Software development based on artificial neural networks for fitness club

Vadim S. Tynchenko [1,2,3], Vladimir V. Bukhtoyarov [1,2], Aleksey N. Bocharov [1,2], Alexander V. Lavrishchev [1] and Yuriy N. Seregin [1]

[1]*Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarskiy rabochiy pr., Krasnoyarsk, 660037, Russian Federation*
[2]*Siberian Federal University, 79, Svobodny pr., Krasnoyarsk, 660041, Russian Federation*
[3]*Bauman Moscow State Technical University, 5/1, 2nd Baumanskaya st., Moscow, 105005, Russian Federation*

Abstract
An artificial neural network is a system of simple processors (artificial neurons) that are connected and interact with each other. Programming neural networks means training the network, not writing program code. Thanks to training, the network is able to identify dependencies between data (input and output), generalize, simplify the results, and use knowledge to break down complex problems into simpler ones. The aim of the presented work is to improve the efficiency of managing the needs of the clients of the fitness club through the use of modern methods of data analysis. As a technology for the implementation of such an analysis, the work uses neural networks of a special type - Kohonen networks. On the basis of the proposed approach, the final software product is being developed to improve the quality of management of a fitness club.

Keywords
Artificial neural networks, Kohonen maps, mathematical models, software system

1. Introduction

An artificial neural network is a mathematical model, as well as its software and hardware implementation, built on the principle of biological neural networks - the nerve cells of a living organism. This concept arose when trying to simulate the processes occurring in the human brain [1].

An artificial neural network is a system of simple processors (artificial neurons) that are connected and interact with each other. Each of the network processors deals with signals that are periodically received or transmitted to other processors. A large network is capable of solving the most difficult tasks in the shortest possible time [2].

From a mathematical point of view, neural networks are a way to solve nonlinear optimization problems. Cybernetics uses the theory of neural networks in solving problems of adaptive control, building algorithms for robotics [3].

Neural networks are a model of the structure and processes occurring in the cerebral cortex. An artificial neural network (ANN) is a set of formal neurons (FN) interconnected [4].

Biological neuron. The human nervous system is built from elements called neurons. The unique ability of a neuron is the reception, processing and transmission of electrochemical signals along the nerve pathways that form the communication system of the brain [5].

Proceedings of MIP Computing-V 2022: V International Scientific Workshop on Modeling, Information Processing and Computing, January 25, 2022, Krasnoyarsk, Russia
EMAIL: vadimond@mail.ru (Vadim S. Tynchenko); vladber@list.ru (Vladimir V. Bukhtoyarov); sibalexbo@gmail.com (Aleksey N. Bocharov); lav@optilink.pro (Alexander V. Lavrishchev); ius_ceregin@mail.ru (Yuriy N. Seregin)
ORCID0000-0002-3959-2969(Vadim S. Tynchenko); 0000-0003-4505-2851 (Vladimir V. Bukhtoyarov); 0000-0002-8021-0352 (Aleksey N. Bocharov); 0000-0003-4309-8637 (Yuriy N. Seregin)

Dendrites run from the nerve cell body to other neurons, where they receive signals at junction points called synapses. The input signals received by the synapse are fed to the body of the neuron. Here they are summed up, with some inputs striving to excite the neuron, others - to prevent its excitation. When the total excitation in the body of a neuron exceeds a certain threshold, the neuron is excited, sending a signal to other neurons along the axon. This basic functional scheme has many complications and exceptions, however, most artificial neural networks model only these simple properties [6].

There are several methods for training neural networks that we will focus on in this work - supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning.

The object of the research is the process of optimizing the selection of services and increasing profits by analyzing the system data, based on the fitness club "NRGym". The subject of research is an intelligent data analysis system based on the fitness club "NRGym". The scientific novelty in this study lies in the use of neural network technologies (information models), as well as a data analysis optimization system.

2. Material and methods

Let's consider in more detail the methods of training neural networks mentioned above.

2.1. Supervised learning

Supervised learning assumes the availability of a complete set of labeled data for training the model at all stages of its construction [7]. Figure 1 shows an example of supervised learning.

The presence of a fully labeled dataset means that each example in the training set corresponds to the answer that the algorithm should receive. Thus, a tagged dataset of flower photographs will train a neural network where roses, daisies or daffodils are depicted. When the network receives a new photo, it will compare it with examples from the training dataset to predict the answer [8].

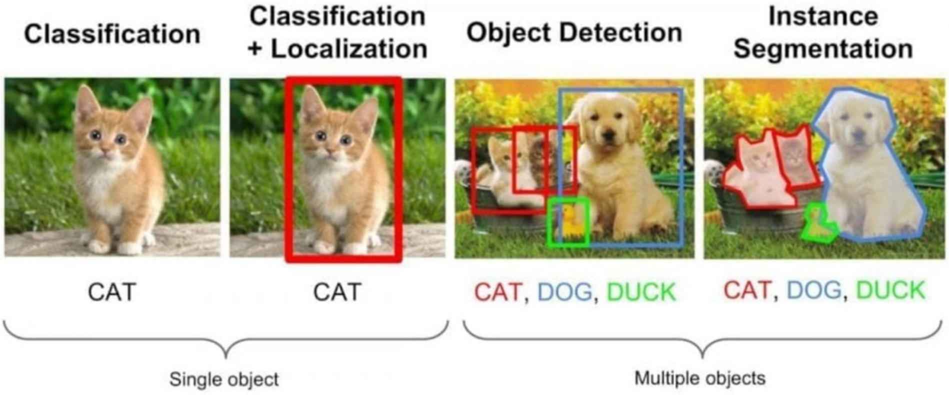


Figure 1: An example of supervised learning - classification (left), and its further use for segmentation and object recognition

Basically, supervised learning is used to solve two types of problems: classification and regression [9].

In classification problems, the algorithm predicts discrete values corresponding to the numbers of the classes to which the objects belong. In a training dataset with animal photos, each image will have a corresponding label - "cat", "koala" or "turtle". The quality of the algorithm is assessed by how accurately it can correctly classify new photos with koalas and turtles. [ten]

Regression tasks, on the other hand, are associated with continuous data. One example, linear regression, calculates the expected value of a variable y given specific x values.

More utilitarian machine learning tasks involve a large number of variables. For example, a neural network that predicts the price of an apartment in San Francisco based on its area, location, and public transport availability. The algorithm performs the work of an expert who calculates the price of an apartment based on the same data.

Thus, supervised learning is most suitable for tasks when there is an impressive set of reliable data for training the algorithm.

2.2. Unsupervised learning

Perfectly labeled and clean data is not easy to come by. Therefore, sometimes the algorithm is faced with the task of finding previously unknown answers. This is where learning without a teacher is needed [11].

In unsupervised learning, the model has a dataset, and there is no explicit indication of what to do with it. The neural network tries to independently find correlations in the data, extracting useful features and analyzing them., An example of such training is shown in Figure 2.

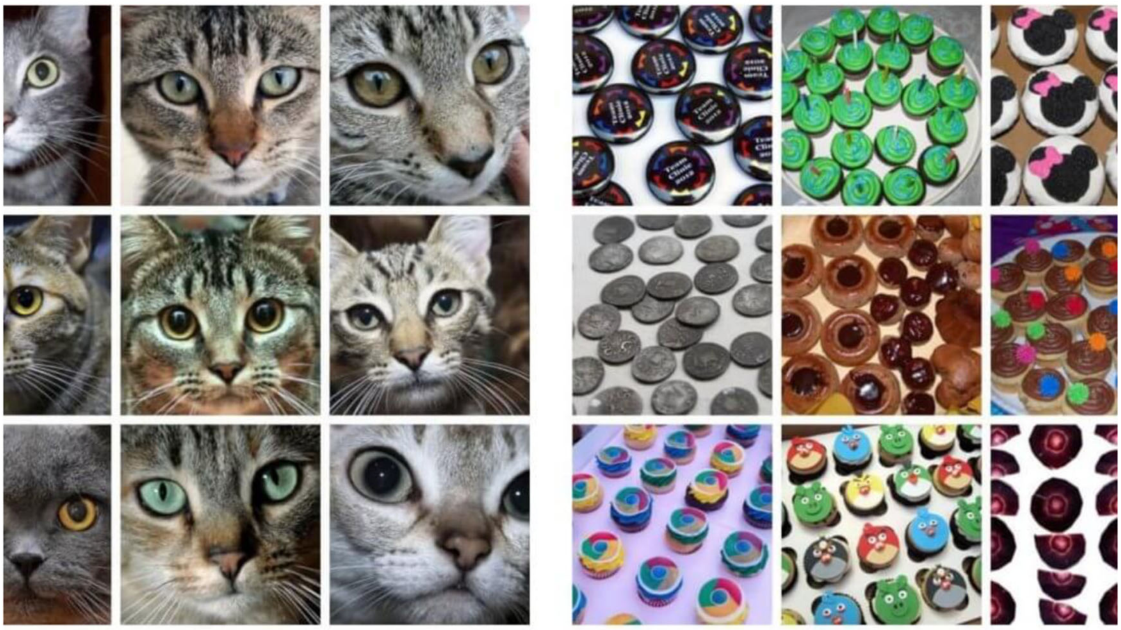


Figure 2: Clustering data based on common characteristics

Depending on the task, the model organizes the data in different ways. [12]:

- Clustering. Even without the special knowledge of an expert ornithologist, you can look at a collection of photographs and divide them into groups by bird species based on feather color, beak size or shape. The algorithm picks up similar data, finds common features, and groups them together
- Detection of anomalies. Banks can detect fraudulent transactions by identifying unusual behavior in customer buying behavior. For example, it is suspicious if the same credit card is used in California and Denmark on the same day. Similarly, unsupervised learning is used to find outliers in data
- Associations. Choose diapers, applesauce and baby sippy mug from the online store and the site will recommend that you add a bib and baby monitor to your order. This is an example of associations: some characteristics of an object are correlated with other characteristics. Considering a couple of key features of an object, the model can predict others with which there is a connection
- Autoencoders. Autoencoders take input, encode it, and then try to recreate the initial data from the resulting code. Using noisy and raw versions of images for training, autoencoders can remove noise from video data, images, or medical scans to improve data quality

2.3. Semi-supervised learning

Semi-supervised learning is characterized by its name: the training dataset contains both tagged and unlabeled data. This method is especially useful when it is difficult to extract important features from the data or to mark up all objects - a laborious task [13].

This machine learning technique is common for the analysis of medical images such as CT scans or MRI scans. An experienced radiologist can mark up a small subset of scans that show tumors and diseases. But manually marking up all scans is too time-consuming and expensive task. However, a neural network can extract information from a small fraction of labeled data and improve prediction accuracy compared to a model that trains exclusively on untagged data.

A popular training method that requires a small set of labeled data is to use a generative adversarial network (GAN).

2.4. Reinforcement learning

Video games are based on an incentive system. Complete the level and get a reward. Defeat all monsters and earn a bonus. Trapped - game over, don't get caught. These incentives help players understand how best to act in the next round of the game. Without feedback, people would just make random decisions and hope to move to the next level of play. Reinforcement learning follows the same principle (Figure 3). Video games are a popular test environment for research [14].

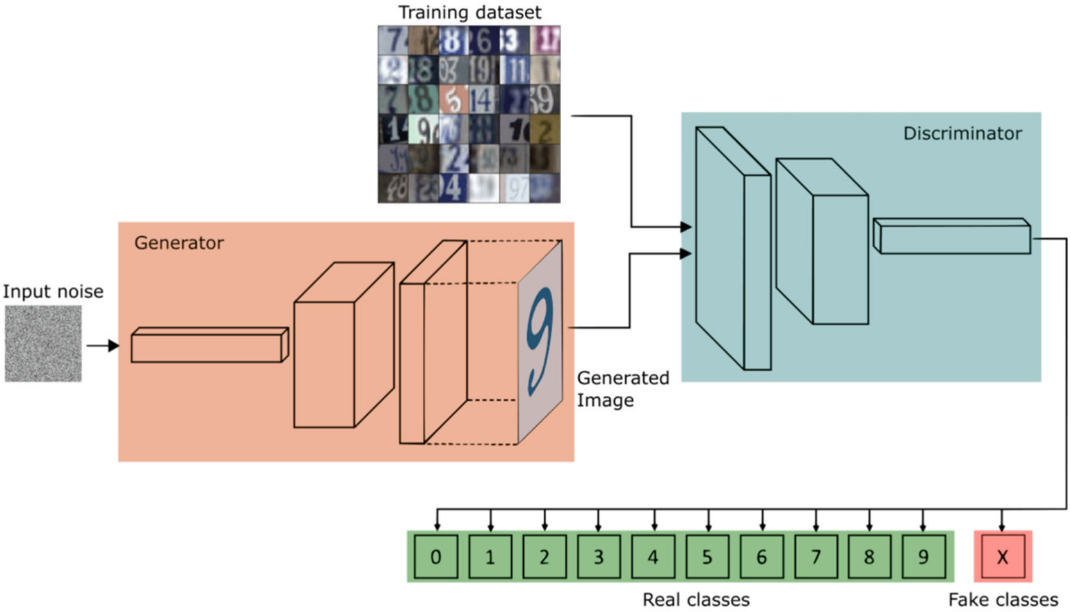


Figure 3: How GAN works

AI agents try to find the best way to achieve a goal or improve performance for a specific environment. When an agent takes actions that contribute to the achievement of a goal, he receives a reward. The global goal is to predict the next steps in order to earn the maximum reward in the end.

When making a decision, the agent studies feedback, new tactics and solutions that can lead to greater gains. This approach uses a long-term strategy - just like in chess: the next best move may not help you win in the long run. Therefore, the agent tries to maximize the total reward.

This is an iterative process. The more levels of feedback, the better the agent's strategy becomes. This approach is especially useful for training robots that operate autonomous vehicles or inventory in a warehouse.

Just like the students at school, each algorithm learns in a different way. But thanks to the variety of methods available, the question is to choose the right one and teach your neural network to understand the environment.

3. Result and discussion

A self-organizing map defines an ordered mapping, a kind of projection from a set of specified data items onto a regular, usually two-dimensional, grid. The model is associated with each grid point. These models are computed using the SOM algorithm. The item will be mapped to the node whose model is most similar to the item, that is, has the smallest distance from the item in some metric.

When implementing the SOM algorithm, the grid configuration (rectangular or hexagonal), as well as the number of neurons in the network, are preset. Some sources recommend using as many neurons as possible in the map. At the same time, the initial learning radius (neighborhood in the English-language literature) significantly affects the ability to generalize using the resulting map.

In the case when the number of map nodes exceeds the number of examples in the training set, then the success of using the algorithm largely depends on the appropriate choice of the initial training radius. However, in the case when the size of the map is tens of thousands of neurons, the time required to train the map is usually too long for solving practical problems. Thus, it is necessary to reach an acceptable compromise when choosing the number of nodes.

Before starting training the map, it is necessary to initialize the weights of the neurons. A well-chosen initialization method can significantly speed up learning and lead to better results. There are three ways to initiate initial weights.

Initialization with random values, when all weights are given small random values. Initialization by examples, when the values of randomly selected examples from the training set are set as initial values

Linear initialization. In this case, the weights are initiated by the values of vectors linearly ordered along a linear subspace between the two principal eigenvectors of the original dataset. Eigenvectors can be found, for example, using the Gram-Schmidt procedure.

Training consists of a sequence of corrections of vectors representing neurons. At each step of training, one of the vectors is randomly selected from the initial data set, and then a search is made for the vector of neurons coefficients that is most similar to it. In this case, the winner neuron is selected, which is most similar to the vector of inputs. Similarity in this problem is understood as the distance between vectors, usually calculated in Euclidean space.

After the winner neuron is found, the neural network weights are adjusted. In this case, the vector describing the winning neuron and the vectors describing its neighbors in the grid move in the direction of the input vector.

With this rendering method, the resulting map can be represented as a puff cake. Each layer of which is a coloring generated by one of the components of the original data. The resulting set of colorings can be used to analyze the patterns that exist between the components of the dataset.

After forming the map, we get a set of nodes that can be displayed as a two-dimensional image. In this case, each node of the map can be assigned a section in the figure, four or hexagonal, the coordinates of which are determined by the coordinates of the corresponding node in the lattice. Now, for visualization, it remains only to determine the color of the cells of this picture. For this, the values of the components are used.

The resulting colorings together form an atlas showing the location of the components, the connections between them, as well as the relative location of the various values of the components.

A cluster will be a group of vectors, the distance between which within this group is less than the distance to neighboring groups. The cluster structure when using the SOM algorithm can be displayed by visualizing the distance between the support vectors (neuron weights).

The most common use of this method is the unified distance matrix (u-matrix). For this, the distance between the vector of neuron weights in the grid and its nearest neighbors is calculated. These values are then used to determine the color that this node will be drawn with.

Usually grayscale is used, and the greater the distance, the darker the node is drawn. When used in this way, the nodes with the greatest distance between them and their neighbors correspond to black, and to nearby nodes - white.

In Figure 4, we observe the visualization of the neural network operation using Kohonen maps.

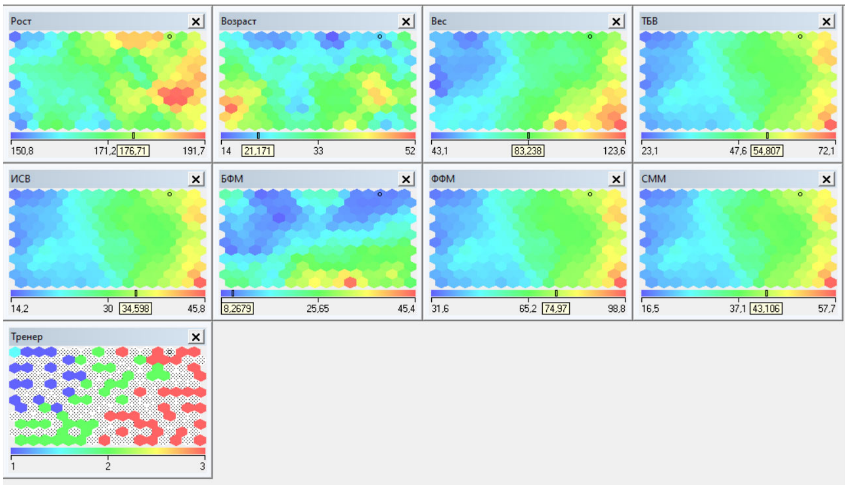


Figure 4: Kohonen Maps

In order to correctly design a software structure, it is necessary to form and analyze various requirements.

Business Requirements:

- Reducing the cost of enterprise management
- Achieve effective resource management
- Obtaining results of the required quality
- Ensuring the implementation of the most profitable initiatives
- Ensuring continuous interaction with the client

User requirements:

- Providing a choice of work scenario
- Ability to work with reports, depending on the selected user

Functional requirements (functions):

- Display of work progress
- Postponement of works execution
- Coordination of terms of performance and the fact of work performance
- Sorting, adding / deleting / editing data
- Data output in the form of reports

Based on these requirements, the menu structure of the software product was developed, which is shown in Figure 5.

The program consists of a single form in which the main functions are highlighted in separate tabs. User-editable lists open when loaded.

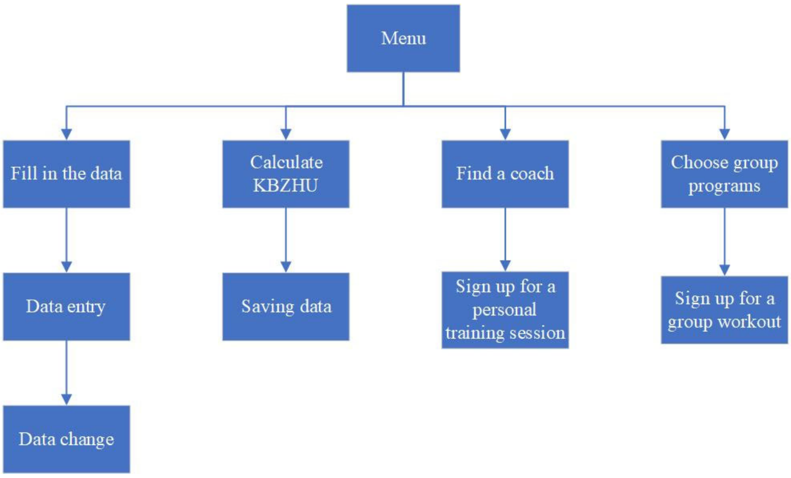


Figure 5: Menu structure

At the moment, the program consists of three forms, on which the main form, parameters of the modeled objects are displayed.

By changing the parameters of the simulated objects, the user has the ability to customize the results of the model, depending on the goals.

The system allows you to perform the following actions:

- View the main menu
- Entering data into the system
- View the trainers database
- Viewing the database of group programs
- Registration in the system for new users

Figure 6, 7, 8, 9 show the system interface.

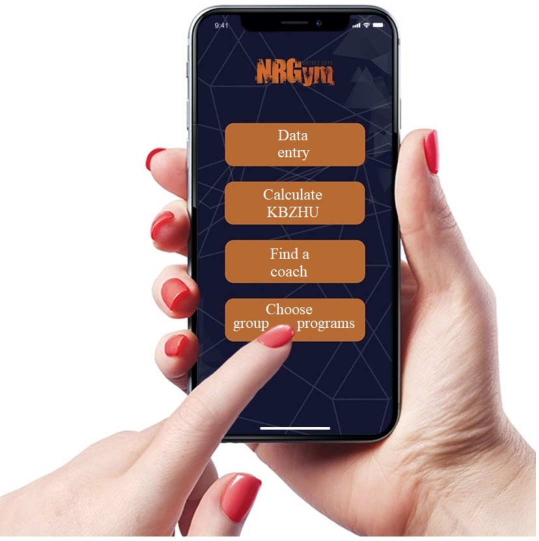


Figure 6: Home page, view of the application from a smartphone

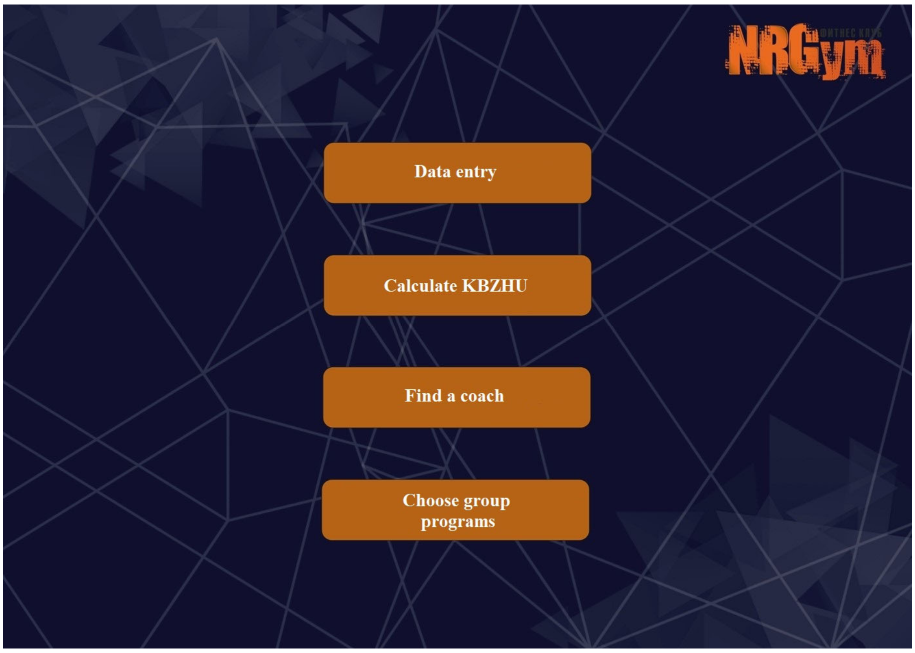
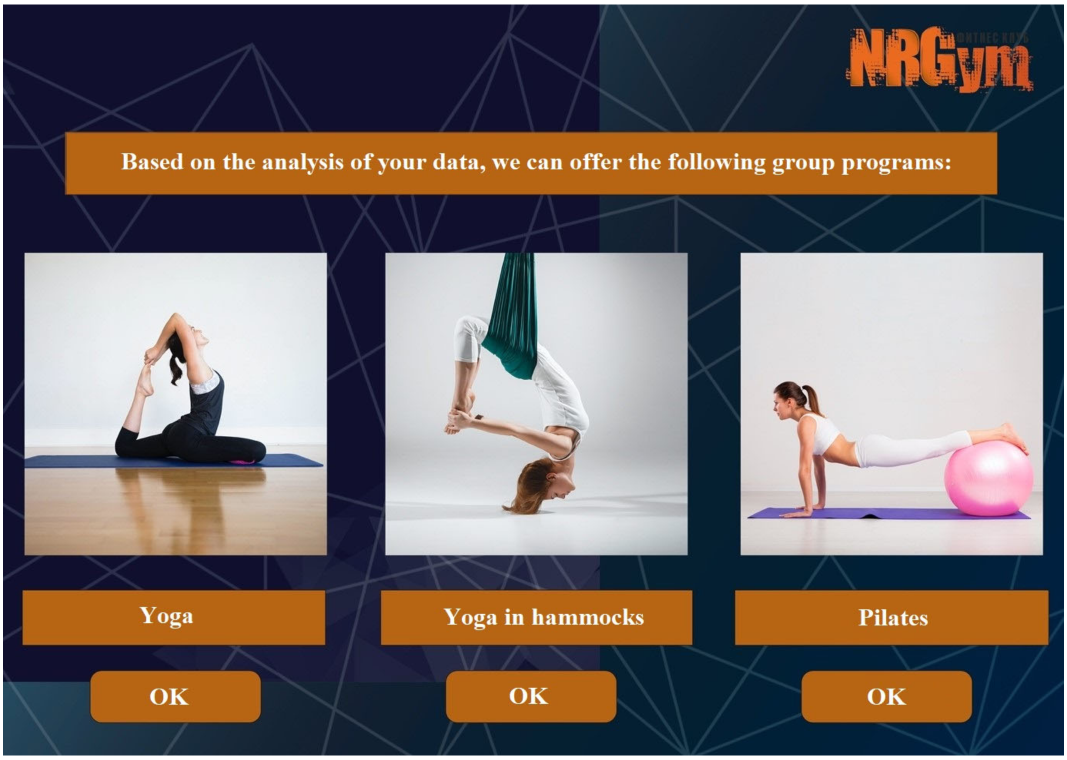


Figure 7: Application home page



Based on the analysis of your data, we can offer the following group programs:

Figure 8: Function interface “Selection of group programs”

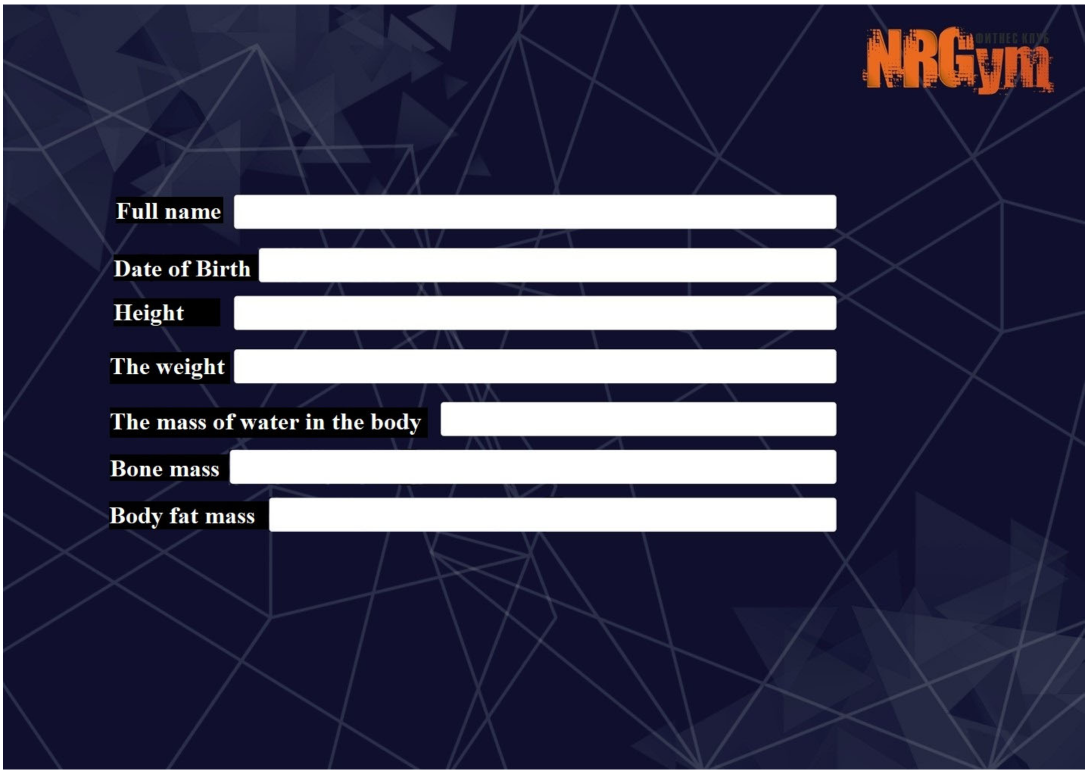


Figure 9: Data Entry function interface

4. Conclusion

An algorithm of the system's operation was developed and thought out, in particular, during the conceptual design, 12 entities were identified, of which 2 are reference. At the stage of logical design, all attributes and attribute domains were formed, and a logical model was built. A physical model of the system has been developed using the dbForge for MySQL database administration program with the structure of the database tables. The structure of the software product as a whole was developed, and the structure of interaction between the tabs of the main window was described in detail and the process of working with the system was described.

As a result of the current work, the authors have developed an Intelligent Data Analysis System to manage the needs of clients of the fitness club “NRGym”, which allows accumulating data on the health of clients, assessing the effectiveness of sports activities with various details, compiling, constructing and printing various reports and graphs for a certain time period, to optimize the process of assessing sports activity, to reduce the search time for the necessary information due to the ability to search and filter.

5. References

[1] P. G. Circle, Neural networks and neurocomputers, MPEI, Moscow, 2009.

[2] V. V. Kruglov, M. I. Dli, R. Y. Golunov, Fuzzy logic and artificial neural networks, Fizmatlit, Moscow, 2001.

[3] D. Graupe, Principles of Artificial Neural Networks, World Scientific Publishing Co. Pte. Ltd., New York, NY, 2013.

[4] D. Cox, A. Biel, F. Hoque, Experimental Design of Artificial Neural-Network Solutions for Traffic Sign Recognition, in: Proceedings of SAI Intelligent Systems Conference, Springer, Cham, 2021, pp. 362-372.

[5] A. Sallom, R. Fatahi, Z. Zamani, A. Ebadi, Optimization in vitro conditions for plum × apricot embryo rescue and modeling some critical factors by using artificial neural networks technology, Scientia Horticulturae 289 (2021) 110487.

[6] A. Monié, S. Franceschi, S. Balayssac, M. Malet-Martino, M. Delample, E. Perez, J. C. Garrigues Study of rapeseed oil gelation induced by commercial monoglycerides using a chemometric approach, Food Chem 1 (2021) 130870.

[7] R. Caruana, A. Niculescu-Mizil, An empirical Comparison of Supervised Learning Algorithms, in: Proc. 23rd Int. Conf. Mach. Learn. - ICML ’06, 2006, New York, NY, United States, pp. 161-168.

[8] Z. Huang, W. Xie, W. Liu, C. Song, Y. Zhang, J. Jing, TSCDNet +: A Highly Robust Substation Anomaly Detection Method, Optik 1 (2021) 167808.

[9] Q. Wang, W. Min, Q. Han, Q. Liu, C. Zha, H. Zhao, Z. Wei, Inter-Domain Adaptation Label for Data Augmentation in Vehicle Re-identification, IEEE Trans, Multimed (2021) 1.20.

[10] F. Ghani, U. Sattar, M. Narmeen, H. Wazir Khan, A. Mehmood, A Methodology for Glaucoma Disease Detection Using Deep Learning Techniques, Int. J. Comput. Digit. Syst (2021) 1-11.

[11] R. Ma, R. Angryk, Tiered Clustering for Time Series Data, in: Int. Conf. Deep Learn, Big Data Blockchain, Springer, Cham, 2021, pp 3-14.

[12] C. Sandoval, E. Pirogova, M. Lech, Adversarial Learning Approach to Unsupervised Labeling of Fine Art Paintings, IEEE Access 5 (2021) 81969-81985.

[13] W. Wang, T. Lin, D. He, F. Li, S. Wen, L. Wang, J. Liu, Semi-Supervised Temporal Action Proposal Generation via Exploiting 2-D Proposal Map, IEEE Trans, Multimed 10 (2021) 1-15.

[14] D. Daoun, F. Ibnat, Z. Alom, Z. Aung, M. A. Azim, Reinforcement Learning: A Friendly Introduction, in: Int. Conf. Deep Learn. Big Data Blockchain, Springer, Cham, 2021, pp 134-46.