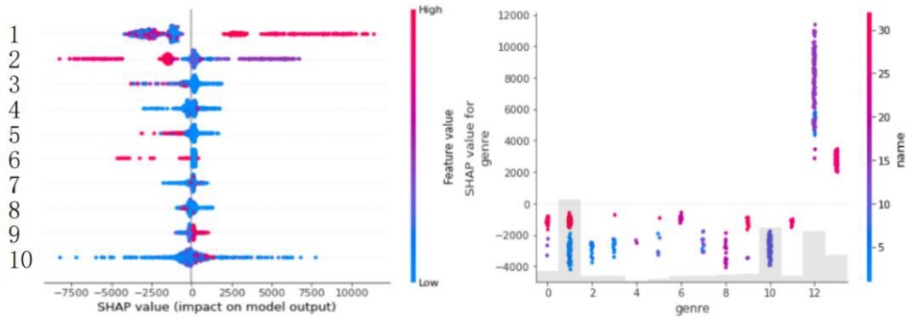

Figure 6. LiveLab dataset. In the left section is global explanation; In the right section is inter-feature explanation. Left, 1: genre; 2: App; 3: lag D14; 4: lag D1; 5: lag D20; 6: price; 7: lag D21; 8: lag D26; 9: month; 10: sum of 40 other features. The results in the left part of the image show that "genre" has the greatest impact on the forecast results. The results on the right side of the image show that "App" has the most influence on "genre".

7. Conclusions

This paper's not only compares different forecast models for time series problems based on the Daily Phone Usage and LiveLab datasets, but also from the perspective of XAI, thus making users more trustful of the models used, and helping them to find features that have a high and almost no impact on the forecast results. On the one hand based on the forecast results, we found that the LightGBM algorithm is currently the optimal forecast model for the App Usage time series forecast problem. On the other hand, the forecast results based on the LightGBM algorithm are explained using the SHAP algorithm, and we are able to obtain the contribution of each feature in the forecast results. Taking the work in this paper as an example, firstly, the duration of users' daily use of different apps in the Daily phone usage and LiveLab datasets is forecast, and secondly, the forecast results are explained. Based on the output of the SHAP algorithm, we find that in the time series forecast problem of App Usage type, the frequency of App usage, category and the lag of the target variable are important for the forecast of App duration. At the same time, the SHAP algorithm can also calculate the most relevant features (in

this paper, the second-order lag of App and App) to the most important features obtained (in this paper, Count and genre) and show the relationship between the most important features and the most relevant features to them. In summary, these are more helpful for users to know the inside of the black box model (LightGBM) and how it works.

In this work, we compare and explain the explanation results of ensemble models and neural network models in a time series forecasting problem, and the chosen XAI algorithm is performed for the global situation. In the future, we will forecast and explain traditional time series forecasting models, ensemble models and neural network models on different time series datasets, and focus on local explanations. It is hoped that the user's confidence in the results generated by the forecast model can be better improved.

Acknowledgments

This work was supported by the Ministry of Science and Higher Education of the Russian Federation by the Agreement № 075-15-2020-933 dated 13.11.2020 on the provision of a grant in the form of subsidies from the federal budget for the implementation of state support for the establishment and development of the world-class scientific center «Pavlov center» "Integrative physiology for medicine, high-tech healthcare, and~stress-resilience" technologies.

References

[1] Robinzonov N, Tutz G, Hothorn T. Boosting techniques for nonlinear time series models. AStA 2012; 1:99–122.
[2] Inoue A, Kilian L. How useful is bagging in forecasting economic time series? A case study of US consumer price inflation. J. Am. Stat. Assoc. 2008; 482: 511–522.
[3] Georg D. Neural networks for time series processing. Neural Netw. World 1996; 6: 447–468.
[4] Zhang Y, Xu F, Zou J, Petrosian OL, Krinkin KV. XAI evaluation: evaluating black-box model explanations for prediction. In Proceedings of the IEEE NeuroNT, Saint Petersburg, Russia. 2021 June; pp. 13–16.
[5] Tian Y, Zhou K, Pelleg D. What and how long: prediction of mobile app engagement. ACM Transactions on Information Systems (TOIS), 2021.
[6] Hossein F, Ratul M, Srikanth K, Dimitrios L, Ramesh G, Deborah E. Diversity in smartphone usage. In Proceedings of the 8th International Conference on Mobile Systems, Applications and Services. ACM, 2010; 179–194.
[7] Xu Q, Jeffrey E, Alexandre G, Mao ZQ, Pang J, Venkataraman S. Identifying diverse usage behaviors of smartphone apps. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference. ACM, 2011; 329–344.
[8] Li HR, Lu X, Liu XZ, Xie T, Bian KG, Lin XZ, Mei QZ, Feng F. Characterizing smartphone usage patterns from millions of android users. In Proceedings of the 2015 Internet Measurement Conference. ACM, 2015; 459–472.
[9] Matthias B, Brent H, Johannes S, Antonio K, Gernot B. 2011. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. ACM, 47–56.
[10] Tian Y, Zhou K, Pelleg D. What and how long: prediction of mobile app engagement, ACM Transactions on Information Systems, 2022.
[11] https://www.kaggle.com/johnwill225/daily-phone-usage/code
[12] http://yecl.org/livelab/traces.html
[13] Ke G, Meng Q. Lightgbm: A highly efficient gradient boosting decision tree. Neural Comput. 2017; 30: 3146–3154.
[14] Tin K. The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 1998; 20: 832–844.

[15] Schuster M, Paliwal K. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 1997; 45: 2673–2681.
[16] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997; 9: 1735–1780.
[17] Cho K, Van M. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv 2014, arXiv:1406.1078.
[18] Zhang Y, Ma R, Liu J, Liu X, Petrosian O, Krinkin K. Comparison and explanation of forecasting algorithms for energy time series. Mathematics. 2021 Нояб. 1;9(21). 2794. https://doi.org/10.3390/math9212794.
[19] Arrieta, A. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. Inf. Fusion 2020; 58: 82–115.
[20] Das A, Rad P. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. arXiv 2020, arXiv:2006.11371.
[21] Zou J, Xu F, Zhang Y. High-dimensional explainable AI for cancer detection. ResearchGate 2021. Available online: https://www.researchgate.net/publication/349715168_High_Dimensional_Explainable_AI_for_Cancer_Detection (accessed on 20 September 2021).
[22] Holzinger A, Biemann C, Pattichis CS, Kell DB. What do we need to build explainable AI systems for the medical domain? arXiv 2017, arXiv:1712.09923.
[23] Folke T, Yang SCH, Anderson S, Shafto P. Explainable AI for medical imaging: Explaining pneumothorax diagnoses with Bayesian teaching. arXiv 2021, arXiv:2106.04684.
[24] Zakharov V, Balykina Y, Petrosian O, Gao H. CBRR Model for predicting the dynamics of the COVID-19 epidemic in real time. Mathematics, 2020; 8(10), [1727]. https://doi.org/10.3390/math8101727.
[25] Hossain M, Muhammad G, Guizani N. Explainable AI and mass surveillance system-based healthcare framework to combat COVID-I9 like pandemics. IEEE Netw. 2020; 34: 126–132.
[26] Lin YC, Liang YJ, Chen MS, Chen XM. A comparative study on phenomenon and deep belief network models for hot deformation behavior of an Al-Zn-Mg-Cu alloy. Appl. Phys. Mater. Sci. Proc. 2017; 123: 68.
[27] Lin H, Dai Q, Zheng L, Hong H, Deng W, Wu F. Radial basis function artificial neural network able to accurately predict disinfection by-product levels in tap water: Taking haloacetic acids as a case study. Chemosphere 2020; 248: 125999.
[28] Deng Y, Zhou X, Shen J, Xiao G, Hong H, Lin H, Liao BQ. New methods based on Back Propagation (BP) and Radial Basis Function (RBF) Artificial Neural Networks (ANNs) for predicting the occurrence of haloketones in tap water. Sci. Total Environ. 2021; 772: 145534.
[29] Makridakis S, Spiliotis E, Assimakopoulos V. The M5 accuracy competition: Results, findings, and conclusions. Int. J. Forecast. 2020, in press.
[30] Zou J, Petrosian O. Explainable AI: using shapely value to explain complex anomaly detection ML-based systems. A. J. Tallon-Ballesteros, C-H. Chen (Ed.), Machine Learning and Artificial Intelligence: Proceedings of MLIS 2020 (. 152-164). (Frontiers in Artificial Intelligence and Applications; 332. IOS Press. https://doi.org/10.3233/FAIA200777.
[31] Liu X, Ai W, Li H, Tang J, Huang G, Feng F, Mei Q. Deriving user preferences of mobile apps from their management activities. ACM Trans. Inf. Syst. 2017; 35: 1–32. https://doi.org/10.1145/3015462.
[32] Yan T, Chu D, Ganesan D, Kansal A, Liu J. Fast app launching for mobile devices using predictive user context. ACM, 2012.
[33] Ke G, Meng Q. Lightgbm: A highly efficient gradient boosting decision tree. Neural Comput. 2017; 30: 3146–3154.
[34] Schuster M, Paliwal K. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 1997; 45: 2673–2681.
[35] Lundberg S, Lee S. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4768–4777.
[36] Das A, Rad P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. (2020).
[37] https://github.com/slundberg/shap.
[38] Ribeiro MT, Singh S, Guestrin C. Why Should I Trust You? explaining the predictions of any classifier// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. 2016.