Real-time Power 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 usage 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 desire to enjoy this data too. For example, how much riders tactically save energy, which group uses more power, etc. Revealing these insights helps to entertain fans more. However, it is difficult to obtain the power information from each rider directly. Although most teams attach power meters on their bikes, the performance information is usually confidential within the team. In cooperation with one of the professional teams in cycling sports and Dimension Data's data analytics platform which collects GPS data from all riders, we deployed a machine learning module that predicts power using the GPS data. This paper discusses 1. feature design method and 2. real-time machine learning model analysis. First, the proposed feature design method leverages both hand-made 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 so that the system can predict the power of 198 riders per one second. Our module was used during the *Tour de France 2017* in a real-time manner to indicate *an effort index* that was shared with fans via media.

Keywords: Machine Learning, Recurrent Neural Network, Autoencoder, Real-time System, Spatiotemporal Data

1 Introduction

Understanding muscle fatigue has a significant effect on many sports. It is the key for victory to judge the timing to save or exert muscular strength particularly in Cycling and Speed skate(Short track, Mass start, Pursuit). Such muscle management is crucial for athletes and audiences in professional competitions. For athletes, knowing the performance of the opponent makes it possible to strategize their tactics against the other party's one proactively. For audiences, knowing the player's muscle fatigue in real time allows them to enjoy the sports more deeply.

Among many sports, this paper focuses on cycling. 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 effectiveness of training and set accurate training programs 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 predict whether Peloton, the main group of the race, will catch the front of the race.

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. As a result, there is increasing interest in understanding the power output of the professional cyclists. Second, data on muscle usage is usually highly confidential, and it is not easily accessible. It is difficult to directly obtain the power data from professional riders due to its competitive sensitivity. Although most professional teams attach power meters, the performance information is usually confidential within the team.

The purpose of this paper is to create the real-time muscle power prediction tool for cycling sports to enable people to get access to power data. This applied data science paper discusses the design process of our real-time machine learning system that predicts the power usage of professional riders at *Tour de France*[3].

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* [4] which collects GPS data from all riders, we obtained the labeled dataset for this purpose.

This paper, in general, discusses the data-driven approach that also fuses the physics knowledge with the focus on 1. feature design method and 2. real-time machine learning model analysis. First, the proposed feature design method leverages both hand-made feature engineering using physics knowledge and automatic feature generation using autoencoder. Beyond the previous studies of muscle fatigue analytics for cyclists, the feature inspired by deep learning enables trajectory patterns to be embedded to Machine Learning 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 various machine learning models are analyzed under the latency constraints. The tree-based model and time-series deep learning models are compared regarding latency and error rate.

[3] *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

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

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* 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. This section shows articles on muscle fatigue and machine learning application using cycling data.

2.1 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.

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] Contrary to the past studies, our ultimate aim of this paper is mainly for fan engagement, which is also important in sports industry from the business perspective.

2.2 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-made feature engineering.[8]

In our limited but best knowledge, there is no research report about muscle fatigue prediction using both hand-made feature and generated feature by deep learning. Moreover, this paper also reports the comparison between model-drive and data-driven approaches for muscle power prediction in cycling sports.

3 Dataset

This section describes the input data and the labeled data obtained by sensors.

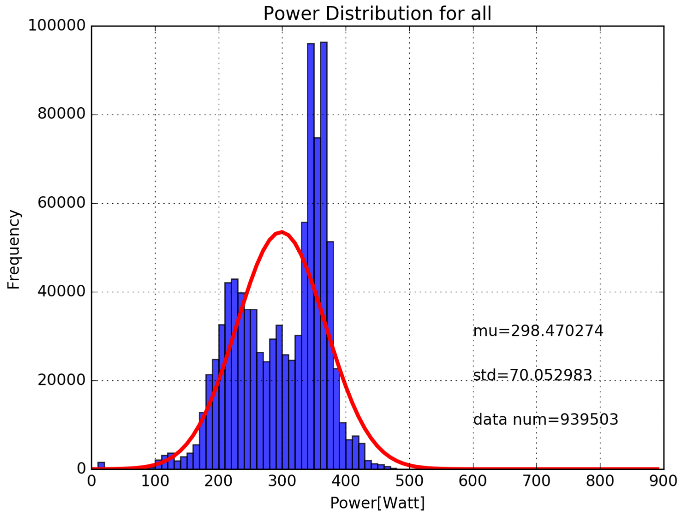


Fig. 1. Power Distribution: The imbalanced data makes hard to predict the high or low power range

3.1 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-second 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 data such as the current gradient of the road at that point, as well as the wind conditions at that location.

The detail of the backend system to collect these real-time data is described in Appendix A.

3.2 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 distribution of the power data is shown

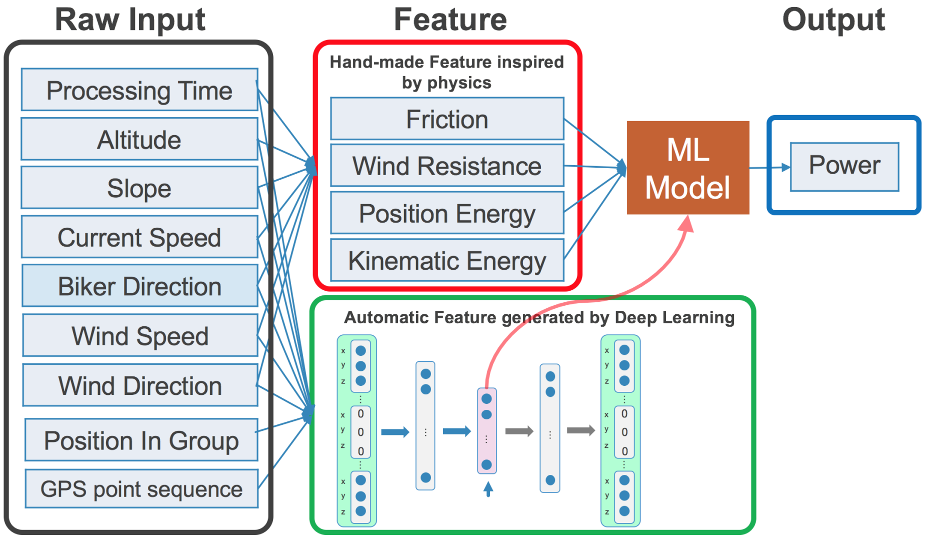


Fig. 2. Machine Learning pipeline: Raw input is obtained from *Dimension Data's data analytics platform*. Both hand-made feature by physics and automatic feature generated by Deep Learning is concatenated for Machine Learning model.

in Fig. 1. Since this dataset is rare for the machine learning project, this research project could be the first study on power prediction based on the GPS data of professional cyclists in our best knowledge.

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-made 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.2: a hand-made feature inspired by physics knowledge and an automatically generated feature using deep learning.

Hand-made Feature Physically, the power of the rider is determined by four factors:

1. "friction with the ground" denoted as $P_f = C_f v_b mg$, where C_f is the friction coefficient, v_b is the velocity of a bicycle, m is the mass of a rider and a bicycle, 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 cross-section 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 $t = now$[sec], and v_p is the previous velocity at $t = now - \Delta T$[sec].
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.

For each, theoretical values of these coefficients are known.[5] However, we realized that the power calculated by these theoretical values is greatly different from the data from real sensors. Therefore, the power prediction model is designed by machine learning, which identifies the desired coefficients to fit with real sensor values.

Since it is known that P_w is the most dominant factor to compute muscle power, the wind data around rider v_w is also considered for precise wind resistance estimation. Plus, it is worthwhile to remark that the position in the cluster is also considered as one of the features because it dramatically affects power use. For example, it is said that a rider in the second row uses approximately 50% of the power that a rider in the first row uses.

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. Therefore, consideration of trajectory pattern should improve power prediction accuracy. However, we found a problem in applying original GPS trajectory data to learn machine learning model. Since GPS trajectory data tends to be very sparse, there was little impact on the accuracy improvement. There are two reasons for this problem: 1. direction diversity, 2. high dimensionality.

First, normalization of the trajectory direction is performed in the (x, y) plane. As shown in Fig.3, the rotation transformation is applied so that the direction of the vector towards the position after $N\Delta$ 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 autoencoder. There are a variety of autoencoders such as denoising autoencoder[9], deep autoencoder[10], and stacked deep autoencoder[11]. They are all compared and analyzed when applied to this trajectory embedding problem. The input vector is the GPS points from

[5] one example of the physical constants
`http://socrates.berkeley.edu/~fajans/Teaching/CalcsWeb.htm`

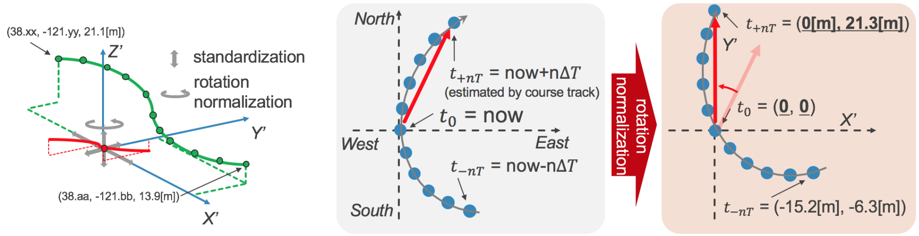

(38.xx, -121.yy, 21.1[m])
Z'
standardization
rotation
normalization
Y'
(38.aa, -121.bb, 13.9[m])
X'
North
t_{+nT} = now+nΔT
(estimated by course track)
t_0 = now
West
East
t_{-nT} = now-nΔT
South
rotation normalization
t_{+nT} = (0[m], 21.3[m])
Y'
t_0 = (0, 0)
X'
t_{-nT} = (-15.2[m], -6.3[m])

Fig. 3. Data normalization for GPS trajectory

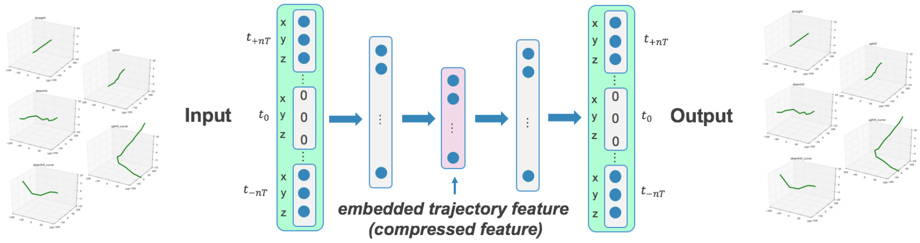


Fig. 4. embedded trajectory feature by autoencoder

$t - N\Delta T$ to $t + N\Delta T$, where each GPS point has x, y, and z value as shown in Fig.4. This ends up a total $3(2T{+}1)$ dimension for input space. In the deep autoencoder, weights are learned so that the output becomes same as the input with multiple intermediate layers sandwiched therebetween. 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-made feature. Then, this overall feature can be used as an input to regression models.

4.2 Regression Model

Several regression models are tested with the aim of the real-time scenario. The challenge of the model choice is, in general, to find the best model regarding latency and error rate.

In our real-time power prediction application, the latency is a critical issue. Our system 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 at 5 msec. Within this 5 msec, the following process needs to be completed: extract dataset from the database, run feature engineering, run inference, and send prediction outcome to the database. Although the distributed computing can solve this latency issue in the real scenario by the parallel computation, we consider the latency requirement 2.0[msec] in consideration of the limited project budget.

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 afffects 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 Feature Engineering

The purpose of this section is to analyze the effect of feature engineering by both hand-made 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: Model-Based Model (baseline)
   This is the conventional power model which only relies on only fundamental physics. This model is designed by the sum of four power factors, $P = P_f + P_w + P_k + P_p$, where coefficients and parameters are determined by our best knowledge with literature previously cited in this paper.

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	0.0%
M2	37.55	89.90	35.40%
M3	24.36	66.82	51.99%
M4	21.86	60.13	56.79%

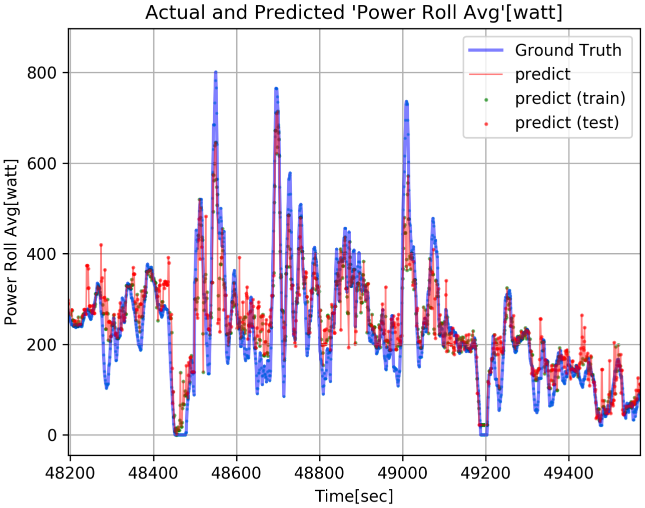


Fig. 5. An example of power prediction with the comparison to ground truth

2. M2: Data-Driven Model without Feature Engineering
   This is a data-driven model without any additional feature engineering. This machine learning model simply uses raw input that is obtained from *Dimension Data's data analytics platform* described in 3.1.
3. M3: M2 + Hand-made Feature
   In addition to M2, the model M3 considers the hand-made feature designed in 4.1.
4. M4: M3 + Embedded Trajectory Feature by denoising stacked autoencoder
   In addition to M3, the model M4 considers the embedded trajectory feature designed in section 4.1.2. In this experiment, the parameters are set as N=5 and 'the dimensions of layers' = [33, 20, 10, 20, 33] from input to output. The autoencoder type is chosen to be the denoising stacked autoencoder.

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. 5 indicates our proposed model can work accurately

Table 2. Performance comparison between autoencoder Model

Model Type	Parameter	Error Rate to AE
autoencoder	[33, 10, 33]	100.0%
Denoising autoencoder	[33, 10, 33]	97.8%
Denoising Deep autoencoder	[33, 20, 10, 20, 33]	96.82%
Denoising Stacked (Deep) autoencoder	[33, 20, 10, 20, 33]	92.54%

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-made feature and embedded trajectory feature should help to capture important factors to predict power use in cycling.

Analysis of embedded trajectory feature by various autoencoder In this section, various autoencoder performances are compared here. In addition to the labeled data of power data, there are a larger amount of the trajectory path data available that is utilized in this experiment. On every autoencoder models, the space for the embedded trajectory feature is constrained to 10 dimensions. The results are summarized in Table. 2. First, the Denoising autoencoder improves MAE by 2.2% compared to the regular autoencoder. Thus, adding denoising effect seems to have the advantage to extract key feature. Next, While the deep autoencoder simply increases the layer, stacked autoencoder train the hidden layer one by one. This ends up improving MAE by 7.46% compared to the regular autoencoder. Therefore, the denoising Stacked autoencoder is used for the extracting trjectory embbed feature from the GPS sequence data.

5.2 Regression Model

The tree-based models and time-series deep learning models are analyzed compared regarding inference latency and error rate.

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. 6 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 models, the latency widely varies and takes +10[msec] 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

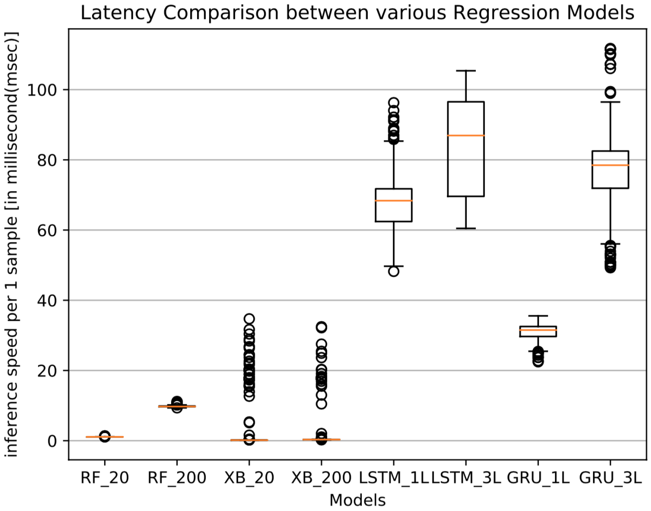


Fig. 6. Latency Comparison between different regression models (RF=Random Forest, XB=XGBoost, _20,_200= estimation by 20,200 trees, LSTM_1,_3=LSTM by 1,3 layers, GRU_1,_3=GRU with 1,3 layers)

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[msec] as described in 4.2.

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

Eror Rate - MAE In this section, the error rate is argued. The result is shown in Table. 3.

Originally, the time-series deep learning models, LSTM and GRU, were expected to perform much better than this results. In our practical research project, it surely ends up underfitting. One training example is shown in Fig.7

One famous way to avoid underfitting is to change the model to deeper structure. Thus, multiple layers such 3, 5, 7 layers were also tested. However, it did not get even close to 100 by MAE. As a conclusion in this project, the more labeled dataset needs to be collected to train the effective model. While it may get outperform tree-based models in future, the latency result indicates another challenge. The latency performance is less stable and takes a longer period for inference on average as shown in Fig. 6.

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

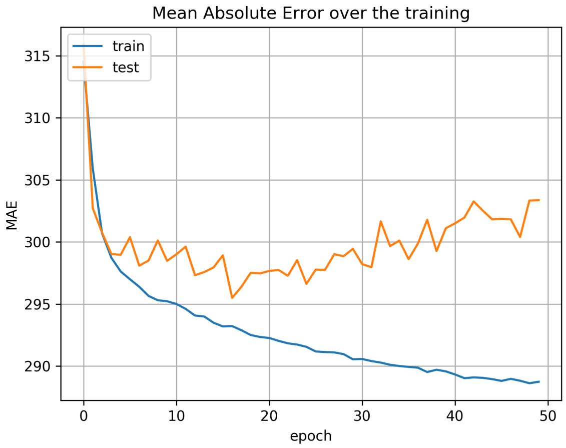


Fig. 7. Stacked LSTM: validation MAE performance over epochs:

Table 3. Performance Comparison between Regression Models by MAE and latency

Model Type	MAE (Train)	MAE (Test)	Average Latency[msec]
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

of the unstable latency issue. Thus, the Random Forest (n_est.=20) is considered to be the best regression model in our practical situation.

5.3 Qualitative Analysis

Real Deployment in Tour de France 2017 Our proposed machine learning model was successfully implemented in Dimension Data's data analytics platform. In *Tour de France 2017*, we converted the power[watt] to *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. In the practical data science application that predicts personal data and opens to the public, this process was very important.

As a result, *an effort index* is successfully predicted for 198 riders every one second in a real-time manner in *Tour de France 2017*. This was the first trial to

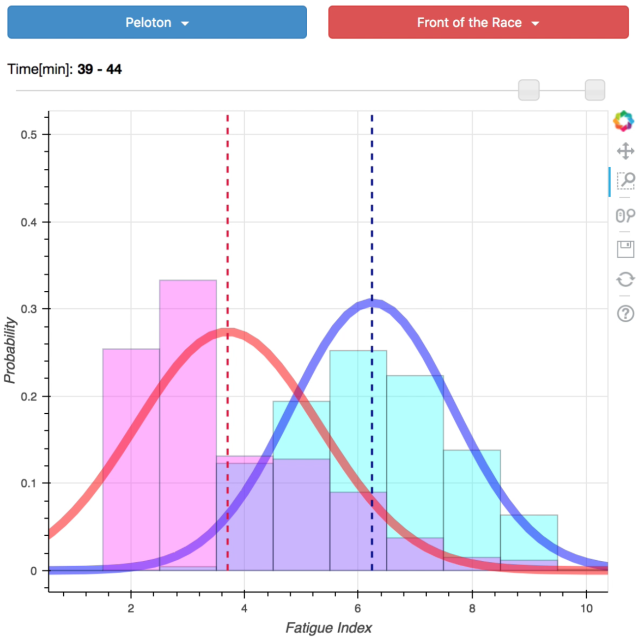

Time[min]: 39 - 44

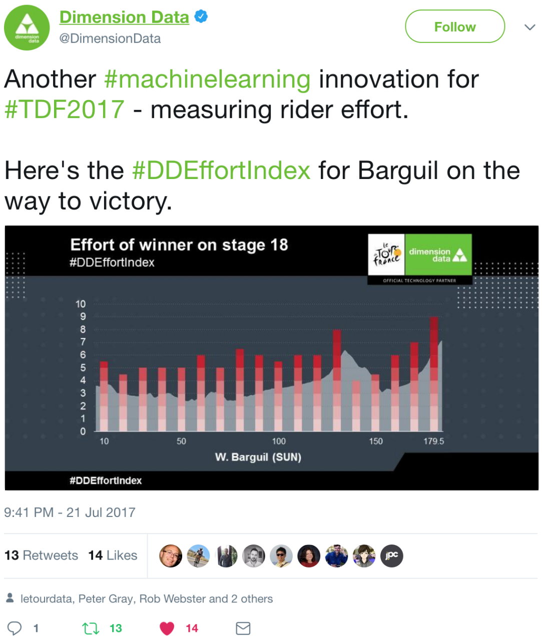
Fig. 8. Power Distribution for two group: x-axis means fatigue index, y-axis means probability, each bar chart means power distribution for two different rider groups.

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

Fig. 8 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 minutes. This can visualize how enthusiastically peloton saves energy during a race or tries to catch up the front group. In future, this predicted value can be utilized as a significantly important feature in order to predict if the catch happens by Peloton or not. Apparently, the accumulated muscle fatigue is an important feature for this prediction task.

Fig. 9 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.

Evaluating the impact on Fan Engagement The inclusion of machine learning based insights along with other innovations contributed towards over 20% year on year growth in social media engagement with fans, with data and insights from this program being referenced regularly across television broadcast, print



Dimension Data
@DimensionData

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. 9. Example of Social Media Exposure: Visualization of the winner's performance in accordance with the terrain variation

and digital media, and social media. This is a strong indication of the audience appetite for enhanced data and analytics in professional cycling.

Our proposed methodology and the deployment of it to *Tour de France 2017* were acknowledged by key business leaders involved in the sport.

"The introduction of Machine Learning into the live race analytics at the Tour de France has enhanced our ability to create engaging content for fans and commentators, and metrics such as the effort index allow us to better explain the race tactics." - Scott Gibson, Group Executive - Digital Practice, Dimension Data

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-made feature based on physics and generated features based on denoising stacked autoencoder. Considering the GPS trajectories by Deep Learning, it implicitly considers the factors of rider's intuitive judgment such as the tendency to loosen the force in the curve of the descending slope. 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. In our applied data science project with the limited dataset, it is concluded that Random Forest is the best performing regression model. This power prediction application contributes to fan engagement of cycling sports, as evidenced by the increased social and digital engagement by both fans and the cycling press. In the future, we are planning to gather amateur riders' datasets for further deep learning analytics and sensorless power prediction products at a lower price than the power meters, which are often unaffordable for the ordinary consumer.

References

1. Castronovo, Anna Margherita, et al. "How to assess performance in cycling: the multivariate nature of influencing factors and related indicators." Frontiers in physiology 4 (2013): 116.
2. Abbiss, Chris R., and Paul B. Laursen. "Models to explain fatigue during prolonged endurance cycling." Sports medicine 35.10 (2005): 865-898.
3. Theurel, J., et al. "Effects of different pedalling techniques on muscle fatigue and mechanical efficiency during prolonged cycling." Scandinavian journal of medicine & science in sports 22.6 (2012): 714-721.
4. Martin, James C., et al. "Validation of a mathematical model for road cycling power." Journal of applied biomechanics 14.3 (1998): 276-291.
5. Kataoka, Yasuyuki, and Douglas Junkins. "Mining Muscle Use Data for Fatigue Reduction in IndyCar." 11th Annual MIT Sloan Sports Analytics Conference
6. Zheng, Yu, et al. "Learning transportation mode from raw gps data for geographic applications on the web." Proceedings of the 17th international conference on World Wide Web. ACM, 2008.
7. Endo, Yuki, et al. "Classifying spatial trajectories using representation learning." International Journal of Data Science and Analytics 2.3-4 (2016): 107-117.
8. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
9. Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning. ACM, 2008.
10. Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.
11. Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research 11.Dec (2010): 3371-3408.
12. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.
13. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
14. Hochreiter, Sepp, and J ü rgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
15. Chung, Junyoung, et al. "Gated feedback recurrent neural networks." International Conference on Machine Learning. 2015.

A Data Collection System Backend Architecture

Over the past three years, Dimension Data has worked with Amaury Sport Ogranization(A.S.O.), the owners of the Tour De France, to implement a live GPS tracking and analytics solution. This allows the GPS tracking of position and speed for all riders in the *Tour de France* at a 1-second frequency. The GPS sensor is mounted under the bicycle saddles for all riders during the competition. The overall architecture of this data analytics platform is shown in 10

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 data such as the current gradient of the road at that point, as well as the wind conditions at that location. This enrichment is undertaken using an in-memory streaming analytics platform, which then pushes live data into the television graphics system and a web API that is used to service digital platforms.

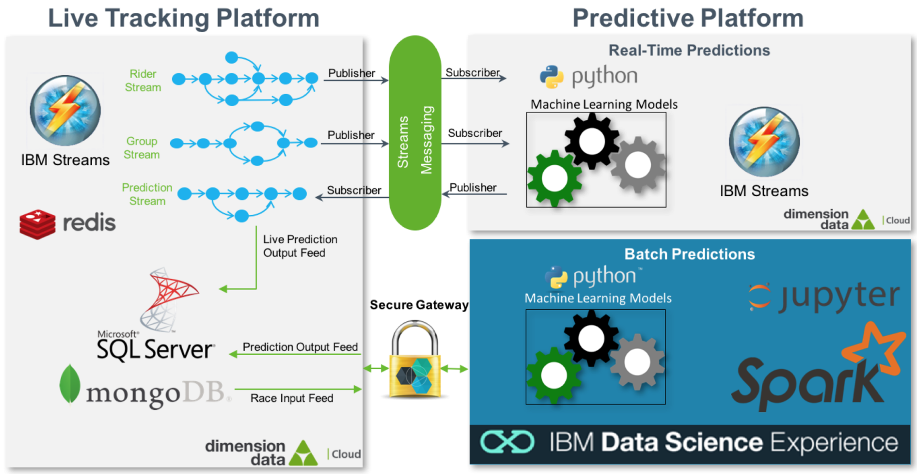


Fig. 10. Real-time data streaming architecture of data collection platform that was used in *Tour de France 2017*