

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2015, Article ID 735060, 9 pages
<http://dx.doi.org/10.1155/2015/735060>



Research Article

Predictive Modeling in Race Walking

**Krzysztof Wiktorowicz,¹ Krzysztof Przednowek,²
Lesław Lassota,² and Tomasz Krzeszowski¹**

¹Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, 35-959 Rzeszów, Poland

²Faculty of Physical Education, University of Rzeszów, 35-959 Rzeszów, Poland

Correspondence should be addressed to Krzysztof Wiktorowicz; kwiktor@prz.edu.pl

Received 15 December 2014; Accepted 18 June 2015

Academic Editor: Okyay Kaynak

Copyright © 2015 Krzysztof Wiktorowicz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the use of linear and nonlinear multivariable models as tools to support training process of race walkers. These models are calculated using data collected from race walkers' training events and they are used to predict the result over a 3 km race based on training loads. The material consists of 122 training plans for 21 athletes. In order to choose the best model leave-one-out cross-validation method is used. The main contribution of the paper is to propose the nonlinear modifications for linear models in order to achieve smaller prediction error. It is shown that the best model is a modified LASSO regression with quadratic terms in the nonlinear part. This model has the smallest prediction error and simplified structure by eliminating some of the predictors.

1. Introduction

The level of today's high-performance sport is very high and very even. Both coaches and competitors are forced to search for and use newer and sometimes innovative solutions in the process of sports training [1]. A solution supporting this process may be the application of various types of regression models.

Prediction in sport concerns many aspects including the prediction of performance results [2, 3] or predicting sporting talent [4, 5]. Models predicting results in sport, taking into account the seasonal statistics of each team, were also constructed [6]. The application of predictive models in athletics was described by Maszczyk et al. [2], where the regression was used to predict results in a javelin throw. These models were applied to support the choice and selection of prospective javelin throwers.

Prediction of sports results using linear regression was also presented in the work by Przednowek and Wiktorowicz [7]. A linear predictive model, implemented by ridge regression, was applied to predict the outcomes of a walking race after the immediate preparation phase. As input for the model, the basic somatic features (height and weight) and training loads (training components) for each day of training

were provided, and the output was the result expected over a distance of 5 km. In addition to linear models, artificial neural networks, whose parameters were specified in cross-validation, were also used to implement this task.

In the paper by Drake and James [8], the regressions estimating the results over distances of 5, 10, 20, and 50 km and the levels of the selected physiological parameters (e.g., $VO_2\max$) were presented. The regressions applied were the classical linear models, and the R^2 criterion was chosen for the quality evaluation. This study included 23 women and 45 men. The amount of collected data was different depending on the task and ranged from 21 to 68 records.

A nonlinear regression equation to predict the maximum aerobic capacity of footballers was proposed by Chatterjee et al. [9]. The data came from 35 young players aged from 14 to 16. The experiment was to verify the use of the 20 m MST (Multistage Shuttle Run Test) in evaluating the performance of $VO_2\max$. The talent of young hockey players was identified by Rocznioek et al. [5] using a regression equation. The research involved 60 boys aged between 15 and 16, who attended selection camps. The applied regression model classified candidates for future training, based on selected parameters of the players. The logistic regression was used in the model as the classification method.

The nonlinear predictive models used in sport are also based on the selected methods of “data mining” [10]. Among them, an important role is played by fuzzy logic expert systems. Papić et al. [4] described practical application of such a system. The proposed system was based on knowledge of experts in the field of sport, as well as the data obtained as a result of motor tests. The model suggested the most suitable sport and it was designed to search for prospective sports talents.

The application of fuzzy modeling techniques in sports prediction was also presented by Mężyk and Unold [11]. The goal of their paper was to find the rules that can express swimmer’s feelings the day after in-water training. The data was collected for two months among competitors practicing swimming. The swimmers were characterized by a good level of sports attainment (2nd sport class). The material obtained consisted of 12 attributes, and the total number of models was 480, out of which 136 were used in the final stage. The authors proved that their method was characterized by better predictive ability than the traditional methods of classification.

Other papers also concern the use of artificial neural networks in sports prediction [6]. Neural models are used to analyze the effectiveness of the training of swimmers, to identify handball players’ tactics, or to predict sporting talent [12]. Many studies present the application of neural networks in various aspects of sports training [13–15]. These models support the planning of training loads, practice control, or the selection of sports.

An approach developed by the authors is the construction of models performing the task of predicting the results achieved by a competitor in the proposed sports training. This allows for the proper selection of training components and thus supports the achievement of the desired result. The aim of this study is to determine the effectiveness of selected linear and nonlinear models in predicting the outcome in a 3-kilometer walking race for the proposed training. The research hypothesis of the paper is stated as follows: the prediction error of 3 kilometers’ result in race walking for nonlinear models can be smaller than for linear models.

The paper is organized as follows. In Section 2, the training data of the race walkers recorded during annual training cycle is described. Section 3 contains the methods used to build the linear and nonlinear predictive models, including ordinary least squares regression, regularized methods, that is, ridge, LASSO, and elastic net regressions, nonlinear least squares regression, and artificial neural networks as multilayer perceptron and radial basis function network. In Section 3, the criterion used to evaluate the performance of the models, calculated using mean square error in the process of cross-validation, is also defined. Section 4 describes the procedures used for building models and their evaluation in R language and STATISTICA software. The obtained results are analyzed and discussed in Section 5. Finally, in Section 6, the performed work is concluded.

2. Material

The predictive models were built using the training data of athletes practising race walking. The analysis involved

a group of colts and juniors from Poland. Among the competitors were the finalists in the Polish Junior Indoor Championships and the Polish Junior Championships. The data of race walkers was recorded during the 2011-2012 season in the form of training means and training loads. The training mean is the type of work performed while the training load is the amount of work at a particular intensity done by an athlete during exercise [1]. In the material, which has been collected, 11 means of training were distinguished. The material was drawn from the annual training cycle for the following four phases: transition, general preparation, special preparation, and starting phase. The training data has the form of sums of training loads completed in one month of the chosen training phase. The material included 122 training patterns made by 21 race walkers.

Control of the training process in race walking requires different tests of physical fitness at every training level. Because this research concerns the competitors in colt and junior categories, thus in order to determine a unified criterion of the level of training, a result for 3000 m race walking was used. The choice of the distance of 3000 m is valid because this is the indoor walking competition.

The description of the variables under consideration and their basic statistics are presented in Table 1. The variables are as follows: arithmetic mean of \bar{x} , minimum value x_{\min} , maximum value x_{\max} , standard deviation SD, and coefficient of variation $V = SD/\bar{x} \cdot 100\%$. The qualitative variables are X_1, X_2, X_3, X_4 , which take their values from the set $\{0, 1\}$. The other variables, that is, X_5, \dots, X_{18} , are quantitative variables. If the value at inputs X_1, X_2, X_3 is 0, it means that the transitional period is considered. Setting the value 1 on one of the inputs X_1, X_2, X_3 , it means the training period is selected. The variable X_4 represents the gender of the competitor, where the value 0 denotes a female, while the value 1 denotes a male, and the age is represented by X_5 . Basic somatic features of race walkers such as weight and height are presented in the form of BMI (X_6) expressed by the formula

$$\text{BMI} = \frac{M}{H^2} \left[\text{kg/m}^2 \right], \quad (1)$$

where M is the body weight [kg] and H is the body height [m]. The variable X_7 denotes the current result over 3 km in seconds. Training loads are characterized by the following variables: running exercises (X_8), walking with different levels of intensity (X_9, X_{10}, X_{11}), exercises forming different types of endurance (X_{12}, X_{13}, X_{14}), exercises forming techniques (X_{15}), exercises forming muscle strength (X_{16}), exercises forming general fitness (X_{17}), and warming up exercises (X_{18}).

An example of data used for building the model has the form

$$\begin{aligned} \mathbf{x}_5 &= [0, 1, 0, 0, \\ &23, 22.09, 800, 32, 400, 112, 20, 16, 32.4, 48, 8, \\ &280, 640, 400], \\ y_5 &= 800. \end{aligned} \quad (2)$$

TABLE I: The variables and their basic statistics.

Variable	Description	\bar{x}	x_{\min}	x_{\max}	SD	V [%]
Y	Result over 3 km [s]	936.9	780	1155	78.4	8.4
X_1	General preparation phase	—	—	—	—	—
X_2	Special preparation phase	—	—	—	—	—
X_3	Starting phase	—	—	—	—	—
X_4	Competitor's gender	—	—	—	—	—
X_5	Competitor's age [years]	18.9	14	24	3.0	15.6
X_6	BMI (body mass index) [kg/m ²]	19.3	16.4	22.1	1.7	8.7
X_7	Current result over 3 km [s]	962.6	795	1210	87.7	9.1
X_8	Overall running endurance [km]	30.9	0	56	10.6	34.4
X_9	Overall walking endurance in the 1st intensity range [km]	224.6	57	440	96.1	42.8
X_{10}	Overall walking endurance in the 2nd intensity range [km]	53.2	0	120	34.6	65.1
X_{11}	Overall walking endurance in the 3rd intensity range [km]	7.9	0	30	9.4	119.7
X_{12}	Short tempo endurance [km]	8.9	0	24	5	56.0
X_{13}	Medium tempo endurance [km]	8.3	0	32.4	8.6	103.2
X_{14}	Long tempo endurance [km]	12.9	0	56	16.1	125.0
X_{15}	Exercises forming technique (rhythm) of walking [km]	4.4	0	12	4.2	96.0
X_{16}	Exercises forming muscle strength [min]	90.2	0	360	104.8	116.3
X_{17}	Exercises forming general fitness [min]	522.0	120	720	109.9	21.0
X_{18}	Universal exercises (warm up) [min]	317.3	150	420	72.5	22.8

The vector \mathbf{x}_5 represents a 23-year-old race walker with BMI = 22.09 kg/m², who completes training in the special preparation phase. The result both before and after the training was the same and is equal to 800 s.

3. Methods

In this study, two approaches were considered. The first approach was based on white box models realized by modern regularized methods. These models are interpretable because their structure and parameters are known. The second approach was based on black box models realized by artificial neural networks.

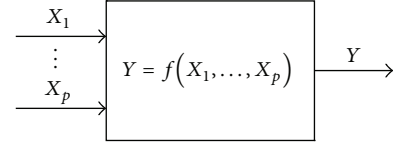


FIGURE 1: A diagram of a system with multiple inputs and one output.

3.1. *Constructing Regression Models.* Consider a multivariable regression model with the inputs (*predictors* or *regressors*) X_j , $j = 1, \dots, p$, and one output (*response*) Y shown in Figure 1. We assume that the model is linear and has the form

$$\begin{aligned} \hat{Y} &= w_0 + X_1 w_1 + \dots + X_p w_p \\ &= w_0 + \sum_{j=1}^p X_j w_j, \end{aligned} \quad (3)$$

where \hat{Y} is the estimated response and w_0, w_j are unknown weights of the model. The weight w_0 is called *constant term* or *intercept*. Furthermore, we assume that the data is standardized and centered and the model can be simplified to the form (see, e.g., [16])

$$\begin{aligned} \hat{Y} &= X_1 w_1 + \dots + X_p w_p \\ &= \sum_{j=1}^p X_j w_j. \end{aligned} \quad (4)$$

Observations are written as pairs (\mathbf{x}_i, y_i) , where $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]$, $i = 1, \dots, n$, x_{ij} is the value of the j th predictor in the i th observation, and y_i is the value of the response in the i th observation. Based on formula (4), the i th observation can be expressed as

$$\begin{aligned} \hat{y}_i &= x_{i1} w_1 + \dots + x_{ip} w_p \\ &= \sum_{j=1}^p x_{ij} w_j = \mathbf{x}_i \mathbf{w}, \end{aligned} \quad (5)$$

where $\mathbf{w} = [w_1, \dots, w_p]^T$. Introducing matrix \mathbf{X} in the form of

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \quad (6)$$

formula (5) can be written as

$$\hat{\mathbf{y}} = \mathbf{X} \mathbf{w}, \quad (7)$$

where $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_n]^T$.

In order to construct regression models, an error (residual) is introduced as the difference between the real value y_i and the estimated value \hat{y}_i in the form of

$$e_i = y_i - \hat{y}_i = y_i - \sum_{j=1}^p x_{ij} w_j = y_i - \mathbf{x}_i \mathbf{w}. \quad (8)$$

Using matrix (6), the error can be written as

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\mathbf{w}, \quad (9)$$

where $\mathbf{e} = [e_1, \dots, e_n]^T$ and $\mathbf{y} = [y_1, \dots, y_n]^T$.

Denoting by $J(\mathbf{w}, \cdot)$ the *cost function*, the problem of finding the optimal estimator can be formulated as to minimize the function $J(\mathbf{w}, \cdot)$, which means solving the problem

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} (J(\mathbf{w}, \cdot)), \quad (10)$$

where $\hat{\mathbf{w}}$ is the vector of solutions.

Depending on the function $J(\mathbf{w}, \cdot)$, different regression models can be obtained. In this paper, the following models are considered: ordinary least squares regression (OLS), ridge regression, LASSO (least absolute shrinkage and selection operator), elastic net regression (ENET), and nonlinear least squares regression (NLS).

3.2. Linear Regressions. In OLS regression (see, e.g., [16–18]) the model is calculated by minimizing the sum of squared errors:

$$\begin{aligned} J(\mathbf{w}) &= \mathbf{e}^T \mathbf{e} \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2, \end{aligned} \quad (11)$$

where $\|\cdot\|_2$ denotes the Euclidean norm (L_2). Minimizing the cost function (11), which is the quadratic function of \mathbf{w} , we get the following solution:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (12)$$

It should be noted that solution (12) does not exist if the matrix $\mathbf{X}^T \mathbf{X}$ is singular (due to correlated predictors or if $p > n$). In this case, different methods of regularization, including the previously mentioned ridge, LASSO, and elastic net regressions, can be used.

In ridge regression by Hoerl and Kennard [19], the cost function includes a penalty and has the form

$$\begin{aligned} J(\mathbf{w}, \lambda) &= \mathbf{e}^T \mathbf{e} + \lambda \mathbf{w}^T \mathbf{w} \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w} \\ &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2. \end{aligned} \quad (13)$$

The parameter $\lambda \geq 0$ determines the size of the penalty: for $\lambda > 0$, the model is penalized, for $\lambda = 0$, ridge regression

reduces to OLS regression. Solving problem (10) for ridge regression, we get

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}, \quad (14)$$

where \mathbf{I} is the identity matrix with the size of $p \times p$. Because the diagonal of the matrix $\mathbf{X}^T \mathbf{X}$ is increased by a positive constant, the matrix $\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}$ is invertible and the problem becomes nonsingular.

In LASSO regression by Tibshirani [20], similarly to ridge regression, the penalty is added to the cost function, where the L_1 -norm (the sum of absolute values) is used:

$$\begin{aligned} J(\mathbf{w}, \lambda) &= \mathbf{e}^T \mathbf{e} + \lambda \mathbf{z}^T \mathbf{w} \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{z}^T \mathbf{w} \\ &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1, \end{aligned} \quad (15)$$

where $\mathbf{z} = [z_1, \dots, z_p]^T$, $z_j = \operatorname{sgn}(w_j)$, and $\|\cdot\|_1$ denotes the Manhattan norm (L_1). Because problem (10) is not linear in relation to \mathbf{y} (due to the use of L_1 -norm), the solution cannot be obtained in the compact form as in ridge regression. The most popular algorithm used in this case is the LARS algorithm (least angle regression) by Efron et al. [21].

In elastic net regression by Zou and Hastie [22], the features of ridge and LASSO regressions are combined. The cost function in the so-called *naive elastic net* has the form of

$$\begin{aligned} J(\mathbf{w}, \lambda_1, \lambda_2) &= \mathbf{e}^T \mathbf{e} + \lambda_1 \mathbf{z}^T \mathbf{w} + \lambda_2 \mathbf{w}^T \mathbf{w} \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda_1 \mathbf{z}^T \mathbf{w} \\ &\quad + \lambda_2 \mathbf{w}^T \mathbf{w} \\ &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2. \end{aligned} \quad (16)$$

To solve the problem, Zou and Hastie [22] proposed the LARS-EN algorithm, which was based on the LARS algorithm developed for LASSO regression. They used the fact that elastic net regression reduces to LASSO regression for the augmented data set $(\mathbf{X}^*, \mathbf{y}^*)$.

3.3. Nonlinear Regressions. To take into account the nonlinearity in the models, we can apply the transformation of predictors or use nonlinear regression. In this paper, the latter solution is applied.

In OLS regression, the model is described by formula (5), while in more general nonlinear regression the relationship between the output and the predictors is expressed by a certain nonlinear function $f(\cdot)$ in the form of

$$\hat{y}_i = f(\mathbf{x}_i, \mathbf{w}). \quad (17)$$

In this case, the cost function $J(\mathbf{w})$ is formulated as

$$\begin{aligned} J(\mathbf{w}) &= \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^n (y_i - f(\mathbf{x}_i, \mathbf{w}))^2. \end{aligned} \quad (18)$$

Since the minimization of function (18) is associated with solving nonlinear equations, numerical optimization is used in this case. The main problem connected with the construction of nonlinear models is the choice of the appropriate function $f(\cdot)$.

3.4. Artificial Neural Networks. Artificial neural networks (ANNs) were also used for building predictive models. Two types of ANNs were implemented: a multilayer perceptron (MLP) and networks with radial basis function (RBF) [18].

The MLP network is the most common type of neural models. The calculation of the output in 3-layer multiple-input-one-output network is performed in *feed-forward* architecture. In the first step, m linear combinations, or the so-called *activations*, of the input variables are constructed as

$$a_k = \sum_{j=1}^p x_j w_{kj}^{(1)}, \quad (19)$$

where $k = 1, \dots, m$ and $w_{kj}^{(1)}$ denotes the weights for the first layer. From the activations a_k , using a nonlinear *activation function* $h(\cdot)$, hidden variables z_k are calculated as

$$z_k = h(a_k). \quad (20)$$

The function $h(\cdot)$ is usually chosen as logistic or “tanh” function. The hidden variables are used next to calculate the *output activation*

$$a = \sum_{k=1}^m z_k w_k^{(2)}, \quad (21)$$

where $w_k^{(2)}$ are weights for the second layer. Finally, the output of the network is calculated using an activation function $\sigma(\cdot)$ in the form of

$$y = \sigma(a). \quad (22)$$

For regression problems, the function $\sigma(\cdot)$ is chosen as identity function, and so we obtain $y = a$. The MLP network utilizes iterative supervised learning known as *error back-propagation* for training the weights. This method is based on *gradient descent* applied to the sum of squares function. To avoid the problem with *overtraining* the network, the number m of hidden neurons, which is a free parameter, should be determined to give the best predictive performance.

In the RBF network, the concept of radial basis function is used. Linear regression (5) is extended by linear combinations of nonlinear functions of the inputs in the form of

$$\hat{y}_i = \sum_{j=1}^p \phi_j(x_{ij}) w_j = \boldsymbol{\varphi}(\mathbf{x}_i) \mathbf{w}, \quad (23)$$

where $\boldsymbol{\varphi} = [\phi_1, \dots, \phi_p]^T$ is a vector of *basis functions*. Using nonlinear basis functions, we get a nonlinear model, which is, however, a linear function of parameters w_j . In the RBF network, the hidden neurons perform a radial basis function whose value depends on the distance from selected center c_j :

$$\boldsymbol{\varphi}(\mathbf{x}_i, \mathbf{c}) = \boldsymbol{\varphi}(\|\mathbf{x}_i - \mathbf{c}\|), \quad (24)$$

where $\mathbf{c} = [c_1, \dots, c_p]$ and $\|\cdot\|$ is usually the Euclidean norm. There are many possible choices for the basis functions, but the most popular is Gaussian function. It is known that RBF network can exactly interpolate any continuous function; that is, the function passes exactly through every data point. In this case, the number of hidden neurons is equal to the number of observations and the values of coefficients w_j are found by simple standard inversion technique. Such a network matches the data exactly, but it has poor predictive ability because the network is overtrained.

3.5. Choosing the Model. In this paper, the best predictive model is chosen using leave-one-out cross-validation (LOOCV) method [23], in which the number of tests is equal to the number of data n and one pair (\mathbf{x}_i, y_i) creates a testing set. The quality of the model is evaluated by means of the square root of the mean square error (RMSE_{CV}) defined as

$$\text{MSE}_{\text{CV}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_{-i})^2, \quad (25)$$

$$\text{RMSE}_{\text{CV}} = \sqrt{\text{MSE}_{\text{CV}}},$$

where \hat{y}_{-i} denotes the output of the model built in the i th step of validation process using a data set containing no testing pair (\mathbf{x}_i, y_i) and MSE_{CV} is the mean square error.

In order to describe the measure to which the model fits the training data, the root mean square error of training (RMSE_T) is considered. This error is defined as

$$\text{MSE}_{\text{T}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (26)$$

$$\text{RMSE}_{\text{T}} = \sqrt{\text{MSE}_{\text{T}}},$$

where \hat{y}_i denotes the output of the model built in the i th step using the full data set and MSE_{T} is the mean square error of training.

4. Implementation of the Predictive Models

All the regression models were calculated using R language with additional packages [24].

The `lm.ridge` function from “MASS” package [25] was used for calculating OLS regression (where $\lambda = 0$) and ridge regression (where $\lambda > 0$). With the function `enet` included in the package “elastic net” [26], LASSO regression and elastic net regression were calculated. The parameters of the `enet` function are $s \in [0, 1]$ and $\lambda \geq 0$, where s is a fraction of the L_1 norm, whereas λ denotes λ_2 in formula (16). The parameterization of elastic net regression using the pair (λ, s) instead of (λ_1, λ_2) in formula (16) is possible because elastic net regression can be treated as LASSO regression for an augmented data set $(\mathbf{X}^*, \mathbf{y}^*)$ [22]. Assuming that $\lambda = 0$, we get LASSO regression with one parameter s for the original data (\mathbf{X}, \mathbf{y}) .

All the nonlinear regression models were calculated using the `nls` function coming from the “stats” package [27]. It

calculates the parameters of the model using the nonlinear least squares method. One of the parameters of the `nls` function is a formula that specifies the function $f(\cdot)$ in model (18). To calculate the weights, Gauss-Newton's algorithm was used which was selected by default in the `nls` function. In all the calculations, it was assumed that the initial values of the weights are equal to zero.

For the implementation of artificial neural networks, StatSoft STATISTICA [28] software was used. The learning of MLP networks was implemented using the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm [18]. While calculating the RBF network, the parameters of the basis functions were automatically set by the learning procedure.

The parameters in all models were selected using leave-one-out cross-validation. In the case of regularized regressions, the penalty coefficients were calculated, while, in the case of neural networks, the number of neurons in the hidden layer was calculated. The primary performance criterion of the model was $RMSE_{CV}$ error. Cross-validation functions in the STATISTICA program were implemented using Visual Basic language.

5. Results and Discussion

From a coach's point of view, the prediction of results is very important in the process of sport training. A coach using the model, which was constructed earlier, can predict how the training loads will influence the sport outcome. The presented models can be used for predictions based on the proposed monthly training introduced as the sum of the training loads of each type implemented in a given month.

The results of the research are presented in Table 2; the description of the selected regressions will be presented in the next paragraphs. Linear models such as OLS, ridge, and LASSO regressions have been calculated by the authors in work [3]. They will be briefly described here. The nonlinear models implemented using nonlinear regression and artificial neural networks will be discussed in greater detail. All the methods will be compared taking into account the accuracy of the prediction.

5.1. Linear Regressions. The regression model calculated by the OLS method generated the prediction error $RMSE_{CV} = 26.90$ s and the training error $RMSE_T = 22.70$ s (Table 2). In the second column of Table 2, the weights w_0 and w_j are presented.

The search for the ridge regression model is based on finding the parameter λ , for which the model achieves the smallest prediction error. In this paper, ridge regression models for λ changing from 0 to 2 with step of 0.1 were analyzed. Based on the results, it was found that the best ridge model is achieved for $\lambda_{opt} = 1$. The prediction error $RMSE_{CV} = 26.76$ s was smaller than in the OLS model, while the training error $RMSE_T = 22.82$ s was greater (Table 2). The obtained ridge regression improved the predictive ability of the model. It is seen from Table 2 that as in the case of OLS regression, all weights are nonzero and all the input variables are used in computing the output of the model.

TABLE 2: Coefficients of linear models and linear part of nonlinear model NLS1 and error results.

Regression	OLS	RIDGE	LASSO, ENET	NLS1
w_0	237.2	325.7	296.6	2005
w_1	45.67	34.67	32.75	41.24
w_2	90.61	74.84	71.91	77.12
w_3	39.70	27.49	24.45	-3.439
w_4	-2.838	2.424		15.45
w_5	-0.9755	-1.770	-1.416	-22.44
w_6	1.072	0.5391		-24.71
w_7	0.7331	0.6805	0.7069	-1.782
w_8	-0.2779	-0.3589	-0.3410	-1.500
w_9	-0.1428	-0.1420	-0.1364	-0.0966
w_{10}	-0.1579	-0.0948	-0.0200	0.7417
w_{11}	0.7472	0.4352	0.0618	0.6933
w_{12}	0.4845	0.3852	0.1793	-0.6726
w_{13}	0.1216	0.1454	0.1183	-0.0936
w_{14}	-0.1510	-0.0270		2.231
w_{15}	-0.5125	-0.3070		0.7349
w_{16}	-0.0601	-0.0571	-0.0652	-0.2685
w_{17}	-0.0153	-0.0071		0.0358
w_{18}	-0.0115	-0.0403	-0.0220	-0.0662
$RMSE_{CV}$ [s]	26.90	26.76	26.20	28.83
$RMSE_T$ [s]	22.70	22.82	22.89	20.21

The LASSO regression model was calculated using the LARS-EN algorithm, in which the penalty is associated with the parameter s changing from 0 to 1 with step of 0.01. It was found that the optimal LASSO regression is calculated for $s_{opt} = 0.78$. The best LASSO model generates the error $RMSE_{CV} = 26.20$ s, which improves the results of OLS and ridge models. However, it should be noted that this model is characterized by the worst data fit with the greatest training error $RMSE_T = 22.89$ s. The LASSO method is also used for calculating an optimal set of input variables. It can be seen in the fourth column of Table 2 that the LASSO regression eliminated the five input variables (X_4 , X_6 , X_{14} , X_{15} , and X_{17}), which made the model simpler than for OLS and ridge regression.

The use of elastic net regression model has not improved the value of the prediction error. The best model was obtained for a pair of parameters $s_{opt} = 0.78$ and $\lambda_{opt} = 0$. Because the parameter λ is zero, the model is identical to the LASSO regression (fourth column of Table 2).

5.2. Nonlinear Regressions. Nonlinear regression models were obtained using various functions $f(\cdot)$ in formula (18). It was assumed that the function $f(\cdot)$ consists of two components: the linear part, in which the weights are calculated as in OLS regression, and the nonlinear part containing expressions of higher orders in the form of a quadratic function of selected predictors:

$$f(\mathbf{x}_i, \mathbf{w}, \mathbf{v}) = \sum_{j=1}^p x_{ij} w_j + \sum_{j=1}^p x_{ij}^2 v_j, \quad (27)$$

TABLE 3: Coefficients of nonlinear part of nonlinear models and error results (all coefficients have to be multiplied by 10^{-2}).

Regression	NLS1	NLS2	NLS3	NLS4
v_5	53.35	-0.3751	-0.3686	-0.6995
v_6	59.43	-1.0454	-1.3869	
v_7	0.1218	0.0003	0.0004	0.0001
v_8	1.880	0.0710	0.0372	-0.0172
v_9	-0.0016	0.0093	0.0093	0.0085
v_{10}	-0.6646	-0.0577	-0.0701	-0.1326
v_{11}	-3.0394	-0.3608	0.0116	0.8915
v_{12}	4.8741	0.3807	0.4170	1.0628
v_{13}	0.4897	-0.2496	-0.2379	-0.1391
v_{14}	-4.7399	-0.1141	-0.1362	
v_{15}	-13.6418	1.3387	0.8183	
v_{16}	0.0335	-0.0015	-0.0003	-0.0004
v_{17}	-0.0033	-0.0006	-0.0006	
v_{18}	0.0054	0.0012	0.0013	-0.0002
RMSE _{CV} [s]	28.83	25.24	25.34	24.60
RMSE _T [s]	20.21	22.63	22.74	22.79

where $\mathbf{w} = [w_1, \dots, w_p]^T$ is the vector of the weights of the linear part and $\mathbf{v} = [v_1, \dots, v_p]^T$ is the vector of the weights of the nonlinear part. The following cases of nonlinear regression were considered (Table 3), wherein each of the following models does not take into account the squares of qualitative variables X_1, X_2, X_3 , and X_4 ($v_1 = v_2 = v_3 = v_4 = 0$):

- (i) NLS1: both the weights of the linear part and the weights v_5, \dots, v_{18} of the nonlinear part are calculated.
- (ii) NLS2: the weights of the linear part are constant, and their values come from the OLS regression (the second column of Table 2); the weights v_5, \dots, v_{18} of the nonlinear part are calculated (the third column of Table 3).
- (iii) NLS3: the weights of the linear part are constant, and their values come from the ridge regression (the third column of Table 2); the weights v_5, \dots, v_{18} of the nonlinear part are calculated (the fourth column of Table 3).
- (iv) NLS4: the weights of the linear part are constant, and their values come from the LASSO regression (the fourth column of Table 2); the weights $v_5, v_7, \dots, v_{13}, v_{16}, v_{18}$ of the nonlinear part are calculated (the fifth column of Table 3).

Based on the results shown in Table 3, the best nonlinear regression model is the NLS4 model, that is, the modified LASSO regression. This model is characterized by the smallest prediction error and the reduced number of predictors.

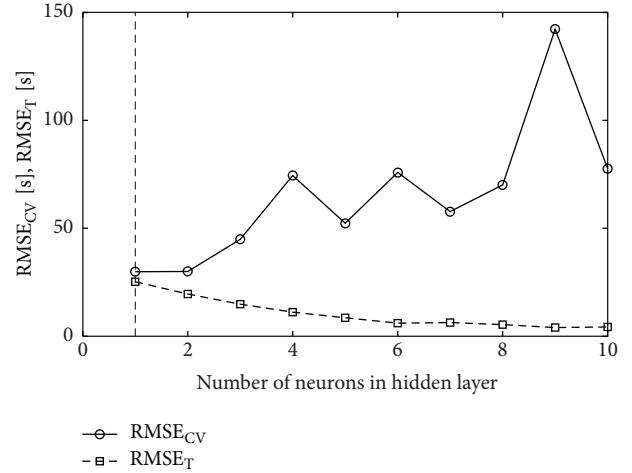


FIGURE 2: Cross-validation error (RMSE_{CV}) and training error (RMSE_T) for MLP(tanh) neural network; vertical line drawn for $m = 1$ signifies the number of hidden neurons chosen in cross-validation.

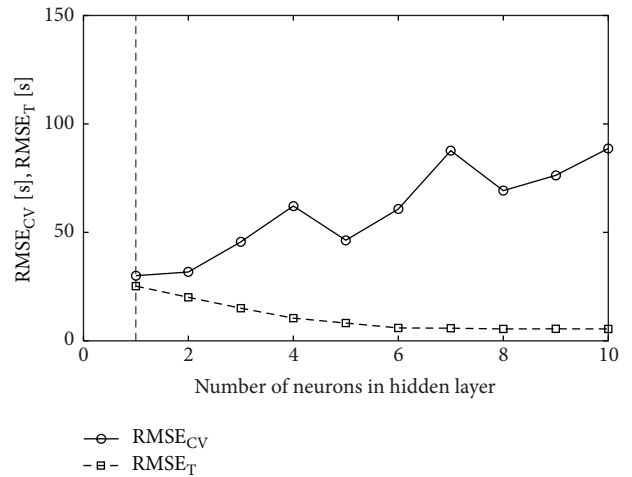


FIGURE 3: Cross-validation error (RMSE_{CV}) and training error (RMSE_T) for MLP(exp) neural network; vertical line drawn for $m = 1$ signifies the number of hidden neurons chosen in cross-validation.

5.3. Neural Networks. In order to select the best structure of a neural network, the number of neurons $m \in [1, 10]$ in the hidden layer was analyzed. In Figures 2, 3, and 4, the relationships between cross-validation error and the number of hidden neurons are presented. The smallest cross-validation errors for the MLP(tanh) and MLP(exp) networks were obtained for one hidden neuron (18-1-1 architecture) and they were, respectively, 29.89 s and 30.02 s (Table 4). For the RBF network, the best architecture was the one with four neurons in the hidden layer (18-4-1) and cross-validation error in this case was 55.71 s. Comparing the results, it is seen that the best model is the MLP(tanh) network with the 18-1-1 architecture. However, it is worse than the best regression model NLS4 (Table 3) by more than 5 seconds.

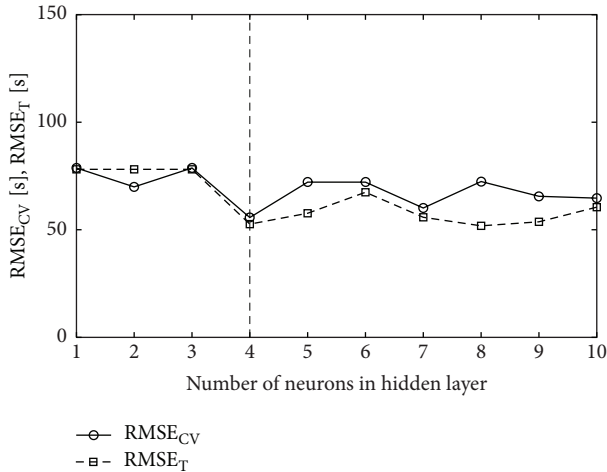


FIGURE 4: Cross-validation error ($RMSE_{CV}$) and training error ($RMSE_T$) for RBF neural network; vertical line drawn for $m = 4$ signifies the number of hidden neurons chosen in cross-validation.

TABLE 4: The number of hidden neurons and error results for the best neural nets.

ANN	MLP(tanh)	MLP(exp)	RBF
m	1	1	4
$RMSE_{CV}$ [s]	29.89	30.02	55.71
$RMSE_T$ [s]	25.19	25.17	52.63

6. Conclusions

This paper presents linear and nonlinear models used to predict sports results for race walkers. Introducing a monthly training schedule for a selected phase in the annual cycle, a decline in physical performance may be predicted based on the generated results. Thanks to that, it is possible to take into account earlier changes in the scheduled training. The novelty of this research is the use of nonlinear models, including modifications of linear regressions and artificial neural networks, in order to reduce the prediction error generated by linear models. The best model was the nonlinear modification of LASSO regression for which the error was 24.6 seconds. In addition, the method has simplified the structure of the model by eliminating 9 out of 32 predictors. The research hypothesis was confirmed. Comparing with other results is difficult because there is a lack of publications concerning predictive models in race walking.

Experts in the fields of sports theory and training were consulted during the construction of the models in order to maintain the theoretical and practical principles of sport training. The importance of the work is that practitioners (coaches) can use predictive models for planning of training loads in race walking.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of the paper.

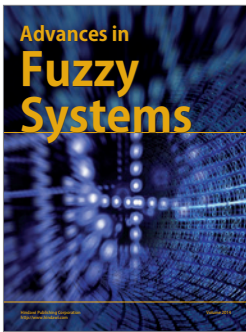
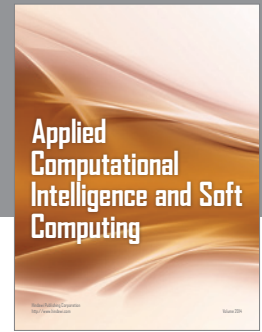
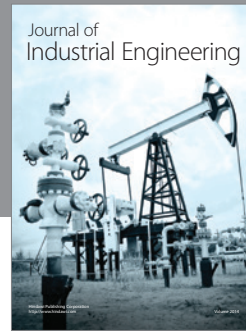
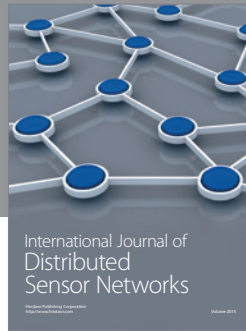
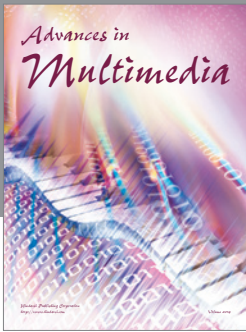
Acknowledgment

This work has been partially supported by the Polish Ministry of Science and Higher Education under Grant U-649/DS/M.

References

- [1] T. O. Bompa and G. Haff, *Periodization: Theory and Methodology of Training*, Human Kinetics, Champaign, Ill, USA, 1999.
- [2] A. Maszczyk, A. Zajac, and I. Ryguła, "A neural network model approach to athlete selection," *Sports Engineering*, vol. 13, no. 2, pp. 83–93, 2011.
- [3] K. Przednowek and K. Wiktorowicz, "Prediction of the result in race walking using regularized regression models," *Journal of Theoretical and Applied Computer Science*, vol. 7, no. 2, pp. 45–58, 2013.
- [4] V. Papić, N. Rogulj, and V. Pleština, "Identification of sport talents using a web-oriented expert system with a fuzzy module," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8830–8838, 2009.
- [5] R. Rocznio, A. Maszczyk, A. Stanula et al., "Physiological and physical profiles and on-ice performance approach to predict talent in male youth ice hockey players during draft to hockey team," *Isokinetics and Exercise Science*, vol. 21, no. 2, pp. 121–127, 2013.
- [6] M. Haghghat, H. Rastegari, N. Nourafza, N. Branch, and I. Esfahan, "A review of data mining techniques for result prediction in sports," *Advances in Computer Science*, vol. 2, no. 5, pp. 7–12, 2013.
- [7] K. Przednowek and K. Wiktorowicz, "Neural system of sport result optimization of athletes doing race walking," *Metody Informatyki Stosowanej*, vol. 29, no. 4, pp. 189–200, 2011 (Polish).
- [8] A. Drake and R. James, "Prediction of race walking performance via laboratory and field tests," *New Studies in Athletics*, vol. 23, no. 4, pp. 35–41, 2009.
- [9] P. Chatterjee, A. K. Banerjee, P. Dasb, and P. Debnath, "A regression equation to predict VO_2 max of young football players of Nepal," *International Journal of Applied Sports Sciences*, vol. 21, no. 2, pp. 113–121, 2009.
- [10] B. Ofoghi, J. Zeleznikow, C. MacMahon, and M. Raab, "Data mining in elite sports: a review and a framework," *Measurement in Physical Education and Exercise Science*, vol. 17, no. 3, pp. 171–186, 2013.
- [11] E. Męzyk and O. Unold, "Machine learning approach to model sport training," *Computers in Human Behavior*, vol. 27, no. 5, pp. 1499–1506, 2011.
- [12] M. Pfeiffer and A. Hohmann, "Applications of neural networks in training science," *Human Movement Science*, vol. 31, no. 2, pp. 344–359, 2012.
- [13] I. Ryguła, "Artificial neural networks as a tool of modeling of training loads," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 2985–2988, IEEE, September 2005.
- [14] A. J. Silva, A. M. Costa, P. M. Oliveira et al., "The use of neural network technology to model swimming performance," *Journal of Sports Science and Medicine*, vol. 6, no. 1, pp. 117–125, 2007.
- [15] A. Maszczyk, R. Rocznio, Z. Waškiewicz et al., "Application of regression and neural models to predict competitive swimming performance," *Perceptual and Motor Skills*, vol. 114, no. 2, pp. 610–626, 2012.

- [16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2009.
- [17] G. S. Maddala, *Introduction to Econometrics*, Wiley, Chichester, UK, 2001.
- [18] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.
- [19] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [20] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society, Series B: Methodological*, vol. 58, no. 1, pp. 267–288, 1996.
- [21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [22] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society—Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.
- [23] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.
- [24] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Development Core Team, Vienna, Austria, 2011.
- [25] B. Ripley, B. Venables, K. Hornik, A. Gebhardt, and D. Firth, *Package "MASS", Version 7.3–20*, CRAN, 2012.
- [26] H. Zou and T. Hastie, *Package "elasticnet", version 1.1*, CRAN, 2012.
- [27] R Development Core Team and Contributors Worldwide, *The R "Stats" Package*, R Development Core Team, Vienna, Austria, 2011.
- [28] StatSoft, *Statistica (Data Analysis Software System), Version 10*, StatSoft, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

