Leverage the Predictive Power Score of Lifelog Data's Attributes to Predict the Expected Athlete Performance

Anh-Vu Mai-Nguyen[1], Van-Luon Tran[1], Minh-Son Dao[⋆2], and Koji Zettsu[2]

[1] University of Science, VNU-HCMC, Vietnam
{1612904,1612362}@student.hcmus.edu.vn
[2] National Institute of Information and Communications Technology, Japan
{dao,zettsu}@nict.go.jp

Abstract. Doing exercises regularly and scientifically can bring better health for people and improve sports performance for athletes. Many investigations have carried on to build necessary models that utilize data collected from people to predict sports performance. Nevertheless, most researches use data directly related to the moment when exercises happen to build prediction models, even though more data related to people's daily activities also impact on sports performance. Thanks to lifelogging, we can now have more data reports not only on people's exercises but also on people's daily activities (both mental and physical aspects). Unfortunately, finding out which data attributes correlate to the changing of sports performance and leveraging these correlated attributes to build a precise prediction model is not a trivial problem. In this paper, we introduce the solution that utilized the predictive power score of lifelog data attributes during a long time to predict an athlete who trained for a sporting event. We evaluate our solution using the dataset and evaluation metric given by imageCLEFlifelog task 2: sports performance lifelog.

1 Introduction

Having long-term and regular exercise can bring many benefits to people's health and daily activities [1]. This argument is also valid for athletes who want to improve their sports performance [2]. Hence, if we can monitor training periods and other factors that could impact both the mental and physical aspects of a person, we can predict that person's sports performance.

In [3], the authors used SVM with particle swarm optimization to tackle the problem of athlete performance prediction. They used that dataset of 500 records of 100 meters running for training and testing their model. The comparison between the proposed method and the linear regression and neural networks

⋆ corresponding author

confirmed the better performance. This method's vital contribution is to apply chaotic theory on the historical data of the athletes to discover the hidden rules towards improving the productivity of prediction models. Unfortunately, the description of the dataset was not clarified enough. Hence, the re-producing of this method could be difficult.

In [4], the authors introduced the problem of post hoc analysis (i.e., a process to analyze the athlete's performance after the performed sports activity) using artificial intelligence. The historical data of the athlete's performance was analyzed, mostly using heart rate, to automatically make up the time deficit on the running competition by using a differential evolution (DE) algorithm. Unfortunately, the model did not concern the environmental conditions (e.g., weather, altitude, topography, humidity) that logically influences the athlete's performance.

In [5], the authors utilized computational intelligence and visualization to analyze heart rate and GPS data to better understand cycling and fitness physical activities. The authors discovered the positive correlation between heart rate and altitude gradient, the negative correlation between heart rate and speed, and the correlation between The mean heart rate change delay and changes in the altitude gradients associated with cycling up and down.

In [6], the authors introduced a new algorithm based on the behavior of micro-bats for association rule mining (BatMiner) to explore the athlete's characteristics that have the most significant positive impact on performance. Based on the results, an athlete can practice alone without the appearance of his/her coach. There were two kinds of data sources: (1) activity datasets obtained by sports trackers or other wearable mobile devices, and (2) subjective information about psycho-physical characteristics of the athlete during training sessions through conversations between the athlete and the trainer. The former was a duration of the training session, distance of the training session, average heart-rates, and calorie consumption. The latter were external factors (e.g., weather conditions, the type of training session), sports nutrition, rest time (e.g., afternoon, night), and overall health (e.g., fatigue, cramping, welfare). The data were captured from TCX files of a professional, 32-years-old, male cyclist with many years of experience, who underwent training sessions during the first half of 2014, and he prefers to remain anonymous. The result was that the BatMiner algorithm is slightly better than the HBCS-ARM (a family of SI-based algorithms) for all measures when comparing the best ten association rules discovered by both algorithms obtained in 25 independent runs.

The imageCLEFlifelog 2020 (task sports performance lifelog) [7] provides such a lifelog dataset and raises the exciting challenge to predict the change in running time and weight of a person between the beginning and end of training periods. The challenge here, in our opinion, is to find out which data attributes correlate to the changing of sports performance, and leveraging these correlated attributes to build a precise prediction model is not a trivial problem. In this paper, we introduce the solution that utilized the predictive power score of lifelog

data attributes during a long time to predict an athlete who trained for a sporting event.

The paper is organized as follows: section 2 introduces our methodology, section 3 reports our results and discussions, and section 4 concludes our contribution.

2 Methodology

In this section, we describe the dataset given by the organizers and our approach used to predict the athlete's expected performance.

2.1 Data Prepossessing

We have to process a multi-modal dataset collected from Fitbit versa 2, PMSYS, and Google form [8]. This dataset consists of various interval observed attributes such as minute-observed attributes (e.g., calories burned, heart rate, step), day-observed attributes (e.g., weight, meals), and event-observed attributes (e.g., sleep, activity).

First, we synchronized different intervals of time into one interval of time for further processing. To do that, we summarized minute-observed attributes following the time length of event-observed attributes and day-observed attributes. For instance, we calculated the total calories burned for each activity and total calories consumed per day.

Then, we normalized the unit of attributes sharing the same meaning to one basic unit. For example, we converted the 'duration' attribute from millisecond to second and active action attributes (e.g., lightly active, very active) from minutes to seconds. Besides, we used one-hot encoding with category attributes such as 'meal.'

Next, we dealt with missing values by filling them with previous value if they are not in the first order of time and replacing the first time-order values with their following values. Especially with non-value attributes, we filled with the average value of these attributes from all participants. We also detected and deleted outliers to decrease their impact on the final results.

Finally, we generated a new attribute representing time per kilometer of running activity in exercise data using speed attributes.

2.2 Feature Selection

As mentioned in the previous section, we use a predictive power score (PPS) to find a correlation coefficient among dataset's attributes. We are utilizing PPS to imply the (hidden) correlations among attributes so that the following things can be detected and summarized (1) non-linear relationships, (2) asymmetric correlation, and (3) predictive value among categorical variables and nominal data. Clearly, the dataset we deal with has all three characteristics mentioned above. Another reason is that the PPS has some advantages over correlation

Table 1: Chosen attributes for time prediction models

		LR	MLP	First CNN	Second CNN	Third CNN	Vanilla LSTM	Stack LSTM	GRU	Condition LSTM	Data Type
Multi-attribute Models	active_duration	x	x	x	x	x	x	x	x	x	time-series
	average_heart_rate	x	x	x	x	x	x	x	x	x	auxiliary
	calories	x	x	x	x	x	x	x	x	x	time-series
	distance	x	x	x	x	x	x	x	x	x	time-series
	elevantion_gain	x	x	x	x	x	x	x	x	x	time-series
	steps	x	x	x	x	x	x	x	x	x	time-series
	time_in_cardio_heart_rate_zone	x	x	x	x	x	x	x	x	x	auxiliary
	time_in_fat_burn_heart_rate_zone	x	x	x	x	x	x	x	x	x	auxiliary
	time_in_peak_heart_rate_zone	x	x	x	x	x	x	x	x	x	auxiliary
	very_active_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	id_parti									x	auxiliary
	time_per_km	x	x	x	x	x	x	x	x	x	time-series
	age									x	auxiliary
	height									x	auxiliary
	gender									x	auxiliary
One-attribute Models	id_parti								x	x	auxiliary
	time_per_km	x	x	x	x	x	x	x	x	x	time-series
	age									x	auxiliary
	height									x	auxiliary
	gender									x	auxiliary

for finding predictive patterns in the data. That leads to the cue to select the suitable features for our prediction models.

After building the PPS matrix from cleaned data, we remove all attributes that do not have any relation to running time and weight. Besides, we also ignore attributes that can be predicted by another attribute to reduce the complexity of feature sets. Moreover, we keep pairs of mutually predictive attributes to maintain a strong correlation. Finally, we come up with a set of attributes denoted in Table 1, 2, and 3

Table 2: Chosen attributes for weight prediction models (part 1)

		LR	MLP	First CNN	Second CNN	Third CNN	Vanilla LSTM	Stack LSTM	GRU	Condition LSTM	Data Type
Multi-attribute Models	weight	x	x	x	x	x	x	x	x	x	time-series
	glasses	x	x	x	x	x	x	x	x	x	auxiliary
	very_active	x	x	x	x	x	x	x	x	x	auxiliary
	lightly_active	x	x	x	x	x	x	x	x	x	auxiliary
	sedentary	x	x	x	x	x	x	x	x	x	auxiliary
	calories	x	x	x	x	x	x	x	x	x	auxiliary
	distance	x	x	x	x	x	x	x	x	x	auxiliary
	steps	x	x	x	x	x	x	x	x	x	auxiliary
	heart_rate	x	x	x	x	x	x	x	x	x	auxiliary
	fatigue	x	x	x	x	x	x	x	x	x	auxiliary
	mood	x	x	x	x	x	x	x	x	x	auxiliary
	readiness	x	x	x	x	x	x	x	x	x	auxiliary
	sleep_duration_h	x	x	x	x	x	x	x	x	x	auxiliary
	sleep_quality	x	x	x	x	x	x	x	x	x	auxiliary
	soreness	x	x	x	x	x	x	x	x	x	auxiliary
	stress	x	x	x	x	x	x	x	x	x	auxiliary
	breakfast	x	x	x	x	x	x	x	x	x	auxiliary
	lunch	x	x	x	x	x	x	x	x	x	auxiliary
	dinner	x	x	x	x	x	x	x	x	x	auxiliary
	evening	x	x	x	x	x	x	x	x	x	auxiliary
	efficiency	x	x	x	x	x	x	x	x	x	auxiliary
	end_time	x	x	x	x	x	x	x	x	x	auxiliary
	overall_score	x	x	x	x	x	x	x	x	x	auxiliary
	composition_score	x	x	x	x	x	x	x	x	x	auxiliary
	revitalization_score	x	x	x	x	x	x	x	x	x	auxiliary
	duration_score	x	x	x	x	x	x	x	x	x	auxiliary
	resting_heart_rate	x	x	x	x	x	x	x	x	x	auxiliary
	restlessness	x	x	x	x	x	x	x	x	x	auxiliary
	deep_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	deep_thirty_day_avg_seconds	x	x	x	x	x	x	x	x	x	auxiliary

2.3 Prediction Models

We consider data attributes that reported exercise activities (e.g., running, jogging) as time-series data because athletes are in the training time prepared for sports events. It means that their exercises repeat regularly and seasonally. We also consider the rest of the data attributes as auxiliary data such as age, gender, and height.

As discussed in previous sections, two main directions of research in an athlete's performance prediction topic have investigated. The first one considers only

Table 3: Chosen attributes for weight prediction models (part 2)

		LR	MLP	First CNN	Second CNN	Third CNN	Vanilla LSTM	Stack LSTM	GRU	Condition LSTM	Data Type
Multi-attribute Models	light_thirty_day_avg_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	rem_count	x	x	x	x	x	x	x	x	x	auxiliary
	rem_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	rem_thirty_day_avg_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	wake_count	x	x	x	x	x	x	x	x	x	auxiliary
	wake_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	wake_thirty_day_avg_seconds	x	x	x	x	x	x	x	x	x	auxiliary
	id_parti	x	x	x	x	x	x	x	x	x	auxiliary
	age									x	auxiliary
	height									x	auxiliary
	gender									x	auxiliary
One-attribute Models	weight	x	x	x	x	x	x	x	x	x	time-series
	id_parti									x	auxiliary
	age									x	auxiliary
	height									x	auxiliary
	gender									x	auxiliary

data collected during the exercise and ignore other data even these data probably correlate to the athlete's performance. The second one concern all correlated and related data. Followed these directions, we design two types of models. The first one, called the univariate time-series model, utilizes one attribute. The second one uses a set of attributes.

First, we build two baseline methods: (1) a simple linear regression model and (2) a multilayer perception model using the ReLU activation function.
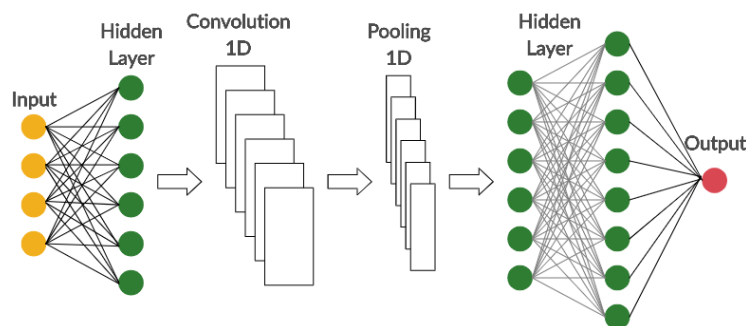
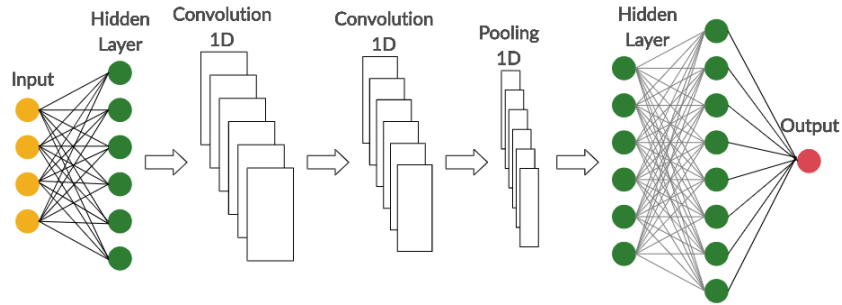

Fig. 1: First CNN model architecture
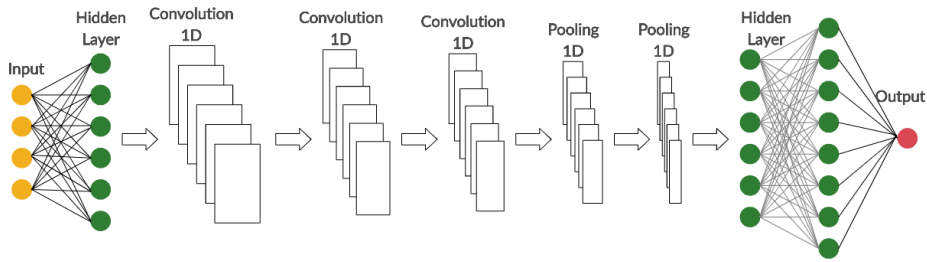
Fig. 2: Second CNN model architecture

Fig. 3: Third CNN model architecture

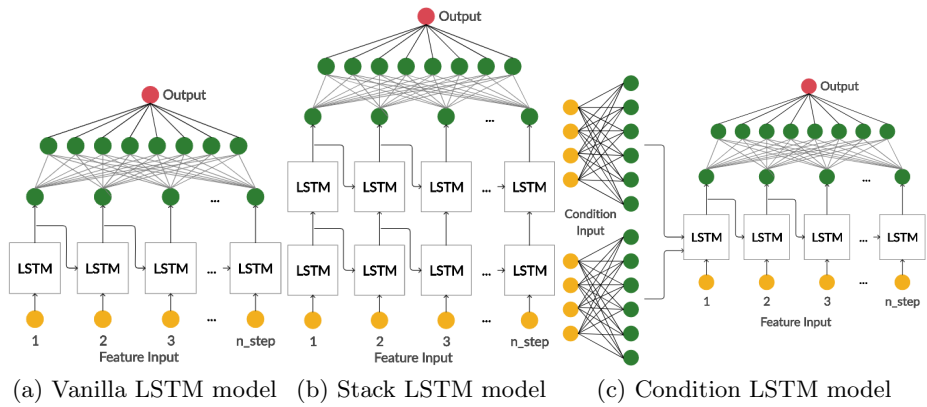(a) Vanilla LSTM model   (b) Stack LSTM model   (c) Condition LSTM model

Fig. 4: LSTM-like model architectures

Then, we build three CNN models: (1) the first one consists of one 1D-convolution and 1D-max-pooling layers (Fig. 1), (2) the second one contains

more 1D-convolution layers than the first (Fig. 2), and (3) the third one includes more 1D-convolution 1D and 1D-max pooling layers than the second (Fig. 3).

Next, we create three LSTM-like models: (1) the Vanilla LSTM model that has the number of units in the hidden layer equaled to the number of time steps (Fig. 4.a), (2) the Stack LSTM model (Fig. 4.b)., and (3) the conditional-LSTM model with initial condition attributes such as age, gender, and height (Fig. 4.c).
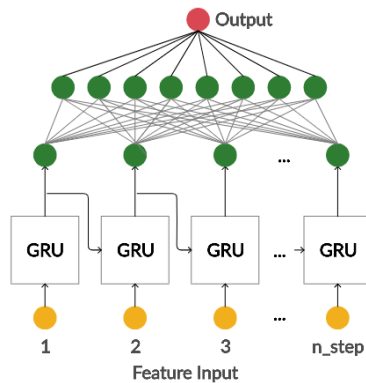
Finally, we build the GRU model (Fig. 5).

Fig. 5: GRU model architecture

3 Experimental Results

In this section, we describe how to utilize our models with selected data attributes to predict the change in running speed and weight since the beginning of the reporting period to the end of the reporting period.

We use the dataset and evaluation metric provided by the imageCLEFlifelog organizers [9]. The readers could refer to the paper written by the organizers for more details [7]. For short, the evaluation metric is defined as follow: "*For the evaluation of the tasks the main ranking will be based on whether there is a correct positive or negative change (a point per correct) - and if there is a draw, the difference between the predicted and actual change will be evaluated and used to rank the task participants.*"[3].

The organizers define three subtasks: "*(1) predict the change in running speed given by the change in seconds used per km (kilometer speed) from the initial run to the run at the end of the reporting period., (2) predict the change in weight since the beginning of the reporting period to the end of the reporting period in kilos (1 decimal), and (3) predict the change in weight from the beginning of February to the end of the reporting period in kilos (1 decimal) using the images.*"

[3] https://www.imageclef.org/2020/lifelog

Table 4: The ten best models for time prediction

Time Prediction Model	Validation MSE	Validation MAE	Train MSE	Train MAE
VanillaLSTMmodel_one_attribute_time-steps_5	3872.775	43.5962	3593.101	43.40623
VanillaLSTMmodel_one_attribute_time-steps_7	3922.615	47.3889	3554.712	42.87104
Condition_LSTM_one_attribute_time-steps_5	4023.349	43.10283	4874.171	51.20744
StackLSTMmodel_one_attribute_time-steps_5	4039.655	44.82555	4191.725	45.65004
StackLSTMmodel_one_attribute_time-steps_3	4044.094	46.14255	2586.746	35.38247
Second_CNN_model_one_attribute_time-steps_5	4046.895	44.3603	4435.651	48.17548
GRU_model_one_attribute_time-steps_7	4076.255	47.21735	4922.799	52.2997
Condition_LSTM_one_attribute_time-steps_7	4216.521	48.09542	4940	52.39553
First_CNN_model_one_attribute_time-steps_7	4263.523	48.82619	2918.934	39.79049
MLPmodel_one_attribute_time-steps_7	4265.8	46.76684	4693.186	50.9879

3.1 Predict the change in running time

We train different prediction models for predicting the change of running time, both for a univariate attribute (i.e., time) and a set of attributes. We use three different numbers of input time steps, which are 3, 5, and 7. Table 1 denotes which attributes are used for training which models.

After training these models, we evaluate them and select the ten best models with the lowest validation loss as the official models for this subtask. Table 4 shows information of these models during training and validating stages.

3.2 Predict the change in weight

Like the first subtask, we build two types of models that use one or a set of attributes. These models have the same architecture as the models of the first subtask. However, we use four different input time steps: 7, 14, 21, and 30.

After training these models, we evaluate them and select the ten best models with the lowest validation loss as the official models for this subtask. Table 5 shows information of these models during training and validating stages.

For subtask 3 "*Predict the change in weight from the beginning of February to the end of the reporting period in kilos (1 decimal) using the images*", we use the app, namely Calorie Mama[4], to approximately calculate the calories from meal/food images. Then, we add this attribute to the set of current attributes. Then, we run as subtask 2.

The table 6 shows our results evaluated by the organizers. We apply ten different models with the best accuracy filtered at the training stage, as described in Table 4 and 5, to run at the testing stage (i.e., the model's IDs expressed in the table are the same with the run's IDs denoted in table 6). As we can see in the table 6, our models cannot reach the optimal stage where both accuracy and the absolute difference can be optimal at the same time. For example, with subtask 1 (i.e., predict the change in running time), run 8 got the best accuracy (i.e., 1) while 10 received the smallest absolute difference (i.e., 96). With subtask 2 (i.e., predict the change in weight without image data), run 9 reached the best accuracy (i.e., 0.9), while 6 and 8 gained the smallest absolute difference (i.e.,

---

[4] https://www.caloriemama.ai/CalorieMama

Table 5: The ten best models for weight prediction

Weight Prediction Model	Validation MSE	Validation MAE	Train MSE	Train MAE
Condition_LSTM_weight_one_attribute_time-steps_14	0.204387	0.320392	0.350457	0.260382
LR_model_weight_one_attribute_time-steps_7	0.211679	0.318684	0.426128	0.305971
VanillaLSTMmodel_weight_one_attribute_time-steps_7	0.211884	0.326093	0.379312	0.285081
LR_model_weight_one_attribute_time-steps_14	0.215518	0.33098	0.448237	0.296486
Condition_LSTM_weight_one_attribute_time-steps_7	0.216421	0.318217	0.353498	0.25426
VanillaLSTMmodel_weight_one_attribute_time-steps_14	0.217808	0.341755	0.402532	0.285623
StackLSTMmodel_weight_one_attribute_time-steps_7	0.218527	0.337486	0.38896	0.288394
StackLSTMmodel_weight_one_attribute_time-steps_14	0.218925	0.330796	0.328408	0.267264
StackLSTMmodel_weight_one_attribute_time-steps_21	0.220059	0.346851	0.387996	0.269111
GRU_model_weight_one_attribute_time-steps_7	0.221908	0.323945	0.363954	0.24998

11). With subtask 3 (i.e., predict the change in weight with image data), the results look more stable than other subtasks; at run 8, the accuracy got the maximum (e.g., 1) while at run 10 the smallest absolute difference was received (i.e., 1). Nevertheless, we can accept one model that can balance between the accuracy and the absolute difference, according to the purpose of users.

The table 7 illustrates the results for three subtasks conducted by organizers, namely baseline. Regarding the first subtask, our result is far better than the baseline on both metrics. For instance, except run 5, the rest of our runs have a higher accuracy score than the best accuracy of baseline from 0.2 to 0.4 points, while run 10 and run 8 have absolute differences fewer 100 points than the best one of baseline. Besides, the baseline cannot precisely perform on both accuracy and absolute difference categories at the same time. Considering the second subtask, although our best run for absolute difference (run 1) has a slightly lower score than the baseline does, about 3 points, its accuracy is twice better than that of the baseline (0.4). When coming to the third subtasks, our result has double accuracy (run 8) and half absolute difference (run 10) compared to the baseline.

3.3 Discussions

After we gain an insight into the given data, we find that people subjectively provide plenty of information such as stress, fatigue, mood, and sleep score. It means that these data are likely to be irrelevant to what we want to predict and irrelevant among each person who provided the data. This subjective data could lead to the unstable accuracy of our models.

Moreover, the data collected from fitbit has many noises, making it more difficult to generalize models. Furthermore, the amount of data for each participant is limited and inconsistent. For instance, there are approximately twenty running activities for participants 1, 2, and 4, especially only three of these activities for participants 3 and 5. Another illustration of this is that the interval of day-observer attributes is not equal, or some participants like 12 lack much information such as sleep data. These things also prevent our models from reaching optimal accuracy.

Additionally, regarding the time running prediction task, the given data does not have the direct attribute representing time running. However, there is an ini-

Table 6: The results of ten different runs for three subtasks returned by the task's organizers

Run ID	SubTask ID	Accuracy	Abs_difference
1	1	0.8	291
	2	0.8	11.3
	3	0.5	4.6
2	1	0.6	290
	2	0.5	14.5
	3	0.5	4.6
3	1	0.6	238
	2	0.6	12.1
	3	0.5	4.6
4	1	0.6	356
	2	0.6	12
	3	0.5	4.6
5	1	0.4	358
	2	0.6	12.6
	3	0.5	4.6
6	1	0.6	304
	2	0.7	11
	3	0.5	4.6
7	1	0.8	234
	2	0.7	11.6
	3	0.5	4.6
8	1	1	232
	2	0.7	11
	3	1	2.6
9	1	0.8	112
	2	0.9	11.4
	3	0.5	4.6
10	1	0.8	96
	2	0.6	15
	3	0.5	1

Table 7: The results of two different runs for three subtasks using the baseline of task's organizers

Run ID	SubTask ID	Accuracy	Abs_difference
1	1	0.4	192.6
	2	0.4	8.5
	3	0.5	2
2	1	0.6	302.8
	2	0.4	8.5
	3	0.5	2

tial 5km run time for each participant. We find that this initial time is randomly extracted from each participant's running activities by dividing the distance attribute by speed attribute. Although the initial running time is claimed to belong to 5km running time, the distance attribute shows the much shorter run. Meanwhile, we are informed that data are collected from 16 people who train for a 5km run, almost running activities have done less than 5km. Moreover, despite the requirement of predicting the difference between seconds used per km (kilometer speed) from the initial run to the run at the end of the reporting period, there is no information to indicate which run or day is at the end of the reporting period for each participant. To cope with these problems, we have to apply some ad-hoc methods to preprocess data that prevent us from generalizing our models.

4 Conclusions

We introduced the solution for predicting an athlete's performance during the training period using neural networks and predictive power score. The predictive power score supports us in enhancing the quality of attributes/features sets towards improving the accuracy of prediction models. We built different prediction models and tested with various parameters and hyperparameters to find the best one. The gained results are auspicious. We will compare our solution with others and thoroughly consider the predictive power score of data attributes to discover hidden patterns useful for improving the accuracy of prediction models.

Acknowledgement

References

1. M. Reiner, C. Niermann, D. Jekauc, and A. Woll, "Long-term health benefits of physical activity–a systematic review of longitudinal studies," *BMC public health*, vol. 13, no. 1, pp. 1–9, 2013.
2. R. P. Bunker and F. Thabtah, "A machine learning framework for sport result prediction," *Applied computing and informatics*, vol. 15, no. 1, pp. 27–33, 2019.
3. P. Zhu and F. Sun, "Sports athletes performance prediction model based on machine learning algorithm," in *International Conference on Applications and Techniques in Cyber Security and Intelligence*. Springer, 2019, pp. 498–505.
4. I. Fister, D. Fister, S. Deb, U. Mlakar, and J. Brest, "Post hoc analysis of sport performance with differential evolution," *Neural Computing and Applications*, pp. 1–10, 2018.
5. H. Charvátová, A. Procházka, S. Vaseghi, O. Vyšata, and M. Vališ, "Gps-based analysis of physical activities using positioning and heart rate cycling data," *Signal, Image and Video Processing*, vol. 11, no. 2, pp. 251–258, 2017.

6. I. Fister, I. Fister Jr, and D. Fister, "Batminer for identifying the characteristics of athletes in training," in *Computational intelligence in sports*. Springer, 2019, pp. 201–221.

7. V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, and D.-T. Dang-Nguyen, "Overview of ImageCLEF Lifelog 2020:Lifelog Moment Retrieval and Sport Performance Lifelog," in *CLEF2020 Working Notes*, ser. CEUR Workshop Proceedings. Thessaloniki, Greece: CEUR-WS.org <http://ceur-ws.org>, September 22-25 2020.

8. V. Thambawita, S. A. Hicks, H. Borgli, H. K. Stensland, D. Jha, M. K. Svensen, S.-A. Pettersen, D. Johansen, H. D. Johansen, S. D. Pettersen *et al.*, "Pmdata: a sports logging dataset," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 231–236.

9. B. Ionescu, H. Müller, R. Péteri, A. B. Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. D. Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, D.-T. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. D. Ştefan, and M. G. Constantin, "Overview of the ImageCLEF 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), vol. 12260. Thessaloniki, Greece: LNCS Lecture Notes in Computer Science, Springer, September 22-25 2020.