2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT) | 979-8-3503-4110-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/COMNETSAT59769.2023.10420746

Dietary Food Ingredients Recommendation for Patients with Hypertension Using a Genetic Algorithm

Nalurisa Izma Mardiana
School of Computing
Telkom University
Bandung, Indonesia
nalurisaizmamardiana@student.telkomuniversity.ac.id

Z. K. A. Baizal*
School of Computing
Telkom University
Bandung, Indonesia
baizal@telkomuniversity.ac.id

***Abstract*—Hypertension is a disease of the cardiovascular system. People with hypertension are often not aware of their condition because hypertension does not show apparent symptoms. Measuring blood pressure is an excellent habit for identifying indications of hypertension symptoms quickly. Obesity is a risk factor for hypertension. Calorie restriction and a healthy diet can manage blood pressure. The Dietary Approach to Stop Hypertension (DASH) is a dietary program for maintaining blood pressure. Hypertension sufferers need to pay attention to the amount of sodium intake. Many researchers have successfully researched food recommendation systems using knowledge base methods. This method has the disadvantage of requiring manual updating if there is new data or knowledge. In this study, we develop a recommender system that uses a genetic algorithm to recommend dietary ingredients to people with hypertension. The genetic algorithm has the characteristics of providing optimal results and can adapt when the data is updated. The recommender system produces recommendations for food ingredients suitable for consumption by people with hypertension. We are testing the accuracy using the MAPE (Mean Absolute Percentage Error) method. The results indicate that the recommender system provides accurate recommendations for food ingredients for 45 people with hypertension, with an average MAPE of 25.668%.**

***Keywords*—hypertension, obesity, recommender system, genetic algorithm**

I. INTRODUCTION

Hypertension is one of the factors that contribute to early mortality worldwide. Hypertension is a condition where blood pressure exceeds normal limits [1]. The World Health Organization (WHO) claims 46% of adults are unaware of their hypertension [1]. Obesity has a risk of cardiovascular disease, such as hypertension [2]. One form of prevention and control of hypertension is to regulate diet. Lifestyle changes to be healthier play an essential role in reducing blood pressure and preventing hypertension for individuals with normal blood pressure [3]. Therefore, people with hypertension must manage a good diet and do a balanced physical activity so that hypertensive patients can control their blood pressure.

Adopting a healthier lifestyle in people with hypertension results in more controlled blood pressure compared to patients who do not adopt a healthy lifestyle [4]. Patients with hypertension should follow a diet to reduce blood pressure. Dietary Approaches to Stop Hypertension (DASH) is a diet pattern that treats and prevents hypertension [5]. Healthy food ingredients aligned with our body's needs are crucial for supporting our overall health. Thus, a recommender system is needed to determine dietary ingredients for people with hypertension.

Alian et al. [6] Developed a knowledge-based recommender system to recommend food intake and physical exercise to support adopting a healthy lifestyle for people with diabetes using ontology. Subramaniyaswamy et al. [7] developed a food availability recommender system for chronic disease patients following a strict diet taking into account nutritional value and user characteristics using ontology. Knowledge-based recommendation systems provide deterministic recommendations, but there are barriers to constructing this knowledge [8], [9]. Based on previous research, many researchers have succeeded in recommending food using the knowledge base method. However, the weakness of the knowledge base cannot provide optimal solutions when dealing with new situations, so manual updating is needed to adjust the updated knowledge because data cannot be changed automatically. These weaknesses can affect the accuracy of the recommendations.

Many studies have developed recommender systems using machine learning. Manoharan et al. [10] developed a diet recommender system for patients based on their health conditions using the K-clique method. Rosli et al. [11] recommended food intake according to ideal glucose levels and predicted subsequent meals according to blood glucose levels for patients with gestational diabetes mellitus using a genetic algorithm. Vasanth et al. [12] implemented a genetic algorithm to recommend foods based on the calorie content of food ingredients. Recommender system research using machine learning has the advantage of learning and adapting to new data. In unknown situations, machine learning models can provide recommendations. In this work, we use a genetic algorithm to suggest food ingredients for people with hypertension. Genetic algorithms provide optimal search solutions by producing the best results from natural evolution processes [13].

II. RELATED WORK

A. Hypertention

Hypertension is classified as cardiovascular disease when a person's systolic blood pressure is more than 140 mmHg and diastolic is more than 90 mmHg [1]. Table I is the blood pressure categories published by the American Heart

*Corresponding Author

Association (AHA) and the American College of Cardiology (ACC). Blood pressure readings are taken at least twice at different times [14].

TABLE I. Blood Pressure Category

Classification of blood pressure	Systolic blood pressure (mmHg)	Diastolic blood pressure (mmHg)
Normal	<120	<80
Prehypertension	120 - 129	<80
Stage 1 hypertension	130 -139	80 - 89
Stage 2 hypertension	≥140	≥90

The main non-pharmacological treatments for hypertension include reducing the amount of salt in food, avoiding excessive alcohol consumption, stopping smoking, eating a balanced diet, exercising, and losing weight [15]. One of the recommended diets for people with hypertension is DASH (Dietary Approaches to Stop Hypertension) [5]. The DASH diet aims to reduce sodium levels consumed by a person [5]. The sodium limit individuals consume is 2.300 mg, while for hypertensive patients, it is 1.500 mg per day [5]. In addition to limiting the amount of sodium consumption, the DASH diet also pays attention to eating lots of fruits and vegetables and choosing foods that are low in sugar and fat [5]. To follow the DASH diet, a person must know the calories they need to carry out daily activities.

The Harris-Benedict equation calculates BMR (Body Metabolic Rate) [16]. Table II of the Harrist-Benedict formula considers gender, age, weight, and height [16].

TABLE II. BMR Formula

	Male	Female
Harrist-Benedict equation	66 + (13.7 x Weight (kg)) + (5.0 x Height (cm)) – (6.8 x Age (years))	655 + (9.6 x Weight (kg)) + (1.8 x Height (cm)) – (4.7 x Age (years))

After obtaining the results of BMR calculations using the Harrist-Benedict equation, the next step is to estimate daily caloric needs by multiplying them by the PAL (Physical Activity Level) value in Table III [16].

TABLE III. Physical Activity Level

Activity	PAL Value
Sedentary	1.2
Lightly active	1.375
Moderately active	1.55
Very active	1.725

B. *Genetic Algorithm*

An algorithm with a genetic component iteratively selects the top individual from the population set until it produces the best result [11]. The genetic algorithm's properties are intelligence, parallelism, self-organization, ease of use, and robustness in solving optimization problems [17]. Genetic algorithms have a genetic structure consisting of three main components, namely individuals, chromosomes, and genes [18]. The genetic algorithm generates the initial population randomly [17].

The genetic algorithm generates a population to solve the problem [19]. The next step is to calculate the fitness function [17]. The fitness function is a criterion for assessing chromosome performance and is the only criterion for optimizing the algorithm [17]. The fitness value represents the ability of the individual assigned to produce a solution [20]. A higher fitness value increases the probability of selecting individuals from the population [12]. After obtaining the fitness value of the chromosomes, the next step is the selection operation. The selection operation is the elimination of low-fitness chromosomes and the selection of high-fitness chromosomes from a population with a given selection probability to create new individuals [17]. The higher the chromosome fitness value, the higher the chance of selection [17]. The section gives individuals better choices and may make it possible to pass on their genes to the next generation [21]. Parents with the best fitness value will be mated [18]. After selecting chromosomes using the selection operation, the next step is crossover [18]. A genetic operator known as crossover combines the genes of two specific chromosomes to produce a new chromosome or offspring [18]. The new chromosomes are expected to inherit the best qualities from their parents through crossover [18]. The mutation is a genetic operator for increasing diversity and preventing premature convergence of algorithms [18]. The mutation of gene values at specific positions in the coding string of chromosomes maintains population diversity [17]. In most cases, the operator selects a gene position randomly and replaces it with a new gene not in the pathway [18].

Alcaraz-Herrera et al. [22] developed EvoRecSys (Evolutionary Recommender System), a meal-physical activity recommender system for professional athletes and patients with chronic diseases. EvoRecSys reformulates the recommendation problem as a multi-objective optimization problem and uses genetic algorithms as an evolutionary technique to optimize recommendations. In achieving a balance of the three main dimensions, namely user preferences, user needs, and goals set by the user, EvoRecSys is an effective solution. Zubair et al. [23] developed a model to improve the prediction accuracy of recommender systems by combining the results of cosine similarity and pearson correlation similarity based on genetic algorithms. The genetic algorithm provides better performance and accuracy compared to other similarity methods.

III. EXPERIMENT AND IMPLEMENTATION

A. *Datasets*

This study of the recommender system for dietary food ingredients for people with hypertension uses a food ingredient dataset from the Indonesian Food Composition Table accessed on the *Andra Farm* website. The website manages agriculture, nutrition, food processing, and research and development. Food ingredients are selected based on food ingredients suitable for people with hypertension, considering the sodium content in these foodstuffs. We collected 288 data covering the categories of carbohydrates, proteins, vegetables, and fruits. The data includes information about the names of food ingredients, energy, carbohydrates, protein, fat, sodium, and weight of food ingredients.

At a hospital in Bandung, we collected medical data from patients diagnosed with hypertension who underwent the DASH diet. The medical data includes information such as the patients' names, genders, weights, heights, activity levels, systolic blood pressures, diastolic blood pressures, calorie needs, carbohydrate needs, protein needs, and fat needs. We employed the data of 45 patients as actual data to assess the accuracy of the recommender system.

B. Calorie Needs

A man aged 45 years, height 163 cm, weight 73, and has a sedentary activity. On checking blood pressure, the man had a systolic blood pressure of 150 mmHg and a diastolic blood pressure of 90 mmHg. Based on the information from the man, here are the steps to recommend a suitable diet for the man. Calorie calculation, using the Harrist-Benedict equation to determine the BMR value.

BMR = 66 + (13.7 x Weight (kg)) + (5.0 x Height (cm)) – (6.8 x Age (years))
= 66 + (13.7 x 73) + (5.0 x 163) – (6.8 x 45)
= 1575 kkal

Based on information on the PAL value of a man in the case example, he has a sedentary activity.

Calorie = BMR X PAL
= 1575 x 1.2
= 1890 kkal

After knowing the calorie needs of the man, the next step is calculating the grams needed for carbohydrates, proteins, and fats. The division of carbohydrates, proteins, and fats in the number of grams of calories is as follows.

grams_of_carbohydrate = (60% x Calorie) / 4
= (60% x 1890) / 4
= 283 gram
grams_of_protein = (15% x Calorie) / 4
= (15% x 1890) / 4
= 70 gram
grams_of_fat = (25% x Calorie) / 9
= (25% x 1890) / 9
= 52 gram

C. Foodstuff Recommendations

Chromosomes generated population initialization with 12 genes representing the category Id of each food item. Each gene is randomly selected based on the food category. On chromosomes, genes 1 to 4 represent breakfast, genes 5 to 8 represent lunch, and genes 9 to 12 represent dinner. Examples of chromosomes for one day, representing food IDs according to their respective categories, are found in Table IV. Fig. 1 is the implementation of initialization of the population.

TABLE IV. Chromosomes

Breakfast				Lunch				Dinner			
C	P	V	F	C	P	V	F	C	P	V	F
10	45	133	193	6	46	182	231	17	58	139	230

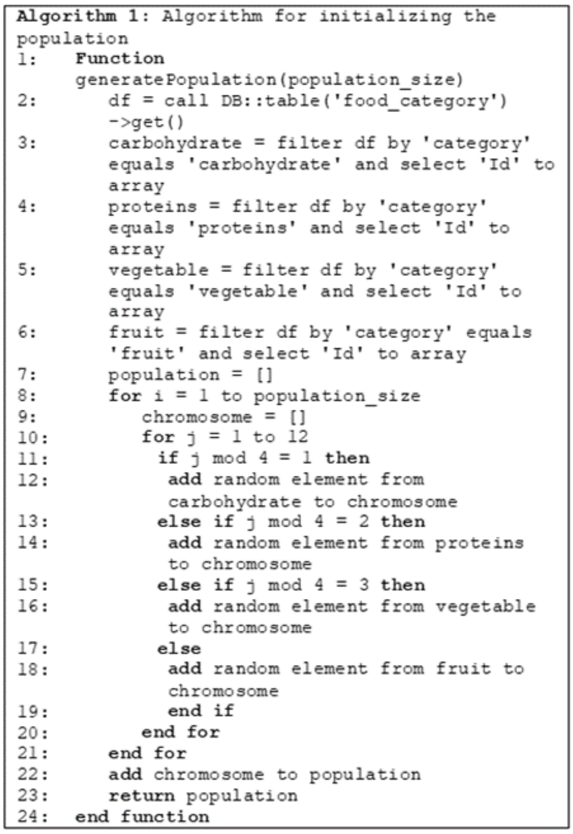
Information:

C: Carbohydrates

P: Proteins

V: Vegetables

F: Fruits



```
Algorithm 1: Algorithm for initializing the population
Function generatePopulation(population_size)
    df = call DB::table('food_category') ->get()
    carbohydrate = filter df by 'category' equals 'carbohydrate' and select 'Id' to array
    proteins = filter df by 'category' equals 'proteins' and select 'Id' to array
    vegetable = filter df by 'category' equals 'vegetable' and select 'Id' to array
    fruit = filter df by 'category' equals 'fruit' and select 'Id' to array
    population = []
    for i = 1 to population_size
        chromosome = []
        for j = 1 to 12
            if j mod 4 = 1 then
                add random element from carbohydrate to chromosome
            else if j mod 4 = 2 then
                add random element from proteins to chromosome
            else if j mod 4 = 3 then
                add random element from vegetable to chromosome
            else
                add random element from fruit to chromosome
            end if
        end for
    end for
    add chromosome to population
    return population
end function
```

Fig. 1. Algorithm for Initializing the Population

The fitness value determines the quality of the chromosome according to the specified criteria. We calculate the fitness value by considering the nutritional content of the food ingredients.

$$fitness = \frac{1}{\begin{gathered}(|calorie - total_{calories}| \\ + |grams_{carbohydrate} - total_{carbohydrates}| \\ + |grams_{protein} - total_{proteins}| \\ + |grams_{fat} - total_{fat}| \\ + |sodium - total_{sodium}|\end{gathered}} \quad (1)$$

Information:

calorie: daily calorie requirement

$total_{calories}$: total calories in one chromosome

$grams_{carbohydrate}$: daily carbohydrate requirement

$total_{carbohydrates}$: total carbohydrates in one chromosome

$grams_{protein}$: daily protein requirement

$total_{proteins}$: total protein in one chromosome

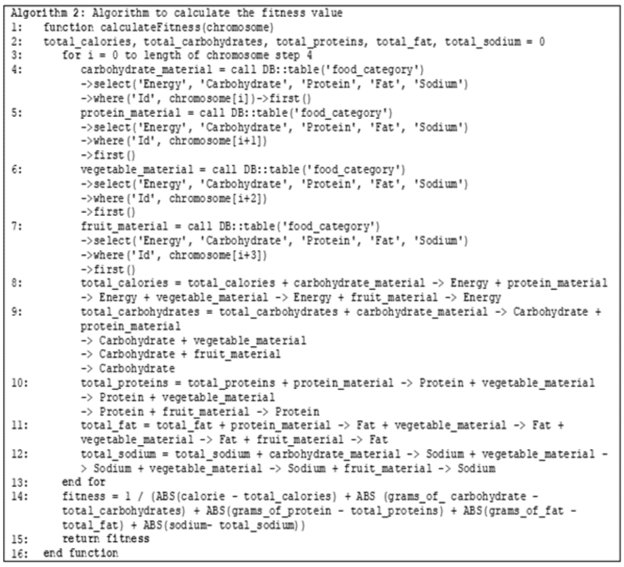
$grams_{fat}$: daily fat requirement

$total_{fat}$: total fat in one chromosome

sodium: daily sodium requirement

$total_{sodium}$: total sodium in one chromosome

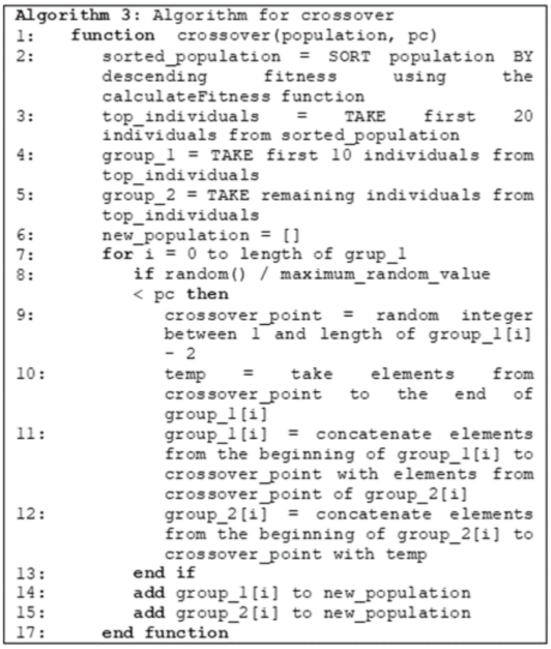
Chromosomes with large fitness values become parents for crossover. Thus, chromosomes with greater fitness values will have more excellent crossover opportunities. For people with hypertension, consuming foodstuffs consisting of carbohydrates, protein, fat, and sodium is regulated. We set a sodium value of 1500 mg. In Fig. 2, we demonstrate how to calculate the fitness value based on equation (1).



```
Algorithm 2: Algorithm to calculate the fitness value
    function calculateFitness(chromosome)
    total_calories, total_carbohydrates, total_proteins, total_fat, total_sodium = 0
        for i = 0 to length of chromosome step 4
            carbohydrate_material = call DB::table('food_category')
            ->select('Energy', 'Carbohydrate', 'Protein', 'Fat', 'Sodium')
            ->where('Id', chromosome[i])->first()
            protein_material = call DB::table('food_category')
            ->select('Energy', 'Carbohydrate', 'Protein', 'Fat', 'Sodium')
            ->where('Id', chromosome[i+1])
            ->first()
            vegetable_material = call DB::table('food_category')
            ->select('Energy', 'Carbohydrate', 'Protein', 'Fat', 'Sodium')
            ->where('Id', chromosome[i+2])
            ->first()
            fruit_material = call DB::table('food_category')
            ->select('Energy', 'Carbohydrate', 'Protein', 'Fat', 'Sodium')
            ->where('Id', chromosome[i+3])
            ->first()
            total_calories = total_calories + carbohydrate_material -> Energy + protein_material
            -> Energy + vegetable_material -> Energy + fruit_material -> Energy
            total_carbohydrates = total_carbohydrates + carbohydrate_material -> Carbohydrate +
            protein_material
            -> Carbohydrate + vegetable_material
            -> Carbohydrate + fruit_material
            -> Carbohydrate
            total_proteins = total_proteins + protein_material -> Protein + vegetable_material
            -> Protein + vegetable_material
            -> Protein + fruit_material -> Protein
            total_fat = total_fat + protein_material -> Fat + vegetable_material -> Fat +
            vegetable_material -> Fat + fruit_material -> Fat
            total_sodium = total_sodium + carbohydrate_material -> Sodium + vegetable_material -
            > Sodium + vegetable_material -> Sodium + fruit_material -> Sodium
        end for
        fitness = 1 / (ABS(calorie - total_calories) + ABS (grams_of_ carbohydrate -
        total_carbohydrates) + ABS(grams_of_protein - total_proteins) + ABS(grams_of_fat -
        total_fat) + ABS(sodium- total_sodium))
        return fitness
    end function
```

Fig. 2. Algorithm to Calculate the Fitness Value

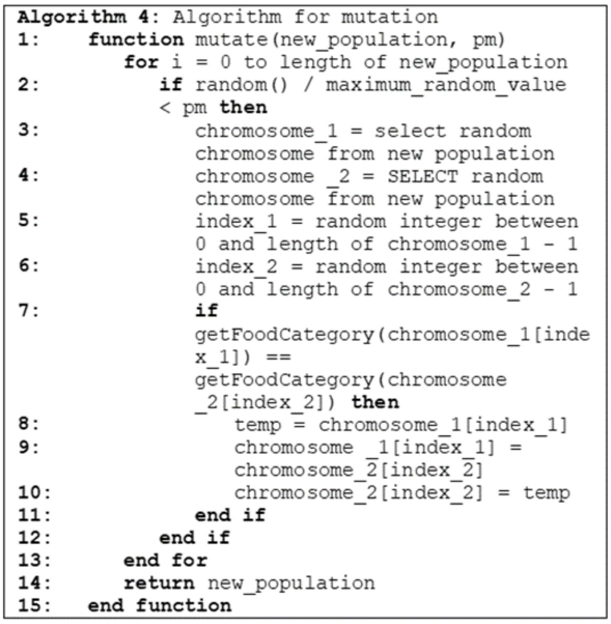
The next step is to select the fitness value of each individual. Sort the fitness values from highest to lowest. Some individuals with the highest fitness value are assigned to be parents.



```
Algorithm 3: Algorithm for crossover
    function  crossover(population, pc)
        sorted_population = SORT population BY
        descending fitness using the
        calculateFitness function
        top_individuals = TAKE first 20
        individuals from sorted_population
        group_1 = TAKE first 10 individuals from
        top_individuals
        group_2 = TAKE remaining individuals from
        top_individuals
        new_population = []
        for i = 0 to length of grup_1
            if random() / maximum_random_value
            < pc then
                crossover_point = random integer
                between 1 and length of group_1[i]
                - 2
                temp = take elements from
                crossover_point to the end of
                group_1[i]
                group_1[i] = concatenate elements
                from the beginning of group_1[i] to
                crossover_point with elements from
                crossover_point of group_2[i]
                group_2[i] = concatenate elements
                from the beginning of group_2[i] to
                crossover_point with temp
            end if
            add group_1[i] to new_population
            add group_2[i] to new_population
        end function
```

Fig. 3. Algorithm for Crossover

Fig. 3 illustrates the algorithm for performing crossover by selecting the top 20 individuals as parents from a pool of 20 individuals with the highest scores, divided into two groups. The first group contains ten individuals. The second group includes ten individuals. The crossover process is carried out randomly with crossover probability (*pc*). The process occurs when the *pc* value is smaller than the random value of the chromosome pair. The crossover results form a new population. After carrying out the crossover process, the next step is to perform the mutation. Mutation aims to change the new offspring.



```
Algorithm 4: Algorithm for mutation
    function mutate(new_population, pm)
        for i = 0 to length of new_population
            if random() / maximum_random_value
            < pm then
                chromosome_1 = select random
                chromosome from new population
                chromosome _2 = SELECT random
                chromosome from new population
                index_1 = random integer between
                0 and length of chromosome_1 - 1
                index_2 = random integer between
                0 and length of chromosome_2 - 1
                if
                getFoodCategory(chromosome_1[inde
                x_1]) ==
                getFoodCategory(chromosome
                _2[index_2]) then
                    temp = chromosome_1[index_1]
                    chromosome _1[index_1] =
                    chromosome_2[index_2]
                    chromosome_2[index_2] = temp
                end if
            end if
        end for
        return new_population
    end function
```

Fig. 4. Algorithm for Mutation

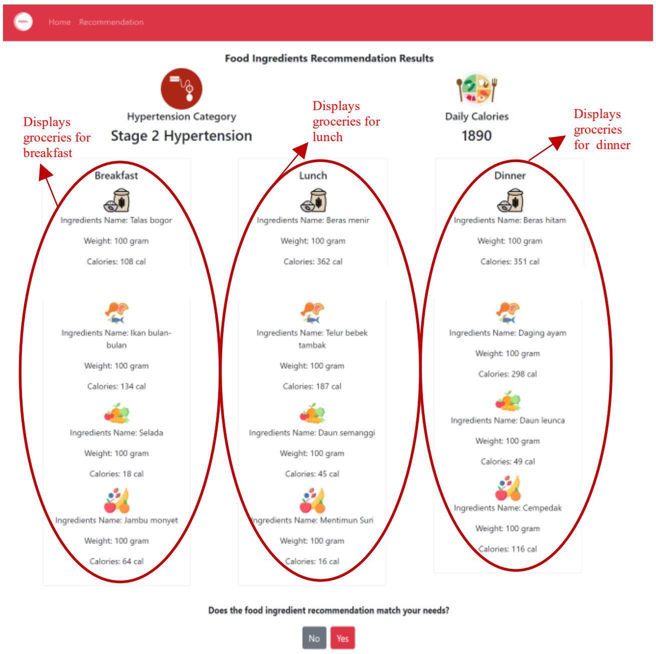
The algorithm for mutation, as depicted in Fig. 4, selects two chromosomes randomly to perform individual exchanges. They define a separate index for gene exchange. We ensured that the genes to be mutated were in the same category. Mutation probability (*pm*) for mutations that determine the occurrence of mutations in individuals. If the random value of an individual is smaller than the *pm* value, a mutation process will occur in that individual. Next, calculate the fitness value of the new individual who has completed the mutation process. The individual with the highest mutational fitness value is the result of the recommender system.

D. Result and Evaluation

Based on the case examples above, Fig. 5 shows the categories of hypertension, daily caloric needs, and food recommendations for people with hypertension. The recommender system displays the food name, weight, and calories. As a form of user satisfaction, the recommender system will ask whether the food ingredients produced are by the user's wishes. If it doesn't match, the system will recommend another component.

Testing the parameters of the genetic algorithm consists of initial population size, number of generations, crossover probability (*pc*), and mutation probability (*pm*). After mutation and program running time, the test produces an average fitness value for each individual. Table V of genetic algorithm parameter test results.

The results of testing the parameters of the genetic algorithm show that the more generations, the longer the program's running time. In the population table with 100 individuals, the number of generations is 10, the *pc* value is 0.8, and the *pm* is 0.8, producing the highest average fitness value with fast running time. We use the best results from testing the parameters of the genetic algorithm to build a recommender system.



Food Ingredients Recommendation Results

Hypertension Category
Stage 2 Hypertension

Daily Calories
1890

Displays groceries for breakfast

Displays groceries for lunch

Displays groceries for dinner

Breakfast

Ingredients Name: Talas bogor
Weight: 100 gram
Calories: 108 cal

Ingredients Name: Ikan bulan-bulan
Weight: 100 gram
Calories: 134 cal

Ingredients Name: Selada
Weight: 100 gram
Calories: 18 cal

Ingredients Name: Jambu monyet
Weight: 100 gram
Calories: 64 cal

Lunch

Ingredients Name: Beras menir
Weight: 100 gram
Calories: 362 cal

Ingredients Name: Telur bebek tambak
Weight: 100 gram
Calories: 187 cal

Ingredients Name: Daun semanggi
Weight: 100 gram
Calories: 45 cal

Ingredients Name: Mentimun Suri
Weight: 100 gram
Calories: 16 cal

Dinner

Ingredients Name: Beras hitam
Weight: 100 gram
Calories: 351 cal

Ingredients Name: Daging ayam
Weight: 100 gram
Calories: 298 cal

Ingredients Name: Daun leunca
Weight: 100 gram
Calories: 49 cal

Ingredients Name: Cempedak
Weight: 100 gram
Calories: 116 cal

Does the food ingredient recommendation match your needs?

Fig. 5. Food Ingredients Recommendation Results

TABLE V. Parameter Experiment Results

Population	**Generation**	***pc***	***pm***	**Fitness population average**	**Time (second)**
50	20	0.8	0.8	0.000615	12
50	20	0.9	0.9	0.000595	11
50	20	0.8	0.9	0.000605	11
50	20	0.9	0.8	0.000614	11
100	10	0.8	0.8	0.000708	5
100	20	0.8	0.8	0.000638	12
100	30	0.8	0.8	0.000626	28
100	40	0.8	0.8	0.000602	44
80	20	0.8	0.8	0.000597	13
100	20	0.8	0.8	0.000664	16
120	20	0.8	0.8	0.000661	13
140	20	0.8	0.8	0.000650	14

We use the MAPE (Mean Absolute Percentage Error) method to measure the accuracy of the recommender system. MAPE serves to measure the level of error in predictions. The MAPE value is essential for evaluating prediction accuracy by indicating how much error there is in the forecast compared to the actual data. Equation (2) is the MAPE formula where a = actual data; b = result data; n amount of data.

$$MAPE = \frac{\sum_{t=1}^{n} \left|\frac{a-b}{a}\right|}{n} \times 100\% \tag{2}$$

The recommendation model was evaluated using actual data from 45 patients with hypertension. In actual data, we took information on calorie, carbohydrate, protein, and fat needs from hypertensive patients. We use the results of the recommender system as the result data. In the result data, we retrieved the total calorie, carbohydrate, protein, and fat information from the recommendation system results. Based on actual data and result data, we perform MAPE calculations. We calculated calorie MAPE, carbohydrate MAPE, protein MAPE, and fat MAPE based on essential patient needs data and food product outcome data based on the results of the recommender system. Next, we performed calculations to determine the average MAPE for each parameter. Of the 45 hypertensive patients, the average MAPE result was 25.668%. The MAPE average indicates that the average prediction error is 25.668% of the actual value. Based on MAPE averages, the recommendation system accurately recommends ingredients.

IV. CONCLUSIONS

The system we propose aims to recommend food ingredient combinations for individuals with hypertension. In this study, we calculated daily caloric needs using the Harris-Benedict Equation involving age, weight, height, and level of physical activity. In hypertensive patients, blood pressure categories are determined based on systolic and diastolic blood pressure values. The system recommends dietary food ingredients for people with hypertension by considering the energy content, carbohydrates, protein, fat, and sodium of food ingredients. In this study, a genetic algorithm was used to find combinations of food ingredients based on the categories of carbohydrates, proteins, vegetables and fruit. On the chromosome, we built 12 genes. Genes 1 to 4 represent food ingredients for breakfast. Genes 5 to 8 represent ingredients for lunch. Gen 9 to 12 represents ingredients for dinner. The genetic algorithm method produces more varied food ingredients. Based on experiment, the number of generations affects the running time of the program. We conduct tests to assess the accuracy of the recommending system using the MAPE (Mean Absolute Percentage Error) method. The recommender system shows relatively good accuracy in recommending food ingredients, with an average MAPE value of 25.668%.

REFERENCES

[1] World Health Organization, "Hypertension," *World Health Organization*, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/hypertension. [Accessed: 25-Nov-2022].

[2] P. Parto and C. J, Lavie, "Obesity and CardiovascularDiseases," *Curr. Probl. Cardiol.*, vol. 42, no. 11, pp. 376–394, 2017, doi: 10.1016/j.cpcardiol.2017.04.004.

[3] S. A. S. Alsaigh, M. D. Alanazi, and M. A. Alkahtani, "Lifestyle Modifications for Hypertension Management," *Egypt. J. Hosp. Med.*, vol. 70, no. 12, pp. 2152–2156, 2018, doi: 10.12816/0045044.

[4] S. Akbarpour, D. Khalili, H. Zeraati, M. A. Mansournia, A. Ramezankhani, and A. Fotouhi, "Healthy lifestyle behaviors and control of hypertension among adult hypertensive patients," *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, 2018, doi: 10.1038/s41598-018-26823-5.

[5] R. Sookrah, J. D. Dhowtal, and S. D. Nagowah, "A DASH diet recommendation system for hypertensive patients using machine learning," *2019 7th Int. Conf. Inf. Commun. Technol. ICoICT 2019*, 2019, pp. 1–6. doi: 10.1109/ICoICT.2019.8835323.

[6] S. Alian, J. Li, and V. Pandey, "A Personalized Recommendation System to Support Diabetes Self-Management for American Indians," *IEEE Access*, vol. 6, no. c, pp. 73041–73051, 2018, doi: 10.1109/ACCESS.2018.2882138.

[7] V. Subramaniyaswamy *et al.*, "An ontology-driven personalized food recommendation in IoT-based healthcare system," *J. Supercomput.*, vol. 75, no. 6, pp. 3184–3216, 2019, doi: 10.1007/s11227-018-2331-8.

[8] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, "Query refinement in recommender system based on product functional requirements," *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS,* 2016, pp. 309–314, doi:

10.1109/ICACSIS.2016.7872760.

[9] Z. K. A. Baizal, A. Iskandar, and E. Nasution, "Ontology-based Recommendation Involving Consumer Product Reviews," *Fourth Int. Conf. Inf. Commun. Technol.*, vol. 4, no. c, 2016, pp. 5–10. doi: 10.1109/ICoICT.2016.7571890.

[10] D. S. Manoharan and Prof. Sathish, "Patient Diet Recommendation System Using K Clique and Deep learning Classifiers," *J. Artif. Intell. Capsul. Networks*, vol. 2, no. 2, pp. 121–130, 2020, doi: 10.36548/jaicn.2020.2.005.

[11] M. M. Rosli, N. S. M. Yusop, and A. S. Fazuly, "Design of meal intake prediction for gestational diabetes mellitus using genetic algorithm," *IAES Int. J. Artif. Intell.*, vol. 9, no. 4, pp. 591–599, 2020, doi: 10.11591/ijai.v9.i4.pp591-599.

[12] R. D. Vasanth, P. Gokul, T. Balamurugan, and S. Nivedha, "Nutrition Recommendation System Using Genetic Algorithm," *Int. J. Emerg. Technol. Innov. Eng.*, vol. 5, no. 3, pp. 133–138, 2019.

[13] H. Zhi and S. Liu, "Face recognition based on genetic algorithm," *J. Vis. Commun. Image Represent.*, vol. 58, no. December, pp. 495–502, 2019, doi: 10.1016/j.jvcir.2018.12.012.

[14] J. M. Flack and B. Adekola, "Blood pressure and the new ACC/AHA hypertension guidelines," *Trends Cardiovasc. Med.*, vol. 30, no. 3, pp. 160–164, 2020, doi: 10.1016/j.tcm.2019.05.003.

[15] J. Jordan, C. Kurschat, and H. Reuter, "Arterial hypertension-diagnosis and treatment," *Dtsch. Arztebl. Int.*, vol. 115, no. 33–34, pp. 557–558, 2018, doi: 10.3238/arztebl.2018.0557.

[16] J. M. Tallon, J. Narciso, R. Saavedra, A. Barros, A. J. Silva, and A. M. Costa, "Reported energy intake versus estimated energy requirements of Portuguese adolescents and young adults," *Prog. Nutr.*, vol. 22, no. 3, 2020, doi: 10.23751/pn.v22i3.8309.

[17] L. Deng, P. Yang, and W. Liu, "An Improved Genetic Algorithm," *2019 IEEE 5th Int. Conf. Comput. Commun. ICCC 2019*, no. 3, 2019, pp. 47–51. doi: 10.1109/ICCC47050.2019.9064374.

[18] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Comput. Sci.*, vol. 127, pp. 180–189, 2018, doi: 10.1016/j.procs.2018.01.113.

[19] Nasruddin, Sholahudin, P. Satrio, T. M. I. Mahlia, N. Giannetti, and K. Saito, "Optimization of HVAC system energy consumption in a building using artificial neural network and multi-objective genetic algorithm," *Sustain. Energy Technol. Assessments*, vol. 35, no. October 2018, pp. 48–57, 2019, doi: 10.1016/j.seta.2019.06.002.

[20] G. Zheng *et al.*, "DRN: A deep reinforcement learning framework for news recommendation," *Web Conf. 2018 - Proc. World Wide Web Conf. WWW 2018*, vol. 2, 2018, pp. 167–176. doi: 10.1145/3178876.3185994.

[21] M. Syahputra, V. Felicia, R. Rahmat, and R. Budiarto, "Scheduling diet for diabetes mellitus patients using genetic algorithm," *J. Phys. Conf. Ser.*, vol. 801, no. 1, 2017, doi: 10.1088/1742-6596/755/1/011001.

[22] H. Alcaraz-Herrera, J. Cartlidge, Z. Toumpakari, M. Western, and I. Palomares, *EvoRecSys: Evolutionary framework for health and well-being recommender systems*, vol. 32, no. 5. Springer Netherlands, 2022.

[23] S. Zubair and M. A. Al Sabri, "Hybrid Measuring the Similarity Value Based on Genetic Algorithm for Improving Prediction in A Collaborative Filtering Recommendation System.," *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.*, vol. 10, no. 2, pp. 165–182, 2021, doi: 10.14201/adcaij2021102165182.