Real-Time Power Performance Prediction in Tour de France

Yasuyuki Kataoka[1](✉) and Peter Gray[2]

[1] NTT Innovation Institute Inc., East Palo Alto, CA 94089, USA
kataoka.yasuyuki@ntti3.com
[2] Dimension Data Australia, Port Melbourne, VIC 3207, Australia
peter.gray@dimensiondata.com

Abstract. This paper introduces the real-time machine learning system to predict power performance of professional riders at *Tour de France*. In cycling races, it is crucial not only for athletes to understand their power output but for cycling fans to enjoy the power usage strategy too. However, it is difficult to obtain the power information from each rider due to its competitive sensitivity. This paper discusses a machine learning module that predicts power using the GPS data with the focus on feature design and latency issue. First, the proposed feature design method leverages both hand-crafted feature engineering using physics knowledge and automatic feature generation using autoencoder. Second, the various machine learning models are compared and analyzed with the latency constraints. As a result, our proposed method reduced prediction error by 56.79% compared to the conventional physics model and satisfied the latency requirement. Our module was used during the *Tour de France 2017* to indicate *an effort index* that was shared with fans via media.

Keywords: Machine learning · Recurrent neural network · Autoencoder · Real-time system · Spatiotemporal data · Sports analytics

1 Introduction

Measurement of muscle power in cycling has become an attractive tool for professional riders, coaches, and amateurs to improve the riding performance. For instance, it allows coaches to help monitor the training effectiveness when combined with heart rate measurement. It also allows riders to tactically determine energy use by analyzing other's muscle fatigue level. Moreover, it helps audiences to enjoy the competition by monitoring the rider's performance.

However, there are two issues in power data collection in cycling sports. First, power sensors tend to be expensive. Reducing cost of power meters means that they are becoming more accessible to competitive and even recreational amateur cyclists. Second, data on muscle usage is usually highly confidential, and it is not easily accessible. Although most professional teams attach power meters, the performance information is usually confidential within the team.

U. Brefeld et al. (Eds.): MLSA 2018, LNAI 11330, pp. 121–130, 2019.
https://doi.org/10.1007/978-3-030-17274-9_10

The purpose of this paper is to discuss the real-time machine learning system that predicts muscle power in cycling competition to enable people to get access to power performance information in *Tour de France*[1].

Conventionally, the power data is analyzed by the physics model which heavily relies on not data-driven but model-driven approach. This approach heavily depends on the physical constants, which tends to be less accurate. The challenge of the data-driven approach is collecting the labeled data with power information along with GPS data. Fortunately, in cooperation with one of the professional cycling teams and *Dimension Data's data analytics platform*[2] which collects GPS data from all riders, we obtained the labeled dataset for this purpose.

This paper proposes the data-driven power prediction that fuses the physics model with 1. feature design method and 2. real-time machine learning model analysis. First, the proposed feature design method leverages both hand-crafted feature engineering using physics knowledge and automatic feature generation using deep autoencoder. Beyond the previous studies of muscle fatigue analytics for cyclists, the feature inspired by deep learning enables trajectory patterns to be embedded into the model. This generated feature allows us to implicitly consider the rider's behavior such that the power use is loosened in the context of turning a sharp corner on a downhill slope. Second, the tree-based machine learning models and time-series deep learning models are compared regarding latency and error rate.

As a result, our ultimate model reduced prediction error by 56.79% compared to the conventional model-based model that depends on the prior knowledge of physics. Our Machine Learning module was used during the *Tour de France 2017* in a real-time manner to create *an effort index*, the power indicator, that was shared with fans via social media. Our proposed method can be used for amateur riders too who want to know the power performance but does not want to purchase a real power meter which tends to be very expensive.

2 Related Work

The performance of cycling riders has been studied across various academic fields [1]. Among them, this paper focuses on muscle fatigue and addresses the problem of predicting it by machine learning.

Fatigue Analytics. In the study of muscle fatigue in cycling, models considering various factors have been proposed. For example, one proposed model considers physiological, biomechanical, environmental, mechanical and psychological factors and integrates them into nonlinear complex system models [2]. While many researchers take the model-based approach [3,4], this paper focuses on a data-driven approach using the limited available dataset such as GPS.

[1] *Tour de France* is one of the three major European professional cycling stage races in road bicycle racing. https://en.wikipedia.org/wiki/Tour_de_France.

[2] *Dimension Data's data analytics platform*, https://www2.dimensiondata.com/tourdefrance/analytics-in-action.

The fatigue analytics is, in general, used for performance or safety improvement in the sports industry. One example is finding the relaxation place during a race [5]. This paper aims for fan engagement, which is also important in sports industry from the business perspective.

Machine Learning Application Using Cycling Data. The most famous machine learning applications using cycling GPS data is the transportation mode prediction: classifying user's activity to bicycle, car, train, walk, run and others [6]. One study reported that the generated feature from GPS trajectory using Deep Learning improves accuracy [7]. This is because the Deep Learning automatically captures important features which are difficult to be designed explicitly by hand-crafted feature engineering [8]. In our limited but best knowledge, there is no research report about muscle fatigue prediction using both hand-crafted feature and generated feature by deep learning in cycling sports.

3 Dataset

Input Data: GPS and Wind Sensor. *Dimension Data's data analytics platform* has a live GPS tracking system. This system provides the GPS tracking of position and speed for all riders at a 1 Hz frequency from the GPS sensors mounted under the bicycle saddle. This data is processed in real time, and enriched to calculate key metrics such as distance to finish, position in the race, time gaps, clustering of individual riders into groups, and the additional environmental data such as the current gradient of the road and the wind conditions.

Labeled Data: Power Sensor. Power is the measurement of how much force is being pushed through the pedals by the rider and is measured using dedicated sensors usually built into cranks, pedals or rear wheel hub. Most power meters connect wirelessly to the rider's bike computer allowing them to monitor their power output during a training session or race and manage their effort accordingly. In this project, a training dataset was obtained from one of the professional cycling teams in previous professional races. This dataset includes the data *Dimension Data's data analytics platform* provides as well as the power sensor data in accordance with the time stamp. The total count of the valid labeled data is 68849 after cleansing the dataset.

4 Methodology

This section mainly describes how the machine learning model is designed for power prediction with the focus on 1. feature design method and 2. real-time machine learning model analysis. In the feature engineering part, the hand-crafted feature is designed by mechanical factors using fundamental physics. Also, the generated feature by autoencoder is concatenated to the feature space. In the regression model, the various machine learning models are introduced with the arguments of advantages and latency perspectives.

4.1 Feature Engineering

Our proposed feature design is shown in Fig. 1: a hand-crafted feature inspired by physics knowledge and an automatically generated feature using deep learning.

Hand-Crafted Feature. Rider's power is physically determined by four factors:

1. *friction with the ground* denoted as $P_f = C_f v_b m g$, where C_f is the friction coefficient, v_b is bicycle velocity, m is total mass, and g is standard gravity.
2. *wind resistance* denoted as $P_w = 1/2 C_d A \rho (v_b - v_w)^2$, where C_d is drag coefficient, A is frontal area, ρ is air density, and v_w is wind velocity.
3. *kinetic energy* denoted as $P_k = \frac{m}{2\Delta T}(v_n^2 - v_p^2)$, where v_n is the velocity at $t = now$, and v_p is the previous velocity at $t = now - \Delta T$.
4. *potential energy* denoted as $P_p = mg\frac{\Delta h}{\Delta T}$, where ΔT is sampling time interval, and h is height variation within ΔT.

Each coefficient are surveyed in various reports[3]. However, we realized that the power calculated by these values is greatly different from the data from real sensors for our dataset. Therefore, the power prediction model is designed by machine learning, which identifies the desired coefficients to fit with our professional rider's dataset.

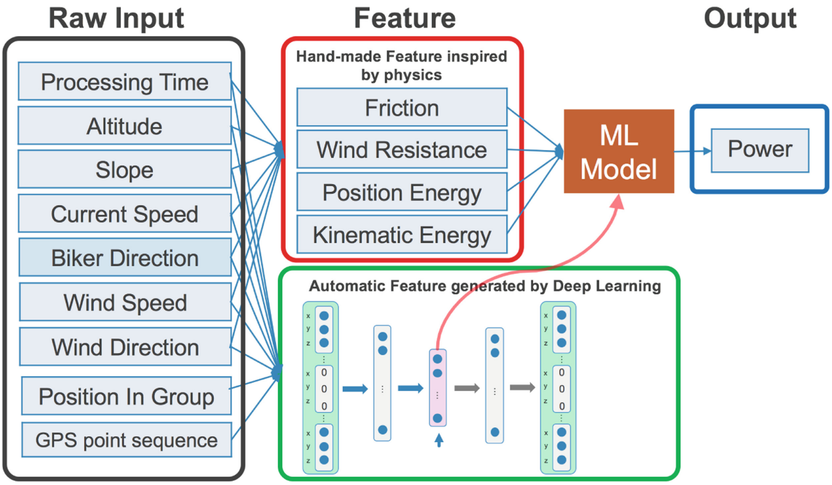


Fig. 1. Machine Learning pipeline: both hand-crafted feature by physics and automatic feature generated by Deep Learning is concatenated for Machine Learning model.

[3] One example of the physical constants https://www.cyclingpowerlab.com/CyclingAerodynamics.aspx.

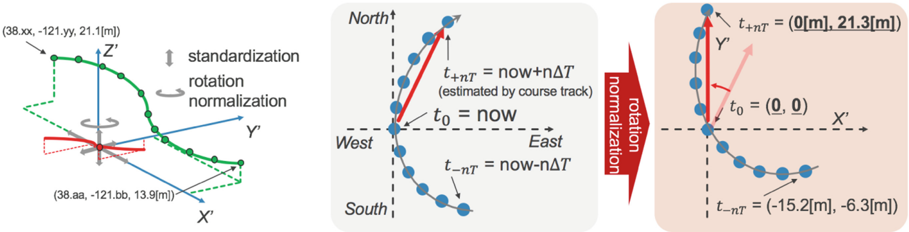

(38.xx, -121.yy, 21.1[m])
Z'
standardization
rotation normalization
Y'
(38.aa, -121.bb, 13.9[m])
X'
North
t_{+nT} = now+nΔT
(estimated by course track)
t_0 = now
West
East
t_{-nT} = now-nΔT
South
rotation normalization
t_{+nT} = (0[m], 21.3[m])
Y'
t_0 = (0, 0)
X'
t_{-nT} = (-15.2[m], -6.3[m])

Fig. 2. Data normalization for GPS trajectory: The main idea of this preprocess is to align the direction for all of the data.

Generated Feature by Deep Learning. It is assumed that the rider's power use is influenced by past and future trajectory of a rider. For example, it is observed that the pedal is stopped in the context of turning a sharp corner on a downhill slope. When fusing GPS trajectory data to the model, we found two issues: 1. direction diversity, 2. high dimensionality.

First, normalization of the trajectory direction is performed in the (x, y) plane. As shown in Fig. 2, the rotation transformation is applied so that the direction of the vector towards the position after $N\Delta T$ seconds corresponds to the positive direction of the y-axis. Here, the future GPS points are predicted based on the assumption that the current speed is maintained along with the course track. After applying rotation normalization, standardization is applied for each x, y, and z-axis.

Second, a dimensional reduction is applied by various autoencoders such as denoising autoencoder [9], deep autoencoder [10], and stacked deep autoencoder [11]. The input vector is the GPS points from $t - N\Delta T$ to $t + N\Delta T$, where each GPS point has x, y, and z value. This ends up a total $3(2T{+}1)$ dimension for input space, which tends to be sparse. These deep layers make it accurate to restore the input, meaning that implicit but powerful feature of the trajectory is extracted automatically. In this paper, the compressed feature vector by deep autoencoder is called the 'embedded trajectory feature'. This embedded trajectory feature is concatenated to the hand-crafted feature.

4.2 Regression Model

The challenge of the model choice is, in general, to optimize the model with respect to latency and error rate. The latency issue is critical in this real-time power prediction application, because it must predict power for each of 198 riders within one second. In the case of simple scenario by one machine, it is necessary to complete one prediction approximately in 5 ms. Within this 5 ms, the following process needs to be completed: subscribe incoming data, compute feature, run inference, and send outcome to the database. Although the distributed computing can solve this challenge, we set the latency requirement to 2 ms in model optimization task.

Tree-Based Models. Random Forest [12] and XGBoost [13] are considered as part of the regression model candidates. The advantage of the decision tree type model is that the number of trees in the model can easily be adjusted. This parameter affects the inference latency. Plus, the tree-based models have a chance to outperform the deep learning models when data is not sufficiently adequate. In addition to it, the tree-based model is explanatory to analyze the cause of the muscle fatigue. The hyperparameters are tuned by grid search through several experiments except for the number of trees.

Time-Series Deep Learning Models (Recurrent Neural Net). Stacked Long Short-Term Memory (LSTM) [14] and Gated Recurrent Units (GRU) [15] are considered as part of the regression model candidates from Recurrent Neural Net (RNN) models. The advantage of RNN is that predictive performance may outperform other models by extracting effective features over time-series information. After several experiments, some hyperparameters are fixed, e.g., the number of the past time-series data = 10, dropout ratio = 0.4. In this paper, the number of the layer numbers is treated as hyperparameter.

5 Result

First, this section quantitatively evaluates the accuracy of the power prediction regarding feature engineering, embedded trajectory feature, and regression models. Moreover, this section qualitatively evaluates the impact of the use of this machine learning model on fan engagement at the *Tour de France 2017*.

In the evaluation, *stratified* 5-fold cross validation is applied, because the dataset is imbalanced data. The metrics for the evaluation is mean absolute error (MAE), which computes the absolute value between the predicted value and the ground truth.

5.1 Evaluation on Feature Engineering

The purpose of this section is to analyze the effect of feature engineering by both hand-crafted features inspired by physics and generated feature by trajectory embedding autoencoder. In this comparison analysis, the following four different model types are considered:

1. **M1**: Physic-based Model (baseline)
   M1 is the conventional power model, the sum of four power factors $P = P_f + P_w + P_k + P_p$, where coefficients are determined by other articles.
2. **M2**: Data-Driven Model without feature engineering
   M2 is a machine learning model without any additional feature engineering. This simply uses raw input described in Fig. 1.
3. **M3**: M2 + hand-crafted Feature
   In addition to M2, M3 considers the hand-crafted feature designed in Sect. 4.1.
4. **M4**: M3 + Embedded Trajectory Feature by denoising stacked autoencoder
   In addition to M3, M4 considers the embedded trajectory feature designed in Sect. 4.1. The parameters $N = 5$, dimension of autoencoder's layers = [33, 20, 10, 20, 33].

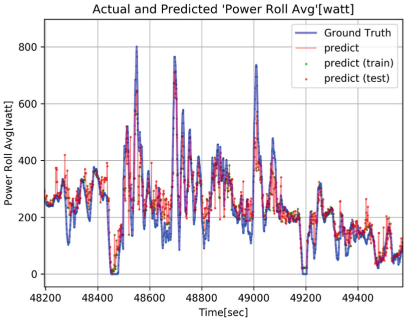


Fig. 3. Power prediction with the comparison to ground truth using model M4

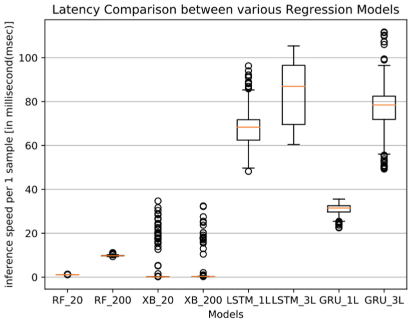


Fig. 4. Latency evaluation (RF = Random Forest, XB = XGBoost, _X= # of estimator or layers)

The result is shown in Table 1. Although the prediction may be difficult in high power range (>400 watt) or low power range (<100 watt) due to the imbalanced training dataset, Fig. 3 indicates our proposed model can work accurately in these challenging ranges too. Then, the comparative evaluation is shown in Table 1. Our proposed method, M4, outperforms the simple model-based model using only physics by 56.79% error reduction in MAE. Compared to M2, the simple data-driven model, our feature design improves machine learning model by 35.40% error reduction in this experiment. Thus, both hand-crafted feature and embedded trajectory feature should help to capture important factors to predict power use in cycling.

5.2 Evaluation on Regression Models

Inference Latency. In this experiment, the inference process was run on 198 samples data on GPU server (Tesla K80) whose status is idle except for this experiment. Note that the computation of 198 samples by matrix must not be run at once, because it needs to be done one by one in the real scenario. The results of the inference latency analysis are shown in Fig. 4 by box plot.

The best latency performance measured by median is XGBoost with the 200 trees. While the average performance of XGBoost outperforms other regression

Table 1. Performance comparison between four different feature engineering by MAE.

Model type	MAE (Train)	MAE (Test)	Error reduction to baseline
M1(Baseline)	-	139.17	-
M2	37.55	89.90	35.40%
M3	24.36	66.82	51.99%
M4	21.86	60.13	**56.79%**

models, the latency widely varies and takes +10 ms for some cases. This anomaly causes the negative impact on the backend system. One negative effect is missing values. The backend system terminates the inference process and then returns NaN for some cases. Contrary to XGBoost, Random Forest fairly performed stably. Random Forest with 20 tree trees satisfies the latency requirement, which is set to be 2 ms as described in Sect. 4.2.

The time-series deep learning models, LSTM and GRU did not satisfy our latency requirement. When the multiple layers are stacked, obviously the latency gets worse due to the additional computation.

Error Rate - MAE. The result is shown in Table 2. Among the tree-based models, XGBoost (n_est = 200) has shown the best performance with the satisfactory latency on average. However, it has the problem with the unstable latency. Thus, the Random Forest (n_est. = 20) is considered to be the best regression model in our practical situation. The time-series deep learning models, LSTM and GRU, turned out not to be better than tree-based models. It surely ends up underfitting. One famous way to avoid underfitting is to change the model to deeper structure. However, this approach did not work in this experiment. It may be because of the less labeled dataset to train the effective deep learning model. Even if it may get outperform tree-based models by adding more dataset available in future, the unstable and longer latency performance would not be solved.

5.3 Qualitative Analysis - Real Deployment in Tour de France 2017

Our proposed machine learning model was successfully deployed in Dimension Data's data analytics platform. In *Tour de France 2017*, the analytics platform team decided to use the output of power prediction as *an effort index* which indicates the power level from 1 to 10 for better visualization to fans and for respects to rider's semi-private data. This was the first trial to compute the effort

Table 2. Performance comparison between regression models by MAE and latency

Model type	MAE (Train)	MAE (Test)	Average latency [ms]
Random Forest (n_est. = 20)	24.35	**66.81**	**1.09**
Random Forest (n_est. = 200)	23.48	**66.02**	9.76
XGBoost (n_est = 20)	32.02	70.72	2.94
XGBoost (n_est = 200)	0.37	**63.97**	**1.07**
LSTM (1 Layer)	289.18	296.01	67.79
LSTM (3 Layer)	280.49	289.21	84.22
GRU (1 Layer)	289.78	290.78	30.72
GRU (3 Layer)	280.49	289.22	77.02

index in the history of *Tour de France* or any other cycling competition in our best knowledge.

Figure 5 shows one of the real-time visualization tools using the power prediction. This tool enables a user to compare the performance of two different groups at specific time range: e.g., Peloton vs Front Group in the past 5 min. This can visualize how enthusiastically peloton saves energy during a race or tries to catch up the front group.

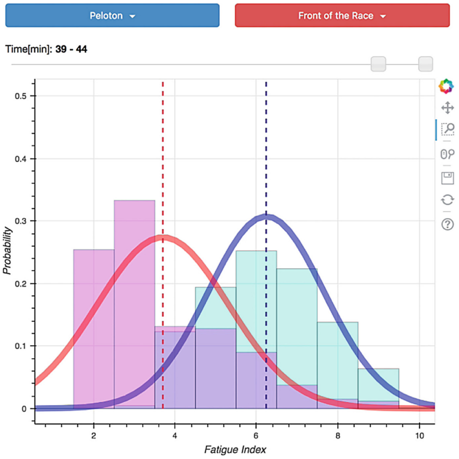
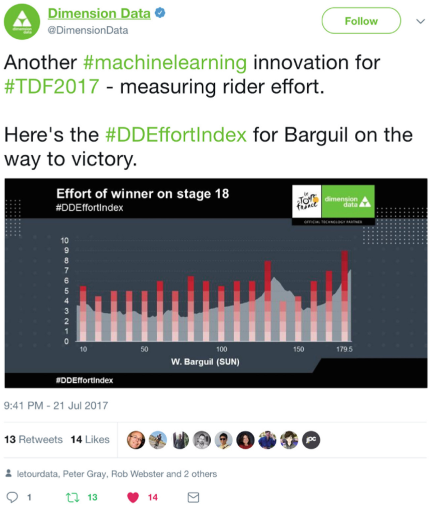


Fig. 5. Real-time visualization of power distribution for two group: x-axis is fatigue index, y-axis is probability


Dimension Data
@DimensionData
Follow

Another #machinelearning innovation for #TDF2017 - measuring rider effort.

Here's the #DDEffortIndex for Barguil on the way to victory.

9:41 PM - 21 Jul 2017

13 Retweets 14 Likes

letourdata, Peter Gray, Rob Webster and 2 others

1 13 14

Fig. 6. Example of social media exposure: visualization of the winner's performance in accordance with the terrain variation

Figure 6 shows the social media exposures of our technology: one tweet by Dimension Data that describes how the winner on stage 18 expends energy in accordance with terrain variation of the course. This graph indicates how the winner saved at downhill before the final uphill and used the peak effort at the end of the race.

6 Conclusion

This paper presented a machine learning application of power prediction used in *Tour de France 2017*. The characteristic approach of this paper is the feature design combined with both hand-crafted feature based on physics and generated features based on deep autoencoder using GPS trajectory. As a result, the error (MAE) rate is reduced by 56.79% compared to the physical model, and by 21.39% compared to the basic machine learning model. Moreover, several regression models are investigated regarding error rate and latency. This power prediction application contributes to fan engagement in cycling sports, as evidenced by social media. In the future, we plan to gather amateur riders' datasets for further sensorless power prediction products at a lower price than the power meters, which are often unaffordable for the ordinary consumer.

References

1. Castronovo, A.M., Conforto, S., Schmid, M., Bibbo, D., D'Alessio, T.: How to assess performance in cycling: the multivariate nature of influencing factors and related indicators. Front. Physiol. **4**, 116 (2013)
2. Abbiss, C.R., Laursen, P.B.: Models to explain fatigue during prolonged endurance cycling. Sports Med. **35**(10), 865–898 (2005)
3. Theurel, J., Crepin, M., Foissac, M., Temprado, J.J.: Effects of different pedalling techniques on muscle fatigue and mechanical efficiency during prolonged cycling. Scand. J. Med. Sci. Sports **22**(6), 714–721 (2012)
4. Martin, J.C., Milliken, D.L., Cobb, J.E., McFadden, K.L., Coggan, A.R.: Validation of a mathematical model for road cycling power. J. Appl. Biomech. **14**(3), 276–291 (1998)
5. Kataoka, Y., Junkins, D.: Mining muscle use data for fatigue reduction in IndyCar. In: 11th Annual MIT Sloan Sports Analytics Conference (2017). https://www.semanticscholar.org/paper/Mining-Muscle-Use-Data-for-Fatigue-Reduction-in-Kataoka-Junkins/abac602b26f7dd97655aa6443efde0ab9e90b186
6. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw GPS data for geographic applications on the web. In: Proceedings of the 17th International Conference on World Wide Web, pp. 247–256. ACM, April 2008
7. Endo, Y., Toda, H., Nishida, K., Ikedo, J.: Classifying spatial trajectories using representation learning. Int. J. Data Sci. Anal. **2**(3–4), 107–117 (2016)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
9. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103. ACM, July 2008
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
11. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(Dec), 3371–3408 (2010)
12. Liaw, A., Wiener, M.: Classification and regression by randomForest. R news **2**(3), 18–22 (2002)
13. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, August 2016
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Gated feedback recurrent neural networks. In: International Conference on Machine Learning, pp. 2067–2075, June 2015