

# The impact of the weighting techniques on MultiMOORA-based ranking on patients using ambiguous medical data★

Antoni Jaszcz<sup>1</sup>

<sup>1</sup> Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, POLAND

## Abstract

MCDM methods have proven to be a powerful tool for selecting, prioritizing and sorting ambiguous data. In emergency situations, when resources are in short supply, such methods can be used to provide said limited resources to the most in need, without human bias. In this paper, a scenario, where there is a large number of patients awaiting treatment is considered. Each patient suffers from several COVID-19 symptoms and has an AI-assessed probability of COVID-related pneumonia. Said attributes are used to simulate real-life scenarios where patients with different symptoms and test results are ranked by the state-of-art MCDM method. To further enhance the performance of the method, different weighting methods have been tested, analyzed and compared. In the experiments, a MultiMOORA decision-maker was chosen. Among the tested weighting methods, Standard-deviation, MEREC and CRITIC methods performed better, than the basic mean method. However, the entropy method improved the performance of the MCDM exceptionally.

## Keywords

MCDM methods, MultiMOORA, weighting methods, medical data, patient ranking

## 1. Introduction

The enormous growth of big data in past decades has resulted in an ever-growing need for better and more reliable methods of data analysis. Finding an optimal solution may prove to be a challenging task, given a large number of usually ambiguous criteria for assessment. Such a problem significantly limits the human ability to efficiently make decisions. Over recent years, more and more attention has been brought to MCDM (Multiple-Criteria Decision-Making) methods and techniques, facilitating this task. From choosing the best and worst option, through highlighting the most important criterion, to ranking all alternatives, MCDM methods have proven to be highly effective and vaguely universal. In recent years, the focus has been on the use of aggregation or averaging methods. This is evident in one of the core federated learning solutions. Federated learning is based on the aggregation of models in order to obtain a single, common one. However, not all models can be used as there may be an attack on one of them. Through the development of federated learning, more and more attention is being paid to MCDM and bringing a consensus of different results. An example is a fuzzy controller that analyzes the results from a large number of methods [1]. The fuzzy approach is widely used, which is also useful in hybridizing this solution with probabilistic models, as shown in[2]. Another approach is to use the ensemble model for such analysis [3]. The integration of different solutions is also used through various measures, including entropy. One such study is shown in[4], where the authors used the MULTIMOORA method, which is based on selecting the best results from many rankings. Decision-based solutions are widely used in industry and the Internet of Things. An example is hybridization or the use of optimization algorithms such as heuristic algorithms. Optimization is used to find the optimal solution, most often based on a specific objective function. Examples of such implementation solutions are presented in[5, 6, 7]. In these papers, the focus was on the construction of various functions and the

---

28<sup>th</sup> Conference on Information Society and University Studies (IVUS'2023), May 12, 2023, Kaunas, Lithuania

EMAIL: aj303181@student.polsl.pl (A. Jaszcz)

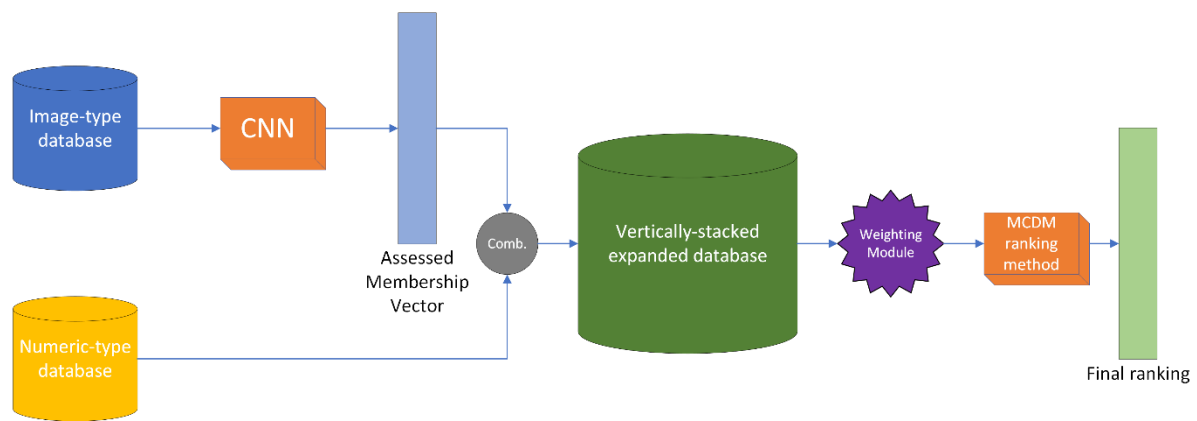
ORCID: 0000-0002-8997-0331 (A. Jaszcz)



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

search for the most optimal solution in a finite time. Such solutions are used in the Internet of Things, which allows you to create a network of connections between various objects that can download data, process it and communicate. In [8, 9], the role of blockchain in communication between objects and the place of data storage was emphasized. Access to the last blocks also allows you to analyze other solutions and make decisions [10]. The correct use of models to make decisions is an essential step in any kind of system. This allows for further integration with speech recognition[11, 12] and face recognition for improving security [13, 14]. Based on the analysis of current research, in this paper, I present a solution based on a real-life scenario, where a medical emergency is considered and patients need to be assessed and ranked based on their need for help, to provide those of the most need with the limited resources. The main contribution of this paper are:

- a new approach for further enhancing the MCDM method on ranking ambiguous data,
- providing a universal well-performing solution for binary ranking problems.



**Figure 1:** Graphical representation of the proposed pipeline

## 2. Methodology

This section describes the proposed method along with the description of methods and techniques used in the framework.

### 2.1. Creating Multiple-Criteria Patient database

In order to extend the accessible data and unify it into one, attribute-extensive database, artificial samples can be created by mixing samples from each dataset together, creating a new set of artificially created samples. In the case of class-distinguishable samples, objects of the given class are mixed only with the samples of the same class. In this paper, only two classes are considered: healthy and ill.

### 2.2. Creating Multiple-Criteria Patient database

Given a decision matrix  $X$ , with  $m$  criteria and  $n$  alternatives (see Eq. 1), each criterion  $C$  needs to be assigned weight and objective. The objective of the criterion is to maximize the values if it is beneficial or to minimize the values when it is not. For each criterion in  $X$ , its objective needs to be chosen manually. For example, if one were to consider buying a car, he would aim to lower the cost, therefore the price criterion would be non-beneficial and its objective set to minimize the value. Determining weights, however, is usually a very subjective matter, requiring expertise. Nevertheless, there are methods of criteria weighting using sole information contained in the decision matrix.

$$X = \begin{matrix} & C_1 & C_2 & \dots & C_j & \dots & C_m \\ A_1 & (x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1m}) \\ A_2 & (x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2m}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_i & (x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{im}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_n & (x_{n1} & x_{n2} & \dots & x_{nj} & \dots & x_{nm}) \end{matrix} \quad (1)$$

### 2.2.1. Mean method

The mean method is the most basic one. It assigns to all criteria the same value, equal to the fraction of the number of criteria. So in decision matrix  $X$  with  $m$  criteria  $C$ , each weight is determined as:

$$w_{C_1} = w_{C_2} = w_{C_3} = \dots = w_{C_m} = \frac{1}{m} \quad (2)$$

### 2.2.2. Standard deviation method

In the standard deviation method, first, we need to normalize the values in the Decision matrix. In order to do so, for each criterion  $C_j$ , each value  $x_{ij}$  is normalized using min-max scalar:

$$n_{ij} = \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}} \quad (3)$$

Next, for each criterion  $C_j$ , the standard deviation of its corresponding set of normalized values  $N_j$  is counted. By calculating the standard deviation for the criteria we consequently determine their weights.

$$w_{c_j} = \sigma_j = \sqrt{\frac{\sum_{i=1}^n (n_{ij} - \bar{n}_j)^2}{n - 1}} \quad (4)$$

### 2.2.3. Entropy method

In this method, again, the decision matrix first needs to be standardized. We can do this by calculating standardized values  $p$  for each criterion set in the decision matrix by the formula:

$$p_{ij} = \frac{x_{ij}}{\sum_{k=1}^n x_{kj}} \quad (5)$$

With the normalized matrix formed, we can now calculate the entropy value for each criterion  $C_j$ , which can be described as:

$$E_j = -\frac{\sum_{i=1}^n p_{ij} \cdot \ln p_{ij}}{\ln n} \quad (6)$$

## 2.2.4. MEREC method

Given the decision matrix  $X$ , normalization of its values must be performed. We can do so by scaling values of each criterion set accordingly with its objective using. For each element of the matrix  $X$ , this process of linear normalization can be presented as follows:

$$n_{ij} = \begin{cases} \frac{x_{min,j}}{x_{ij}} & \text{if } C_j \text{ is beneficial} \\ \frac{x_{ij}}{x_{max,j}} & \text{if } C_j \text{ is non-beneficial} \end{cases} \quad (8)$$

Next, the overall performance of the alternatives  $S_i$  is calculated. This is done by measuring the values of logarithmic function Eq. 9 for each sum of alternative  $A_i$  values, with mean weights applied to them. The given formula is:

$$S_i = \ln \left( 1 + \left( \frac{1}{m} \sum_{j=1}^m |\ln n_{ij}| \right) \right) \quad (9)$$

Proceeding further, the performance of the alternatives with each criterion separately removed is measured. Thus, for every alternative  $A_i$  set  $S'_i$  (see 10) of size  $m$  is obtained. This set contains the overall performances of the  $A_i$  alternative concerning each case of ignoring the  $j$ -th criterion  $C_j$ .

$$S'_i = \{S'_{i1}, S'_{i2}, \dots, S'_{ij}, \dots, S'_{im}\} \quad (10)$$

$$S'_{ij} = \ln \left( 1 + \left( \frac{1}{m} \sum_{k=1, k \neq j}^m |\ln n_{ik}| \right) \right) \quad (11)$$

Lastly, for each criterion  $C$  the sum of absolute deviations  $E$  of the overall performances  $S$  is calculated. This can be presented by the formula:

$$E_j = \sum_{i=1}^n |S'_{ij} - S_i| \quad (12)$$

The resulting weights are:

$$w_{C_j} = \frac{E_j}{\sum_{k=1}^m E_k} \quad (13)$$

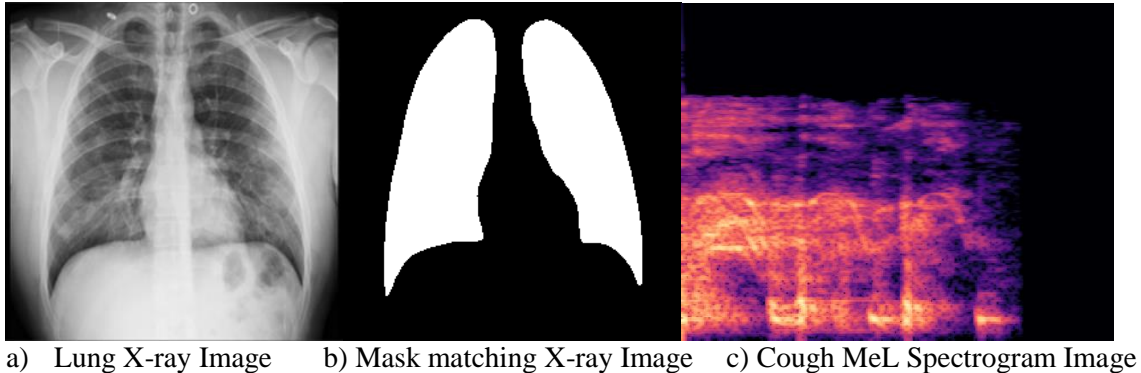
## 2.2.5. CRITIC method

In the CRITIC method, first, we need to normalize the values in the Decision matrix using the min-max scaler, the formula of which is shown in Eq. 3. Next, weights for each criterion  $C_j$  are calculated accordingly to the following equation:

$$w_{C_j} = \frac{\sigma_j \sum_{k=1}^n (1 - r_{jk})}{\sum_{j=1}^n \sigma_j \sum_{k=1}^n (1 - r_{jk})} \quad (14)$$

### 3. Experiments

#### 3.1. Data



**Figure 2:** Example of image data used in the experiments, described in section 3.1

For this research three different databases were used, each including covid positive and negative cases:

1. Covid-19 Symptom patient dataset with information about 2575 patients suffering symptoms with different intensities. The 5 criteria are:
  - a. Fever [94, 108] °
  - b. Body Pain {0, 1}
  - c. Age [1, 100]
  - d. Runny Nose {0, 1}
  - e. Difficulties with breathing {-1, 0, 1}

This data is accessible at Kaggle, under this link.

2. Coughing Mel-Spectrogram Image dataset, which is documented under this link. An example of a sample from this database is shown in fig. 2.
3. Lung X-ray dataset, along with image-matching masks. This data can be accessed on Kaggle, under this link. An example of a sample from this database is shown in fig. 2.

#### 3.2. Convolutions Neural Network models

In order to obtain a usable piece of information for multiple-criteria decision analysis from image-type data, a simple convolutional neural network was used. By processing the images throughout the model, a set of floating-point numbers in the range between 0 and 1 indicating the possibility of each patient being Covid-positive was gathered.

##### 3.2.1. Training X-ray model

###### Input layer

Input layer consisted of dual input of two 256x256 grayscale images, one being a lung image and the other being its matching mask, which is then combined in the minimum layer. Then a signal of the same size as the images (that is, a lung image with the mask applied) is passed onto the hidden layers.

###### Hidden layers

In the model, the following layer configuration, was used:

- Convolutional 2D layer (filters = 64, kernel size = 3, activation function = ReLU)

- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 64, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 128, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 256, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Flatten Layer
- Dense layer (64 neurons, activation function = ReLU)
- Dense layer (8 neurons, activation function = ReLU)

## Output layer

As an output layer a single neuron with a sigmoid activation function was used. So the output of the model for each case was a floating point number.

## Training settings

The training was performed on a dedicated dataset with the Adam optimization algorithm. After five epochs, the model reached satisfactory accuracy of 94% on a validation set.

### 3.2.2. Training cough spectrogram model

#### Input layer

Input layer consisted of a single input of shape 256x256x4 RGBA images. The signal is then passed onto the hidden layers.

#### Hidden layers

In the model, the following layer configuration, was used:

- Convolutional 2D layer (filters = 64, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 64, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 128, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Convolutional 2D layer (filters = 256, kernel size = 3, activation function = ReLU)
- Max Pooling 2D layer (pool size = 3x3)
- Flatten Layer
- Dense layer (64 neurons, activation function = ReLU)
- Dense layer (8 neurons, activation function = ReLU)

#### Output layer

As an output layer a single neuron with a sigmoid activation function was used. So the output of the model for each case was a floating point number.

### 3.3. Creating database of artificial patients

In order to study the capabilities of MCDM methods well, three additional databases were created. The symptoms dataset, as a core, was expanded by probability tables obtained in Sec. 3.2.1 and Sec. 3.2 in the following fashion: each positive sample from the core database was assigned a positive sample from one of the probability tables. That way, two new databases were created (with their size grown exponentially). The third database was obtained by simply assigning to each positive sample from one of the already mixed datasets, another positive probability table (so the mix of symptoms and x-ray datasets was again, an expanded probability table obtained from the cough database). It should be noted, that the same was done with negative cases.

Due to the large quantity of data in the X-ray and Symptoms set and their imbalance towards exceptionally sparser data in the Cough dataset, only 48 positive and 73 negative samples were randomly drafted from the former datasets, in order to match the exact numbers of the latter one. Only those selected samples were then shuffled together, using the fore-mentioned fashion. Having explained the process of creating new datasets of artificial patients, the databases used in the experiments can be listed as follows:

- Symptoms only database (48 positive and 73 negative samples)
- Symptoms x Cough database (482 = 2, 304 positive and 732 = 5, 329 negative samples)
- Symptoms x X-ray database (482 = 2, 304 positive and 732 = 5, 329 negative samples)
- Symptoms x X-ray x Cough database (483 = 110, 592 positive and 733 = 389, 017 negative samples)

After assembling the databases, for each one of them (except for the Symptoms only database, where 48 samples of each class were chosen) only 1000 positive and 1000 negative samples have been randomly chosen for further assessment.

In each dataset, the objective of every single criterion was to maximize the value. That is because the goal was to detect Covid-positive patients and greater values of the criteria were beneficial for such purpose.

### 3.4. MultiMOORA method

As for the MCDM method, MultiMOORA was chosen for the experiments. The robustness and objectiveness of the method makes it perfect for universal application [15].

## 4. Results

In the experiments, I wanted to create a list of patients, ordered by the severity of Covid related symptoms. It is worth mentioning, that the main goal was to separate Covid-positive patients from the negative ones suffering similar symptoms. In other words, the desired ranking would have Covid-positive patients at the very top of it and the majority of negative cases at the bottom.

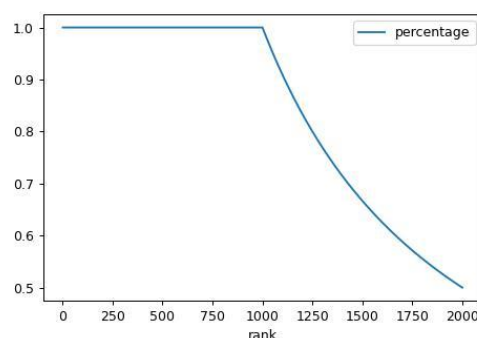
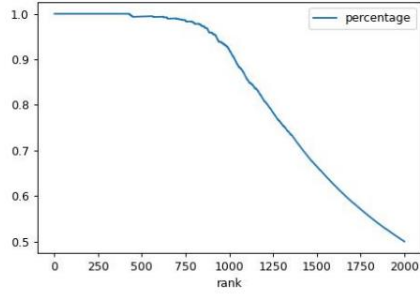
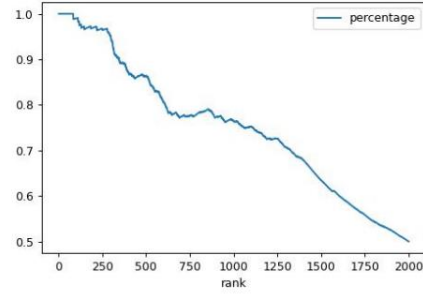


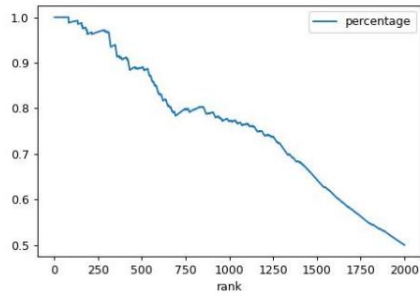
Figure 3: Ideal TOP(x) graph (for 2000 samples)



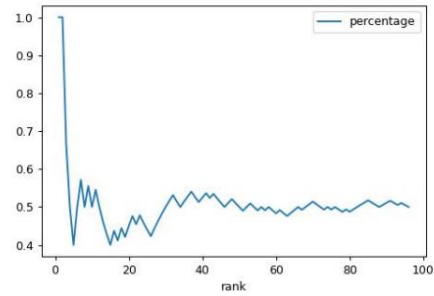
(a) Symptoms x Cough x X-ray



(b) Symptoms x Cough

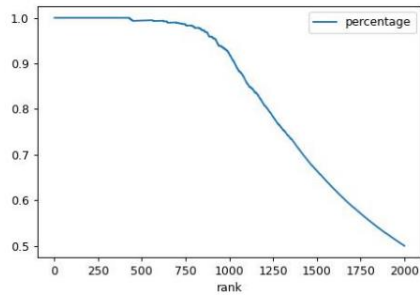


(c) Symptoms x X-ray

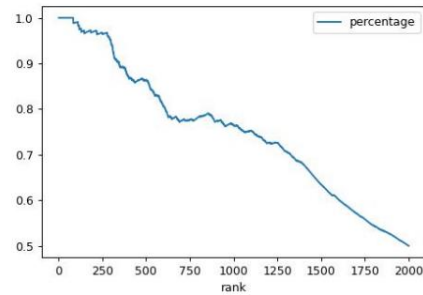


(d) Symptoms only

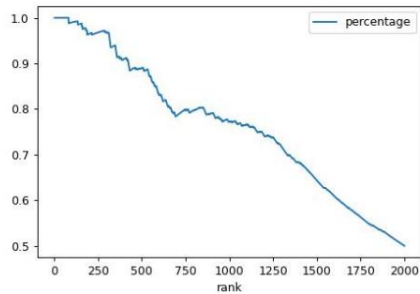
**Figure 4:** Comparison of MultiMOORA TOP(x) ranking graphs for each database with mean weighting method applied.



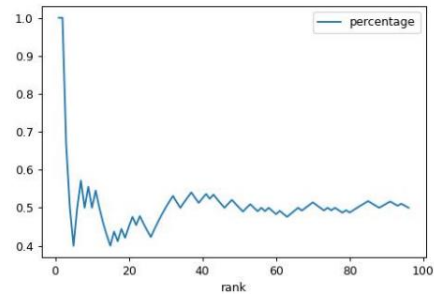
(a) Symptoms x Cough x X-ray



(b) Symptoms x Cough



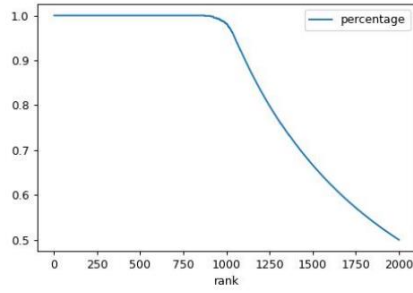
(c) Symptoms x X-ray



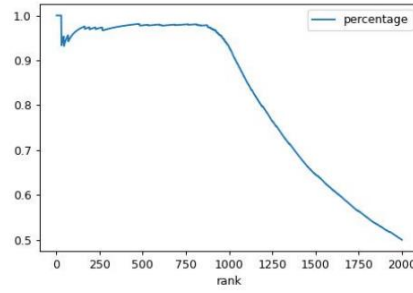
(d) Symptoms only

**Figure 5:** Comparison of MultiMOORA TOP(x) ranking graphs for each database with standard deviation weighting method applied.

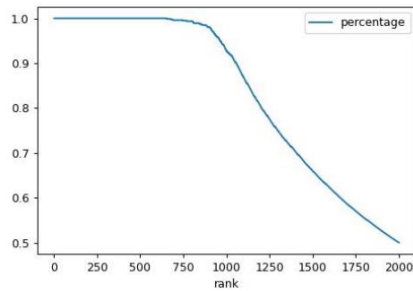




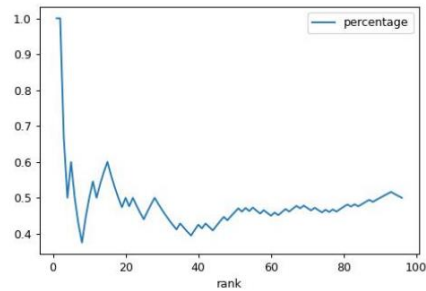
(a) Symptoms x Cough x X-ray



(b) Symptoms x Cough

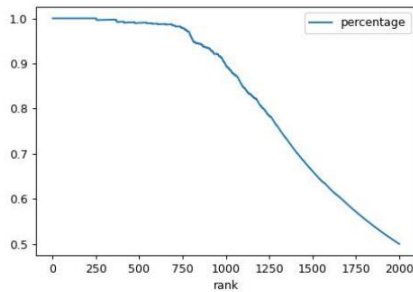


(c) Symptoms x X-ray

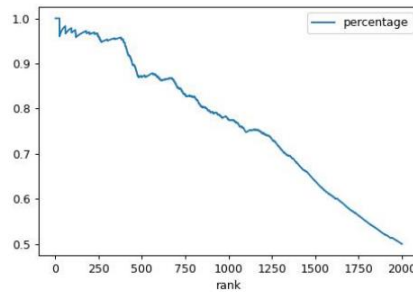


(d) Symptoms only

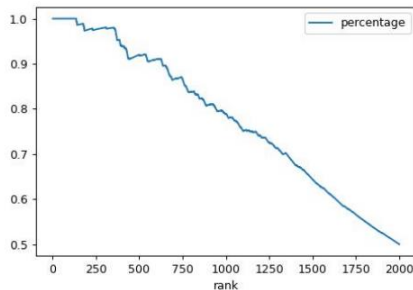
**Figure 6:** Comparison of MultiMOORA TOP(x) ranking graphs for each database with entropy weighting method applied.



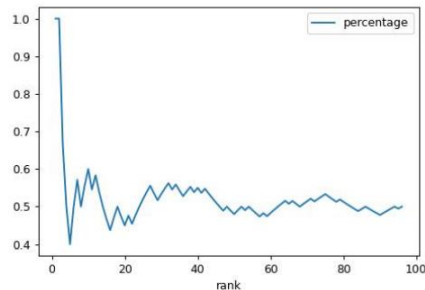
(a) Symptoms x Cough x X-ray



(b) Symptoms x Cough

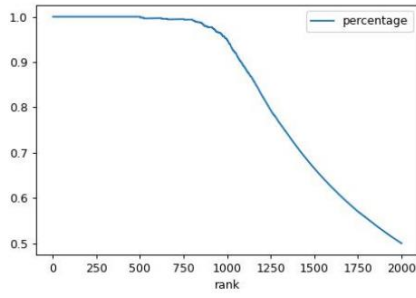


(c) Symptoms x X-ray

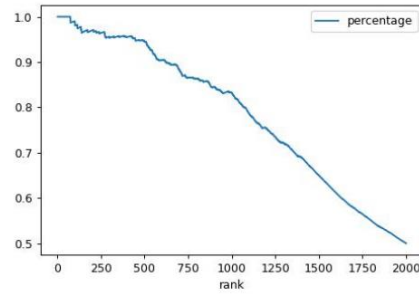


(d) Symptoms only

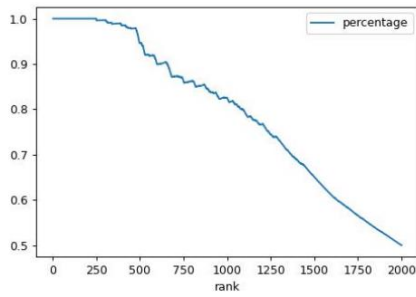
**Figure 7:** Comparison of MultiMOORA TOP(x) ranking graphs for each database with CRITIC weighting method applied.



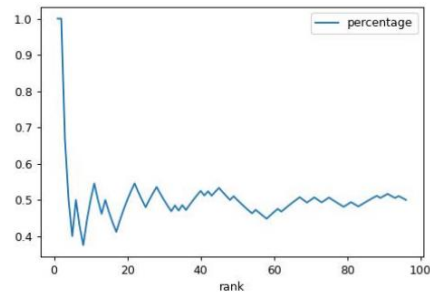
(a) Symptoms x Cough x X-ray



(b) Symptoms x Cough



(c) Symptoms x X-ray



(d) Symptoms only

**Figure 8:** Comparison of MultiMOORA TOP(x) ranking graphs for each database with MEREC weighting method applied.

#### 4.1. Calculated weights

Weights obtained by using each method (Sec. 2.2) are presented visually (Fig. 9 - Fig. 12) as a collection of five radar graphs and one collective graph, for each database considered in the experiments.

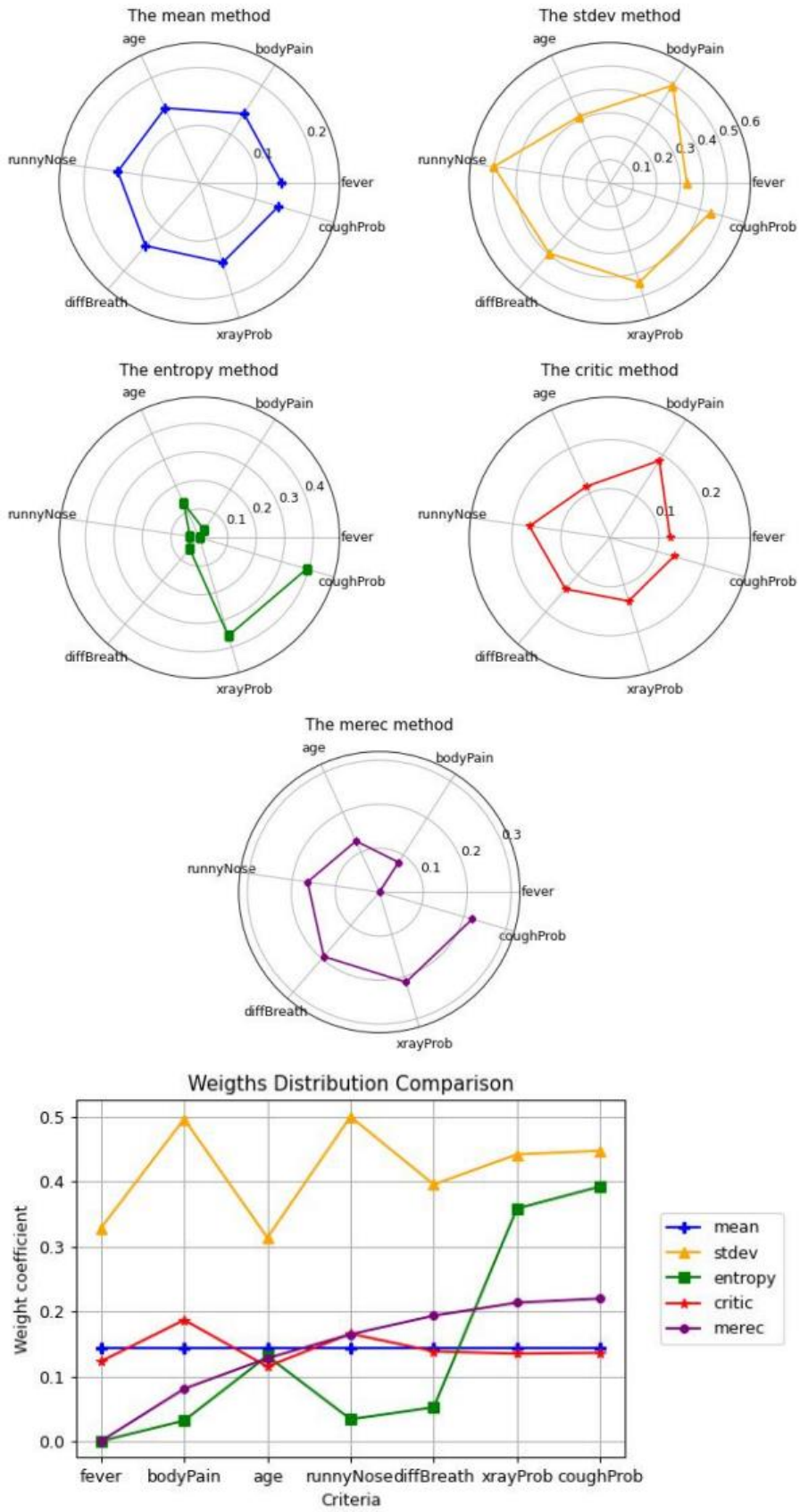


Figure 9: Weighting methods comparison for Symptoms x X-ray x Cough database.

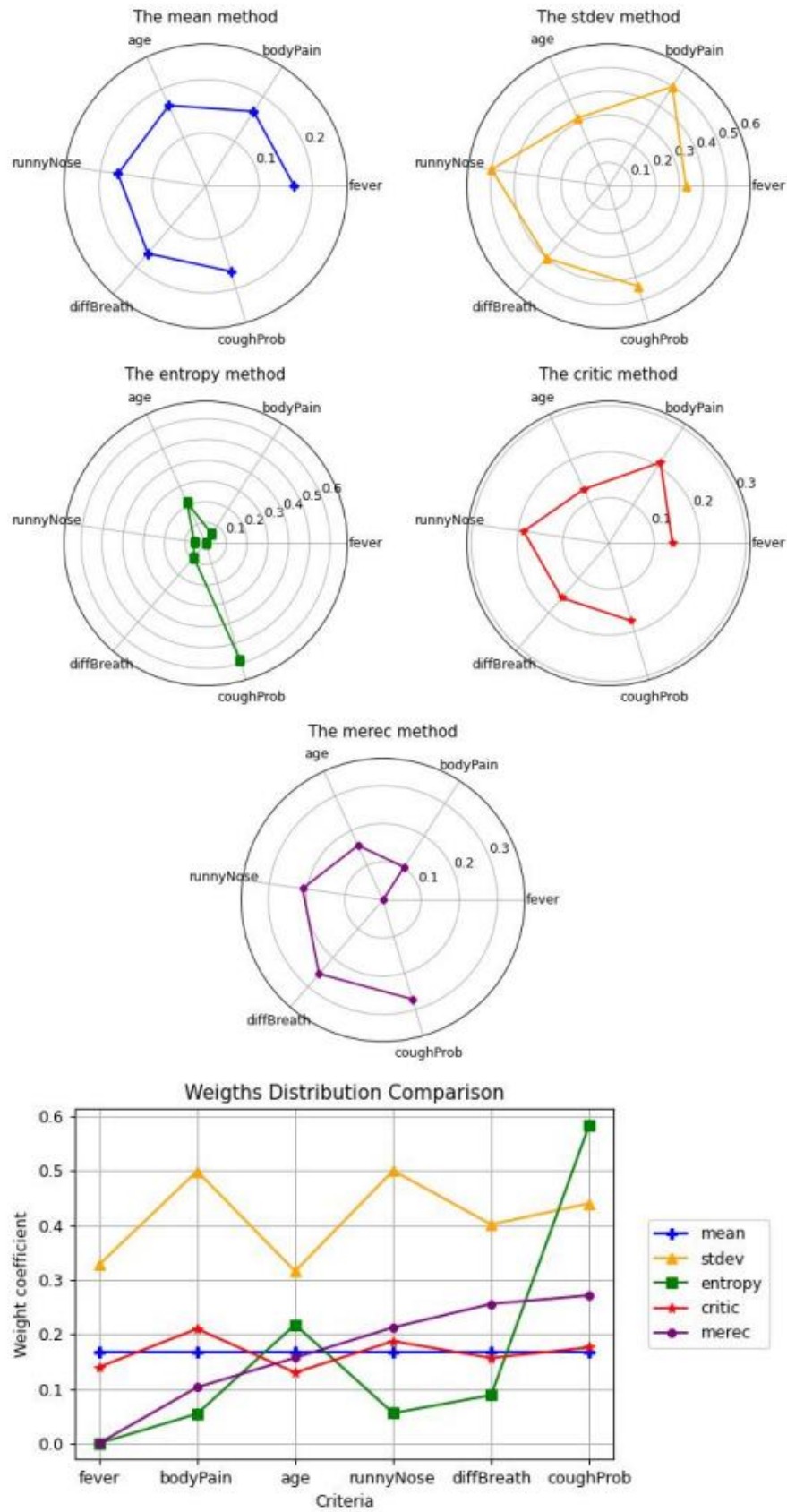


Figure 10: Weighting methods comparison for Symptoms x Cough database

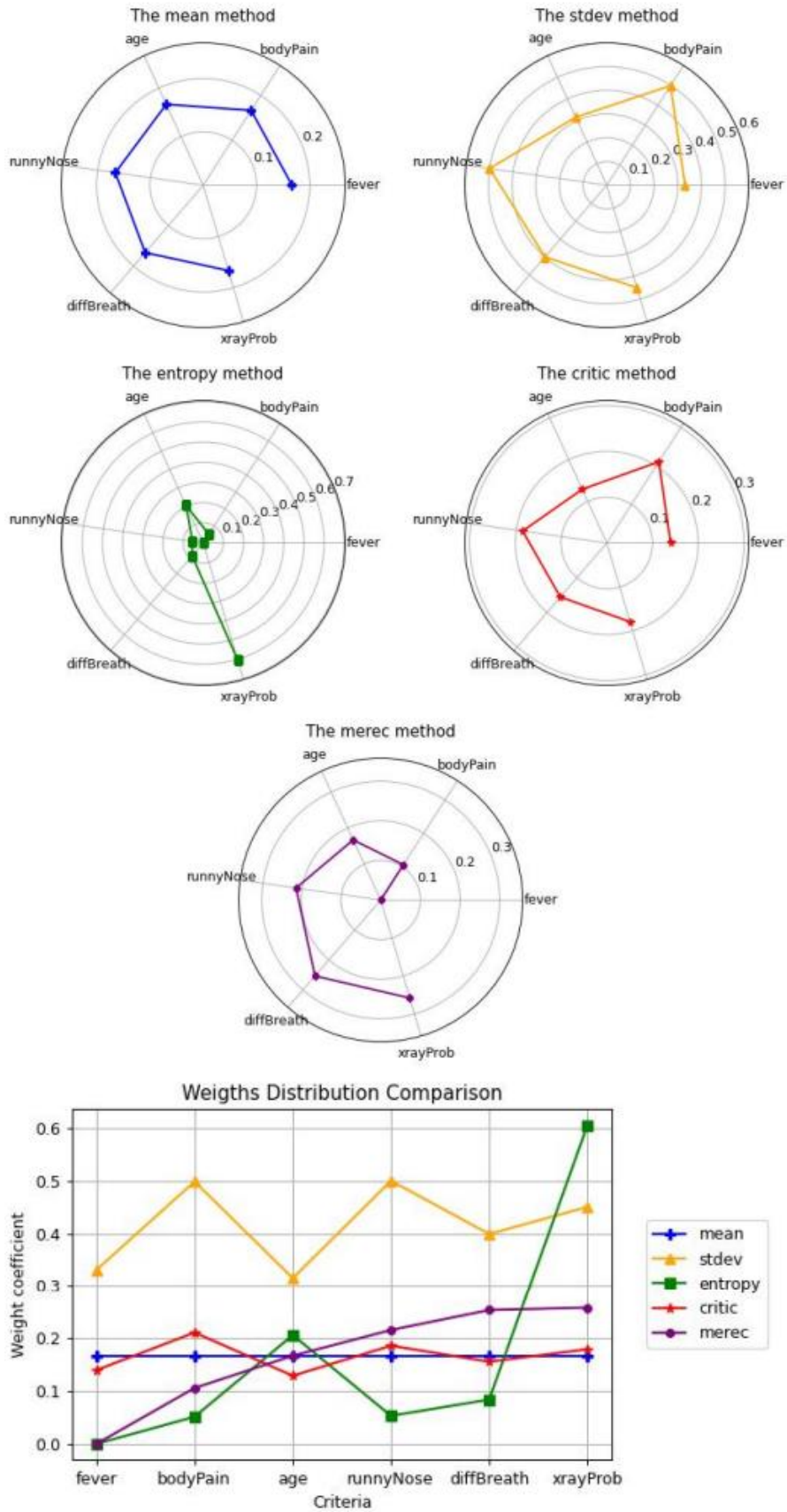


Figure 11: Weighting methods comparison for Symptoms x X-ray database

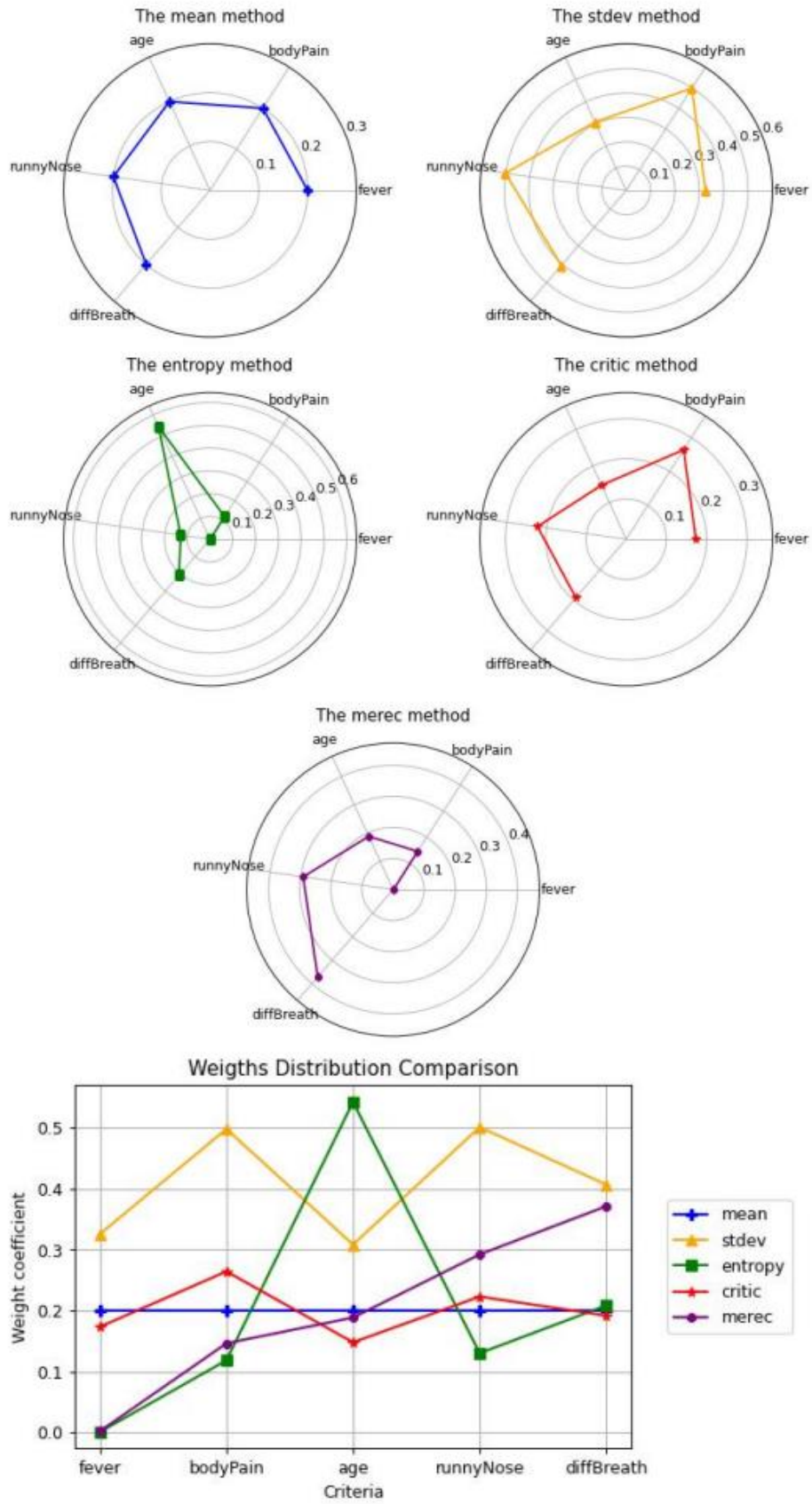


Figure 12: Weighting methods comparison for Symptoms database

## TOP x accuracy graph

In order to visualize the effectiveness of ranking, TOP x accuracy graphs for ranking vector  $R$  were created which can be presented as a function:

$$TOP(x) = \frac{\sum_{i=1}^x \begin{cases} 1 & \text{if } R_i \text{ is a positive sample} \\ 0 & \text{if } R_i \text{ is a negative sample} \end{cases}}{x} \quad (15)$$

The TOP(x) function graph representing ideal ranking  $R$  should then look like Fig. 3. This function, and metrics directly derived from it were used to assess the effectiveness of weighting methods.

## 4.2. Metrics

All of the obtained results can be listed as:

- TOP x% accuracy.
- Mean Square Error (MSE) (for the TOP(x) function).
- Mean Absolute Error (MAE) (for the TOP(x) function).
- Difference between the perfect and the obtained area under the first half of the TOP(x) function.

As can be observed in Fig. 4 - 8, using more sophisticated weighting techniques before the decision-making module was beneficial for the final ranking in most cases. Even though all of the methods did very poorly with the symptoms-only database, considerable improvement was made with the other three. In terms of Symptoms x Cough and Symptoms x X-ray databases, mean and standard-deviation methods scored seemingly and both MEREC and CRITIC methods did slightly better than the previous two. However, the entropy weighting method performed exceptionally better (despite the initial drop in the Symptoms x Cough database), than any other method. The same conclusion can be drawn from the ranking of Symptoms x Cough x X-ray database, where, while all the other methods performed equally or better than mean weighting, the entropy weighting method performed far better than any other technique, with the mean squared error (comparing to the ideal ranking in Fig. 3) equal to  $7e-6$ , while mean method had the mean squared error of  $5.52e-4$ , as can be read in Tab. 3 and 1.

**Table 1**

Metrics (Mean weighting)

|          | Symptoms x Cough x X-ray | Symptoms x Cough | Symptoms x X-ray | Symptoms only |
|----------|--------------------------|------------------|------------------|---------------|
| AreaDiff | 12.396811                | 135.255527       | 118.51648        | 23.075534     |
| MAE      | 0.012733                 | 0.095735         | 0.083449         | 0.33708       |
| MSE      | 0.000552                 | 0.01622          | 0.013124         | 0.152024      |
| TOP1%    | 1.0                      | 1.0              | 1.0              | 1.0           |
| TOP5%    | 1.0                      | 0.990099         | 0.990099         | 0.5           |
| TOP5%    | 1.0                      | 0.970149         | 0.965174         | 0.545455      |

**Table 2**

Metrics (Standard-deviation weighting)

|          | Symptoms x Cough x X-ray | Symptoms x Cough | Symptoms x X-ray | Symptoms only |
|----------|--------------------------|------------------|------------------|---------------|
| AreaDiff | 10.099273                | 106.527543       | 96.056515        | 21.041559     |
| MAE      | 0.011004                 | 0.078946         | 0.070611         | 0.322613      |
| MSE      | 0.000452                 | 0.011103         | 0.00978          | 0.135964      |
| TOP1%    | 1.0                      | 1.0              | 1.0              | 1.0           |
| TOP5%    | 1.0                      | 0.960396         | 0.990099         | 0.666667      |
| TOP5%    | 1.0                      | 0.960199         | 0.9801           | 0.636364      |

**Table 3**

Metrics (Entropy weighting)

|          | Symptoms x Cough x X-ray | Symptoms x Cough | Symptoms x X-ray | Symptoms only |
|----------|--------------------------|------------------|------------------|---------------|
| AreaDiff | 0.833911                 | 27.632805        | 6.328391         | 24.01915      |
| MAE      | 0.000898                 | 0.025707         | 0.010469         | 0.359783      |
| MSE      | 7e-06                    | 0.000916         | 0.000369         | 0.169561      |
| TOP1%    | 1.0                      | 1.0              | 1.0              | 1.0           |
| TOP5%    | 1.0                      | 0.960396         | 1.0              | 0.5           |
| TOP5%    | 1.0                      | 0.970149         | 1.0              | 0.545455      |

**Table 4**

Metrics (CRITIC weighting)

|          | Symptoms x Cough x X-ray | Symptoms x Cough | Symptoms x X-ray | Symptoms only |
|----------|--------------------------|------------------|------------------|---------------|
| AreaDiff | 20.197108                | 107.848585       | 85.901988        | 21.772522     |
| MAE      | 0.017986                 | 0.07938          | 0.06741          | 0.323382      |
| MSE      | 0.000993                 | 0.01107          | 0.00884          | 0.138473      |
| TOP1%    | 1.0                      | 1.0              | 1.0              | 1.0           |
| TOP5%    | 1.0                      | 0.970297         | 1.0              | 0.5           |
| TOP5%    | 1.0                      | 0.965174         | 0.975124         | 0.545455      |

**Table 5**

Metrics (MERECA weighting)

|          | Symptoms x Cough x X-ray | Symptoms x Cough | Symptoms x X-ray | Symptoms only |
|----------|--------------------------|------------------|------------------|---------------|
| AreaDiff | 6.433779                 | 78.672207        | 67.39159         | 23.024529     |
| MAE      | 0.006169                 | 0.059859         | 0.052489         | 0.340997      |
| MSE      | 0.000157                 | 0.006404         | 0.006109         | 0.154003      |
| TOP1%    | 1.0                      | 1.0              | 1.0              | 1.0           |
| TOP5%    | 1.0                      | 0.980198         | 1.0              | 0.5           |
| TOP5%    | 1.0                      | 0.970149         | 1.0              | 0.545455      |

## 5. Conclusion

The proposed solution can be used to solve real-life problems, such as the one considered in this paper. MCDMs can be used as accurate, unbiased ranking agents, which can be further enhanced by using adequate weighting methods prior. Based on the obtained results, the use of entropy weighting along with the MultiMOORA method is one of the best criteria methods. The proposal can be used in various types of systems where there is a lot of information on the basis of which a decision should be made, e.g. medical systems or federated learning. In future work, we plan to focus on adapting new multicriteria methods in the federated learning process.

## Acknowledgements

This work is supported by the Rector's mentoring project at the Silesian University of Technology.



## References

- [1] D. Połap, Fuzzy consensus with federated learning method in medical systems, *IEEE Access* 9 (2021) 150383–150392.
- [2] P. S. Abril, R. Plant, The patent holder's dilemma: Buy, sell, or troll?, *Communications of the ACM* 50 (2007) 36–44. doi:10.1145/1188913.1188915.
- [3] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, S. M. S. Islam, Machine learning for detecting covid19 from cough sounds: An ensemble-based mcdm method, *Computers in Biology and Medicine* 145 (2022) 105405.
- [4] Z. Shang, X. Yang, D. Barnes, C. Wu, Supplier selection in sustainable supply chains: Using the integrated bwm, fuzzy shannon entropy, and fuzzy multimora methods, *Expert Systems with Applications* 195 (2022) 116567.
- [5] A. Krawinkler, R. J. Breiteneker, D. Maresch, Heuristic decision-making in the green energy context: Bringing together simple rules and data-driven mathematical optimization, *Technological Forecasting and Social Change* 180 (2022) 121695.
- [6] D. Połap, M. Woźniak, Meta-heuristic as manager in federated learning approaches for image processing purposes, *Applied Soft Computing* 113 (2021) 107872.
- [7] C. T. Kamanga, E. Bugingo, S. N. Badibanga, E. M. Mukendi, A multi-criteria decision making heuristic for workflow scheduling in cloud computing environment, *The Journal of Supercomputing* 79 (2023) 243–264.
- [8] M. El Khatib, A. Al Mulla, W. Al Ketbi, The role of blockchain in e-governance and decision-making in project and program management, *Advances in Internet of Things* 12 (2022) 88–109.
- [9] K. Prokop, D. Połap, G. Srivastava, J. C.-W. Lin, Blockchain-based federated learning with checksums to increase security in internet of things solutions, *Journal of Ambient Intelligence and Humanized Computing* (2022) 1–10.
- [10] I. Erol, I. M. Ar, I. Peker, Scrutinizing blockchain applicability in sustainable supply chains through an integrated fuzzy multi-criteria decision making framework, *Applied Soft Computing* 116 (2022) 108331.
- [11] V. Rudzionis, A. Lopata, S. Gudas, R. Butleris, I. Veitaite, D. Dilijonas, E. Grišius, M. Zwitterloot, K. Rudzioniene, Identifying irregular financial operations using accountant comments and natural language processing techniques, *Applied Sciences* 12 (2022) 8558.
- [12] M. H. Ali, M. M. Jaber, S. K. Abd, A. Rehman, M. J. Awan, D. Vitkute-Adžgauskienė, R. Damaševičius, S. A. Bahaj, Harris hawks sparse auto-encoder networks for automatic speech recognition system, *Applied Sciences* 12 (2022) 1091.
- [13] T. Hussain, D. Hussain, I. Hussain, H. AlSalman, S. Hussain, S. S. Ullah, S. Al-Hadhrami, Internet of things with deep learning-based face recognition approach for authentication in control medical systems, *Computational and Mathematical Methods in Medicine* 2022 (2022).
- [14] A. Sardar, S. Umer, R. K. Rout, S.-H. Wang, M. Tanveer, A secure face recognition for iot-enabled healthcare system, *ACM Transactions on Sensor Networks (TOSN)* (2022).
- [15] W. K. M. Brauers, E. K. Zavadskas, Robustness of multimora: A method for multi-objective optimization, *Informatica* 23 (2012) 1–25. doi:10.15388/Informatica.2012.346.