Win Your Race Goal: A Generalized Approach to Prediction of Running Performance

Authors
Sandhyarani Dash

Affiliations
Electrical and Computer Engineering, Portland State University, Portland, United States

Keywords
running performance prediction, deep learning, running logs, ultramarathon, marathon

received 20.05.2024
revised 30.07.2024
accepted 31.07.2024
published online 2024

Bibliography
Sports Medicine International Open 2024; 8: a24016234
DOI 10.1055/a-2401-6234
ISSN 2367-1890

Georg Thieme Verlag, Rüdigerstraße 14
70469 Stuttgart, Germany

Correspondence
Sandhyarani Dash
Portland State University
Electrical and Computer Engineering
1825 SW Broadway
97201 Portland
United States
Tel.: 5037252817
dashs@pdx.edu

ABSTRACT

We introduce a novel approach for predicting running performance, designed to apply across a wide range of race distances (from marathons to ultras), elevation gains, and runner types (front-pack to back of the pack). To achieve this, the entire running logs of 15 runners, encompassing a total of 15,686 runs, were analyzed using two approaches: (1) regression and (2) time series regression (TSR). First, the prediction accuracy of a long short-term memory (LSTM) network was compared using both approaches. The regression approach demonstrated superior performance, achieving an accuracy of 89.13 % in contrast, the TSR approach reached an accuracy of 85.21 %. Both methods were evaluated using a test dataset that included the last 15 runs from each running log. Secondly, the performance of the LSTM model was compared against two benchmark models: Riegel formula and UltraSignup formula for a total of 60 races. The Riegel formula achieves an accuracy of 80 %, UltraSignup 87.5 %, and the LSTM model exhibits 90.4 % accuracy. This work holds potential for integration into popular running apps and wearables, offering runners data-driven insights during their race preparations.

Introduction

Accurately forecasting a runner's performance in long distance races such as marathons is challenging, but it's particularly difficult for ultramarathons. Ultramarathon or ultra-distance running is a race that exceeds the traditional marathon distance of 42.195 km. Common ultramarathon distances include 50 km, 80 km, 100 km, 162 km, and even up to 1,600 km. As highlighted in [1], the sport has seen increasing popularity. A survey of the Deutsche Ultramarathon Vereinigung (DUV), which houses the largest database on ultramarathons, revealed that there were fewer than 20 ultra races annually from 1798 to 1969. However, this number surged to 7,465 events worldwide in 2019.

Participating in these races often involves significant expenses and extensive preparation, which can span years depending on fac-

tors like distance, elevation gain, weather conditions, and the technicality of the terrain. Notably, premier events like the Western States 162 km, Ultra-Trail du Mont-Blanc (UTMB), Hardrock 162 km, and Badwater 217.26 km have qualifying races and entry lotteries. Almost all ultramarathons enforce cut- off times, as referenced in [2]. Given these challenges, runners benefit from regular feedback during their race preparations. At present, a significant challenge exists due to the lack of a single, unified, and precise method capable of predicting a runner's performance across any race, regardless of its distance or elevation gain.

There is extensive research focused on predicting a runner's performance in shorter distance races using physiological data as shown in [3–5]. where the primary performance predictors are peak velocity at VO_2 max, baseline blood pressure and blood lactate response to exercise. However, it is worth noting that most of this research focused mainly on track and treadmill trials for distances up to 5000 m. Addressing the terrain factor, Nicot et al. [6] demonstrated that running on technical trails consumes more energy compared to road, track, and treadmill running. Specifically, the oxygen cost of running increases by 11.4 %, heart rate by 5 %, total volume of inhaled and exhaled air by 14.42 %, and the overall energy expenditure is 5 % higher on trails.

Numerous techniques exist for predicting race times up to marathon distances.

1) Riegel Formula [7]

$$T_2 = T_1 * \left(\frac{D_2}{D_1}\right)^{1.06} \quad (1)$$

In eq. 1, T_2 denotes the predicted race time, whereas T_1 signifies the time for a shorter race. Likewise, D_1 corresponds to the distance of the shorter race, and D_2 indicates the distance of the race for which we aim to predict the time. For comparison against this benchmark, if forecasting is conducted for longer ultra races, then the shorter race is set to a 50-km race. Each of the 15 runners has at least one 50-km race in their data.

2) Modified Riegel Formula [8]

$$time_{marathon} = time_{r2} * \left(\frac{\frac{42195}{distance_{r2}}}{60}\right)^{k_{marathon}} \quad (2)$$

where $distance_{r2}$ and $time_{r2}$ are distance and time for second race. $k_{marathon}$ is a constant that the authors calculated based on the runner's typical weekly mileage.

3) Matrix Completion Method [9]

$$logt = \lambda_1 f_1(s) + \lambda_2 f_2(s) + \ldots . \lambda_n f_n(s) \quad (3)$$

where $f_1, f_2 \ldots f_n$ are components that are same for every runner and different for different distances, while $\lambda_1, \lambda_2 \ldots \lambda_n$ are coefficients which summarize the runner under consideration.

4) Regression Shrinkage [10] – Ridge regression and LASSO regression with nonlinear part performed the best.
5) Simple Statistical Methods [11] – The best performing model in this paper is as follows: the author segmented a race distance and then calculated the speed in different segments based on the slope of the segment. Then for each speed-slope pair in the test set, they identified a corresponding pair in the histogram of training slopes. Based on this technique, they predicted the total time to finish the distance.
6) Case-based Reasoning (CBR) [12–15] – This method relies on generating a database of cases using the following equation.

$$c_{ij}(r, m_{i,} m_j) = \langle nPB(r, m_i) PB(r, m_j) \rangle \quad (4)$$

where r is the runner m_i, m_j are two marathons that the runner has run. *nPB* is the non-personal best and PB is the personal best times for m_i and m_j respectively. The database is queried with *nPB* race record and the system retrieves a set of k cases with similar *nPB* records. Then the system averages the *PB* of these $k$cases to generate a *PB* finish time for a race.

7) Multi-Layer Perceptron Model [16–18]
8) Statistical Analysis [19] – Turkey test and standard t-test.
9) Bagged Ensemble Learning Algorithm [20] – Predict injuries by analyzing running logs of 74 middle- to long-distance runners over a seven-year period.

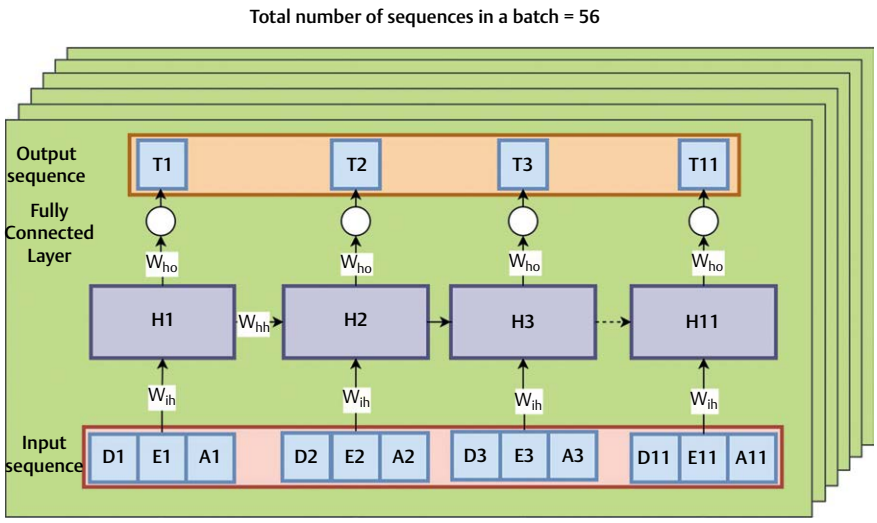
As mentioned above, accurately predicting a runner's performance for ultramarathons is more challenging than other distances and therefore there are very few publications. Fogliato et al. [21] introduced a predictive framework aimed at optimizing ultra-distance race organization by forecasting a runner's ability to reach the next checkpoint. CBR is one of the techniques used to predict race times for ultras. McConnell et al. [22] utilized UTMB race results from 2013–2017, creating a case base of 3,222 for 1,266 runners, and achieved a pacing error rate of ≈ 7–9 % in race time prediction. The effectiveness of CBR technique hinges on finding similar cases of runners for the same race. However, finding similar cases of runners for ultra-distance races is difficult because factors affecting the race performance such as stroke volume, running cost, oxygen cost of exercise, fat oxidation, core body temperature, hydration, muscle strength, cognitive function, reaction time, and decision- making ability can vary as indicated in [1].

Coquart [23] pinpointed various performance determinants for a 100-km race, including physiological data such as age, body mass index (BMI), recent marathon performance or personal record (PR), and environmental factors like wind speed and barometric pressure during the race. Using these predictors, the author formulated an equation that achieved a standard error of estimate (SEE) of 14 %. This translates to an average standard deviation of 95 minutes across 56 athletes.

UltraSignup tracks ultra endurance athletes' performances by assigning a rank based on their ultra-distance race results. Upon registering for a new race, runners receive a target time calculated using the formula [24].

$$\frac{\frac{1}{4}\sum_{i=1}^{4} Winner's\ Time_i}{Runner's\ Rank} \quad (5)$$

In equation 5, the average of the race winner's time from the four years preceding the race is computed. Runner's rank is determined by UltraSignup, based on runners' performances in previous ultra



▶ **Fig. 1** LSTM model. A unidirectional sequence-to-sequence (Seq2Seq) multivariate LSTM architecture was chosen. This figure shows the regression model and therefore it lacks the fourth input, i. e., time passed between runs. The network receives an average (across 15 customized networks, one per runner) sequence length of 11 inputs, namely distance (D) in km, elevation gain (E) in m, and age (A) in years and output, total time to finish the run (T) in minutes. These inputs are channeled through 11 time steps, from H1 to H11, culminating in a fully connected layer that predicts an output sequence of 11 moving times (in minutes) required to complete a run. Notably, the input and output sequence lengths are identical. Training is conducted in batches, with an average batch size of 56, meaning there are 56 sequences in a single batch. Consequently, the total number of hidden units is given by 56 × 11 = 616. Each hidden cell encompasses 17 neurons (17 dimensional hidden state), which consist of input, forget, and output gates connected in the same way as originally invented in [34]. The hidden states are initialized with zeros before each forward pass. This means that the model remembers 11 time steps and updates hidden states for each sequence of 11 time steps independently. After processing the 11th timestep of one sequence, the LSTM initiates the subsequent sequence with a fresh initial hidden state. This approach is deemed appropriate since the training intensity distribution (TID) patterns span across a week, after which the pattern recurs, unless there is a shift in the training phase, such as a tapering period. Wih, Whh, Who represent input-to-hidden, hidden-to-hidden and hidden-to-output weights, respectively. The total number of parameters to train in one LSTM cell is 14. Across 616 hidden cells, the total number of parameters is 14 × 616 = 8624.

races. This is the second equation that is used for benchmark comparison.

In this study, a generalized approach for predicting race times irrespective of distance, elevation gain, and type of runner (front-pack, mid-pack and back of the pack) is introduced by using runners' complete running logs and an autoregressive deep learning model called long short-term memory (LSTM).

This research addresses the following queries in the specified order:

1) Regression vs. time series regression (TSR) approach: Since running log is a time series dataset, we must therefore first identify which approach performs better using LSTM.
2) Performance of LSTM vs. benchmark models: Comparison of LSTM model with two benchmark models – the Riegel formula and UltraSignup formula – across 60 races.

Given the intended use of this study's outcomes in running apps and wearables, the selected network must prioritize fast training and energy efficiency.

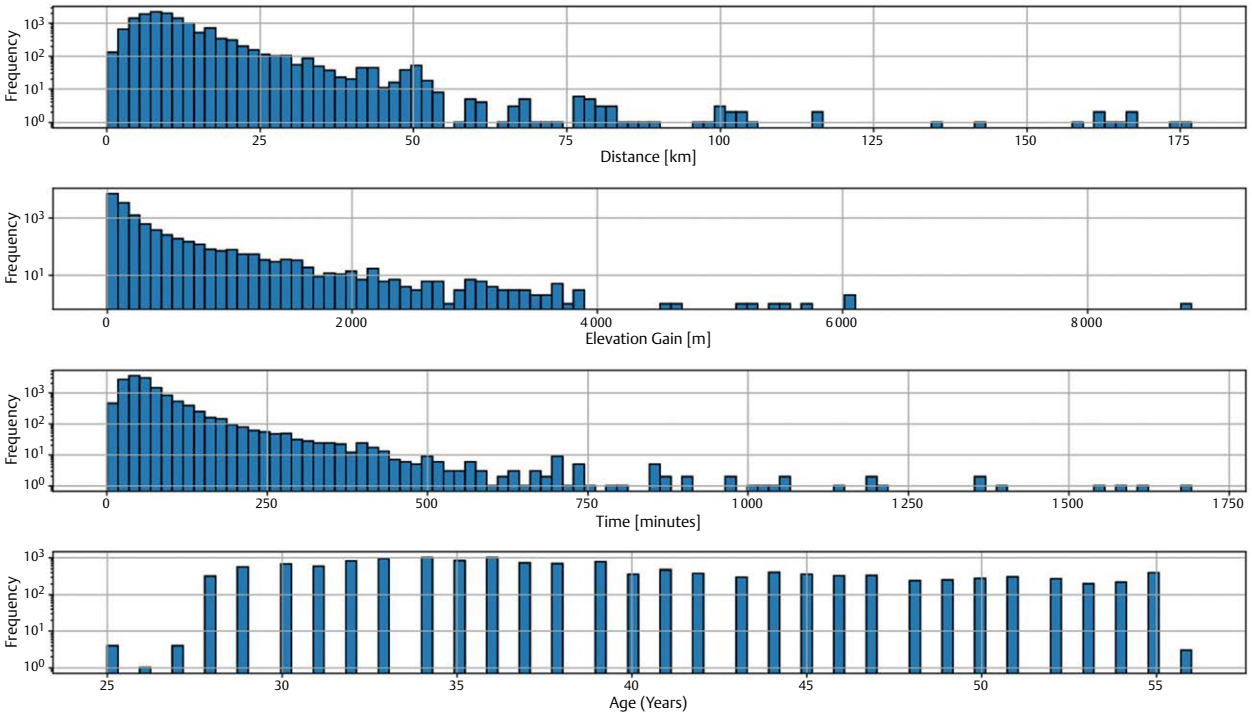
The network inputs are distance, elevation gain, and age of the runner for each run in their running log and the network output is the total time to complete the run as shown in ▶ **Fig. 1**.

Materials and Methods

Data preparation

The data for this study was collected in adherence to IRB Protocol #207107–18. A survey, hosted on the servers of an institution affiliated with the author, was shared across various running forums on Facebook, with participants voluntarily completing it. Through this process, running logs and UltraSignup pages of 15 recreational ultra runners comprising 13 men and 2 women aged 28 to 60 years, were collected. Their recent paces for a 50-km race, occurring in 2022–2023, ranged from 5.44 mins/km to 12.44 mins/km. The total dataset encompasses 15,686 runs. Each log spans between three to ten years, averaging 6.6 years of running history across the 15 runners. On average, each runner logged about 1,046 runs.

The following data from the running logs was utilized for this study: activity date, activity name, activity type, distance, moving time, elevation gain, and age. However, the "activity type" can sometimes be mislabeled. For instance, activities like running, cycling, skiing, hiking, and walking may all be labeled as "run." The onus of accurate labeling lies with the runner. Feeding mislabeled



▶ **Fig. 2** Data distribution of running logs. The running logs present a challenge due to their non-Gaussian data distribution. Transforming this distribution to a normal one is infeasible because the study also focuses on information from the third and fourth quartiles, which pertain to longer distance runs. Therefore, the data distribution is used as is, without normalization.

data, such as other activities inaccurately marked as "run" into the neural network can lead to model confusion. For example, downhill skiing can reach speeds of up to 97 km/h, while the record for running is 45 km/h. Hence, data filtering from running logs is imperative before input to the model. Employing typical outlier removal methods, such as the z-score or interquartile range, is unsuitable for the purposes of this study. Consequently, a specialized algorithm was developed to refine the running logs.

1) Activity type: If the activity type for any row is not "run," that data is discarded.
2) Missing data: Rows with NaN, empty cells, zero elevation gain, or zero moving time are dropped.
3) Short activities: Activities with moving times less than 3 minutes are discarded.
4) Short runs: Runs less than 1.6 km are excluded, as the study focuses on longer runs.
5) Mislabeled walks: To identify mislabeled walking data, we assess distance, pace, and elevation gain for each activity. Activities slower than a runner's average pace are discarded.
6) Mislabeled bike and ski activities: To spot mislabeled biking and skiing data, we compare the pace of the supposed run against the world record for the fastest mile, which is 3 minutes and 43 seconds. If a runner's pace surpasses this, it is inferred the activity is not a run, and such data is omitted.

This reduced the total number of runs by 4.4 % but improved the performance of the LSTM model by 12 %.

Since this study focusses on a single method that can predict runner's performance in any type of race, regardless of distance or elevation gain, the data in the third and fourth quartiles of the distribution is also of interest, as shown in ▶ **Fig. 2**.

Understanding the seasonal decomposition of the dataset helps with the decision of model selection, therefore the next two subsections describe this in detail.

Missing data

Upon filtering and plotting the data, it became evident that there were instances of missing completely at random (MCAR) data. This inconsistency was characterized by the absence of consistent running patterns. The analysis identified two distinct types of MCAR issues: (i) data absent for multiple years, and (ii) sporadic missing data across several months within a single year.

For the first category, one approach is to omit the incontiguous data from certain years and focus on continuous periods. However, deleting incontiguous data from multiple years will lead to substantial data reduction, particularly for runners with smaller logs. As for the second category, polynomial imputation could address the sporadic missing data. Yet, this approach faces two significant challenges:

1) The non-linear relationship in a multivariate dataset presents a problem: Attempting to impute variables like distance, elevation gain, activity date, and time to complete a run, individually, can result in the loss of inherent relationships among variables. For example, an increase in distance and elevation gain typically

correlates with a non-linear increase in time to complete the run.
2) Accounting for rest days during the imputation period is complex: A runner's decision to take rest days, often influenced by the intensity of previous runs (considering factors like distance and elevation gain), varies from individual to individual, making it challenging to accurately predict and incorporate into the dataset.

Consequently, it was resolved to fill the missing data with zeros only to perform seasonal decomposition. Since neural networks are proven to be effective for discontinuous time series [25], the running logs were therefore sorted by date of the runs and used as is for the LSTM model.

Seasonal decomposition

In this dataset, the independent variables include the distance, total elevation gain of each run as well as the age of the runner. The dependent variable is the total time taken to complete the run. Applying seasonal decomposition to both the independent and dependent variables will reveal their trends, seasonality, and residuals. An additive model is assumed for this decomposition as shown in equation 6, where feature at time t can be decomposed to the sum of seasonality, trend and residual at time t. It is worth noting that for ultra-distance races of up to 162 km, a training plan spanning at least six months is typically required. Consequently, the seasonal decomposition is conducted over a corresponding period to align with this training time frame.

$$Feature_t = S_t + T_t + R_t \quad (6)$$

The residual plot of seasonal decomposition shows stochastic variations after detrending and deseasonalizing the data, where peaks represent prediction errors by the additive model. Ideally if residuals are minimal, then a simpler model will suffice.

Residual variability (RV) was calculated by determining the standard deviation, essentially measuring the spread of the stochastic component of the dependent variable from its mean. A higher RV indicates the need for more complex machine learning models to fit complex patterns in the data. The average RV (across 15 runners) of the output feature, i. e., total time to finish the run is 56.2 minutes.

Most correlated lag and the Pearson correlation value (r value) for each runner was also calculated. The average of most correlated lag is 20.73 and $r = 0.221$. This means the network should remember at least 20 consecutive runs. However, the r is very low, which means that there is more non-linearity in the data that is not captured by the Pearson correlation.

There are two ways to convert a regression problem into a TSR problem:
1) Time delay embedded where lagged versions of output feature is fed as input to the network.
2) Temporal embedding where time passed between the runs is fed as input to the network. Since the Pearson correlation is low for this dataset, option 2 was therefore chosen.

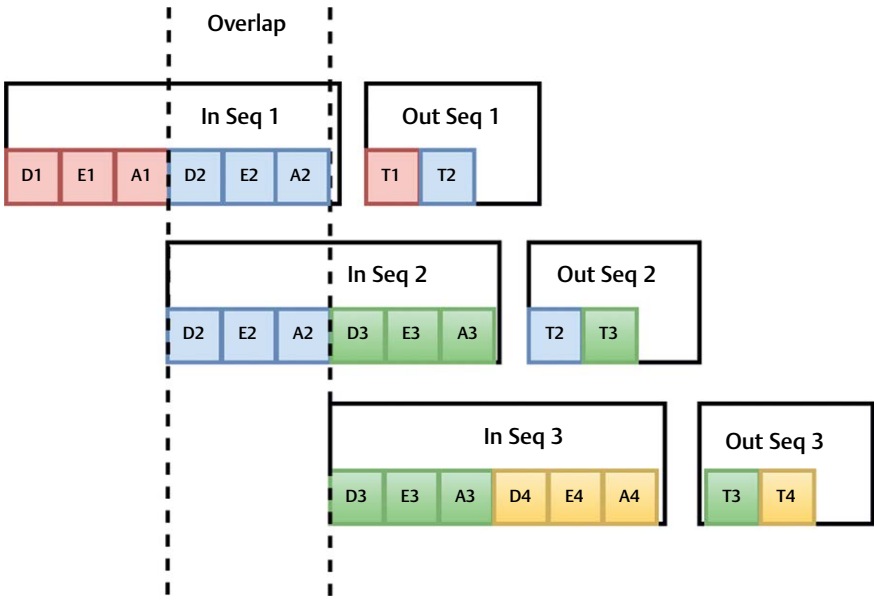
The key challenges with this dataset are:
1) Sparse ultra-distance runs: Athletes typically do not run 50 km or more for their regular training. Consequently, ultra runs/races constitute a mere 1.77 % of the entire dataset.
2) Variable paces for same distances: There are two popular training intensity distributions (TIDs) for middle – and long-distance runners [26–30]:
   a) Polarized training: This approach combines 80 % low-intensity aerobic exercises with 20 % high-intensity anaerobic activities. In the realm of endurance running, it is essential for athletes to develop a robust aerobic foundation through these low-intensity or base runs. Such runs are characterized by a pace that is more relaxed, often lagging behind the intended race pace by 2–3 minutes. For instance, if an athlete's competitive speed is 5.6 min/km, their base runs should ideally hover around 7.6 to 8.6 min/km. The remaining 20 % of the polarized training encapsulates runs that match the race pace, typically aligning with a zone 3 heart rate.
   b) Pyramidal training: This training structure is segmented into 75 % low-intensity, 20 % moderate- intensity, and a mere 5 % high-intensity workouts. The moderate-intensity segment predominantly involves threshold runs. It is essential for the model to recognize these training patterns, understanding that a runner will cover the same distance at varying speeds, depending on the training regimen.
3) Variable paces in ultra-distance runs: Various factors like variations in elevation gain, weather conditions, trail characteristics, and the runner's state on the race day can influence performance. However, the running logs lack details on weather, trail conditions, and perceived exertion, potentially affecting the model's predictive accuracy.
4) Lack of physiological data: Incorporating physiological parameters, such as heart rate, VO_2max, and perceived exertion, can significantly enhance the model's precision.
5) High RV and low Pearson correlation value: As discussed above the dataset's non-linearity and significant stochastic elements necessitate the use of advanced models over simple statistical approaches for precise predictions.

Model selection

This section describes a comprehensive overview of potential models for regression and TSR detailing selection and exclusion criteria for the dataset in this study.

Statistical models

1) Exponential smoothing: This model is effective when the target variable is highly correlated with its recent values and the forecast horizon is short. It handles non-linear dependencies well and does not require stationarity in the data. However, it is not suitable for multivariate time series and is sensitive to outliers. It also struggles with datasets that have non-linear relationships between variables.
2) Moving average: A simple and quick method for observing trends in time series data, applicable to both linear and non-linear patterns. Its main limitation is its ineffectiveness with multivariate time series.



▶ **Fig. 3** Sliding window technique. The sliding or rolling window technique with a stride of one is used to create sequences for LSTM. Here D is distance, E is elevation gain, A is age and T is time. The numbers indicate the sequence of the runs from the running log. For simplicity and to save space, a sequence length of two is shown. This method amplifies the dataset by a factor of 100, offering the network a richer context to learn from and thereby generalizing more effectively.

3) Kalman filter: Known for its simplicity and computational efficiency, the Kalman filter performs well on noisy datasets. It is primarily suited for unimodal distributions; however, it struggles with non- Gaussian distributions and is highly dependent on accurate model parameters and initial conditions.
4) Hidden Markov model: Robust against noise and uncertainty, similar to Kalman filters, and is faster to train. However, it does not generalize well to unseen data and fails to capture non-linear relationships between exogenous and endogenous features, which is crucial for the problem in this study.

Machine learning models

1) K-nearest neighbors: This technique is straightforward, requiring no training time apart from the selection of the “k” parameter. However, it is sensitive to outliers.

Deep neural network (DNN) models

1) Next gen reservoir computing: Offers a reduced number of hyperparameters compared to traditional reservoir computing, leading to faster training times. Nonetheless, it is limited by a very short forecasting horizon [31].
2) Dilated convolutional neural networks (CNNs): These networks have an increased receptive field with fewer parameters, which helps capture long-range dependencies without significantly increasing computational costs. Yet, they cannot handle variable length input sequences or identify complex temporal patterns both in the short term and long term [32].
3) Long short-term memory (LSTM): This autoregressive model handles variable length input sequences and detects complex temporal patterns over both short and long terms. The main drawbacks are its inability to process data in parallel (like transformer networks) and the long training time due to a larger number of parameters. However, given that the dataset is relatively small, consisting of 15,686 runs, parallel processing and training time are not significant concerns.
4) Gated recurrent unit (GRU): With fewer parameters than LSTM, GRU is computationally less demanding. However, it struggles with modeling long-term complex temporal patterns and is prone to overfitting on small datasets.
5) Feedforward multilayer perceptron model: A simpler network for regression tasks compared to other deep neural network (DNN) models, but it requires a substantial amount of data to forecast accurately.

LSTM network architecture

The detailed architecture of LSTM network is shown in ▶ **Fig. 1** and ▶ **Fig. 3**. The LSTM network was individually trained for every runner so the network parameters and hyperparameters are unique to the running log of the runner. During hyperparameter search the validation set is randomly selected using k-fold cross-validation ($k = 5$) and the test set is the last 15 runs of the running log. During the model training phase, the last 30 entries of each running log are equally divided into validation and test datasets, respectively. The test set has a distance range from 1.6 km to 50 km and elevation gain range from 0 m to 2,062 m across 15 running logs. The validation and test datasets were kept separate and were not exposed to the model during training. The model was trained in batches during each epoch and subsequently evaluated on the validation set.

To combat overfitting, two regularization methods were used: dropout at a 50 % rate and the early stopping algorithm. Implementing dropout with a 50 % rate entails that during training, the LSTM network can deactivate 50 % of the neurons in each hidden layer. It is important to note that dropout is disabled during model evaluation, i. e., during validation and testing phases. The early stopping mechanism is designed to halt the training loop if the validation loss begins to rise for a predefined number of consecutive instances. The configurable parameter is called patience, which is treated as a hyperparameter. Additionally, the Optuna optimization framework with 60 trials was used to determine the optimal hyperparameters. To test the stability of the hyperparameters, k-fold cross-validation is performed ($k = 5$) as shown in ▶ **Fig. 4** during hyperparameter search. The Huber loss function served as the chosen metric for assessing the model's performance during each epoch, and Adam's optimization algorithm was used for weight adjustments during backpropagation. Notably, this combination yielded the best results. ▶ **Table 1** shows the average of hyperparameters for the LSTM.

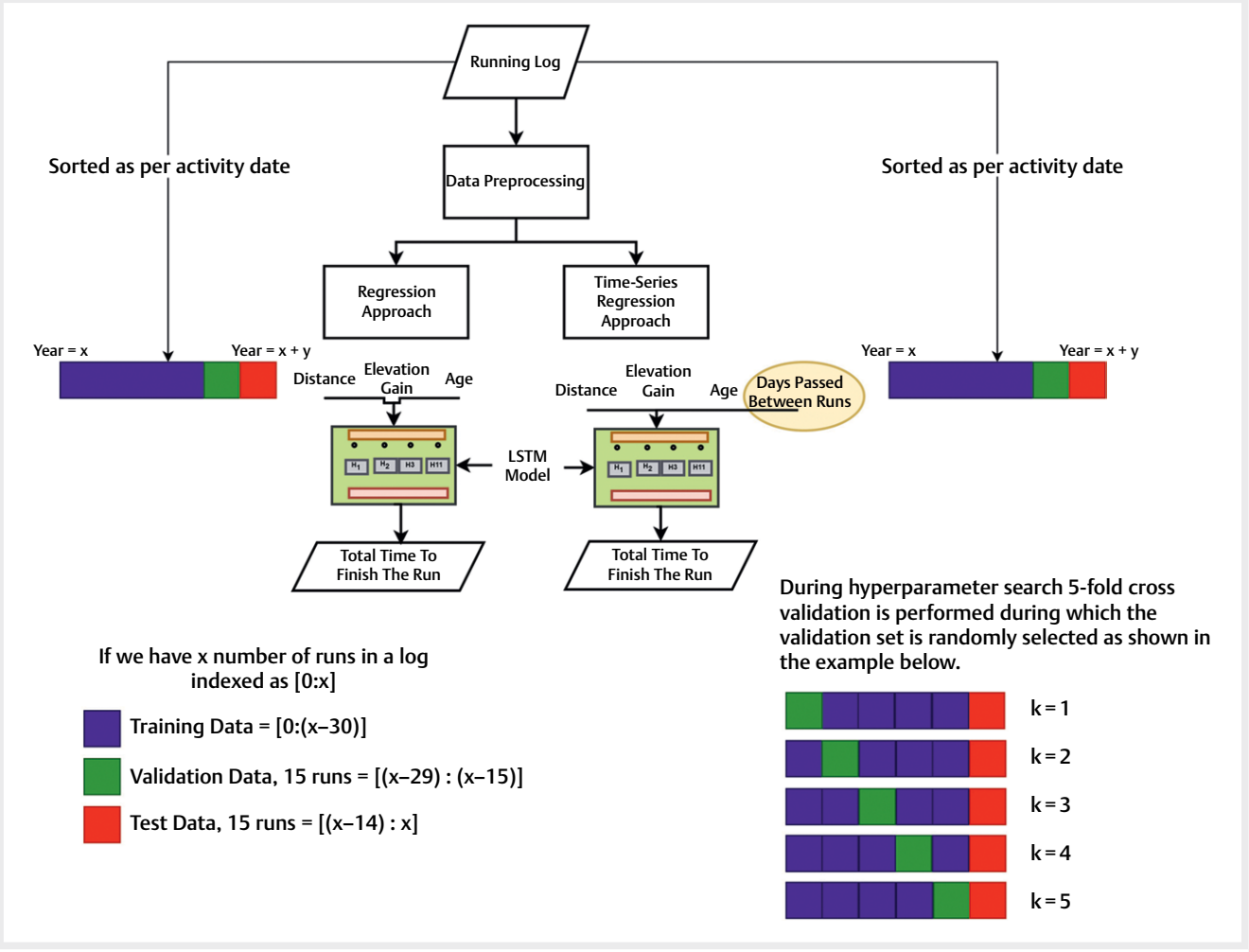
▶ **Table 1** Aggregate hyperparameter settings for LSTM networks evaluated across 15 customized networks.

Hyperparameter	Value
Number of hidden layers	1
Number of neurons per layer	17
Batch size	56
Learning rate	0.018468
Sequence length	11
Patience before early stopping	63
Epochs	300

The LSTM network implementations were coded using the PyTorch NN module. The methodology followed for research question 1 is graphically shown in ▶ **Fig. 4**.

Results

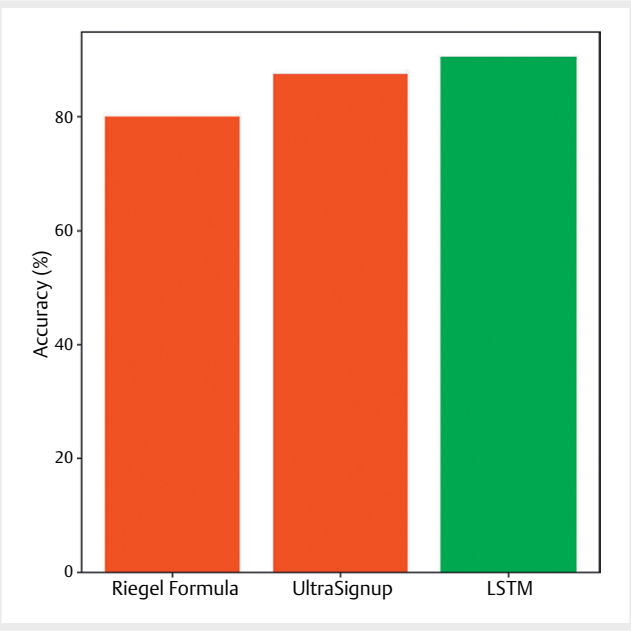
The performance of the LSTM model is assessed using the following metrics:



▶ **Fig. 4** Regression vs. time series regression graphical representation. Shows the graphic version of research question 1: Regression vs. TSR approach. There are two ways to convert a regression problem into a TSR problem: (1) Time delay embedded where lagged versions of output feature are fed as input to the network, and (2) temporal embedding where time passed between the activities is fed as input to the network. Since the Pearson correlation is low for this dataset, option 2 is therefore chosen. Once hyperparameters are identified the training, validation and test indexes are fixed as shown in the figure. These 15 runs in the test set and in validation set could be of any distance or elevation gain.

▶ **Table 2** Distribution of different races included for LSTM model's comparison against benchmark models.

Race Distance (km)	Frequency in benchmark comparison test dataset
25	3
32	1
42.2	1
50	30
60	2
65	1
80	6
100	8
162	7
175	1
Total	60



▶ **Fig. 5** Performance comparison of LSTM and benchmark models. Comparison of LSTM model against two benchmarks for 60 races across 15 runners. Notably, 54 % of these races occurred post the data collection cutoff in December 2021. The test set was formed by excluding the other 46 % documented within the running logs. The accuracies are Riegel: 80 %, UltraSignup: 87.5 %, and LSTM: 90.4 %.

1) Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100\% \quad (7)$$

2) Accuracy

$$Accuracy = 100 - MAPE \quad (8)$$

In equation (7), y_i represents the actual value for the i^{th} observation, while denotes the predicted value for the same observation. n represents the total number of runs in the test set. The reported metrics are averaged across all runners to compute the overall performance of all LSTM networks.

Q1. Regression vs. time series regression approach

The regression approach beats the TSR approach in both accuracy and standard deviation. For the same test set in each running log as shown in ▶ **Fig. 4** the LSTM model predicts running performance at 89.13 % accuracy with a standard deviation of 4.69 % for the regression approach and 85.21 % with a standard deviation of 5.34 % for TSR approach. Due to its autoregressive nature and the use of supervised learning, LSTM can capture both long-term and short-term correlations among exogenous and endogenous variables, even without explicitly incorporating the notion of time into the network.

Q2. Performance of LSTM vs. benchmark models – Riegel formula and UltraSignup formula – across 60 races

The study analyzed 60 races across 15 runners. Notably, 54 % of these races occurred after the data collection cutoff in December 2021. ▶ **Table 2** shows the race distances and the frequency of such races in the test set. This test set was used to compile predictions on total race completion times from the LSTM model, Riegel formula, and UltraSignup formula. ▶ **Fig. 5** illustrates how the LSTM model's performance compares to these established benchmarks. The results highlight the LSTM model's capacity to generalize effectively on unseen races, even when the progress of the runner is unknown to the model.

Discussion

This study offers a comprehensive method for predicting race times across varying distances, elevation gains and types of runners using a runner's entire running log and LSTM. It can be argued that running records of just 15 runners might not accurately represent a wider running population. However, this is a demonstration study, and a comprehensive evaluation with a larger set of running logs is a future objective.

The LSTM model achieved a prediction error (MAPE) of 10.87 %, which aligns closely with the ≈ 7–9 % error in [22]. Coquart [23] presented a standard error of estimate (SEE) of 97 minutes based on the performance of 56 runners in four 100-km races in France. The LSTM model in this study logged an SEE of 5.96 minutes based on the performance of 15 runners in twelve 100-km races. Although the best MAPE is 9.2 % lower than the best MAPE presented for marathon performance in [16], this research encompasses a diverse set of recreational runners with varied race histories. The dataset exhibits a non-Gaussian distribution, contrasting the normal distribution found in [16]. This difference arises from the focus on generalized running performance prediction across various distances, elevation gains, and types of runners.

A noted limitation of this study is the individual customization of the LSTM network for each runner, given the reliance on their running logs rather than just physiological data and performance in marathon and 10-km race. Another constraint is the lengthy hyperparameter optimization process with 5-fold cross-validation for each trial performed by Optuna. However, after the hyperparameter search the training and inference time is less than a minute. Furthermore, data inaccuracies such as elevation discrepancies,

limited inputs, and missing terrain or weather details can hamper model performance.

Another limitation is the choice of a unidirectional Seq2Seq LSTM, which, while superior to GRU according to [33], may not be ideal for Seq2Seq prediction problems. For these types of problems, an attention-based encoder-decoder architecture is more suitable. However, such architectures necessitate a significantly large dataset to achieve optimal accuracy.

One area where the LSTM model could see improvement is by incorporating more input parameters. Enhanced performance could be achieved by factoring in physiological details like heart rate, perceived exertion, VO_2 max and other relevant data such as weather and trail conditions.

Conclusion

The study demonstrates a generalized approach to prediction of running performance across various distances, elevation gains, and recreational runners of varying levels (front-pack, mid-pack, and back of the pack) by using their complete running logs and LSTM. Given the time series nature of running logs, the research initially addressed dataset challenges and determined the suitability of regression versus time series regression (TSR) on a LSTM model. The regression approach outperforms the TSR approach by a higher accuracy of 4.01 % and is favored due to its reduced number of inputs to the network and thereby reduces the number of trainable model parameters. For the regression approach, the LSTM network achieves an accuracy of 89.13 % on the test set (last 15 runs) of each runner's running log.

Furthermore, the LSTM model's performance exceeded two benchmarks, the Riegel formula and UltraSignup formula, in predicting race completion times for 60 races across 15 runners, with the following accuracies: Riegel 80 %, UltraSignup 87.5 %, and LSTM 90.4 %. The LSTM model also demonstrates better performance on unseen data because 54 % of the races in this test set occurred after data collection, i. e., after December 2021.

The findings underscore the LSTM model's capability to discern training intensity distribution (TID) and running patterns, enabling accurate future race performance predictions. This methodology provides runners with data-driven periodic feedback during their race preparation.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] Berger NJA, Best R, Best AW et al. Limits of ultra: towards an interdisciplinary understanding of ultra-endurance running performance. Sports Med 2024; 54: 73–93

[2] UTMB World Series. [Online]. Available from: https://uta.utmb.world/overview Accessed: 28-Jun-2024.

[3] Coates M, Berard J, King TJ et al. Physiological determinants of ultramarathon trail-running performance. Int J Sports Physiol Perform 2021; 16: 1454–1461

[4] Denadai BS, Greco CC. Could middle- and long-distance running performance of well-trained athletes be best predicted by the same aerobic parameters?. Curr Res Physiol 2022; 5: 265–269

[5] Balasekaran G, Loh MK, Boey P, Ng YC. Running energy reserve index (RERI) as a new model for assessment and prediction of world, elite, sub-elite, and collegiate running performances. Sci Rep 2023; 13: 7416

[6] Nicot F, Sabater-Pastor F, Samozino P et al. Effect of ground technicity on cardio-respiratory and biomechanical parameters in uphill trail running. Eur J Sport Sci 2022; 22: 1836–1846

[7] Riegel PS. Athletic records and human endurance: a time-vs-distance equation describing world-record performances may be used to compare the relative endurance capabilities of various groups of people. Am Sci 1981; 69: 285–290

[8] Vickers AJ, Vertosick EA. An empirical study of race times in recreational endurance runners. BMC Sports Sci Med Rehabil 2016; 8: 26

[9] Blythe DAJ, Király FJ. Prediction and quantification of individual athletic performance of runners. PLoS One 2016; 11: e0157257

[10] Przednowek K, Iskra J, Maszczyk A, Nawrocka M. Regression shrinkage and neural models in predicting the results of 400-metres hurdles races. Biol Sport 2016; 33: 415–421

[11] Admetlla AR. Software tools for self-quantifier runners. [M.S. thesis]. Yverdon-les-Bains. Switzerland: Institute for Information and Communication Technologies; ; 2015

[12] Smyth B, Cunningham P. Running with cases: a CBR approach to running your best marathon. in Case-Based Reasoning Research and Development 2017; 10339: 360–374

[13] Smyth B, Cunningham P. Marathon race planning: a case-based reasoning approach. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI'18). 2018 Jul 13–19; Stockholm, Sweden. Washington, DC: AAAI Press; 2018: 5364–5368

[14] Smyth B, Cunningham P. An analysis of case representations for marathon race prediction and planning. In: Cox M, Funk P, Begum S (eds). Case-Based Reasoning Research and Development. Cham: Springer International Publishing; 2018: 369–384

[15] Feely C, Caulfield B, Lawlor A, Smyth B. Modelling the training practices of recreational marathon runners to make personalized training recommendations. In: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization. 2023 Jun 26–29; Limassol, Cyprus. 2023, 183–193

[16] Lerebourg L, Saboul D, Clémençon M, Coquart JB. Prediction of marathon performance using artificial intelligence. Int J Sports Med 2023; 44: 352–360

[17] Maurer J. Race time prediction on individual historical training data for hilly and non-hilly courses. [M.S. thesis] Saarbrücken, Germany: Faculty of Natural Sciences and Technology I, Department of Computer Science,. Saarland University. 2018;

[18] Dracopoulos DC. A better predictor of marathon race times based on neural networks. In: Bramer M, Petridis M (eds) Artificial Intelligence XXXIV. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2017: 293–299

[19] Feely C, Smyth B, Caulfield B, Lawlor A. Estimating the cost of training disruptions on marathon performance. Front Sports Act Living 2023; 4: 1096124

[20] Lövdal SS, Den Hartigh JR, Azzopardi G. Injury prediction in competitive runners with machine learning. Int J Sports Physiol Perform 2021; 16: 1522–1531

[21] Fogliato R, Oliveira NL, Yurko R. TRAP: a predictive framework for the assessment of performance in trail running. J Quant Anal Sports 2021; 17: 129–143

[22] McConnell C, Smyth B. Going further with cases: using case-based reasoning to recommend pacing strategies for ultra-marathon runners. In: Bach K., Marling C (eds). Case-Based Reasoning Research and Development. 2019: 11680: 358–372

[23] Coquart JB. Prediction of performance in a 100-km run from a simple equation. PLoS One 2023; 18: e0279662

[24] Gilligan M. UltraSignup – Frequently Asked Questions. Available from: https://ultrasignup.com/faqs.aspx Accessed: Mar. 18, 2024

[25] Hill T, O'Connor M, Remus W. Neural network models for time series forecasts. Manage Sci 1996; 42: 1082–1092

[26] Filipas L, Bonato M, Gallo G, Codella R. Effects of 16 weeks of pyramidal and polarized training intensity distributions in well-trained endurance runners. Scand J Med Sci Sports 2022; 32: 498–511

[27] Wang Z, Tai Wang Y, Gao W, Zhong Y. Effects of tapering on performance in endurance athletes: A systematic review and meta-analysis. PLoS One 2023; 18: e0282838

[28] González-Ravé JM, González-Mohino F, Rodrigo-Carranza V, Pyne DB. Reverse periodization for improving sports performance: a systematic review. Sports Med Open 2022; 8: 56

[29] Campos Y, Casado A, Vieira JG et al. Training-intensity distribution on middle- and long-distance runners: a systematic review. Int J Sports Med 2022; 43: 305–316

[30] Kenneally M, Casado A, Santos-Concejero J. The effect of periodization and training intensity distribution on middle- and long-distance running performance: a systematic review. Int J Sports Physiol Perform 2018; 13: 1114–1121

[31] Gauthier DJ, Bollt E, Griffith A, Barbosa WAS. Next generation reservoir computing. Nat Commun 2021; 12: 5564

[32] Borovykh A, Bohte S, Oosterlee CW. Dilated convolutions neural networks for time series forecasting. J Comput Finance 2019; 22: 73–101

[33] Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M. A survey on long short-term memory networks for time series prediction. Procedia CIRP 2021; 99: 650–655

[34] Martınez-Llop PG, Sanz Bobi JD, Ortega MO. Time consideration in machine learning models for train comfort prediction using LSTM networks. Eng Appl of Artif Intell 2023; 123: 106303