



*Research article*

## **A bio-inspired weights and structure determination neural network for multiclass classification: Applications in occupational classification systems**

**Yu He<sup>1,2,3</sup>, Xiaofan Dong<sup>1,2,3</sup>, Theodore E. Simos<sup>4,5,6,7,8,\*</sup>, Spyridon D. Mourtas<sup>9,10</sup>, Vasilios N. Katsikis<sup>9</sup>, Dimitris Lagios<sup>11</sup>, Panagiotis Zervas<sup>11</sup> and Giannis Tzimas<sup>11</sup>**

<sup>1</sup> School of Computer Science and Artificial Intelligence, Huanghuai University, Zhumadian 463000, China

<sup>2</sup> Henan Key Laboratory of Smart Lighting, Zhumadian 46300, China

<sup>3</sup> Henan International Joint Laboratory of Behavior Optimization Control for Smart Robots, Henan 463000, China

<sup>4</sup> Center for Applied Mathematics and Bioinformatics, Gulf University for Science and Technology, West Mishref, 32093 Kuwait

<sup>5</sup> Department of Medical Research, China Medical University Hospital, China Medical University, Taichung City 40402, Taiwan, China

<sup>6</sup> Laboratory of Inter-Disciplinary Problems of Energy Production, Ulyanovsk State Technical University, 32 Severny Venetz Street, 432027 Ulyanovsk, Russia

<sup>7</sup> Section of Mathematics, Dept. of Civil Engineering, Democritus Univ. of Thrace, Xanthi 67100, Greece

<sup>8</sup> Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal Univ., Neijiang 641100, China

<sup>9</sup> Department of Economics, Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece

<sup>10</sup> Laboratory “Hybrid Methods of Modelling and Optimization in Complex Systems,” Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia

<sup>11</sup> Data and Media Laboratory, Department of Electrical and Computer Engineering, University of Peloponnese, Patras, Greece

\* **Correspondence:** Email: [simos@ulstu.ru](mailto:simos@ulstu.ru).

**Abstract:** Undoubtedly, one of the most common machine learning challenges is multiclass classification. In light of this, a novel bio-inspired neural network (NN) has been developed to address multiclass classification-related issues. Given that weights and structure determination (WASD) NNs have been acknowledged to alleviate the disadvantages of conventional back-propagation NNs, such

as slow training pace and trapping in a local minimum, we developed a bio-inspired WASD algorithm for multiclass classification problems (BWASDC) by using the metaheuristic beetle antennae search (BAS) algorithm to enhance the WASD algorithm's learning process. The BWASDC's effectiveness is then evaluated through applications in occupational classification systems. It is important to mention that systems of occupational classification serve as a fundamental indicator of occupational exposure. For this reason, they are highly significant in social science research. According to the findings of four occupational classification experiments, the BWASDC model outperformed some of the most modern classification models obtainable through MATLAB's classification learner app on all fronts.

**Keywords:** neural networks; occupational classification; multiclass classification; machine learning; weights and structure determination; metaheuristics; beetle antennae search

**Mathematics Subject Classification:** 68T10, 65F20, 91B40

---

## 1. Introduction

Today's cutting-edge technologies, such as machine learning and natural language processing, significantly minimize the amount of work required to address multiclass classification-related issues [1, 2]. Typically, the classification task in machine learning occurs in a number of disciplines, including engineering [3], medicine [4], economics and finance [5, 6]. In these disciplines, multiclass classification is an important problem. Every instance in the training set belongs to a unique label set that was formerly created for multiclass classification. Building a learning model from a training set of labeled data is the aim of supervised classification techniques, with the goal that it may categorize fresh data with unseen labels [7]. By expanding the binary classification problem, the multiclass classification problem can be resolved using a variety of machine learning techniques, such as  $k$ -Nearest Neighbors, Support Vector Machines, NNs, Naive Bayes and decision trees [8].

Primarily utilized for classification and regression issues, NNs have been successfully applied in a number of disciplines, including engineering, medicine, economics and finance. They are commonly utilized in the field of engineering for linear time varying systems and feedback control system stabilization [9, 10], solar system measurements [11] and alloy behavior analysis [12]. Additionally, NNs are frequently used in the field of medicine to diagnose conditions including flat feet [13], breast cancer [14], lung cancer [15], and diabetic retinopathy [16], while they are commonly utilized in the fields of finance and economics for portfolio management [17], exchange rate analysis [18], time series forecasting [19] and macroeconomic factors prediction [20]. In this paper, a novel bio-inspired NN has been developed to address multiclass classification-related issues, and its effectiveness is then evaluated through applications in occupational classification systems.

Systems of occupational classification serve as frameworks for classifying jobs and related information. They are frequently used by government agencies to standardize the manner in which job descriptions and data on employment are gathered [21]. These systems therefore function as a fairly basic indication of occupational exposure, which is why they are highly significant in social science research [22]. In general, multiclass classification tasks are extensively used in social science research, including analyzing the connection between cancer and changes in occupational characteristics [23], describing occupational mobility [24], carrying out case-control studies in the

healthcare industry [25], and assessing the viability of teleworking in certain occupations [26]. By translating a person's job description and the nature of their business into a standardized numerical code, public health professionals and academics can analyze patterns and trends in work-related illnesses, accidents, and exposures [27]. The systematic monitoring of any population, from the general population of the world to the people who compose a medium or small-sized business or town, is based on a variety of classification systems [28]. In the case of occupational classification, there exist several frequently updated national and international classification schemes [29], such as the Occupational Information Network Standard Occupational Classification (O\*NET-SOC) and the International Standard Classification of Occupations (ISCO). Due of their widespread use, O\*NET-SOC and ISCO are used in this study.

The U.S. Department of Labor developed the Occupational Information Network (O\*NET), a comprehensive system for gathering, organizing, characterizing, and disseminating information on occupational characteristics and worker traits [30]. The O\*NET makes information available online in a searchable database, making it simpler to acquire data on occupations at various levels of detail and enhancing its usefulness for a range of users. O\*NET tools and products can be used by businesses and human resources professionals for a variety of tasks, such as creating job descriptions, increasing the number of qualified applicants for open positions, coordinating organizational development with workplace requirements, and adjusting recruitment and training objectives [31]. To find jobs that match their interests, values, skills, and experience, explore career growth profiles using the most recent labor market data, make wise career decisions to maximize earning potential and job satisfaction, and develop an understanding of what it takes to succeed in their fields and in related occupations, job seekers can use the information provided by O\*NET [32]. Researchers who investigate topics relating to the U.S. workplace and labor market can benefit greatly from O\*NET. To stay current with the jobs currently available on the labor market, the O\*NET-SOC taxonomy is continually updated. Crosswalks between the original professions that were evaluated based on the work values and the new occupations have been created as part of these adjustments. On the other hand, ISCO was developed to make comparing occupational statistics between nations easier and to serve as guidelines for countries developing or adjusting their own national occupational classification schemes. It was created to fulfill a variety of administrative and research needs, and it is fully endorsed by the international community as an accepted norm for global labor statistics [33].

Crosswalks between different systems of occupation classification were formerly only made occasionally because they demand significant effort in terms of time and resources. For example, a crosswalk to go from the O\*NET-SOC 2000 classification to ISCO-88 was used to explore employment consequences of automation and offshoring [34]. A crosswalk to go from the Brazilian Occupation Classification (BOC) 2002 to O\*NET-SOC 2010 was used to analyze occupational profiles in the Brazilian workforce [35]. A crosswalk to go from the ISCO-88 to O\*NET-SOC 2010 was used to analyze the Covid-19 pandemic impact of the global process of automation on employment in a developing economy [36]. A crosswalk to go from the National Classification of Occupations (CNO) 2011 to ISCO-88 was used to investigate job polarization [37]. It is important to note that the most recent versions, ISCO-08 and O\*NET-SOC 2019, were adopted in 2008 and 2019, respectively, while the previous versions, ISCO-88 and O\*NET-SOC 2010, were adopted in 1988 and 2010, respectively.

The main objective of this study is to provide a crosswalk between O\*NET-SOC 2010 and ISCO-08 by using NNs. We will use a feedforward NN (FNN) that can handle multiclass classification tasks

to accomplish this. In particular, a weights and structure determination (WASD) training algorithm will be employed as an alternative to the popular back-propagation technique, which is utilized to train FNNs. The WASD approach, as opposed to the back-propagation strategy, which iteratively adjusts the network's topology, computes the optimal set of weights directly utilizing the weights direct determination (WDD) method. This helps to reduce computational complexity and prevents the occurrence of local minima [38].

Nowadays, one of the rapidly expanding fields of study in evolutionary computation is memetic algorithms [10]. To enhance the performance of the WASD based NNs for multiclass classification-related issues, the metaheuristic beetle antennae search (BAS) algorithm [39] is combined with the WASD algorithm in this research. This is done in consideration of the strong performance of the bio-inspired WASD algorithms for regression-related issues proposed in [20, 40]. As a result, we introduce a new bio-inspired WASD for multiclass classification tasks (BWASDC) algorithm to train a 3-layer FNN. The advantages of the BWASDC algorithm over conventional WASD algorithms are listed below:

- it can more accurately determine the minimum optimal hidden layer structure of the NN with the use of BAS;
- the use of BAS aids the BWASDC in preventing the development of local minima more effectively.

It is noteworthy to highlight that BAS, which is capable of effective global optimization, has recently gained substantial traction in a number of scientific domains, including machine learning [10], robotics [41], engineering [42–44] and finance [20], and that it has undergone a number of modifications, including binary [45] and semi-integer [46] versions, to better address various tasks. Furthermore, taking into account two official ISCO and O\*NET-SOC datasets, we employ the official crosswalk to build concordance between the two codes, O\*NET-SOC 2010 and ISCO-08. This concordance is then utilized for assessing the performance of the BWASDC NN. According to the findings of four occupational classification experiments, the BWASDC model outperformed some of the most modern classification models obtainable through MATLAB's classification learner app on all fronts.

The following is a list of the paper's contributions.

- A new 3-layer bio-inspired WASD FNN for multiclass classification tasks, named BWASDC, is introduced.
- The BWASDC algorithm combines the WASD and BAS algorithms to further enhance the structure and performance of the WASD based NNs when dealing with multiclass classification-related issues.
- The O\*NET-SOC 2010 and ISCO-08 codes' official concordance datasets are utilized to assess the performance of the BWASDC NN.
- In four experiments, the BWASDC model is compared with some of the most modern classification models available through MATLAB's classification learner app. The fine tree (FTR), narrow NN (NNN), ensemble bagged trees (EBT) and fine  $k$ -nearest neighbors (FKN) are these classification models [47]. Furthermore, a comparison is also made between the BWASDC model and the deep learning transformer MATLAB model (TRA) from [48].

The sections that follow provide a description of the paper's structure. A breakdown of the WDD process for multiclass classification tasks and the 3-layer BWASDC FNN structure is given in Section

2. Additionally, Section 2 provides an extensive description of the BWASDC algorithm, along with the entire procedure for training and testing the BWASDC NN model. The findings of four crosswalk experiments between occupational classifications using the BWASDC model and some of the most modern classification models obtainable through MATLAB's classification learner app are displayed and discussed in Section 3. It is important to mention that the datasets utilized in our research and the steps necessary to prepare them for usage with the BWASDC NN are also described in Section 3. Finally, Section 4 includes closing remarks and reflections.

## 2. The bio-inspired WASD neural network model

This section describes the WDD process for multiclass classification-related issues, the structure of the 3-layer bio-inspired FNN and the BWASDC algorithm.

### 2.1. The WDD process for multiclass classification tasks

Because it avoids the need for tedious, time-consuming, and usually inaccurate repetitive computations to obtain the necessary weights matching the present hidden layer structure, the WDD process is a crucial part of any WASD technique. In comparison to conventional weight determination methods, the WDD procedure is said to provide for both lower computational complexity and speed while solving some of the associated problems [38]. It is vital to mention that the WDD only accepts input data in the form of real numbers. The data must also be normalized to a range of  $[0, 1]$  before being fed into the NN model. In this way, the NN can handle over-fitting. If required, we can accomplish that by using the linear transformation shown in [5].

Here, thorough justifications of significant theoretical foundations and research are offered for the development of the BWASDC NN. However, it is crucial to first note a few of the major symbols employed in this paper:  $()^T$  signifies transposition;  $\zeta!$  signifies the factorial of  $\zeta$ ;  $()^\dagger$  signifies pseudoinversion;  $()^\odot$  signifies the elementwise exponential;  $\text{round}(\cdot)$  signifies a round function.

Below is a restatement of the Taylor polynomial approximation (TA) theorem from [49].

**Theorem 2.1.** *When a target function,  $U(\cdot)$ , has the  $(\Theta + 1)$ -order continuous derivative on the range  $[\gamma_1, \gamma_2]$ , and  $\Theta$  is a nonnegative integer, the following is true:*

$$U(\zeta) = B_\Theta(\zeta) + C_\Theta(\zeta), \quad \zeta \in [\gamma_1, \gamma_2], \quad (2.1)$$

where  $C_K(\zeta)$  and  $B_K(\zeta)$  signify the error term and  $\Theta$ -order TA of  $U(\zeta)$ , respectively.

Consider  $U^{(\delta)}(\beta)$  to be the value of the  $\delta$ -order derivative of  $U(x)$  at the point  $\beta$ . Below is shown the approximate representation of  $U(\zeta)$ :

$$U(\zeta) \approx B_\Theta(\zeta) = \sum_{\delta=0}^{\Theta} \frac{f^{(\delta)}(\beta)}{\delta!} (\zeta - \beta)^\delta, \quad \beta \in [\gamma_1, \gamma_2]. \quad (2.2)$$

**Proposition 2.1.** *Theorem 2.1 may be utilized for multivariable function approximation. Consider  $U(\zeta_1, \zeta_2, \dots, \zeta_v)$  to be the target function with  $v$  variables and  $(\Theta + 1)$ -order continuous partial derivatives in an origin's neighborhood  $(0, \dots, 0)$ . Below is shown the  $\Theta$ -order TA  $B_\Theta(\zeta_1, \zeta_2, \dots, \zeta_v)$*

about the origin:

$$B_{\Theta}(\zeta_1, \zeta_2, \dots, \zeta_v) = \sum_{h=0}^{\Theta} \sum_{\delta_1+\dots+\delta_v=h} \frac{\zeta_1 \cdots \zeta_v}{\delta_1 \cdots \delta_v} \left( \frac{\partial^{\delta_1+\dots+\delta_v} U(0, \dots, 0)}{\partial \zeta_1^{\delta_1} \cdots \partial \zeta_v^{\delta_v}} \right), \quad (2.3)$$

where  $\delta_1, \delta_2, \dots, \delta_v$  are nonnegative integers.

Consider the input  $A = [A_1, A_2, \dots, A_m] \in \mathbb{R}^{1 \times m}$  and the target vector  $D \in \mathbb{R}$ . Based on the power activated multi-input NNs presented in [38], the link between the input variables  $A_1, A_2, \dots, A_m$  and the NN's output target  $D$  can be expressed using the nonlinear function presented below:

$$U(A_1, A_2, \dots, A_m) = D. \quad (2.4)$$

Further, in line with Proposition 2.1, the  $\Theta$ -order TA  $B_{\Theta}(A_1, A_2, \dots, A_m)$  may map (2.4) as presented below:

$$B_{\Theta}(A_1, A_2, \dots, A_m) = \sum_{h=0}^{n-1} k_h w_h, \quad (2.5)$$

where  $k_h = G_h(A_1, A_2, \dots, A_m) \in \mathbb{R}^{1 \times mn}$  refers to a power activation function,  $w_h \in \mathbb{R}^{mn}$  is the weight that corresponds to  $k_h$ , and  $h$  implies both the power value and the hidden layer neurons' number.

For a given number of samples  $r \in \mathbb{N}$ , the input becomes a matrix  $A = [A_1, A_2, \dots, A_m] \in \mathbb{R}^{r \times m}$ , where  $A_i \in \mathbb{R}^r$  for  $i = 1, \dots, m$ , and the target vector  $D \in \mathbb{R}^r$ . Thereafter, setting  $k_{r,h} = G_h(A_1, A_2, \dots, A_m) \in \mathbb{R}^{r \times mn}$ , the input-activation matrix  $K$  and the weight vector  $W$  are presented below:

$$K = \begin{bmatrix} k_{1,0} & k_{1,1} & \dots & k_{1,n-1} \\ k_{2,0} & k_{2,1} & \dots & k_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{r,0} & k_{r,1} & \dots & k_{r,n-1} \end{bmatrix} \in \mathbb{R}^{r \times mn}, \quad W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \dots \\ w_{n-1} \end{bmatrix} \in \mathbb{R}^{mn}. \quad (2.6)$$

Then, using the WDD technique described below, the weights of the  $\Theta$ -order TA NN are directly generated, as opposed to utilizing the iterative weight training methods used in conventional NNs [49]:

$$W = K^{\dagger} D. \quad (2.7)$$

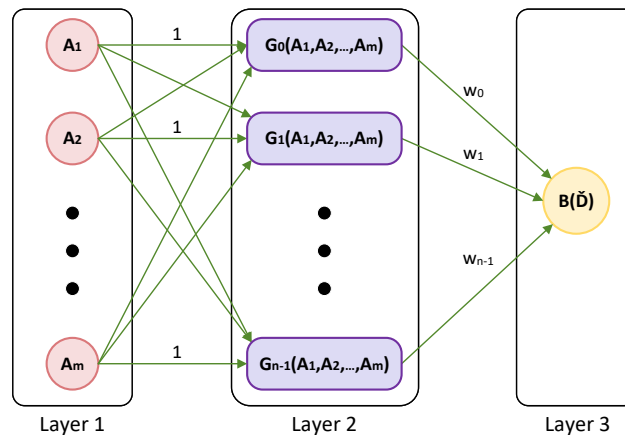
It is crucial to note that the next power maxout elementwise activation function is recommended when dealing with multiclass classification tasks:

$$G_h(A_{ij}) = \begin{cases} A_{ij}^h, & A_{ij}^h = \max(A_i^{\odot h}) \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } i = 1, 2, \dots, r \text{ and } j = 1, 2, \dots, m, \quad (2.8)$$

where  $A_i$  is the  $i$ -th row, and  $A_{ij}$  is the  $ij$ -th element of the input matrix  $A$ .

## 2.2. The neural network structure

Figure 1 depicts the 3-layer FNN architecture. In particular, the NN gets the input values  $A_1, A_2, \dots, A_m$  from Layer 1 (also known as the input layer) and distributes them equally to the



**Figure 1.** Structure of the BWASDC NN.

appropriate neuron of Layer 2. Observe that there is a maximum number  $n$  of active neurons in Layer 2. Additionally, the WDD is utilized to acquire the neurons that connect Layers 2 and 3, and they have weights  $W_c, c = 0, 1, \dots, n - 1$ . The formula described below is utilized to calculate the predictions  $\check{D}$ :

$$\check{D} = \text{round}(KW). \quad (2.9)$$

Finally, one activated neuron is present in Layer 3 (also known as the output layer), and it uses the elementwise function described below:

$$B(\check{D}_i) = \begin{cases} \max(D) & , \check{D}_i > \max(D) \\ \check{D}_i & , \min(D) \leq \check{D}_i \leq \max(D) \\ \min(D) & , \check{D}_i < \min(D) \end{cases}, \quad \text{for } i = 1, 2, \dots, r. \quad (2.10)$$

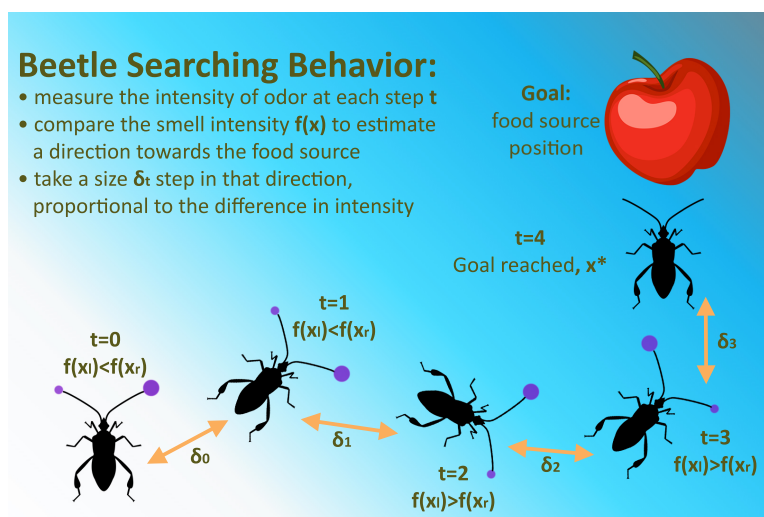
### 2.3. The BWASDC algorithm

The NN model is trained using the BWASDC algorithm, which incorporates the BAS algorithm [39]. It is important to mention that the strength of the odor that beetles perceive on the antennas determines how they use their two antennas to find food (see Figure 2). The BAS algorithm's optimal solution finder mimics this tendency, and this strategy enables the application of cutting-edge optimization techniques (see [50–52]). By imitating the beetle's behavior, BWASDC finds the optimal number of the NN's hidden layer neurons along with their power values. The next steps,  $\mathbf{S}_1$  to  $\mathbf{S}_3$ , outline the BWASDC algorithmic procedure.

$\mathbf{S}_1$ : First, an objective function must be defined. Consider the training set  $X_{tr} \in \mathbb{R}^{r \times m}$ , where  $r$  is the number of samples, and their target  $D_{tr} \in \mathbb{R}^r$ . The  $K$  matrix is created in line with (2.8) and (2.6), and the weights of the NN are calculated by (2.7) using  $D_{tr}$ . Then, the NN predictions  $\check{D}_{tr}$  are obtained by (2.9), and the mean absolute error (MAE) between  $\check{D}_{tr}$  and the target value  $D_{tr}$  is measured through the following formula:

$$\text{MAE} = \frac{1}{r} \sum_{i=1}^r |D_i - \check{D}_i|. \quad (2.11)$$

Note that the MAE measures errors between paired observations that describe the same phenomenon, and it is frequently applied as a loss function in machine learning for classification-related issues. Considering the vector  $x$ , which contains the power values of the NN's hidden layer neurons, the aforementioned process is formulated as an objective function in Algorithm 1.



**Figure 2.** The BAS behavior.

---

### Algorithm 1 Objective function.

---

**Require:** The data input  $X$ , the target  $D$ , and the vector  $x$ .

- 1: **procedure** FITNESS( $X, D, x$ )
- 2:   Set  $r$  the number of the rows of  $X$ .
- 3:   Set in  $N$  only the nonnegative elements of  $x$ .
- 4:   Set the maximum value of  $X_i$ , for  $i = 1, 2, \dots, r$ , to 1.
- 5:   Calculate the matrix  $K$  according to (2.8) and (2.6) under the power values contained in  $N$ .
- 6:   Find  $W$  via the WDD method using  $K$  and  $D$ .
- 7:   Find  $\check{D}$  via (2.9) using  $K$  and  $W$ .
- 8:   Set  $E$  the MAE computed via (2.11) between  $\check{D}$  and  $D$ .
- 9: **end procedure**

**Ensure:** The MAE  $E$ .

---

**S<sub>2</sub>:** Second, the objective function in Algorithm 1 is minimized by mimicking the beetle's behavior. Assume the vector  $x$ , where its elements take the integer values  $0, 1, \dots, n_{\max} - 1$  or  $n_{\max}$  with  $n_{\max}$  denoting the user-specified maximum number of hidden layer neurons. These  $n_{\max} + 1$  numbers correspond to the activation functions' powers for each hidden layer neuron. For example, if  $x = [5, 7]^T$ , it denotes the existence of two hidden layer neurons, the first of which uses the power of 5 in (2.8), and the second of which uses the power of 7.

In our approach, the aforementioned vector  $x$  represents the beetle's position, and the objective function  $f(x)$  in Algorithm 1 represents the odor concentration at the position  $x$ , with the minimum value of  $f(x)$  serving as a link to the odor's source. To determine the hidden layer



neurons' number in the NN at the position  $x$ , the following functions will be used:

$$v(x_i) = \begin{cases} 1, & x_i > 0 \\ 0, & x_i \leq 0 \end{cases}, \quad g(x) = \sum_{i=1}^{n_{\max}} v(x_i). \quad (2.12)$$

Additionally, to represent the beetle's position at the  $t$ -th moment, we use the notation  $x^t$  with  $t = 1, 2, 3, \dots, t_{\max}$  with  $t_{\max}$  denoting the maximum number of iterations specified by the user. The beetle's erratic search path thus defines the paradigm of searching behavior as follows:

$$h = \frac{\lambda}{\epsilon + \|\lambda\|}, \quad (2.13)$$

where  $\lambda \in \mathbb{R}^{n_{\max}}$  denotes a vector of  $n_{\max}$  random elements, and  $\epsilon = 2^{-52}$ . The right ( $x_R$ ) and left ( $x_L$ ) antennae are formulated as below to replicate the searching behaviors of the beetle's antennae:

$$x_R = \text{round}(x^t + \zeta^t h), \quad x_L = \text{round}(x^t - \zeta^t h), \quad (2.14)$$

where  $\zeta^t$  denotes the antennae's sensing width that correlates to exploitation capacity at the  $t$ -th moment. Additionally, consider the candidate optimal solution ( $x_C$ ),

$$x_C = \text{round}(x^t + \xi^t \zeta^t \text{sign}(f(x_L) - f(x_R))), \quad (2.15)$$

where  $\xi^t$  denotes a size step that takes into consideration the convergence's pace following an increase in  $t$  during the search. Then, the behavior of detecting is expressed as below:

$$x^{t+1} = \begin{cases} x_C, & f(x_C) < f(x^t) \\ x_C, & f(x_C) = f(x^t) \text{ and } g(x_C) < g(x^t) \\ x^t, & f(x_C) = f(x^t) \text{ and } g(x_C) \geq g(x^t) \\ x^t, & f(x_C) > f(x^t). \end{cases} \quad (2.16)$$

Last, the update rules for  $\zeta$  and  $\xi$  are described next:

$$\zeta^{t+1} = 0.991\zeta^t + 0.001, \quad \xi^{t+1} = 0.991\xi^t. \quad (2.17)$$

It is significant to note that the following are the initial conditions for the aforementioned strategy:

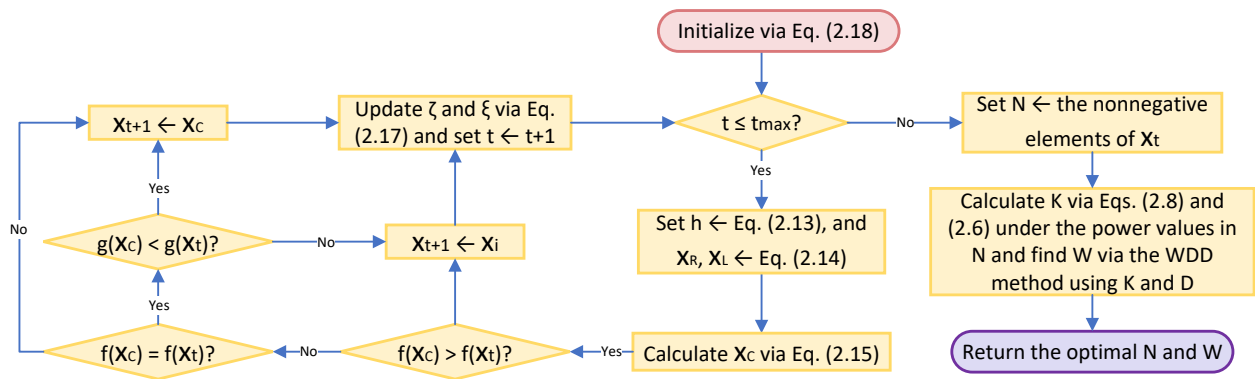
$$x^0 = [1 - k, 2 - k, \dots, n_{\max} - k]^T, \quad (2.18)$$

where  $k = \text{round}(n_{\max}/2)$ .

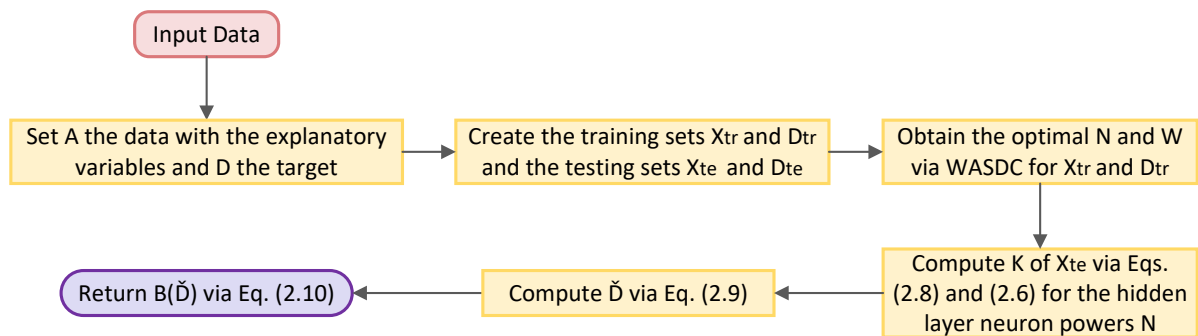
**S<sub>3</sub>:** Last, the BWASD algorithm determines and returns the optimal  $W$  on the entire training data set along with the optimal power value  $N$  of each hidden layer neuron. That is, it sets in  $N$  only the nonnegative elements of  $x^{t+1}$ , it calculates the matrix  $K$  according to (2.8) and (2.6) under the power values contained in  $N$ , and it finds the  $W$  via the WDD method using  $K$  and  $D$ .

In this manner, the NN's MAE can be decreased while the BWASDC algorithm maintains the fewest hidden layer neurons possible. The flowchart in Figure 3 shows the BWASDC algorithm's entire process.

After optimizing the structure of the BWASDC NN model of Figure 1 and determining its optimal weights, we consider the testing set  $X_{te}$  to obtain the NN predictions  $\check{D}_{te}$  through (2.9). The flowchart of Figure 4 shows the step-by-step process for predicting and modeling using the BWASDC NN model.



**Figure 3.** The BWASDC algorithm.



**Figure 4.** Process for predicting and modeling with the BWASDC NN.

### 3. Experiments between occupational classifications

Four crosswalk experiments are performed between the occupational classifications of the ISCO-ONET and ONET-ISCO datasets in this section. Notice that the creation of these datasets is described in the following Section 3.1. In these experiments, the BWASDC NN's performance is examined and contrasted with some of the most modern classification models available in the MATLAB classification learner app. The FTR, NNN, EBT and FKN are these classification models [47]. Notice that the TRA model from [48], which utilizes a pretrained bidirectional encoder with representations from a transformer model, is also compared to the BWASDC model. Additionally, in all experiments, we have set the parameters  $\zeta^0 = 20$ ,  $\xi^0 = 10$ ,  $t_{\max} = 21$  and  $n_{\max} = 10$  for the BWASDC model, while the default settings have been used in MATLAB classification models. It is important to note that you can access GitHub's full development and application of the computational techniques and concepts discussed in Sections 2 and 3.1 by visiting the following link:

<https://github.com/SDMourtas/BWASDC>.

The MATLAB repository includes thorough installation instructions along with the complete implementation.

### 3.1. Datasets and data preparation

The three datasets utilized in our research and the steps necessary to prepare them for usage with the BWASDC NN are described in this section.

ISCO is a tool for categorizing occupations into groups that are precisely specified based on the duties and responsibilities carried out in the work, and it is intended to be utilized in a variety of client-oriented tasks and statistics implementations. This split is advantageous from a social, economic, and medical standpoints [21]. The ISCO's 4-level hierarchy-based categorization system also allows for the categorization of jobs into 436 groupings of 4 digits, 130 minor groupings of 3 digits, 43 sub-major groupings of 2 digits and 10 major groupings of 1 digit. The first dataset, which comes from the International Labor Organization and includes a sample of job titles with associated ISCO-08 codes, may be accessed at the following link: <https://www.ilo.org/ilostat-files/ISCO/newdocs-08-2021/ISCO-08/ISCO-08%20EN%20Structure%20and%20definitions.xlsx>. For ease of use, we will call this dataset ISCOD. It is important to mention that the classification tasks in our approach only include the 4-digit unit groups. As a result, there are 436 distinct classes of professions in the ISCOD, and each class contains a wide range of job titles.

The work carried out in the U.S. is divided into about 1000 occupational groups using the O\*NET-SOC system. Data on the significance and level of a number of occupational variables, such as Knowledge, Skills, Abilities, Tasks, and General Work Activities, are related with these occupations. The 2018 Standard Occupational Classification (SOC) system serves as the foundation for the O\*NET-SOC system. All federal agencies that gather and disseminate occupational data are required to adopt this classification system, according to the U.S. Office of Management and Budget. Anybody can get extensive information on a job inside the O\*NET-SOC system and links to other sources of national, state, and local SOC-based occupational information by using a code and a job title [30]. The second dataset, which comes from the O\*NET resource center and includes a sample of job titles with associated O\*NET-SOC codes, can be accessed at the following link: [https://www.onetcenter.org/dl\\_files/database/db\\_20\\_1\\_excel/Sample%20of%20Reported%20Titles.xlsx](https://www.onetcenter.org/dl_files/database/db_20_1_excel/Sample%20of%20Reported%20Titles.xlsx). For ease of use, we will call this dataset ONETD. It is important to mention that there are 873 distinct classes of professions in the ONETD, and each class contains a wide range of job titles.

The third dataset, which contains the crosswalk between the O\*NET-SOC 2010 and ISCO-08, is taken from the U.S. Bureau of Labor Statistics and can be accessed at the following link: [https://www.bls.gov/soc/ISCO\\_SOC\\_Crosswalk.xls](https://www.bls.gov/soc/ISCO_SOC_Crosswalk.xls). For ease of use, we will refer to the crosswalk from ISCO-08 to O\*NET-SOC 2010 of this dataset as ISCO-ONET, and the crosswalk from O\*NET-SOC 2010 to ISCO-08 as ONET-ISCO. Particularly, ISCO-ONET consists of 1123 samples, each of which comprises an ISCO-08 code and an O\*NET-SOC 2010 code, and the job titles that go with these two codes can be taken from either ISCOD or ONETD. Additionally, ONET-ISCO consists of 1123 samples, each of which comprises an O\*NET-SOC 2010 code and an ISCO-08 code, and the job titles that go with these two codes can be taken from either ISCOD or ONETD.

Only actual numerical data can be utilized as input to train and test the BWASDC NN model owing to the restrictions imposed by the WDD method. Because strings are present in both the ISCO-ONET and ONET-ISCO datasets used in our investigation, the data must be appropriately prepared and processed before being fed into the model. The next phases,  $\mathbf{P}_1$  to  $\mathbf{P}_3$ , outline the procedure that each dataset should follow.

- P<sub>1</sub>**: For each sample  $i, i = 1, \dots, m$  of the dataset (i.e. ISCO-ONET or ONET-ISCO), a foundational vocabulary must be created. To do this, one of the two classification codes (i.e. O\*NET-SOC 2010 or ISCO-08) and the corresponding job titles must be tokenized in accordance with guidelines of Unicode Standard Annex #29 [53], with all letters being changed to lowercase and punctuation being eliminated. We therefore construct the structure vector  $Q = [q_1, q_2, \dots, q_m]$ , where  $q_i, i = 1, \dots, m$ , is the vocabulary of sample  $i$ , provided that there are  $m$  classes.
- P<sub>2</sub>**: Suppose that  $r$  samples, each containing the type of classification code we selected in **P<sub>1</sub>** and one job title, will be used to validate or test the NN model. For each sample  $i, i = 1, \dots, r$ , a vocabulary needs to be created. Each job title must be tokenized, all letters must be changed to lowercase, and punctuation must be eliminated in a manner similar to **P<sub>1</sub>** in order to accomplish this. We therefore construct the structure vector  $Z = [z_1, z_2, \dots, z_r]$ , where  $z_i$  is the vocabulary of a certain class for  $i = 1, \dots, r$ .
- P<sub>3</sub>**: Taking into account that some terms of a vocabulary in  $Z$  may belong to many separate vocabularies of samples in  $Q$ , we generate an input matrix  $A$  that comprises the rates of similarity among the vocabularies in  $Q$  to train or test the model. The input matrix  $A \in \mathbb{R}^{r \times m}$  is specifically created by taking into account the following procedure. Suppose that the aforementioned structure vectors  $Q$  and  $Z$  have vocabularies  $q_j, j = 1, \dots, m$ , and  $z_i, i = 1, \dots, r$ , that have  $k_j$  and  $h_i$  words, respectively. The ratio of the vocabulary  $q_j$  to the vocabulary  $z_i$  is then:

$$J(j, i) = \frac{1}{k_j} \sum_{h=1}^{k_j} \text{strcmp}(z_i, q_j(h)), \quad (3.1)$$

where the function  $\text{strcmp}(\cdot)$  outputs 1 (true) when the input strings are the same and 0 (false) otherwise [54]. Be aware of the fact that  $q_j(h)$  is the  $h$ -th entry (word) of the vocabulary  $q_j$  in (3.1). Additionally, the ratio of the vocabulary  $z_i$  to the vocabulary  $q_j$  is:

$$R(j, i) = \frac{1}{h_i} \sum_{h=1}^{k_j} \text{strcmp}(z_i, q_j(h)). \quad (3.2)$$

Notice that  $J(j, i)$  and  $R(j, i)$  take values in the range  $[0, 1]$ .

It should be stressed that steps **P<sub>1</sub>** through **P<sub>3</sub>** are a heuristic technique that achieves the problem's objectives, namely, text to number conversion as well as matrix standardization to avoid overfitting. For the creation of the training and testing sets, the following phases **P<sub>4</sub>** and **P<sub>5</sub>** are considered.

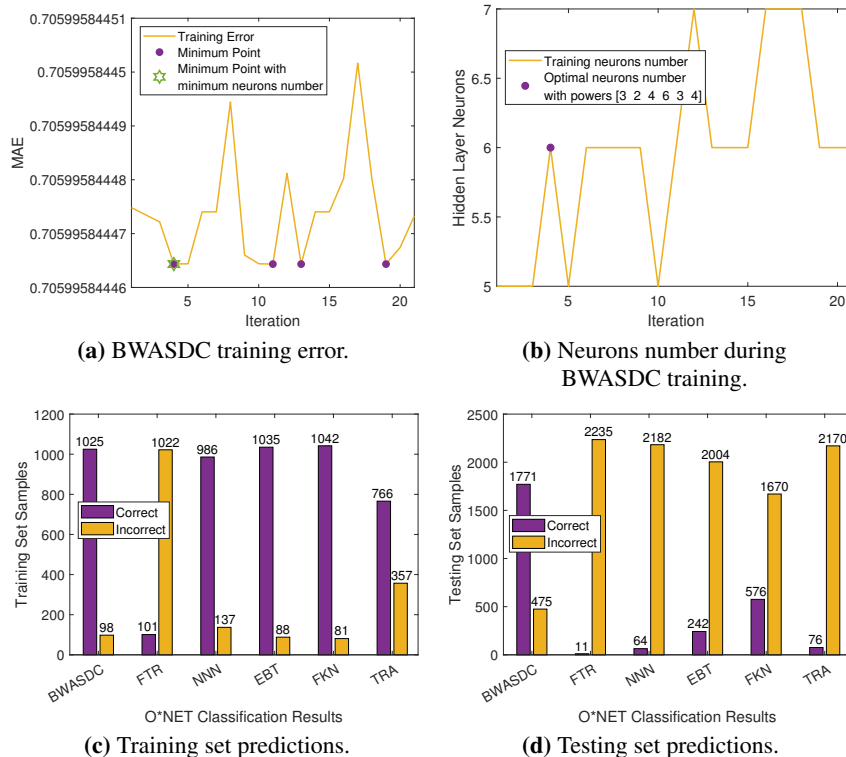
- P<sub>4</sub>**: For training the NN model, the input set  $X_{tr}$  and the target  $D_{tr}$  are created. Particularly, we set  $Z = Q$  in phase **P<sub>3</sub>** to create the input matrices  $J_{tr}, R_{tr} \in \mathbb{R}^{m \times m}$ . Following that, we set  $X_{tr} = pJ_{tr} + (1 - p)R_{tr}$  with a proportion of  $p = 1/2$ , where the  $ji$ -th entry of  $X_{tr}$  reflects the average of (3.1) and (3.2) and takes values in the range  $[0, 1]$ . Furthermore, the target vector of  $X_{tr}$  has been set to  $D_{tr} = [1, 2, \dots, m]^T \in \mathbb{R}^m$ , where each entry refers to a unique class.
- P<sub>5</sub>**: For testing the NN model, the input set  $X_{te}$  and the target  $D_{te}$  are created. To do this, a new dataset  $V_i$  is created using the chosen dataset (i.e., ISCO-ONET or ONET-ISCO). Particularly,  $V_i$  is identical to the chosen dataset with the exception that includes only the  $i$ -th occupation. Next, we set the dataset  $[V_1; V_{\text{last}}]$  in phase **P<sub>2</sub>**, and create the input matrices  $J_{te}, R_{te} \in \mathbb{R}^{2m \times m}$  in phase **P<sub>3</sub>**.

Following that, we set  $X_{te} = pJ_{te} + (1 - p)R_{te}$  with a proportion of  $p = 1/2$ , where the  $ji$ -th entry of  $X_{te}$  reflects the average of (3.1) and (3.2) and takes values in the range  $[0, 1]$ . Furthermore, the target vector of  $X_{te}$  has been set to  $D_{te} = [D_{tr}; D_{tr}] \in \mathbb{R}^{2m}$ .

After implementing the phases  $\mathbf{P}_4$  and  $\mathbf{P}_5$  in both datasets (i.e., ISCO-ONET and ONET-ISCO), we will take  $X_{tr} \in \mathbb{R}^{1123 \times 1123}$ ,  $D_{tr} \in \mathbb{R}^{1123}$ ,  $X_{te} \in \mathbb{R}^{2246 \times 1123}$  and  $D_{te} \in \mathbb{R}^{2246}$ .

### 3.2. Experiment 1

In this experiment, the job titles are taken from ONETD and used with the ISCO-ONET dataset. By using the ISCO-08 code and the job titles as input, the crosswalk from ISCO-08 to O\*NET-SOC 2010 is predicted through the NN models. The BWASDC training error is shown in Figure 5a, while the hidden layer neurons' number of its iteration of the algorithm is shown in Figure 5b. These figures show that the BWASDC algorithm needed 4 iterations to reach the smallest error with the minimum neurons' number. The minimum number of neurons is 6, and their optimal powers are  $[3, 2, 4, 6, 3, 4]^T$ . Furthermore, Figure 5c and 5d, respectively, show the predictions made based on the training and testing sets. Comparing Figure 5c and 5d, we see that the BWASDC, FKN, and EBT perform similarly on the training set, while BWASDC performs significantly better on the testing set. Particularly, Figure 5c shows that FKN has the greatest results on the training set, EBT has the 2nd greatest, BWASDC has the 3rd greatest, and FTR has the worst. On the other hand, Figure 5d shows that BWASDC has the greatest results on the testing set, FKN has the 2nd greatest, and FTR has the worst. In other words, BWASDC outperforms the other models in its ability to classify unknown data.



**Figure 5.** Crosswalk from ISCO-08 to O\*NET-SOC 2010 by NNs in the experiment of section 3.2.

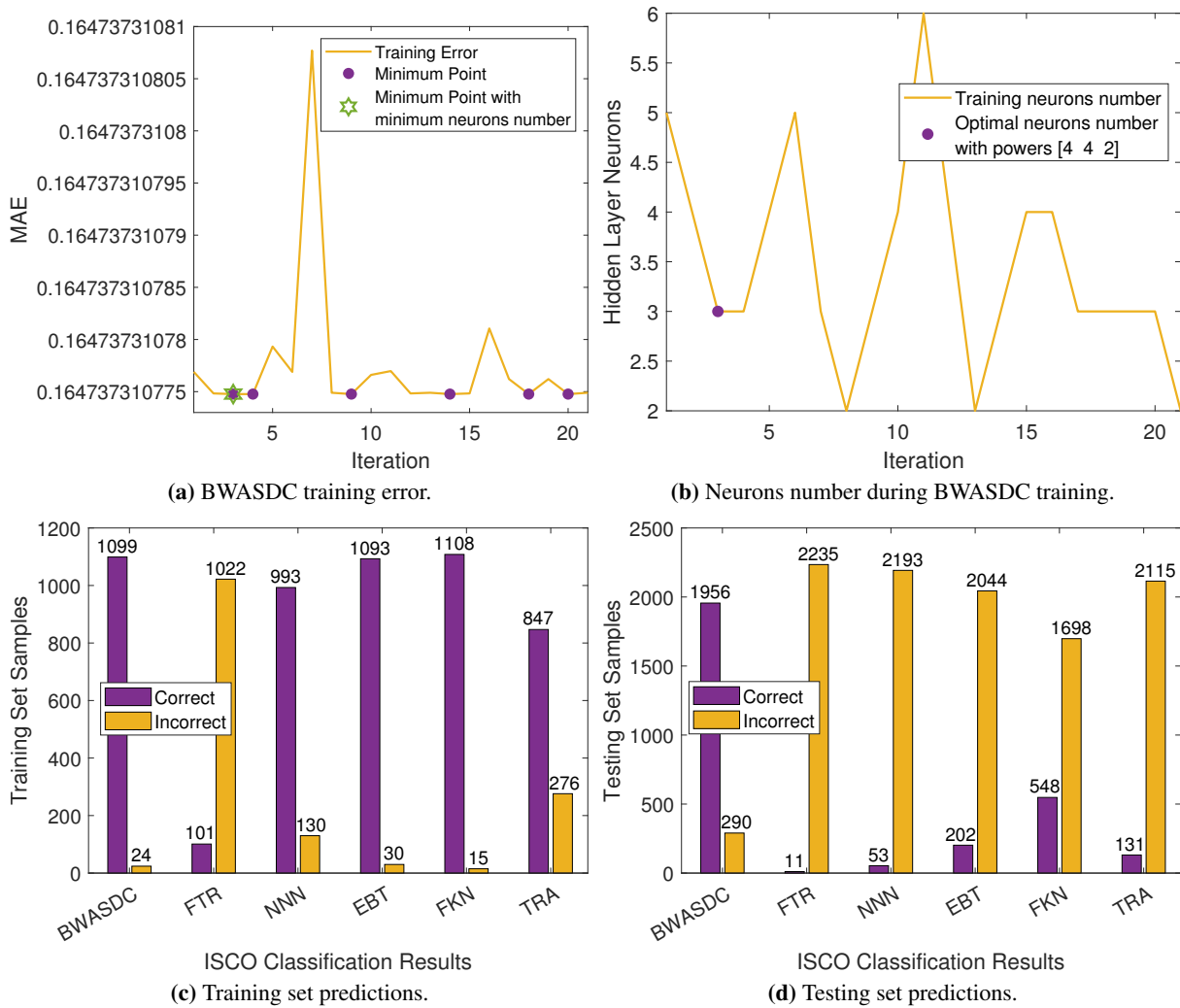
In Table 1, a number of performance metrics are applied to the outputs of the models on the training and testing sets in order to statistically evaluate their performances. The accuracy, MAE, specificity, sensitivity, precision, Matthews correlation coefficient (MCC), F-score, Cohen's  $\kappa$  and false positive rate (FPR) are the performance measures considered in our analysis. See [55, 56] for more information and a thorough study of these measures. On the training set, the MAE and accuracy of the models are measured. We observe that FKN has the best MAE and accuracy, the EBT has the 2nd best, BWASDC has the 3rd best, and FTR has the worst. On the testing set, we note that BWASDC outperforms the other models in the majority of the measures. Particularly, BWASDC has the best MAE, accuracy, specificity, FPR, F-score, Cohen's  $k$  and precision. FKN has the 2nd best accuracy, specificity, FPR, Cohen's  $k$  and precision, and EBT has the 2nd best MAE and F-score, while FTR has the best sensitivity and MCC. Overall, the BWASDC performs better than the other models, while FTR and TRA put on the worst and second-worst performances, respectively.

**Table 1.** Neural network models' statistics in the experiment of section 3.2.

Set	Statistic	Neural Network Models					
		BWASDC	FTR	NNN	EBT	FKN	TRA
Train	MAE	0.7060	345.31	4.72	0.7024	0.6668	36.81
	Accuracy	0.9126	0.0898	0.8796	0.9251	0.9277	0.6821
Test	MAE	11.31	420.37	204.92	125.27	165.33	400.29
	Accuracy	0.7886	0.0049	0.0277	0.1127	0.2565	0.0338
	Specificity	0.9998	0.9991	0.9991	0.9992	0.9993	0.9991
	Sensitivity	0.9468	0.9843	0.7321	0.6577	0.8468	0.9478
	Precision	0.7884	0.0047	0.0275	0.1125	0.2563	0.0338
	FPR	$1.8 \times 10^{-4}$	$8.8 \times 10^{-4}$	$8.6 \times 10^{-4}$	$7.9 \times 10^{-4}$	$6.6 \times 10^{-4}$	$8.6 \times 10^{-4}$
	MCC	0.9418	0.9852	0.7350	0.6671	0.8438	0.9501
	F-score	0.7775	0.0098	0.2481	0.3259	0.2664	0.0361
	Cohen's $k$	0.7884	0.0041	0.0266	0.1118	0.2557	0.0330

### 3.3. Experiment 2

In this experiment, the job titles are taken from ONETD and used with the ONET-ISCO dataset. By using the O\*NET-SOC 2010 code and the job titles as input, the crosswalk from O\*NET-SOC 2010 to ISCO-08 is predicted through the NN models. The BWASDC training error is shown in Figure 6a, while the hidden layer neurons' number of its iteration of the algorithm is shown in Figure 6b. These figures show that the BWASDC algorithm needed 3 iterations to reach the smallest error with the minimum neurons' number. The minimum number of neurons is 3, and their optimal powers are  $[4, 4, 2]^T$ . Furthermore, Figure 6c and 6d, respectively, show the predictions made based on the training and testing sets. Comparing Figure 6c and 6d, we see that the BWASDC, FKN, and EBT perform similarly on the training set, while BWASDC performs significantly better on the testing set. Particularly, Figure 6c shows that FKN has the greatest results on the training set, BWASDC has the 2nd greatest, and FTR has the worst. On the other hand, Figure 6d shows that BWASDC has the greatest results on the testing set, FKN has the 2nd greatest, and FTR has the worst. In other words, BWASDC outperforms the other models in its ability to classify unknown data.



**Figure 6.** Crosswalk from O\*NET-SOC 2010 to ISCO-08 by NNs in the experiment of section 3.3.

In Table 2, a number of performance metrics are applied to the outputs of the models on the training and testing sets in order to statistically evaluate their performances. On the training set, the MAE and accuracy of the models are measured. We observe that FKN has the best accuracy and the 2nd best MAE, while BWASDC has the best MAE and the 2nd best accuracy. Additionally, EBT has the 3rd best MAE and accuracy, while FTR has the worst. On the testing set, we note that BWASDC outperforms the other models in the majority of the measures. Particularly, BWASDC has the best MAE, accuracy, specificity, sensitivity, FPR, F-score, Cohen’s  $k$  and precision. FKN has the 2nd best accuracy, specificity, FPR, Cohen’s  $k$  and precision, and EBT has the 2nd best MAE and F-score, while FTR has the best MCC and the 2nd best sensitivity. Overall, the BWASDC performs better than the other models, while FTR and NNN put on the worst and second-worst performances, respectively.

**Table 2.** Neural network models' statistics in the experiment of section 3.3.

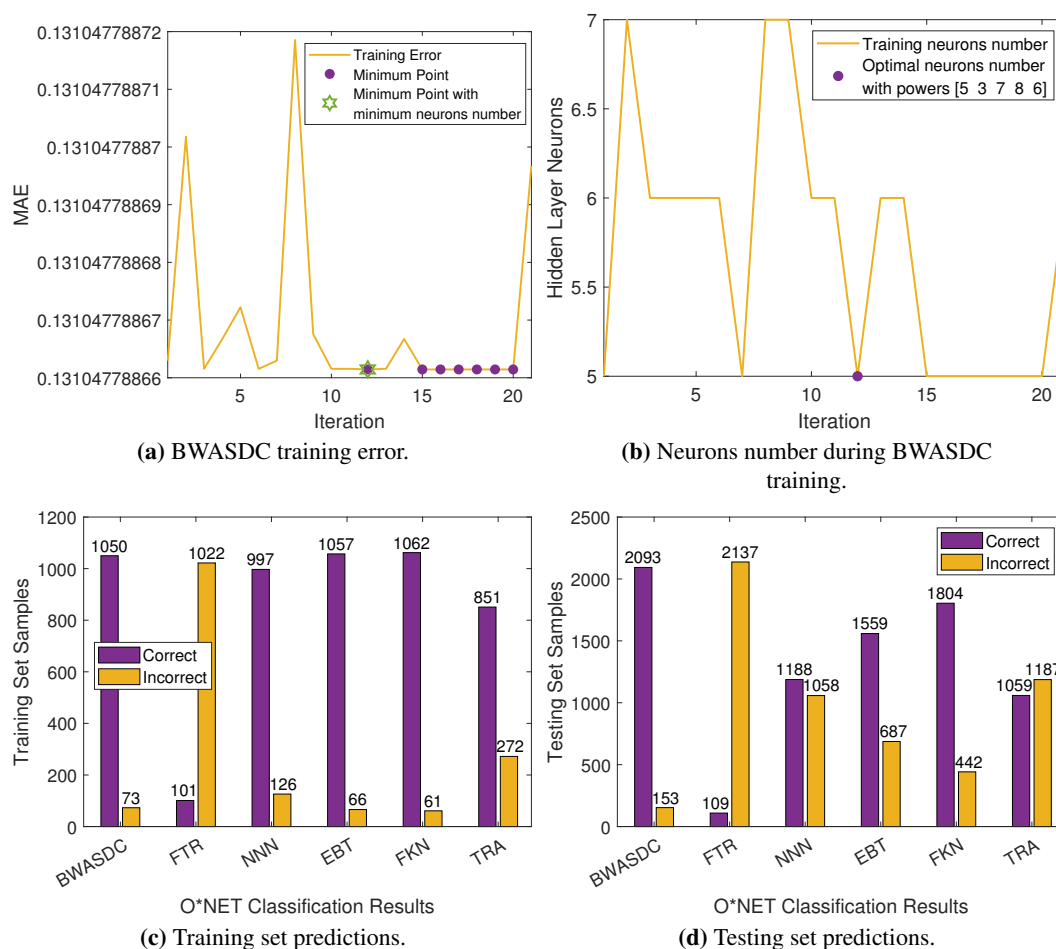
Set	Statistic	Neural Network Models					
		BWASDC	FTR	NNN	EBT	FKN	TRA
Train	MAE	0.1639	345.77	20.40	0.1966	0.1708	18.18
	Accuracy	0.9787	0.0898	0.7783	0.9742	0.9867	0.7542
Test	MAE	6.81	420.86	222.85	151.57	182.54	374.01
	Accuracy	0.8708	0.0047	0.0254	0.0960	0.2438	0.0583
	Specificity	0.9999	0.9991	0.9991	0.9991	0.9993	0.9992
	Sensitivity	0.9854	0.9846	0.7642	0.6517	0.8587	0.9239
	Precision	0.8709	0.0049	0.0254	0.0962	0.2438	0.0583
	FPR	$1.1 \times 10^{-4}$	$8.8 \times 10^{-4}$	$8.6 \times 10^{-4}$	$8.1 \times 10^{-4}$	$6.7 \times 10^{-4}$	$8.3 \times 10^{-4}$
	MCC	0.9808	0.9854	0.7666	0.6576	0.8523	0.9220
	F-score	0.8694	0.0099	0.2196	0.3377	0.2566	0.0744
	Cohen's $k$	0.8709	0.0041	0.0246	0.0955	0.2434	0.0575

### 3.4. Experiment 3

In this experiment, the job titles are taken from ISCOD and used with the ISCO-ONET dataset. By using the ISCO-08 code and the job titles as input, the crosswalk from ISCO-08 to O\*NET-SOC 2010 is predicted through the NN models. The BWASDC training error is shown in Figure 7a, while the hidden layer neurons' number of its iteration of the algorithm is shown in Figure 7b. These figures show that the BWASDC algorithm needed 12 iterations to reach the smallest error with the minimum neurons' number. The minimum number of neurons is 5, and their optimal powers are  $[5, 3, 7, 8, 6]^T$ . Furthermore, Figure 7c and 7d, respectively, show the predictions made based on the training and testing sets. Comparing Figure 7c and 7d, we see that the BWASDC, FKN, and EBT perform similarly on the training set, while BWASDC performs significantly better on the testing set. Particularly, Figure 7c shows that FKN has the greatest results on the training set, EBT has the 2nd greatest, BWASDC has the 3rd greatest, and FTR has the worst. On the other hand, Figure 7d shows that BWASDC has the greatest results on the testing set, FKN has the 2nd greatest, and FTR has the worst. In other words, BWASDC outperforms the other models in its ability to classify unknown data.

In Table 3, a number of performance metrics are applied to the outputs of the models on the training and testing sets in order to statistically evaluate their performances. On the training set, the MAE and accuracy of the models are measured. We observe that FKN has the best accuracy and the 3rd best MAE, while BWASDC has the best MAE and the 3rd best accuracy. Additionally, EBT has 2nd best MAE and accuracy, while FTR has the worst. On the testing set, we note that BWASDC outperforms the other models across all measures. Further, FKN has the 2nd best MAE, accuracy, specificity, FPR, F-score, Cohen's  $k$  and precision, while FTR has the 2nd best sensitivity and MCC. Overall, the BWASDC performs better than the other models, while FTR and TRA put on the worst and second-worst performances, respectively.





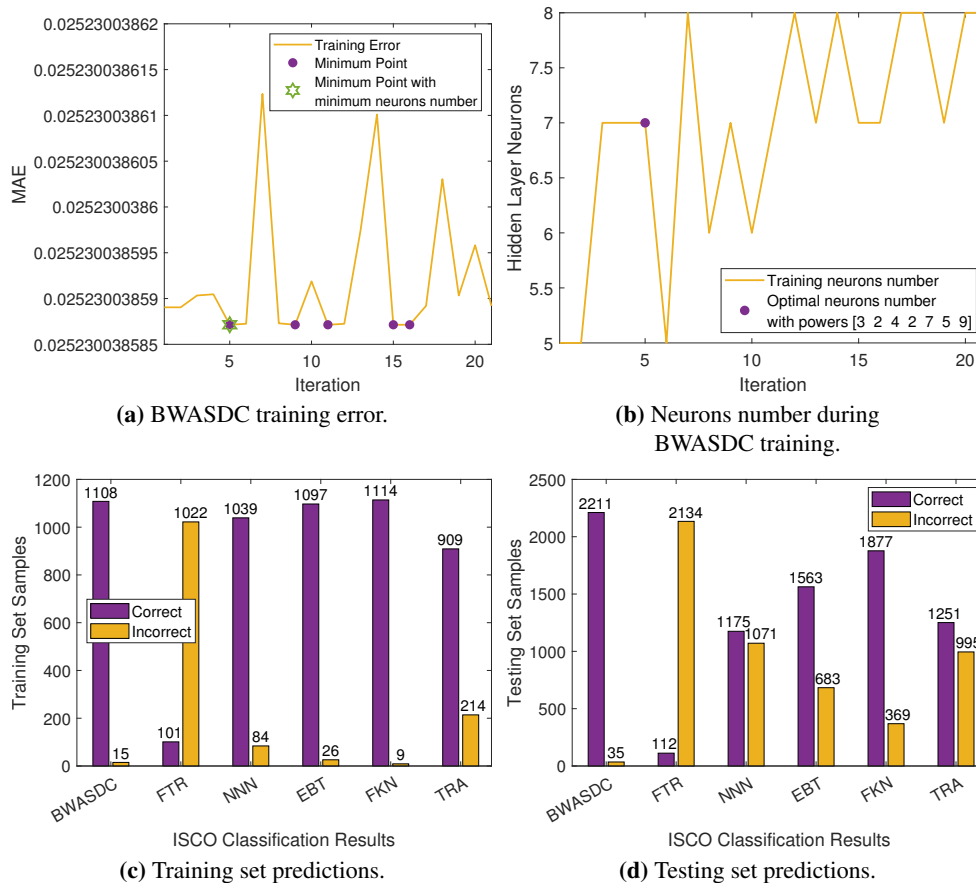
**Figure 7.** Crosswalk from ISCO-08 to O\*NET-SOC 2010 by NNs in the experiment of section 3.4.

**Table 3.** Neural network models' statistics in the experiment of section 3.4.

Set	Statistic	Neural Network Models					
		BWASDC	FTR	NNN	EBT	FKN	TRA
Train	MAE	0.1301	335.51	12.78	0.1452	0.1773	42.09
	Accuracy	0.9351	0.0898	0.8877	0.9422	0.9457	0.7578
Test	MAE	0.5423	370.19	95.87	36.64	25.56	109.83
	Accuracy	0.9319	0.0486	0.5303	0.7017	0.8033	0.4715
	Specificity	0.9999	0.9991	0.9995	0.9997	0.9998	0.9995
	Sensitivity	0.9793	0.9583	0.7785	0.8977	0.9362	0.8287
	Precision	0.9318	0.0486	0.5303	0.7017	0.8033	0.4715
	FPR	$6.0 \times 10^{-5}$	$8.4 \times 10^{-4}$	$4.1 \times 10^{-4}$	$2.6 \times 10^{-4}$	$1.7 \times 10^{-4}$	$4.7 \times 10^{-4}$
	MCC	0.9841	0.9606	0.8021	0.9011	0.9338	0.8563
	F-score	0.9223	0.0438	0.5425	0.6697	0.7764	0.4048
Cohen's <i>k</i>	0.9319	0.0478	0.5299	0.7015	0.8031	0.4710	

### 3.5. Experiment 4

In this experiment, the job titles are taken from ISCOD and used with the ONET-ISCO dataset. By using the O\*NET-SOC 2010 code and the job titles as input, the crosswalk from O\*NET-SOC 2010 to ISCO-08 is predicted through the NN models. The BWASDC training error is shown in Figure 8a, while the hidden layer neurons' number of its iteration of the algorithm is shown in Figure 8b. These figures show that the BWASDC algorithm needed 5 iterations to reach the smallest error with the minimum neurons' number. The minimum number of neurons is 7, and their optimal powers are  $[3, 2, 4, 2, 7, 5, 9]^T$ . Furthermore, Figure 8c and 8d, respectively, show the predictions made based on the training and testing sets. Comparing Figure 8c and 8d, we see that the BWASDC, FKN, and EBT perform similarly on the training set, while BWASDC performs significantly better on the testing set. Particularly, Figure 8c shows that FKN has the greatest results on the training set, BWASDC has the 2nd greatest, EBT has the 3rd greatest, and FTR has the worst. On the other hand, Figure 8d shows that BWASDC has the greatest results on the testing set, FKN has the 2nd greatest, and FTR has the worst. In other words, BWASDC outperforms the other models in its ability to classify unknown data.



**Figure 8.** Crosswalk from O\*NET-SOC 2010 to ISCO-08 by NNs in the experiment of section 3.5.

In Table 4, a number of performance metrics are applied to the outputs of the models on the training and testing sets in order to statistically evaluate their performances. On the training set, the MAE

and accuracy of the models are measured. We observe that FKN has the best accuracy and the 2nd best MAE, while BWASDC has the best MAE and the 2nd best accuracy. Additionally, EBT has 3rd best MAE and accuracy, while FTR has the worst. On the testing set, we note that BWASDC outperforms the other models across all measures. Further, FKN has the 2nd best MAE, accuracy, specificity, FPR, F-score, Cohen's  $k$  and precision, while FTR has the 2nd best sensitivity and MCC. Overall, the BWASDC performs better than the other models, while FTR and NNN put on the worst and second-worst performances, respectively.

**Table 4.** Neural network models' statistics in the experiment of section 3.5.

Set	Statistic	Neural Network Models					
		BWASDC	FTR	NNN	EBT	FKN	TRA
Train	MAE	0.0248	407.85	10.12	0.1461	0.0268	19.79
	Accuracy	0.9867	0.0899	0.9206	0.9822	0.9918	0.8094
Test	MAE	0.1055	443.37	109.80	44.31	23.55	68.54
	Accuracy	0.9845	0.0498	0.5317	0.6817	0.8358	0.5570
	Specificity	0.9999	0.9991	0.9995	0.9997	0.9998	0.9996
	Sensitivity	0.9928	0.9575	0.7801	0.8914	0.9528	0.7849
	Precision	0.9845	0.0498	0.5317	0.6817	0.8358	0.5570
	FPR	$1.3 \times 10^{-5}$	$8.4 \times 10^{-4}$	$4.1 \times 10^{-4}$	$2.8 \times 10^{-4}$	$1.4 \times 10^{-4}$	$3.9 \times 10^{-4}$
	MCC	0.9928	0.9595	0.8012	0.8951	0.9482	0.8171
	F-score	0.9863	0.0510	0.5565	0.6498	0.8165	0.5184
	Cohen's $k$	0.9845	0.0491	0.5313	0.6815	0.8356	0.5566

## 4. Conclusions

This paper presented a new 3-layer bio-inspired WASD FNN for multiclass classification tasks, termed BWASDC. The efficacy of the BWASDC was then assessed by applications in occupational classification systems, specifically through four crosswalk experiments between the O\*NET-SOC 2010 and ISCO-08 codes. The findings of these experiments show that the BWASDC model outperforms some of the most modern classification models obtainable through MATLAB's classification learner app. The BWASDC model has thus proven to be a great substitute for dealing with multiclass classification-related issues. It is significant to note that only actual numerical data may be utilized as input to train and test the BWASDC NN model owing to restrictions imposed by the WDD process, which is utilized by the BWASDC algorithm. The BWASDC model's only testing involved applications in occupational classification systems, which is another drawback. Future study will consequently focus on its suitable adaptation and application to various multiclass classification problems across numerous scientific fields.

### Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

## Acknowledgments

This work was supported by the Key Science and Technology Research of Henan Province, China (Grant No. 222102210279, 232102210129, 232102210076, 232102210074, 232102211038 and 222102210232), and by the Postgraduate Joint Training Base Project of Henan Province, China (Grant No. YJS2022JD45).

## Conflict of interest

The authors declare no conflict of interest. Vasilios N. Katsikis is a Guest Editor for AIMS Mathematics and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

## References

1. E. Felten, M. Raj, R. Seamans, Occupational, industry, and geographic exposure to artificial intelligence: A novel dataset and its potential uses, *Strategic. Manage. J.*, **42** (2021), 2195–2217. <https://doi.org/10.1002/smj.3286>
2. E. W. Felten, M. Raj, R. Seamans, The occupational impact of artificial intelligence: Labor, skills, and polarization, *NYU Stern School of Business*, 2019. <https://doi.org/10.2139/ssrn.3368605>
3. B. Bigdeli, P. Pahlavani, H. A. Amirkolae, An ensemble deep learning method as data fusion system for remote sensing multisensor classification, *Appl. Soft Comput.*, **110** (2021), 107563. <https://doi.org/10.1016/j.asoc.2021.107563>
4. R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, K. Shankar, Optimal feature selection-based medical image classification using deep learning model in internet of medical things, *IEEE Access*, **8** (2020), 58006–58017. <https://doi.org/10.1109/ACCESS.2020.2981337>
5. T. E. Simos, V. N. Katsikis, S. D. Mourtas, A multi-input with multi-function activated weights and structure determination neuronet for classification problems and applications in firm fraud and loan approval, *Appl. Soft Comput.*, **127** (2022), 109351. <https://doi.org/10.1016/j.asoc.2022.109351>
6. G. Varelas, D. Lagios, S. Ntouroukis, P. Zervas, K. Parsons, G. Tzimas, Employing natural language processing techniques for online job vacancies classification, in *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops. AIAI 2022* (eds. I. Maglogiannis, L. Iliadis, J. Macintyre and P. Cortez), vol. 652 of IFIP Advances in Information and Communication Technology, Springer, Cham, 2022, 333–344.
7. A. Rácz, D. Bajusz, K. Héberger, Effect of dataset size and train/test split ratios in QSAR/QSPR multiclass classification, *Molecules*, **26** (2021), 1111. <https://doi.org/10.3390/molecules26041111>
8. R. Venkatesan, M. J. Er, A novel progressive learning technique for multi-class classification, *Neurocomputing*, **207** (2016), 310–321. <https://doi.org/10.1016/j.neucom.2016.05.006>
9. T. E. Simos, V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, Unique non-negative definite solution of the time-varying algebraic Riccati equations with applications to stabilization of LTV systems, *Math. Comput. Simul.*, **202** (2022), 164–180.

10. S. D. Mourtas, V. N. Katsikis, C. Kasimis, Feedback control systems stabilization using a bio-inspired neural network, *EAI Endorsed Trans. AI Robot*, **1** (2022), 1–13.
11. N. Premalatha, A. V. Arasu, Prediction of solar radiation for solar systems by using ANN models with different back propagation algorithms, *J. Appl. Res. Technol.*, **14** (2016), 206–214. <https://doi.org/10.1016/j.jart.2016.05.001>
12. C. Huang, X. Jia, Z. Zhang, A modified back propagation artificial neural network model based on genetic algorithm to predict the flow behavior of 5754 aluminum alloy, *Materials*, **11** (2018), 855.
13. L. Chen, Z. Huang, Y. Li, N. Zeng, M. Liu, A. Peng, et al., Weight and structure determination neural network aided with double pseudoinversion for diagnosis of flat foot, *IEEE Access*, **7** (2019), 33001–33008. <https://doi.org/10.1109/ACCESS.2019.2903634>
14. T. E. Simos, V. N. Katsikis, S. D. Mourtas, A fuzzy WASD neuronet with application in breast cancer prediction, *Neural Comput. Appl.*, **34** (2021), 3019–3031. <https://doi.org/10.1007/s00521-021-06572-9>
15. M. R. Daliri, A hybrid automatic system for the diagnosis of lung cancer based on genetic algorithm and fuzzy extreme learning machines, *J. Medical Syst.*, **36** (2012), 1001–1005. <https://doi.org/10.1007/s10916-011-9806-y>
16. S. Gayathri, A. K. Krishna, V. P. Gopi, P. Palanisamy, Automated binary and multiclass classification of diabetic retinopathy using Haralick and multiresolution features, *IEEE Access*, **8** (2020), 57497–57504. <https://doi.org/10.1109/ACCESS.2020.2979753>
17. S. D. Mourtas, V. N. Katsikis, Exploiting the Black-Litterman framework through error-correction neural networks, *Neurocomputing*, **498** (2022), 43–58. <https://doi.org/10.1016/j.neucom.2022.05.036>
18. S. D. Mourtas, V. N. Katsikis, E. Drakonakis, S. Kotsios, Stabilization of stochastic exchange rate dynamics under central bank intervention using neuronets, *Int. J. Inf. Technol. Decis.*, **22** (2023), 855–883. <https://doi.org/10.1142/s0219622022500560>
19. S. D. Mourtas, A weights direct determination neuronet for time-series with applications in the industrial indices of the federal reserve bank of St. Louis, *J. Forecast.*, **14** (2022), 1512–1524. <https://doi.org/10.1002/for.2874>
20. T. E. Simos, V. N. Katsikis, S. D. Mourtas, Multi-input bio-inspired weights and structure determination neuronet with applications in European Central Bank publications, *Math. Comput. Simul.*, **193** (2022), 451–465. <https://doi.org/10.1016/j.matcom.2021.11.007>
21. R. Boselli, M. Cesarini, S. Marrara, F. Mercorio, M. Mezzanzanica, G. Pasi, et al., WoLMIS: A labor market intelligence system for classifying web job vacancies, *J. Intell. Inf. Syst.*, **51** (2018), 477–502. <https://doi.org/10.1007/s10844-017-0488-x>
22. P. G. Lovaglio, M. Cesarini, F. Mercorio, M. Mezzanzanica, Skills in demand for ICT and statistical occupations: Evidence from web-based job vacancies, *Stat. Anal. Data Min.*, **11** (2018), 78–91. <https://doi.org/10.1002/sam.11372>
23. E. Heinesen, S. Imai, S. Maruyama, Employment, job skills and occupational mobility of cancer survivors, *J. Health Econ.*, **58** (2018), 151–175. <https://doi.org/10.1016/j.jhealeco.2018.01.006>

24. F. Groes, P. Kircher, I. Manovskii, The U-shapes of occupational mobility, *Rev. Econ. Stud.*, **82** (2015), 659–692. <https://doi.org/10.1093/restud/rdu037>
25. M. Khalis, B. Charbotel, E. Fort, V. Chajes, H. Charaka, K. E. Rhazi, Occupation and female breast cancer: A case-control study in Morocco, *Rev. Epidemiol. Sante Publique*, **66** (2018), S302. <https://doi.org/10.1016/j.respe.2018.05.172>
26. I. N. Generalao, Measuring the telework potential of jobs: Evidence from the international standard classification of occupations, *Philipp. Rev. Econ.*, **58** (2021), 92–127. <https://doi.org/10.37907/5erp1202jd>
27. C. Züll, The coding of occupations, *GESIS Survey Guidelines*, Mannheim, Germany: GESIS – Leibniz Institute for the Social Sciences.
28. S. B. Choi, J. H. Yoon, W. Lee, The modified international standard classification of occupations defined by the clustering of occupational characteristics in the Korean working conditions survey, *Ind. Health*, **58** (2020), 132–141. <https://doi.org/10.2486/indhealth.2018-0169>
29. D. T. Marc, P. Dua, S. H. Fenton, K. Lalani, K. Butler-Henderson, *The Health Information Workforce*, chapter Occupational Classifications in the Health Information Disciplines, 71–78, Health Informatics. Springer, Cham., 2021.
30. J. Rounds, P. I. Armstrong, H. Y. Liao, D. Rivkin, P. Lewis, Second generation occupational value profiles for the O\* NET system: Summary, *Raleigh, NC: National Center for O\* NET Development*, 2008.
31. M. P. Wilmot, D. S. Ones, Occupational characteristics moderate personality–performance relations in major occupational groups, *J. Vocat. Behav.*, **131** (2021), 103655. <https://doi.org/10.1016/j.jvb.2021.103655>
32. M. Zhang, Estimation of differential occupational risk of COVID-19 by comparing risk factors with case data by occupational group, *Am. J. Ind. Med.*, **64** (2021), 39–47.
33. W. Uter, *Kanerva's Occupational Dermatology*, chapter Classification of occupations, Springer, Berlin, Heidelberg, 2012.
34. E. Faia, S. Laffitte, M. Mayer, G. Ottaviano, On the employment consequences of automation and offshoring: A labor market sorting view, in *Robots and AI*, Routledge, 2021, 82–122.
35. A. S. Ioshisaqui, R. Attux, I. Luna, Analysis of occupational profiles in the Brazilian workforce based on non-negative matrix factorization, *Big Data Res.*, **29** (2022), 100333. <https://doi.org/10.1016/j.bdr.2022.100333>
36. P. Egana-delSol, G. Cruz, A. Micco, COVID-19 and automation in a developing economy: Evidence from Chile, *Technol. Forecast. Soc. Change*, **176** (2022), 121373. <https://doi.org/10.1016/j.techfore.2021.121373>
37. R. Sebastian, Explaining job polarisation in Spain from a task perspective, *SERIEs*, **9** (2018), 215–248. <https://doi.org/10.1007/s13209-018-0177-1>
38. Y. Zhang, D. Chen, C. Ye, *Deep Neural Networks: WASD Neuronet Models, Algorithms, and Applications*, CRC Press: Boca Raton, FL, USA, 2019.
39. X. Jiang, S. Li, BAS: Beetle antennae search algorithm for optimization problems, *arXiv preprint*, abs/1710.10724, 2017. Available from: <http://arxiv.org/abs/1710.10724>

40. T. E. Simos, S. D. Mourtas, V. N. Katsikis, Time-varying Black-Litterman portfolio optimization using a bio-inspired approach and neuronets, *Appl. Soft Comput.*, **112** (2021), 107767. <https://doi.org/10.1016/j.asoc.2021.107767>
41. Y. Cheng, C. Li, S. Li, Z. Li, Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm, *IEEE Access*, **8** (2020), 138788–138799. <https://doi.org/10.1109/ACCESS.2020.3012564>
42. X. Li, H. Jiang, M. Niu, R. Wang, An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm, *Mech. Syst. Signal Process.*, **142** (2020), 106752.
43. X. Li, Z. Zang, F. Shen, Y. Sun, Task offloading scheme based on improved contract net protocol and beetle antennae search algorithm in fog computing networks, *Mobile Netw. Appl.* **25** (2020), 2517–2526. <https://doi.org/10.1109/ACCESS.2020.3012564>
44. Y. Fan, J. Shao, G. Sun, Optimized PID controller based on beetle antennae search algorithm for electro-hydraulic position servo control system, *Sensors*, **19** (2019), 2727. <https://doi.org/10.3390/s19122727>
45. S. D. Mourtas, V. N. Katsikis, V-shaped BAS: Applications on large portfolios selection problem, *Comput. Econ.*, **60** (2022), 1353–1373. <https://doi.org/10.1007/s10614-021-10184-9>
46. V. N. Katsikis, S. D. Mourtas, Diversification of time-varying tangency portfolio under nonlinear constraints through semi-integer beetle antennae search algorithm, *Appl. Math.*, **1** (2021), 63–73. <https://doi.org/10.3390/appliedmath1010005>
47. P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, Apress: Berkeley, CA, USA, 2017.
48. Transformer models for MATLAB, 2023. Available from: <https://github.com/matlab-deep-learning/transformer-models>
49. Y. Zhang, X. Yu, L. Xiao, W. Li, Z. Fan, W. Zhang, Weights and structure determination of artificial neuronets, in *Self-Organization: Theories and Methods*, New York, NY, USA: Nova Science, 2013.
50. Z. Zhu, Z. Zhang, W. Man, X. Tong, J. Qiu, F. Li, A new beetle antennae search algorithm for multi-objective energy management in microgrid, in *Proc. 13th IEEE Conf. Industrial Electronics and Applications (ICIEA)*, 2018, 1599–1603.
51. Q. Wu, X. Shen, Y. Jin, Z. Chen, S. Li, A. H. Khan, et al., Intelligent beetle antennae search for UAV sensing and avoidance of obstacles, *Sensors*, **19** (2019), 1758. <https://doi.org/10.3390/s19081758>
52. X. Xu, K. Deng, B. Shen, A beetle antennae search algorithm based on Lévy flights and adaptive strategy, *Syst. Sci. Control. Eng.*, **8** (2020), 35–47. <https://doi.org/10.1080/21642583.2019.1708829>
53. M. Davis, L. Iancu, Unicode text segmentation, *Unicode Standard Annex*, **29** (2018), 65. <https://doi.org/10.4324/9780429955600-9>
54. A. K. Gupta, *Numerical methods using MATLAB*, MATLAB solutions series, Apress: Berkeley, CA, USA, New York, NY, 2014.

- 
55. A. Tharwat, Classification assessment methods, *Appl. Comput. Inform.*, **17** (2020), 168–192.  
<https://doi.org/10.1016/j.aci.2018.08.003>
56. M. L. McHugh, Interrater reliability: the kappa statistic, *Biochem. Med.*, **22** (2012), 276–282.  
<https://doi.org/10.11613/bm.2012.031>



© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).