

Development and Comparison of Optimization and Forecasting Methods for Creating an Individual Diet

IEVGEN FEDORCHENKO¹, ANDRII OLIINYK¹, ALEKSANDR STEPANENKO¹, MARIYA MIKHAYLOVA¹, TETIANA FEDORONCHAK¹, YEVGENY GOFMAN¹, MYKOLA KHOKHLOV²

¹Department of Software Tools, National University "Zaporizhzhia Polytechnic", 64 Zhukovskoho str., Zaporizhzhya, Ukraine, 69063.

Email: evg.fedorchenko@gmail.com, olejnikaa@gmail.com, alex@zntu.edu.ua, mariamihajlova31@gmail.com

²Department of Computer systems and networks, National University "Zaporizhzhia Polytechnic", 64 Zhukovskoho str., Zaporizhzhya, Ukraine, 69063.

Email: khokhlov@zp.edu.ua

Corresponding author: Ievgen Fedorchenko (e-mail: evg.fedorchenko@gmail.com).

This research was carried out with the support of the National University "Zaporizhzhia Polytechnic" within the framework of the scientific work "Development and comparison of optimization and forecasting methods for the formation of an individual diet."

ABSTRACT Personalization of the diet is becoming more and more relevant in the modern world, due to the growing interest in a healthy lifestyle and the need to take into account the individual characteristics of the body. Optimization and forecasting methods are used to develop effective personalized nutrition systems. This paper proposes an approach to the creation of an individual diet based on a comparison of four optimization algorithms, as well as three forecasting algorithms. The developed modeling system provides for the determination of the optimal distribution of food components, taking into account the individual daily calorie requirement. While the primary goal of any dietary recommendation system must be ensuring high nutritional quality and balance, the underlying algorithms were comparatively evaluated as a multi-criteria task based on their computational accuracy (compliance with the target calorie content), processing speed, and the variability of possible diet options generated. The results demonstrate the benefits of a combined approach that integrates optimization and machine learning methods to form a balanced and individually adapted diet.

KEYWORDS linear programming, genetic algorithm, Dense Neural Network, individual diet, diet optimization, basal metabolism

I. INTRODUCTION

Diet is one of the key factors that determine a person's physical and mental health, productivity, and overall quality of life. Forming a balanced menu requires taking into account many parameters, such as daily calorie intake, individual dietary restrictions, allergies, and personal food preferences, which makes this process difficult and time-consuming.

In the context of growing interest in a healthy lifestyle and the popularization of a personalized approach to nutrition, the creation of automated systems for planning a daily diet is becoming relevant. Such tools not only reduce the time for compiling a menu, but also contribute to the formation of useful eating habits of users. The use of artificial intelligence methods provides a high level of personalization of recommendations that take into account the individual needs and characteristics of the body.

To achieve optimal efficiency of personal diet planning systems, it is necessary to apply algorithmic approaches that

can take into account individual parameters, limitations and preferences of the user. The primary goal of automated diet formulation is to ensure the high nutritional quality and health adequacy of the recommended meal plans. However, from a computational perspective, the effectiveness of the underlying algorithms in solving this multi-criteria task is evaluated based on their accuracy in achieving the target calorie content, the speed of formation of the diet, and the number of diverse options provided to the user.

To address these challenges and improve existing solutions, the main contributions of this study are presented as follows:

- development of a novel domain-specific genetic algorithm: a modified genetic method was developed to optimize the daily diet. The novelty lies in the modification of the initialization and mutation stages, which directly embed the individual caloric needs of the user and dynamically prioritize dishes with products corresponding to their personal preferences;

- integration of machine learning for forecasting: the framework incorporates predictive models to assess and classify the generated meal plans, ensuring that the recommendations align with actual user dietary patterns and health profiles;
- multi-criteria comparative framework: a comprehensive comparative analysis of various classical optimization (linear programming, greedy algorithm, Monte Carlo) and forecasting methods for the formation of an individual diet was carried out.

The results obtained indicate that the application of optimal algorithms is the most effective in terms of ensuring the accuracy, speed of formation and variety of personalized diet options, confirming the feasibility of integrating optimization and machine learning methods to increase the efficiency of the system.

II. ANALYSIS OF THE SUBJECT AREA

A. TASK STATEMENT

The development of a personal assistant for the diet requires the integration of accurate calculations to determine the individual needs of each person. Formulas that take into account key physiological indicators are the basis for creating a reliable and effective algorithm. The system should provide an accurate calculation of daily caloric requirements based on mathematical models, integrating standardized measures such as body mass index (BMI) and basal metabolic rate (BMR). The use of BMI serves as a fundamental metric for classifying weight-related health risks, such as the growing global prevalence of obesity [1].

BMI is calculated using formula (1):

$$BMI = \frac{Weight}{Height^2}, \quad (1)$$

where *BMI* is the body mass index;

Weight is a person's weight in kilograms;

Height – height in meters.

For a more accurate determination of daily energy needs, BMR is used, which takes into account the individual physiological parameters of the user. There are various formulas for estimating BMR, among which the Mifflin-St Jeor equation has been validated in recent clinical studies as one of the most reliable predictive models for resting metabolic rate [2].

BMR for men is calculated using formula (2):

$$BMR = 88.36 + (13.4 \times Weight) + (4.8 \times (Height \times 100)) - (5.7 \times Age), \quad (2)$$

where *BMR* is basal metabolism;

Weight is a person's weight in kilograms;

Height – height in meters (multiplied by 100 to convert to centimeters);

Age – age in years.

BMR for women is calculated using formula (3) [31]:

$$BMR = 447.6 + (9.2 \times Weight) + (3.1 \times (Height \times 100)) - (4.3 \times Age) \quad (3)$$

The study used standardized anthropometric measures, including body mass index (BMI) and basal metabolic rate

(BMR). Based on their calculation, the need for body weight correction was assessed and the individual daily caloric intake was determined in accordance with the user's data

B. ANALYSIS OF LITERATURE DATA AND PROBLEM STATEMENT

Modern research in the field of personalized diet planning demonstrates a wide range of optimization and forecasting methods. Existing systems use both classical optimization algorithms and modern approaches based on machine learning.

In study [3], a scoping review of the application of linear programming techniques for the development of locally adapted dietary recommendations for children under five years of age was carried out using the Optifood program.

Linear programming (LP) is a mathematical optimization method that allows to obtain an optimal solution (for example, the minimum cost of a diet), adhering to a number of linear restrictions (on the amount of macro- and micronutrients), as indicated in dietary recommendations [4].

The model provided for minimizing deviations from the recommended norms of nutrients with a limitation of energy, cost and local availability of products. The results showed that the vast majority of macro- and micronutrients can be provided, but iron, zinc, calcium, as well as other trace elements remain systemic, depending on the age of the child (for example, folic acid, vitamins B and C). This highlights the need to combine LP applications with complementary strategies.

In the work of Khoshaim [4], a linear programming optimization model was proposed for a diet program. The author formulated the classic problem of “diet problem” in the form of minimizing costs while satisfying the daily norms of macro- and micronutrients. The general statement of the problem can be described using the following expression (4):

$$Z = \sum_{i=1}^n c_i x_i \rightarrow \min, \quad (4)$$

where c_i is the cost of a unit of product i ;

x_i – the amount of product included in the diet.

However, minimization of costs is possible only if certain conditions are met. They can be expressed by a system of constraints (5):

$$\sum_{i=1}^n a_{ij} x_i \geq b_j, j = 1, 2, \dots, m, \quad x_i \geq 0, i = 1, 2, \dots, m, \quad (5)$$

where nutrient a_{ij} in product i ;

b_j – recommended daily intake of nutrient j .

As a result, classical LP allows to obtain mathematically optimal solutions in terms of economy and nutritional value, but difficulties arise in practical application. In particular, the developed diets often fail to take into account cultural food preferences and may be unacceptable to consumers. To overcome these problems, a number of studies have proposed incorporating additional constraints, such as environmental impact (for example, carbon footprint), product diversity or social factors. The linear programming is an effective and transparent method, but for practical use it must be combined with multi-criteria approaches.

In papers [5-7], the Monte Carlo method is presented as a stochastic algorithm for estimating the probabilistic characteristics of complex systems by repeated random sampling. The main idea is to approximate the mathematical expectation of the function $f(x)$ using N independent simulation implementations, which can be expressed by formula (6):

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N f(x_i), \quad (6)$$

where x_i is a random vector of input data generated in accordance with a given probability distribution;

$\hat{\mu}$ – assessment of mathematical expectation;

N is the number of simulation experiments;

$f(x_i)$ is the value of the target function for a specific set of input parameters.

In [5], the application of the method made it possible to assess the influence of random and noisy parameters on the simulation results, which demonstrated high accuracy with a sufficient number of simulations, but required significant computational resources. Paper [6] describes a quasi-Monte Carlo approach that increases convergence through the use of deterministic sequences of points, but its implementation is more complex and not always universal. In [7], the Monte Carlo method was used to model medical processes with high variability of parameters, which made it possible to estimate probabilistic characteristics in detail, but the problem of high computational cost and complexity of model validation remained.

Thus, the Monte Carlo method is an effective tool for modeling systems with uncertain parameters and can be applied to optimize food rations in a stochastic environment, providing transparency and flexibility, but requires optimization of computational complexity.

In work [8], a greedy algorithm was used to form healthy daily menus, taking into account macro- and micronutrient restrictions. The algorithm consistently selected dishes that maximized compliance with the specified nutritional norms, providing basic requirements for calorie content and nutritional value. The results showed that the greedy approach is effective in quickly forming acceptable menu options, but there are limitations in the variety of diets and the need for further improvement to reduce monotony.

Also in study [9], a greedy algorithm was used to initialize initial decisions when planning diets taking into account dietary restrictions. Although the main focus was on machine learning methods, the greedy approach made it possible to quickly form an acceptable starting set of menus.

Therefore, the greedy algorithm demonstrates efficiency in quickly generating acceptable solutions and initial sets of menus, taking into account food restrictions.

However, its use is limited in providing variety and complete personalization of rations, which requires further modifications or combination with other optimization methods.

For complex dietary restrictions, multi-objective optimization models have proven highly effective. In [10], a multi-objective mathematical optimization model was applied to diet planning for people with diabetes, prioritizing the minimization of carbohydrate and lipid consumption while enforcing dietary diversity.

Alongside mathematical modeling, evolutionary algorithms are widely used. In paper [11], the genetic algorithm (GA) is presented as an evolutionary stochastic method of global optimization, operating with a population of decisions, genes and operators of selection, crossover and mutation. The main idea is to consistently improve the set of possible solutions by calculating the fitness function $F(x)$ for each individual and selecting the best ones for the formation of the next generation can be expressed by formula (7):

$$x_{t+1} = \text{Mutation} \left(\text{Crossover}(\text{Selection}(x_t)) \right), \quad (7)$$

where x_t is the current generation of solutions;

Selection, *Crossover*, *Mutation* – corresponding GA operators;

$F(x)$ is a function for assessing the quality of the diet.

In [10], the use of GA made it possible to find optimal daily rations, taking into account caloric and nutritional restrictions, which provided a high variety of acceptable solutions, but required significant computational resources for a large population. In paper [11], the method was improved by adaptive adjustment of the mutation probability and combination with trained models, which increased the convergence rate and optimization accuracy, but the problem of local minima in complex multidimensional spaces remained.

Thus, HA is a powerful tool for global optimization of complex diets, provides flexibility and variety of acceptable solutions, but requires a balanced setting of parameters for effective work.

However, the use of an optimization algorithm alone, even one as powerful as a genetic algorithm, often does not provide sufficient accuracy and adaptability of decisions in complex conditions of multidimensional constraints. To increase the efficiency of optimization, it is advisable to combine algorithms with forecasting methods, which allows to predict the influence of external and variable parameters on the optimization result. This combined use provides greater accuracy, variety, and stability in final solutions.

In modern research, forecasting methods are actively used to improve the accuracy of ration optimization and risk assessment. In particular, in work [12], the Random Forest method was used to predict the risk of developing type 2 diabetes based on changes in HbA1c levels. The model demonstrated high accuracy compared to traditional regression approaches and allowed the identification of important clinical predictors that went unnoticed by other methods. Whereas [12] employed the Random Forest algorithm, [13] adopted XGBoost combined with Bayesian hyperparameter optimization for disease risk classification. The approach reduced the time for selecting optimal hyperparameters and improved the generalization of the model, increasing the accuracy of forecasting. Finally, [14] applied DenseNet's deep neural networks to diagnose nutrient deficiencies in rice from leaf images; The model achieved an accuracy of 97.44%, which highlights the efficiency of neural networks in recognizing complex patterns. Thus, the combination of these prediction techniques can significantly improve the efficiency of optimization algorithms in complex systems, providing accuracy, flexibility, and the ability to process heterogeneous data.

The preceding analysis of the reviewed literature demonstrates that the optimization of food rations is a complex and urgent task that requires the use of different approaches. Classical optimization algorithms – the greedy method, linear programming, the Monte Carlo method and genetic algorithms – have their advantages and limitations:

- greedy algorithms quickly generate acceptable menus, but are limited in variety;
- linear programming is transparent and controllable, but requires modifications to increase the variety of diets;

- genetic algorithms are effective for global search in large and nonlinear spaces and increase the variety of acceptable diets.

Machine learning-based prediction techniques – Random Forest, XGBoost, and neural networks – allow for increased accuracy and flexibility of models: Random Forest highlights key factors, XGBoost improves hyperparameter generalization and optimization, and neural networks model complex nonlinear relationships.

Thus, for personalized ration planning, it is advisable to combine optimization algorithms and forecasting methods, which allows to adapt decisions to specific data, increase the accuracy and variety of rations.

C. PURPOSE AND OBJECTIVES OF THE STUDY AND PLANNED RESULTS

The purpose of the study is to develop and analyze methods for optimizing a personalized diet using classical and intelligent algorithms.

To achieve this goal, it is necessary to solve the following tasks:

- to analyze the existing optimization algorithms and determine their advantages and limitations in creating the diet;
- to explore the possibilities of combining the best optimization algorithm with forecasting methods to improve the accuracy of decisions;
- evaluate the quality of the solutions obtained according to the criteria of accuracy, speed, productivity, variety of dishes and the possibility of practical implementation.

The planned results of the study include the creation of a conceptual model of diet optimization that combines the advantages of different approaches and can become the basis for practical decision support systems in dietetics.

III. DEVELOPMENT AND ANALYSIS OF OPTIMIZATION ALGORITHMS AND METHODS FOR PREDICTING THE DIET TAKING INTO ACCOUNT INDIVIDUAL FOOD PREFERENCES

The analysis of recent scientific studies indicates a significant attention to the problems of optimizing diets, which is due to individual user preferences and practical limitations. There is a wide range of approaches to solving this problem – from classical mathematical methods to modern stochastic and evolutionary algorithms. Among the most common and effective approaches are such as the Monte Carlo method, greedy algorithms and genetic algorithms. To improve prediction accuracy, it is advisable to combine these approaches with machine learning-based forecasting methods. It was these methods that were chosen for the study within the framework of this work.

The linear algorithm is based on the integer linear programming approach [4], which allows to find the optimal diet, taking into account restrictions on the number of meals, portions and calories. Let us denote the number of servings of the dish i , and the binary variable showing whether the dish is selected. Each dish is characterized by a caloric content cal_i .

To ensure consistency in the choice of portions and the availability of the dish, the following restriction (8) is introduced:

$$y_i \leq x_i \leq M * y_i, \quad i = 1, \dots, n, \quad (8)$$

where M is the maximum number of servings of one dish.

This relationship ensures that the selected dish has at least one serving, while the unselected dish has zero servings.

According to food regulations, restrictions on the structure of meals are formulated. For example, if the sets B , L and D contain indices of dishes for breakfast, lunch and dinner, respectively, then (9):

$$\sum_{i \in B} y_i = 1, \sum_{i \in L} y_i = 1, \sum_{i \in D} y_i = 1, \quad (9)$$

where y_i is binary variable that acquires a value of 1 if dish i is included in the diet, and 0 if it is absent;

B is the set of indices of possible breakfast dishes;

L is the set of indices of possible dishes for lunch;

D is the set of indices of possible dishes for dinner.

Thus, conditions (9) ensure that exactly one dish will be chosen in the generated diet for each of the main meals. This restriction corresponds to the standard principles of structuring daily meals and avoids cases when the user is offered several “breakfasts” or their complete absence. To ensure a balanced daily diet, in addition to the main meals (breakfast, lunch and dinner), additional meals are also taken into account – snacks. To avoid an excessive number of selected dishes, additional restrictions are formulated (10):

$$\sum_{i \in S} y_i \leq 2, \sum_{i=1}^n y_i \leq 5, \quad (10)$$

where S is the set of indices of dishes that can act as snacks;

n is the total number of dishes in the database.

This formulation (10) ensures a rational structure of the menu: it does not become overly complex (too many dishes), but at the same time allows for the possibility of including additional snacks within reasonable limits.

To take into account the target caloric content G , an auxiliary variable is introduced $\delta \geq 0$, which characterizes the deviation of the actual caloric content from the desired value. Thus, the conditions for the caloric content of dishes in the diet can be written as (11):

$$\sum_{i=1}^n cal_i x_i - G \leq \delta, \quad G - \sum_{i=1}^n cal_i x_i \leq \delta. \quad (11)$$

where x_i is a binary variable showing the inclusion of dish I in the diet (1 is included, 0 is absent);

cal_i – calorie content of dish I ;

G is the desired daily calorie content;

δ is an auxiliary variable showing the permissible deviation of the calorie content of the diet from the target value.

The purpose of the algorithm is to minimize δ , that is, to find a diet that is as close as possible to the given daily calorie content.

Additionally, a mechanism for prohibiting repeated combinations of dishes is used, which allows to get several different optimal diet options. Formally, this can be expressed through a system of additional restrictions. For a diet already found $\{x_i^*\}$, the following solutions must satisfy the following conditions (12):

$$\sum_{i \in I} 1_{\{x_i = x_i^*\}} \leq |I| - 1, \quad (12)$$

where I is the set of indices of dishes in the previous solution,

$1_{\{x_i\}}$ – indicator function.

This ensures a variety of meal plans without changing the optimality of each individual solution.

The linear approach is advisable for tasks with a limited number of dishes, as it guarantees a globally optimal result and allows to accurately take into account the nutritional requirements of the user. As shown in studies [3, 4], such methods are effective in the creation of rations and resource planning.

After the application of linear optimization methods, which allow obtaining an accurate solution to the problem in cases of well-defined constraints and target functions, there is a need to use more flexible algorithms capable of working in conditions of uncertainty or large dimensionality of the decision space. One such approach is the Monte Carlo method, which is based on stochastic modeling and the use of random samples to approximate the search for optimal solutions.

The Monte Carlo method is a stochastic approach to optimization, which is based on multiple generation of random candidates and assessment of their quality using a target function, an approach successfully applied in modern healthcare modeling [15]. In the context of building a diet, each candidate diet consists of mandatory meals – breakfast, lunch and dinner – and additional snacks. The number of snacks can vary from zero to two, and it is permissible to serve the same dish more than once, which allows to get a larger number of servings of a particular product.

Let x_i be the number of servings of the dish i , and C_i – its calorie content. The total calorie content of the diet is calculated as in (13):

$$T(x) = \sum_{i=1}^n C_i x_i \quad (13)$$

The goal is to minimize the deviation from the given daily calorie content G , which is expressed by formula (14):

$$\delta = |T(x) - G| \quad (14)$$

At each iteration, the algorithm randomly selects basic dishes and generates the initial diet, taking into account the user's priorities for favorite ingredients. Then a local greedy search (hill-climb) is applied to correct the portions: at each step, the portions of each dish can increase or decrease by one if this brings the total calorie content closer to the target value.

The process of multiple generation of random rations and the application of local search allows to obtain a population of N candidates, which can be ranked according to the criterion of proximity to the caloric goal. This can be presented in the form of (15):

$$rank(x^{(k)}) = |T(x^{(k)}) - G| \quad (15)$$

where $x^{(k)}$ is the k -th candidate in the population (generated diet);

$T(x^{(k)})$ – calculated calorie content of the k -th diet;

$rank(x^{(k)})$ – the value of the criterion reflecting the deviation of the caloric content of the generated diet from the target value;

G is the set target daily calorie intake.

After sorting, the best M rations are selected, which ensures both the variability of solutions and their compliance with individual food preferences.

The advantage of the method is its ability to work in large and complex solution spaces, where classical linear methods can be limited [16].

The Monte Carlo method provides flexibility in building a multitude of solutions, allowing both the user's dietary restrictions and personal preferences to be taken into account at the same time, making it an effective tool in personalized meal planning.

The greedy algorithm for optimizing the diet is based on the step-by-step selection of the locally best product in terms of approaching the specified restrictions and target indicators. Unlike linear methods that work with the entire space of admissible solutions at the same time, the greedy approach is used to generate the solution gradually, adding only one element at each step.

The basic idea is that each iteration calculates the “utility” of the product, which is determined by the ratio between nutrient content and calorie content. Formally, such utility can be represented as (16):

$$U_i = \frac{\sum_{j=1}^m \omega_j p_{ij}}{c_i}, \quad (16)$$

where iU_i is the coefficient of “utility” of the i -th product,

p_{ij} – the content of the j -th nutrient in the product,

ω_j – a weight coefficient that determines the significance of this substance for an individual diet,

c_i – calorie content of the product.

Thus, the algorithm prefers those products that provide the maximum increase in nutrients per unit of energy value.

A feature of the greedy approach is that it provides a quick result, since it does not require going through a large number of combinations. However, its disadvantage lies in the risk of falling into the local optimum: early selection of products with a high utility factor can limit the possibility of achieving a more balanced decision in the future. That is why greedy methods are often used as an initial approximation to the solution, which can then be improved with more powerful optimization approaches, in particular genetic algorithms or Monte Carlo methods.

The genetic algorithm used to optimize the diet follows the principles of biological evolution, modeling the process of natural selection [17]. In this approach, each possible meal plan is interpreted as an individual of a population, where its “genes” are individual dishes and the number of servings of them. The primary population is generated randomly, however, taking into account the individual food preferences of the user, which allows the search scope to be narrowed down to more relevant combinations from the very beginning.

The quality of each individual is assessed using the fitness function [17]. It is determined by the deviation of the daily calorie content from the target value (17):

$$F(x) = |Calories(x) - C_{goal}| \quad (17)$$

where $Calories(x)$ is the total calorie content of individual x ;

C_{goal} – a given daily calorie intake.

The lower the $F(x)$ value, the better balanced the meal plan is considered.

At each iteration (generation), the best individuals are selected according to the principle of minimization $F(x)$. Then

the genetic search operators are applied: crossing and mutation. Crossing is implemented as a combination of a part of the diet of one “parent” with a part of the diet of another, i.e. (18):

$$Child = \{x_1, x_2, \dots, x_k\} \cup \{y_{k+1}, \dots, y_n\} \quad (18)$$

Mutation, in turn, consists in accidentally replacing one of the dishes or changing its number of servings. The probability of mutation is given by the parameter $\mu \in [0,1]$. This approach avoids premature convergence of the algorithm to the local minimum, since it introduces additional diversity into the population.

The process is repeated for a fixed number of generations of G, after which the individual with a minimum value of $F(x)$ is considered the best. It is important to note that by incorporating the user’s food preferences into the individual generation procedure, the algorithm not only optimizes the calorie content, but also generates solutions that are realistic in terms of the preferences and habits of a particular individual.

The scientific literature has repeatedly confirmed the effectiveness of genetic algorithms in solving combinatorial problems with a large number of constraints, in particular in the areas of planning and optimization of educational processes. In particular, paper [18] presented a modified genetic algorithm for automatic scheduling of university classes, which makes it possible to take into account numerous limitations and find effective solutions in the conditions of complex and multi-criteria problems.

Applying this approach to solving the problem of optimizing the diet allows to find a balance between the accuracy of compliance with dietary standards and the preservation of individual food preferences.

For a systematic comparison of the effectiveness of optimization algorithms, Table 1 is constructed, which demonstrates the key performance indicators of four methods: linear programming, greedy algorithm, Monte Carlo method, and conventional genetic algorithm. The selection of comparison criteria includes the time to carry out the ration search, the number of variants generated, performance, and the average and best deviation from the target caloric intake. Such a comprehensive presentation makes it possible to evaluate the advantages and limitations of each algorithm in the tasks of personalized selection of rations.

Table 1. Comparison of optimization algorithms by key criteria

Criterion	LP	Greedy algorithm	Monte Carlo Method	Genetic algorithm
Execution time (sec)	69,6	1,9	0,9	7,2
Number of Variants Generated	5	11	14	850
Performance (options/sec)	0,07 1	5,79	15,55	118,06
Average deviation from the target calorie intake	1,75	11,9	1,75	7,4
Deviation of the best diet from the target calorie intake	1,65	8,9	1,65	1,65

The analysis of the data in Table 1 allows us to draw a number of important conclusions about the effectiveness of the considered optimization algorithms. Firstly, according to the indicator of the execution time of the search for rations, the linear algorithm demonstrates the highest value, which is due

to the detailed enumeration of all possible combinations, while the Monte Carlo method is characterized by the minimum execution time. The greedy algorithm and the conventional genetic algorithm show intermediate values, while the execution time of the GA remains quite acceptable for practical application.

Secondly, the analysis of the number of generated variants shows that the genetic algorithm provides a much wider population of solutions (850 options), which increases the likelihood of finding an optimal or close to optimal diet. The linear algorithm and the Monte Carlo method generate a minimum number of options (5 and 14, respectively), which limits their ability to flexibly personalize the diet. Although the greedy algorithm is fast, it also generates a small number of options (11), which makes it less suitable for complex optimization tasks.

In terms of performance measured as the number of options per second, the genetic algorithm is significantly superior to all other methods (118.06 options/sec), which shows its high efficiency in processing a large number of combinations. The Monte Carlo method and the greedy algorithm show average performance, while the linear algorithm remains the least effective.

According to the criterion of average deviation from the target caloric content, the linear algorithm and the Monte Carlo method show the best result (1.75), while the GA demonstrates slightly higher values (7.4), which is explained by the greater variability of the formed rations. The greedy algorithm shows the worst result (11.9), which confirms its limitations for personalization tasks.

Finally, the deviation of the best diet from the target caloric intake is almost the same for the Monte Carlo methods, the linear algorithm and the HA (1.65), indicating that even with a large number of generated solutions, the genetic algorithm is able to provide a highly accurate approximation to the desired daily caloric intake. The greedy algorithm lags behind in this case, showing significantly more deviation.

So, on the basis of the comparison of optimization algorithms as an intermediate solution, it is determined that the genetic algorithm is the most effective for further work. It is advisable to use it in combination with forecasting methods to increase the accuracy and flexibility of the selection of a personalized diet.

At the same time, in order to increase the level of personalization and accuracy of recommendations, it is advisable to combine the chosen optimization method with forecasting methods. This ensures that the individual characteristics of the user are taken into account, the body’s needs are predicted and the diet is adapted to dynamic changes in food preferences and physiological parameters.

To train and evaluate the forecasting models (MLP, Random Forest, XGBoost), a robust and structured dataset was required. For this purpose, the publicly available “Diet Recommendations Dataset” [22] was utilized. This dataset contains 1,000 observations of patient health profiles—including age, gender, body mass index, medical conditions (e.g., diabetes, hypertension), and physical activity levels—along with their dietary preferences and target diet recommendations. The data was preprocessed, vectorized, and split into training and testing subsets to validate the models’ predictive capabilities and classify dietary plans effectively [23-28].

The first predictive method studied is a multilayer fully connected neural network (MLP). This approach belongs to the class of artificial neural networks of direct signal propagation and is characterized by the presence of one or more hidden layers with nonlinear activation functions. With its ability to approximate complex multivariate dependencies, MLP can effectively predict a user's nutrient and calorie needs based on input data such as anthropometric measurements, physical activity levels, and individual food preferences.

A multilayer fully connected neural network (also known as a multilayer perceptron, MLP) is a type of artificial neural network in which each neuron in one layer is connected to each neuron in the next layer.

This architecture allows for the simulation of complex, nonlinear relationships in data, making it highly effective for predicting health risks, such as diabetes, in vulnerable populations [19].

An alternative approach to forecasting, which has demonstrated high efficiency in classification and regression problems with multivariate input data, is the Random Forest method.

Random Forest is an ensemble method based on the construction of a set of solutions in the form of decision trees. Each tree in the ensemble makes a decision about whether the user will approve of a particular diet, based on the characteristics of the dishes and ingredients. The applicability of such machine learning models, including Random Forest, has been widely validated for precision nutrition and processing complex high-dimensional dietary data [20]. The final forecast is generated as an average of the forecasts of all trees in the forest.

The classic Random Forest prediction formula is as follows (19):

$$\hat{y} = \frac{1}{K} \sum_{k=1}^K T_k(x_{mealplan}), k = 1, \dots, K, \quad (19)$$

where T_k is the k -th tree in the ensemble;

$x_{mealplan}$ – input vector that describes the user's diet (includes vectorization of ingredients and names of dishes);

K is the number of trees.

Decision trees are formed on random subsamples of data using bootstrap, and splitting occurs on each node according to the feature that best minimizes the loss function. The most common criterion for binary classification is the Gini index (20):

$$Gini = 1 - \sum_{c=1}^C p_c^2, \quad (20)$$

where p_c is the proportion of class C among the samples in the node.

XGBoost (Extreme Gradient Boosting) is a gradient boosting method used to generate a sequence of decision trees, each of which tries to correct the errors of the previous trees, demonstrating high precision in personalized nutritional recommendation systems [21]. This approach allows to effectively model complex dependencies between the ingredients of dishes and user preferences.

The XGBoost forecast is calculated using formula (21):

$$\hat{y}_i = \sum_{t=1}^T f_t(x_i), f_t \in \mathcal{F}, \quad (21)$$

where T is the total number of trees;

x_i – vector of the incoming diet;

f_t is a function that describes the t -th tree in the ensemble.

For tree training, a regularized loss function is used (22):

$$\mathcal{L} = \sum_{i=1}^N l(y_i \hat{y}_i) + \sum_{t=1}^T \Omega(f_t), \Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^J w_j^2, \quad (22)$$

where $l(y_i \hat{y}_i)$ is the basic loss function (for example, binary cross-entropy);

w_j – weight of the j -th leaf of the tree;

γ, λ – regularization parameters;

J is the number of leaves in the tree.

The analysis of the data in Table 2 allows to assess in detail the advantages and disadvantages of each of the forecasting methods.

Table 2. Comparison of the forecasting methods by key criteria

Criterion	MLP	Random Forest	XGBoost
Training time, (sec)	3,1	0,275	0,187
Sorting time, (sec)	0,335	0,304	0,291
RAM usage, (MB)	0,012	0.0552	0.1355

The analysis of Table 2 shows that:

Firstly, according to the criterion of training time, the XGBoost algorithm is the fastest, which provides high speed due to the use of a boosting strategy and effective optimization mechanisms. Random Forest also shows good results with low training times. In turn, the basic multilayer perceptron (MLP) exhibits the longest learning time (3.1 s), which is due to the more complex structure of the model and the need for numerous iterations to achieve convergence. However, even this time remains acceptable for practical applications, as it is measured in seconds.

Secondly, in terms of sorting time, all three methods show similar results. This indicates that at the stage of classification and forecasting, the difference between them practically does not affect the speed of the system.

The most significant differences are manifested in the use of RAM. Here, MLP shows the best result (0.012 MB), which is significantly lower compared to Random Forest (0.0552 MB) and XGBoost (0.1355 MB). This criterion is the key one in the context of this study, as the system must process large amounts of data when personalizing food rations. Minimal memory usage ensures scalability and stable performance without overloads.

The main advantage of MLP in the context of this study is the minimal use of RAM, which is critical for processing large amounts of data in personalized dietary selection.

IV. MODIFIED OPTIMIZATION ALGORITHM AND FORECASTING METHOD

As part of the study, a modified genetic algorithm (GA) was created, built on the basis of an elitist model (elitist GA) with domain-specific rules for the generation and mutation of individuals. Basic GA involves binary chromosome coding, random population initialization, classical crossing and mutation operators. The proposed approach departs from the classical model, while retaining the general evolutionary paradigm, but applies the principle of adaptation to the subject area – the development of the balanced food plans.

Each individual I in population P is represented in the form of a meal plan (MealPlan), i.e., an ordered set of meals with

appropriate portions, which can be represented by the expression (23):

$$I = \{(d_1, s_1), (d_2, s_2), \dots, (d_k, s_k)\}, k \in \{3, 4, 5\}, \quad (23)$$

where d_i – a dish selected from a variety of available ones (breakfasts, lunches, dinners, snacks);

$s_i \in \{1, 2, \dots, 5\}$ – servings.

Then the total calorie content of an individual is determined by formula (24):

$$C(I) = \sum_{i=1}^k s_i * c(d_i), \quad (24)$$

where $c(d_i)$ is the calorie content of one serving of the dish d_i .

The goal of optimization is to minimize deviation from the given caloric goal C^* . Based on this, the fitness function is formulated as (25):

$$f(I) = |C(I) - C^*| \quad (25)$$

Thus, the lower the value $f(I)$, the better the individual is considered.

The first modification was the initialization of the population adapted to the task. Unlike classical GA, where the initial population is generated randomly, this algorithm uses domain-specific initialization. When creating an individual with a probability of 0.5, dishes are selected that correspond to the individual preferences of the user, i.e. (26):

$$P(d \in D_{fav}) = 0.5, P(d \in D \setminus D_{fav}) = 0.5, \quad (26)$$

where D_{fav} is the set of dishes containing ingredients from the list of user preferences.

The second modification used elite selection, which involves the formation of a new population based on the most suitable individuals of the current generation. This can be expressed by formula (27):

$$P' = \operatorname{argmin}_{I \in P} f(I), |P'| = \frac{1}{2} |P|, \quad (27)$$

where P is the current population, which consists of a set of solutions (rations);

$f(I)$ – target function that assesses the quality of each decision I (for example, deviation from the target caloric content);

$\operatorname{argmin}_{I \in P} f(I)$ – means the selection of those individuals from the population for whom the value of the function $f(I)$ is minimal, i.e., those closest to the optimal solution;

P' – a new population that is formed from the best individuals;

$|P'| = \frac{1}{2} |P|$ – exactly half of the individuals from the initial population passes to the new one, and it is the half that has the best indicators (the smallest deviation from the target function).

Unlike roulette or tournament selection, this method guarantees the safety of the most adapted individuals.

The third modification concerns the crossing operation. It was adapted to the structure of individuals in the task of creating a diet. For this, a single-point cross is used: two parental decisions are taken, a random dividing point is chosen, and a new individual is formed as a combination of the first part of one parent and the second part of the other. This allows to

naturally combine dishes and portions from different solutions, creating more varied and potentially better diets.

The fourth modification refers to the mutation process, which in genetic algorithms is responsible for randomly making changes to current decisions in order to avoid stagnation and allow the exploration of new variants. It can occur in two versions:

- change in the number of servings – for example, for a certain dish, the number of servings is replaced with another (from 1 to 5). This allows to flexibly adjust the calorie content to a given goal;
- meal substitution – the selected dish in the meal plan can be replaced with another from its category (breakfast, lunch, dinner or snack). At the same time, the algorithm controls that there are no repetitions of the same dishes in the diet, that is, uniqueness is always preserved.

Thus, the algorithm is an elite genetic algorithm with modifications at the stages of initialization, selection, crossing and mutation, which take into account the subject specifics of the task (developing balanced food plans) that allows to improve the quality of solutions by reducing the search space due to user preferences and preserving the best individuals at every step with adapted mutation operators that prevent degradation of solutions.

To assess the effectiveness of optimization algorithms and the modified GA algorithm, it is advisable to compare them according to a number of criteria, including accuracy and the number of results obtained, speed, and performance. Such a comparison allows to identify the strengths and weaknesses of each approach, as well as display the improved results achieved by the modified optimization algorithm.

The analysis showed that different optimization algorithms differ significantly in efficiency. In Table 3 we can see this comparison.

Comparison by the criterion of execution time showed a significant difference between the algorithms which can be seen in Table 3. The longest process is observed in the case of linear programming – 69.6 s, which is explained by the need to take into account a significant number of constraints and variables. The greedy algorithm showed a much higher speed of 1.9 s, but its effectiveness is limited by locally optimal solutions. The Monte Carlo method turned out to be the fastest – 0.9 s, due to random generation of solutions, but the quality of such results is not always stable.

Table 3. Comparison of optimization algorithms by performance criteria

Performance criteria	Linear programming	Monte Carlo	Modified genetic algorithm	Greedy algorithm
Lead time (sec)	69.6	0.9	5.5	1.9
Number of Variants Generated	5	14	1000	11
Algorithm performance (options/sec)	0.071	15.55	181.81	5.79
Deviation from the target calorie content on average (calories)	1.75	1.75	7.1	11.9
Deviation of the best diet from the target calorie intake (calories)	1,65	8,9	1,65	1,65

The modified genetic algorithm is inferior in speed to the greedy method and Monte Carlo (5.5 s), but provides a significantly better quality of optimization due to evolutionary selection and the ability to work with a large decision space.

One of the important criteria for the effectiveness of optimization algorithms is the number of solutions generated, which reflects the variety of possible configurations of the optimal diet. This indicator is the key one in case of developing personalized recommendations, since a larger number of options provides more options for choosing according to the individual user preferences. The analysis of the number of generated variants seen in Table 3 shows that the modified genetic algorithm demonstrates the highest ability to generate variable decisions. Thanks to the use of evolutionary operators – crossover mutation and selection – it provides a much larger space of possible combinations, which allows to find both locally and globally optimal options. In quantitative terms, the modified GA generated 1000 variants, while other methods showed much more modest results: linear programming – 5 options, greedy algorithm – 11, Monte Carlo method – 14. Such a gap in favor of GA indicates its high flexibility and suitability for working with multidimensional tasks, where a variety of possible solutions is important. In the practical context of rationing, this means that the user receives a wider set of personalized offers, which increases the quality and usefulness of recommendations.

Another key evaluation criterion is the performance of the algorithms, which is advisable to measure in terms of the number of variants generated per unit of time. This approach allows to assess the efficiency of the use of computing resources and the practical suitability of algorithms for the tasks of creating a diet in real time. The results obtained, and the performance of optimization algorithms reflected in Table 3, confirm that the modified genetic algorithm demonstrates the highest performance among all the methods considered. Its indicator is 181.81 options per second, which is ten times higher than the results of other approaches. For comparison: the Monte Carlo method provided only 15.55 options/sec, the greedy algorithm provided 5.79, and linear programming provided only 0.071 options/sec. This situation has developed because the linear algorithm has a long execution time and, accordingly, a small number of variant generation. The high performance of GA is due to the evolutionary mechanisms of crossover and mutation, which allow for efficient exploration of the decision space while maintaining population diversity. Instead, traditional methods based on rigid deterministic logic generate a much smaller number of alternatives, which limits their applications in practical scenarios where the rapid generation of a large number of personalized options is important.

One of the key indicators of the effectiveness of algorithms for creating an optimal diet is the accuracy of achieving the target values of daily calorie intake. To do this, the average deviation of the resulting rations from the specified caloric content is analyzed, which allows to assess the balance of solutions and their practical suitability. The results of this comparison, presented in Table 3, demonstrate the different ability of the algorithms to achieve accuracy regarding the target caloric intake. The linear programming method and the Monte Carlo method showed the best results, providing a minimum average deviation from the setpoint, which is only 1.75 calories. This indicates their high ability to accurately reproduce the specified parameters. Although the modified

genetic algorithm yielded a slightly higher result (7.1 calories), it remains quite accurate, especially given its ability to generate a large number of alternative options and strike a balance between accuracy and variety of solutions. On the other hand, the greedy algorithm turned out to be the least effective on this criterion, showing an average deviation of 11.9 calories, which indicates its limitation in achieving caloric balance and explains the lack of versatility of this approach in personalized diet selection.

And the last criterion is the deviation of the best generated ration from the target daily calorie content. This indicator allows to assess how the algorithm is able to find the best diet option as close as possible to the specified calorie intake, regardless of the total number of generated options. The comparison results seen in Table 3 demonstrate that the modified genetic algorithm, together with the linear programming method and the Monte Carlo method, provides a minimum deviation from the target caloric intake for the best diet, equal to 1.65 calories. This confirms the high accuracy of these approaches in finding optimal solutions for creating a balanced diet. At the same time, the greedy algorithm showed a much higher result (8.9 calories), which limits its effectiveness in tasks where adhering to a given calorie content is paramount.

The data obtained indicate that the modified genetic algorithm combines the advantages of accuracy and a significant variety of options, which makes it the most suitable for personalized dietary selection tasks.

To increase the accuracy and adaptability of a personalized diet creation, along with the modification of the genetic optimization algorithm, the prediction method was also modified. This allows to more correctly take into account individual food preferences.

In particular, custom pre-processing of data was applied, which made it possible to more flexibly take into account the features of the input features. In addition, the model architecture was extended with two embedding layers (instead of the traditional one), which provided a deeper representation of discrete features. Additionally, we used embedding vectors as a whole, but not as individual elements, which made it possible to better preserve semantic relationships between ingredients. Also, a combination of ingredients and names of dishes in the form of a single vector representation was used for entry, while usually such features are processed separately. To increase the convenience of learning and monitoring the process, our own Logger with custom notifications was integrated, which allowed us to control key parameters and results at each stage. Thanks to these modifications, it was possible to adapt the MLP to the specifics of the task and achieve better prediction accuracy.

MLP consists of an input layer, one or more hidden layers, and an output layer. Each neuron performs a linear transformation of its input data, after which it applies a nonlinear activation function. This allows the network to learn to recognize complex patterns and relationships. Fig. 1 shows the architecture of a neural network.

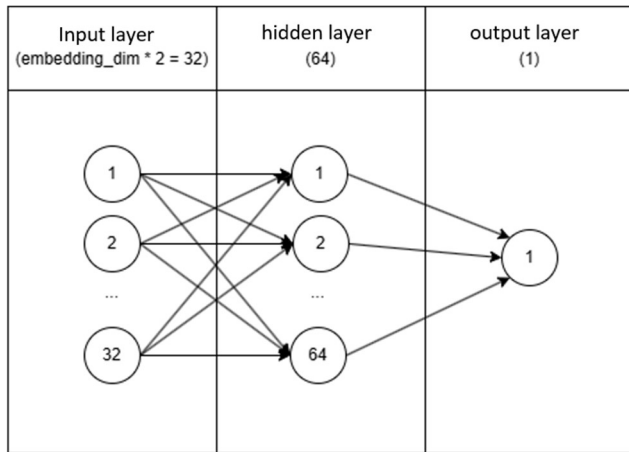


Figure 1. MLP Architecture

The input layer consists of two components: vectorization of ingredients and names of dishes.

Ingredient Vectorization – The Embedded Representations (ingredient_embedding) layer converts ingredient indices into embedding dimension vectors. Calculating the average value of ingredient vectors provides a generalized representation for each dish.

Vectorization of dish names – similarly, the dish_name_embedding layer converts the indexes of dish names into vectors of the dimension embedding.

The resulting vectors are combined by concatenation, forming an input dimension vector, which is transmitted to the hidden layer calculated by formula (28):

$$embedding\_dim = 2 \times embedding, \quad (28)$$

where $embedding\_dim$ is the input vector of dimension;
 $embedding$ – dimension vector.

The latent layer is fully coherent and contains 64 neurons. Each neuron receives input from all neurons in the previous layer. The linear transformation of the input vector is carried out through the Linear operator with the following ReLU activation function, which is calculated using formula (29):

$$h = ReLU(W_x + b), \quad (29)$$

where h is the vector of the outputs of the hidden layer;

b is the displacement vector;

x is the input vector;

W is the matrix of weights dimensions, which is calculated by formula (30):

$$W = (64 \times 2embedding\_dim), \quad (30)$$

The output layer is represented by a single neuron that receives input signals from each of the 64 neurons in the hidden layer. A linear transformation with the Sigmoid activation function converts the total signal into a probability value, which is calculated using formula (31):

$$y = \sigma(w_o h + b_o), \quad (31)$$

where y is the value of probability;

w_o is the weight of the original layer;

b_o – displacement;

$\sigma(x)$ is the matrix of weights dimensions, which is calculated according to formula (32):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (32)$$

The output neuron predicts the probability of a positive assessment of the diet, which allows the ration to be classified as approved or disapproved.

Also, for a better display of the architecture, an image is created in a graphical form in Fig. 2.

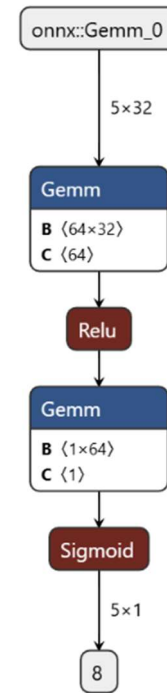


Figure 2. MLP graphical display

To analyze the effectiveness of the modified forecasting method, a comparative analysis was carried out, the results of which can be seen in Table 4.

Table 4. Comparison of forecasting methods by performance criteria

Performance criteria	Modified Multilayer Fully Connected Neural Network	Random Forest	XGBoost
Study time(sec)	2.3 s	0.275 s	0.187 s
Sorting Time(sec)	0.328 s	0.304 s	0.291 s
Memory usage(MB)	0.0081	0.0552	0.1355

One of the key criteria for evaluating the effectiveness of prediction methods is the model training time. This indicator determines the speed at which the algorithm is able to adapt to the available data and form predictive models, which is especially important when working with large amounts of information or in real time. Table 4 shows that the longest training time is demonstrated by a modified multilayer fully connected neural network, but its value remains acceptable and is about 2.3 seconds, which is less than the time in a regular MLP. The Random Forest and XGBoost methods showed fast learning time: 0.275 and 0.187 seconds respectively.

Another important indicator of the effectiveness of forecasting methods is the time of sorting the results obtained. This criterion characterizes the speed of ordering the predicted

options, which is important for further integration with the optimization algorithm and the creation of the final personalized diet. In Table 4, the results show that the longest sorting time takes modified multilayer fully connected neural network about 0.328 seconds, but other methods yield similar results: 0.304 and 0.291 seconds. That is, the difference is insignificant.

Another important criterion for the effectiveness of forecasting methods is the amount of RAM used. This indicator reflects the resource intensity of the algorithm and is important when choosing a method for integration with the algorithm for optimizing and processing large amounts of data. The analysis of the results in Table 4 shows that the most economical in terms of memory consumption is a modified multilayer fully connected neural network. The Random Forest method occupies an intermediate position, while XGBoost shows the highest memory consumption among the three algorithms, which indicates its high resource consumption in data processing.

So, modified MLP has better performance than regular MLP and also other prediction methods.

To summarize the proposed methodology, including the interaction between the modified optimization algorithm and the machine learning forecasting models, the complete workflow is presented in Algorithm 1.

Algorithm 1: integrated diet optimization and forecasting framework.

Input: user profile (age, gender, weight, height, activity level, preferences), target datasets.

Output: top recommended and ML-approved meal plans.

Step 1: Initialization & Baseline Calculation

1. Calculate $BMI = \frac{Weight}{Height^2}$.

2. Calculate Basal Metabolic Rate (BMR) using Mifflin-St Jeor equations (Eq. 2, Eq. 3).

3. Calculate target daily calorie intake (C^*) based on BMR and activity level.

Step 2: Optimization phase (Modified genetic algorithm)

4. Initialize population P based on domain-specific rules & user preferences (Eq. 26).

5. For generation = 1 to MAX_GEN do:

6. Evaluate fitness $f(I) = |C_I - C^*|$ for each meal plan I in P .

7. Select elite population P' retaining the best 50% of plans (Eq. 27).

8. Generate offspring via single-point crossover (Eq. 18).

9. Apply mutation:

- adjust servings (1 to 5);
- substitute meal maintaining category (breakfast, lunch, etc.).

10. Update population P with elites and new offspring.

11. End For.

12. Sort and extract top optimized meal plans from P .

Step 3: Forecasting phase (Machine learning evaluation)

13. Load trained predictive models (MLP, Random Forest, XGBoost).

14. Select the optimal model (modified MLP) for final evaluation.

15. For each optimized meal plan I do:

16. Vectorize ingredients and dish names using embedding layers (Eq. 28).

17. Pass vector to hidden layers with ReLU activation (Eq. 29).

18. Calculate approval probability $y = \sigma(W_o h + b_o)$ (Eq. 31).

19. If $y > \text{approval\_threshold}$ then:

20. Categorize plan I as “Approved”.

21. End For.

22. Return the final list of Approved optimized meal plans.

V. DISCUSSION OF RESULTS

The primary goal of a personalized diet creation is to maximize the nutritional quality and adequacy of the recommended meals. Treating this as a multi-criteria optimization task, the comparison of algorithms showed that the genetic algorithm, especially in its modified form, is the most effective computational tool: it generates a highly diverse range of diet options, demonstrates robust productivity in finding optimal solutions, and provides a balanced deviation from the target calorie content while respecting user-specific data and preferences.

Other methods have demonstrated their advantages in narrow aspects (e.g., the linear method in caloric accuracy), but are inferior to GA in terms of comprehensive efficiency.

Regarding forecasting methods, the comparison showed that the modified MLP is the most suitable for the forecasting task in this study.

It provides low training times, comparable sorting times, and minimal RAM usage, which is critical for working with large amounts of data.

Their application made it possible to significantly improve the results compared to standard approaches: the modified GA provided higher performance and variety of solutions while maintaining acceptable accuracy, while the modified MLP showed better adaptability to data specifics and more efficient use of resources.

Despite the promising results, the proposed framework has several specific limitations. First, the reliance on standardized predictive equations like Mifflin-St Jeor for basal metabolic rate estimation inherently carries physiological inaccuracies, as it does not fully account for individual variations in body composition (e.g., fat-free mass versus fat mass) or metabolic adaptations. Second, the machine learning models were trained on a specific dataset of 1,000 observations, which may limit the generalization of the system across highly diverse global populations with unique cultural dietary patterns or complex concurrent medical conditions. Finally, while the modified genetic algorithm effectively optimizes for caloric targets and basic food preferences, the current fitness function does not deeply evaluate complex micronutrient synergies. To increase the quality and clinical applicability of the proposal, future improvements should focus on integrating real-time physiological feedback from wearable health devices (such as continuous glucose monitors) to dynamically adjust daily caloric targets. Additionally, expanding the training datasets with larger, real-world clinical cohorts and evolving the fitness function into a stricter multi-objective model that penalizes specific vitamin or mineral deficiencies will significantly enhance the framework's robustness.

VI. CONCLUSIONS

During the study, optimization algorithms and forecasting methods for personalized selection of the diet, which consider the individual food preferences of the user, were developed and analyzed.

The scientific novelty of the work lies in the creation of a modified GA for optimization and a modified MLP for prediction.

Thus, the optimal combination for personalized dietary selection includes modified GA as an optimization method and modified MLP as a prediction method.

This allows to achieve high accuracy, productivity and efficient use of resources.

From a practical point of view, the developed system can be integrated into mobile applications or web services for automated selection of a diet, which opens up prospects for its application in the field of dietetics, sports medicine and healthy lifestyle support systems.

References

- [1] NCD Risk Factor Collaboration (NCD-RisC), "Worldwide trends in underweight and obesity from 1990 to 2022: a pooled analysis of 3663 population-representative studies with 222 million children, adolescents, and adults," *The Lancet*, vol. 403, no. 10431, pp. 1027-1050, 2024. [https://doi.org/10.1016/S0140-6736\(23\)02750-2](https://doi.org/10.1016/S0140-6736(23)02750-2).
- [2] A. Rozenberg, et al., "Validation of common predictive equations for resting metabolic rate," *Nutrients*, vol. 15, no. 4, 805, 2023. <https://doi.org/10.3390/nu15040805>.
- [3] Y. X. Miow, W. K. H. Mok, W. Y. Gan, P. Y. Lim, G. Appannah, & S. N. A. Adznam, "The use of linear programming approach in diet optimization among children under five: a scoping review," *BMC Public Health*, vol. 25, art. no. 1279, 2025. <https://doi.org/10.1186/s12889-025-22414-y>.
- [4] A. B. Khoshaim, "A linear programming optimization model for a diet program," *Review of Business and Finance Studies*, vol. 12, no. 1, pp. 31-39, 2021.
- [5] A. Shah, & M. Khan, "A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning and ABC," *International Statistical Review*, vol. 93, no. 1, pp. 18-61, 2024. <https://doi.org/10.1111/insr.12573>.
- [6] L. Wang, & Y. Li, "A review of Monte Carlo and quasi-Monte Carlo sampling techniques," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 5, pp. 1-20, 2021.
- [7] H. Zhang, & J. Liu, "A review of Monte Carlo methods and their application in medical physics," *University of New Hampshire Honors Theses*, pp. 1-15, 2022.
- [8] I. Chatzipantsiou, N. Tsigotakis, & S. Demetriou, "An extensive search algorithm to find feasible healthy menus for humans," *Operational Research*, vol. 22, no. 5, pp. 5231-5267, 2022. <https://doi.org/10.1007/s12351-022-00702-4>.
- [9] D. Khandoker, M. Akter, & R. Rahman, "Diet planning with machine learning: Teacher-forced REINFORCE for composition compliance with nutrition enhancement," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3150-3160. <https://doi.org/10.1145/3447548.3467201>.
- [10] T. F. Soares, M. C. Escarpinati, & P. H. R. Gabriel, "Multi-objective optimization applied to diet planning for people with diabetes," *Pesquisa Operacional*, vol. 44, 2024. <https://doi.org/10.1590/0101-7438.2023.043.00285156>.
- [11] R. Karim, M. B. Biplob, & M. S. Arefin, "Developing a genetic algorithm based daily calorie recommendation system for humans," *International Journal of Computer Science and Information Technology*, vol. 16, no. 3, pp. 45-52, 2024. <https://doi.org/10.5121/ijcsit.2024.16307>.
- [12] T. Ooka, H. Johno, K. Nakamoto, Y. Yoda, & H. Yokomichi, "Random forest approach for determining risk prediction and predictive factors of type 2 diabetes," *BMJ Nutrition, Prevention & Health*, vol. 0, pp. e000200, 2021. <https://doi.org/10.1136/bmjnp-2020-000200>.
- [13] S. Anam, I. N. Purwanto, D. M. Mahanani, F. I. Yusuf, & H. Rasikhun,, "Health risk classification using XGBoost with Bayesian hyperparameter optimization," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 3, pp. 465-473, 2025. <https://doi.org/10.29207/resti.v9i3.6307>.
- [14] Z. Xu, X. Guo, A. Zhu, X. He, X. Zhao, Y. Han, & R. Subedi, "Using deep convolutional neural networks for image-based diagnosis of nutrient deficiencies in rice," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 7307252, 2020. <https://doi.org/10.1155/2020/7307252>.
- [15] T. Velikova, N. Mileva, & E. Naseva, "Method 'Monte Carlo' in healthcare," *World Journal of Methodology*, vol. 14, no. 3, 93930, 2024. <https://doi.org/10.5662/wjm.v14.i3.93930>.
- [16] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2023, 585 p. <https://doi.org/10.1007/978-0-387-21617-1>.
- [17] S. O. Subbotin, A. O. Oliinyk, Ye. M. Fedorchenko, et al., *Mathematical and software tools for decision making, pattern recognition and intelligent diagnostics*, Zaporizhzhia Polytechnic National University, 2020, 271 p.
- [18] I. Fedorchenko, A. Oliinyk, T. Zaiko, K. Medvedev, Y. Fedorchenko, & M. Khokhlov, "Development of a modified genetic method for automatic university scheduling," *CEUR Workshop Proceedings*, vol. 3662, pp. 1-15, 2024. <https://doi.org/10.59018/1223322>.
- [19] T. Sun, et al., "Machine learning predicts diabetes risk in high-risk populations: Analysis of national health and nutrition examination survey," *Archives of Medical Science*, 2025. <https://doi.org/10.5114/aoms/209547>.
- [20] D. Kirk, M. Kok, S. Tufano, B. Tekinerdogan, E. J. Feskens, & G. Camps, "Machine learning in nutrition research," *Advances in Nutrition*, vol. 13, no. 6, pp. 2573-2589, 2022. <https://doi.org/10.1093/advances/nmac103>.
- [21] R. S. Pérez-Baquero, E. B. Pérez-Baquero, I. Palomino-Ramírez, & J. P. Hoyos-Sánchez, "Recommendation system for personalized nutritional solutions using machine learning techniques," *Revista Facultad de Ingeniería*, vol. 33, no. 70, e18720, 2024. <https://doi.org/10.19053/uptc.01211129.v33.n70.2024.18720>.
- [22] Ziya, "Diet recommendations dataset," *Kaggle Repository*, 2023. [Online]. Available at: <https://www.kaggle.com/datasets/ziya07/diet-recommendations-dataset>.
- [23] J. A. J. Alsayayadeh, M. F. Yusof, M. Z. Abdul Halim, M. N. S. Zainudin and S. G. Herawan, "Patient Health Monitoring System Development using ESP8266 and Arduino with IoT Platform" *International Journal of Advanced Computer Science and Applications(IJACSA)*, 14(4), May 2023, pp. 617-624. <http://dx.doi.org/10.14569/IJACSA.2023.0140467>.
- [24] Indra, W. A., Khang, A. W. Y., Yung, Y. T., Abedalrahim, J., & Alsayaydeh, J. A. J. (2019). Radiofrequency identification (RFID) item finder using radio frequency energy harvesting. *ARPN Journal of Engineering and Applied Sciences*, 14(20), 3554-3560. Available online: http://www.arpnjournals.org/jeas/research_papers/rp_2019/jeas_1019_7_964.pdf
- [25] J. A. J. Alsayayadeh, M. F. Yusof, M. Z. Abdul Halim, M. N. S. Zainudin and S. G. Herawan, "Patient Health Monitoring System Development using ESP8266 and Arduino with IoT Platform" *International Journal of Advanced Computer Science and Applications(IJACSA)*, 14(4), May 2023, pp. 617-624. <http://dx.doi.org/10.14569/IJACSA.2023.0140467>.
- [26] S. Lysenko, K. Bobrovnikova, O. Savenko and A. Kryshchuk, "BotGRABBER: SVM-based self-adaptive system for the network resilience against the botnets' cyberattacks," *Communications in Computer and Information Science*, vol. 1039, pp. 127-143, 2019. https://doi.org/10.1007/978-3-030-21952-9_10.
- [27] P. Bykovyy, V. Kochan, A. Sachenko and G. Markowsky, "Genetic algorithm implementation for perimeter security systems CAD," *Proceedings of the 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Dortmund, Germany, 2007, pp. 634-638, <https://doi.org/10.1109/IDAACS.2007.4488498>.
- [28] V. S. K. Kowshik, D. V. S. M. Kumar, P. J. N. D. M. Prakash, Y. Sathwik, J. V. Devasia, "Hybrid TCN-transformer model with multi-head attention for stock price forecasting," *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 18, no. 3, pp. 173-187, 2026. <https://doi.org/10.5815/ijisa.2026.03.11>.



IEVGEN FEDORCHENKO, Senior Lecturer of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics. <https://orcid.org/0000-0003-1605-8066> Email: evg.fedorchenko@gmail.com



PROF. ANDRII OLIINYK, Dr. Sc., Professor of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics. <https://orcid.org/0000-0002-6740-6078> Email: olejnikaa@gmail.com



TETIANA FEDORONCHAK, Assoc. Professor of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics. <https://orcid.org/0000-0001-6238-1177> Email: t.fedoronchak@gmail.com



MARIYA MIKHAYLOVA, Student at the National University of Zaporizhzhia Polytechnic. Field of scientific interests: Optimization Methods, Forecasting Algorithms, Personalized Nutrition Planning. <https://orcid.org/0009-0009-5513-2135> Email: mariamihajlova31@gmail.com



YEVGENY GOFMAN, Assoc. Professor of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Diagnostics, Databases. <https://orcid.org/0009-0001-2885-4185> Email: gofmanjenek@gmail.com



ALEKSANDR STEPANENKO Associate Professor at the Department of Software Tools, National University "Zaporizhzhia Polytechnic". Field of scientific interests: Software Engineering, Optimization Methods, Algorithm Development, Computational Modeling. <https://orcid.org/0000-0002-0267-7440> Email: alex@zntu.edu.ua



MYKOLA KHOKHLOV, Senior Lecturer of Computer systems and networks Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics. <https://orcid.org/0000-0001-8272-9847> Email: khokhlov@zp.edu.ua

...