Journal of Mathematics and Informatics
Vol. 25, 2023, 29-46
ISSN: 2349-0632 (P), 2349-0640 (online)
Published 4 August 2023
www.researchmathsci.org
DOI: http://dx.doi.org/10.22457/jmi.v25a04229

Journal of **Mathematics and Informatics**

Brute Exhaustive Optimization of Intelligent Small Weighted Voting Ensembles in 1EXP(-)Z$^+$ Initial-Term based Arithmetic Sequence's Multi Precision Search Spaces

Augustine Josephat Malamsha[1], ***Mussa Ally Dida***[2] **and** ***Sabine Moebbs***[3]

[1]Department of Information Communication Science and Engineering
Nelson Mandela African Institution of Science and Technology
Arusha, Tanzania. E-mail: malamshaa@nm-aist.ac.tz
[2]Department of Electronics and Telecommunication Engineering
Nelson Mandela African Institution of Science and Technology
Arusha, Tanzania. E-mail: mussa.ally@nm-aist.ac.tz
[3]Department of Business Information Systems,
Cooperative State University Baden Wuerttemberg of Science and Technology
Stuttgart, Germany. E-mail: sabine.moebs@gmail.com

Received 31 May 2023; accepted 10 July 2023

Abstract. Other than the individual machine learning models' capabilities, the weighted voting ensemble (WVE) technique relies on appropriate weight assignment in order to significantly realize prediction performance improvement. Often evolutionary global or grid local search heuristics are being applied for such a challenging optimization task. However, these techniques do not guarantee optimal solution finding. In turn, the surprising outstanding successes of brute exhaustive search procedure in producing similar results shed light on its significance and the need to exploit its possible weights solutions search space(s) with corresponding sizes as a key determinant factor for implementing a successful brute search procedure for finding optimal WVE solution with a trade-off the computational efficiency. This paper formulates an asymptotically WVE weights domain constraints optimal 1EX(-)Z$^+$ initial term-based arithmetic sequences initialization function, and then a computational multi-precision search space-based generation algorithm is developed to find optimal WVE solution as part of the brute exhaustive search procedure. It took 45 minutes for a proposed algorithm to generate 133,192 combinations and find the optimal solution in weights space of precision 0.01.

Keywords: Arithmetic sequences, brute exhaustive search, search spaces, artificial intelligence, machine learning, weighted voting ensemble, weights precision.

AMS Mathematics Subject Classification (2010): 68T20

1. Introduction

Machine learning (ML) ensemble model construction has been implemented in several real-world applications due to their prospective superiority in performance as compared to individual ML models, for that reason, schemes thereof have widely been applied in several studies related to high-performance ML model implementations using ensemble learning strategies. In particular, weighted voting ensemble (WVE), among other ensemble schemes is an ensemble combination strategy that treats all models as unequal and weighs their class probabilities prediction, an act which have theoretical and empirical lead to extensive appreciation due to observed significant model performance improvements by WVE, on the contrary to its former variant, i.e. simple voting which assumes all models to be equal [1].

On the one hand, while the key success of WVE's is highly dependent on the challenging task of assigning appropriate weights to its base ML models predictions [1,2,3,4,5], various state-of-the-art evolutionary algorithms (EA), greedy search (GS) and brute force (BF) based boundless search heuristic methodological procedures are being developed and applied in various WVE optimization experiments to find deemed optimal weights.

However, although both the stochastic population-based evolutionary and greedy-based search heuristic procedures are often more efficient than brute exhaustive search, they may sometimes not guarantee to achieve of global optimum [6,7], whereas greedy and its variant implementation, such as the greedy randomized adaptive search which have been used by (8) may face a hill climbing problem, the evolutionary extremums may be caused by its population-based stochastic search heuristic implementation which may probabilistically select at that one time from a very unfit initialized genes chromosomes of the creature being optimized [9], among other things. As such, as observed in [10], the surprising outstanding successes of the systematic brute force-based exhaustive search counterpart in producing optimal WVE models configuration sets with predictive performances similar to those created by evolutionary-based optimization procedures in conjunction with its theoretical guarantee for finding an optimal solution through a search across systematic search spaces [11], it may become imperative to implement the brute exhaustive search procedures, as given the required high computational effort is available, it guarantees exhaustion of all candidate solutions combinations [11,12], for optimality search problems, such as this of finding the appropriate weights for the most accurate WVE, at a reasonable efficiency tradeoff when the deemed global optima solution estimations has been defined as a key requirement, that is, must occur.

Therefore, given the significance of search spaces in optimization procedure, this paper presents a brute exhaustive search heuristic implementation for optimizing weighted voting ensembles in multi precisions local search spaces formulated from a 1EXP(-)Z^+ Initial-term based arithmetic sequence and to return the deemed optimal weights configuration, particularly emphasizing on a scrutiny across multi precision search spaces for a systematic brute exhaustive based WVE optimization procedure implementation, as an advancement as a contribution to the existing ML WVE models optimization scientific knowledge body. Specifically, a mathematically valid function for computing as search spaces represented by multi precision weights that are asymptotic optimal to WVE weights domain constraints was derived. Then, an algorithmic implementation of the function was developed to computationally generate multi precision weights as search spaces for optimizing ensemble base members using brute search technique. Finally, the performance of the proposed function algorithm in formulating multi-precision search spaces for effectively finding the optimal WVE combination as a configuration set of the individual base expert's weights coefficient values was evaluated, with an objective to maximize accuracy.

The rest of this paper is organized as follows, chapter 2 is preliminary on ensemble learning, WVE, and brute exhaustive search techniques, chapter 3 presents the materials and methods used to develop the proposed weight generation algorithm solution for the brute searching procedure, chapter 4 presents and discusses the results of the experiment carried out to evaluate the proposed algorithm solution, and chapter 5 concludes the paper and provides recommendations of future work.

2. Ensemble learning

Ensemble learning in ML is a method used to combine results of various ML homogeneous or heterogeneous hypotheses or base experts' predictions that answer the same question in order to have more predictive accuracy [13,14,15], have theoretically and empirically proved to have potential for improving the predictive performance of individual learners by combining their predictions, through an ensemble function of all base members, and the ensemble error becomes a decomposition of average individual members errors essentially to compensate for the lower average accuracy of individual members by the higher disagreement weight the ensemble as long as it is correct [16,17,18].

2.1. Weighted voting ensemble (WVE)

While several alternative implementations of ML ensembles could exist, model predictions or voting are the two common ways to combine single base model predictions, whereby averaging mainly reduces variance. Voting selects the class mostly predicted by induvial models. Most importantly, as presented in [19], the WVE whose output y(x) can be expressed through equation (1), is an improved variant of simple voting which was introduced with an understanding that different individual models to form an ensemble cannot in most practical cases have same influence within that formation, in turn specifying a weight coefficient often between 0 and 1 for each member which can be same or different depending on optimality of the ensemble thereof and whose total weight summation should be equal to one as in equation (2) can provide better predictive performance, unlike in simple voting which barely assume models are equal [20,21,22,23].

$$y(x)=\text{argmax} \sum_{i=1}^{k} w_i XA(C_j(x)=j) \tag{1}$$

"where y of all the unknown instances χ in the test sets are evaluated as the argmax function of the respective index with the largest value from array A = {1, 2, . . . , M} denotes the set of exclusive class labels and XA indicates the characteristics function that considered the predictions $j \in$ A of a classifiers Cj on instances and create vectors where the j coordinates take values of one and the remaining takes the value of zero [19]. And, w_i which is the weight of model C_j is constrained by equation (2),

$$\sum_{i=1}^{k} w_i=1, \quad w_i>0, \forall_i=1,\ldots.k, \tag{2}$$

Whereby k is the variable representing the index with the product of the WVE's base classifier's probability prediction and its corresponding weight w_i.

2.2. Brute exhaustive search algorithm

Nearly all science and engineering fields use search algorithms, which automatically explore a search space to find high-performing solutions [24]. Brute or exhaustive search algorithm is a set of instruction used to find optimal solution by examining all possible solution combinations. This search process is not that new at all, it has been applied in several optimization problems to search for the most deemed optimal solution [12,23,25].

With respect to WVE's optimization, the brute-force or exhaustive search algorithm have also been used in various studies, like in [22] were brute search was implemented to perform best ensemble model selection to integrating capabilities of CNN architectures and ensemble learning for decoding EEG signals collected in motor imagery experiments. Also, in [26] static and dynamic predictor weighting strategies were implemented and tested to improve the analog ensemble performance for wind power forecasting at on and offshore wind farms by using a brute force search procedure with error minimization over all possible predictor combinations. Usually, the general basic algorithm that follows an exhaustive or brute force search require two main stages: namely, Listing all the possible candidate solutions in a systematic way, and checking for the optimal solution and reporting it [12]. While the main disadvantage of brute exhaustive technique being its requirement for massive computational resources in order to find solutions in very large search spaces and which may sometimes makes it slow and infeasible [27], a drawback which can be addressed by using the search space reduction and algorithm parallelization strategies such as using parallel CPU–GPU computing structure. Its key advantage being the theoretical simplicity in implementation and ability to always identify global optimal solution given computational resources are available [14], with which may make this algorithm be deemed as a good choice especially when it will not require days, months, or years to locate the required solution in a real-life optimization problem.

3. Materials and methods

This study used ML ensemble techniques, brute exhaustive optimization to implement the optimal WVE solution. Also, the study applied mathematical linear algebra vectors and matrices, as well as arithmetic sequences were used to design and implement the proposed algorithm. Finally, laboratory experimentations were performed to evaluate the effectiveness of proposed solution in optimizing WVEs for soil fertility stratus prediction based on a real world agricultural soil chemical properties ML dataset, using the proposed algorithm's weights coefficients matrices.

3.1. Development of the 1EXP(-)Z^+ search spaces based computational brute Exhaustive WVE Optimization Algorithm

Lemma 3.1. From equation (1) in sub section 2.1.7. of the weighted voting ensemble scheme for model performance improvement, If the WVE combination equation (1) that is described by [19], when expressed as in equation (3) of its matrix form *Y*, that expresses a mathematical system of linear equation's that can be operated through matrix operations to compute the overall prediction outcomes for each WVE's combinations as a summation of the product of weights coefficients Wi and j base experts class probability predictions *C*1 to *C*j on a dataset *D* having *d* unseen targets instances values, where i > 1, and j > 1.

$$Y = \begin{bmatrix} w_1c_1 & w_1c_2 & w_1c_3 & w_ic_j \\ w_1c_1 & w_2c_2 & w_2c_3 & w_ic_j \\ w_2c_1 & w_3c_2 & w_3c_3 & w_2c_j \\ . & . & . & . \\ w_ic_1 & w_ic_2 & w_ic_3 & w_ic_j \end{bmatrix}, \tag{3}$$

Thereby, *Y* can be compared against true classes to score the prediction accuracy of the WVE, which for all other possibly available WVE combinations, the optimal set is chosen based on the one which satisfies an established criteria such as error minimization, accuracy or other performance measure maximization as an objective function.

Proof: The WVE matrix form in equation (3) has been notated by [19]. Where the study referenced values of the weights coefficients as a function of the individual WVE base learners f1_score performances for evaluating the efficiency of individual learners in the ensemble during training.

In addition, by using equation (1) of the WVE scheme in the sub section 2.1., or its corresponding matrix form, the WVE can also be represented into the basic system of linear equations defined in [28,29], as

$$\begin{aligned} Y_1 &= w_1c_1 + w_1c_2 + w_1c_3 + \ldots\ldots\ldots\ldots\ldots.. + w_ic_j \\ Y_2 &= w_1c_1 + w_2c_2 + w_2c_3 + \ldots\ldots\ldots\ldots\ldots.. + w_ic_j \\ Y_3 &= w_2c_1 + w_3c_2 + w_3c_3 + \ldots\ldots\ldots\ldots\ldots.. + w_2c_j \\ &. \\ &. \\ &. \\ Y_k &= w_ic_1 + w_ic_2 + w_ic_3 + \ldots\ldots\ldots\ldots\ldots.. + wicj \end{aligned}$$

These of which can be in matrix form as shown in equation (4)

$$Y[k] = \begin{bmatrix} w_1 & w_1 & w_1 & w_i \\ w_1 & w_2 & w_2 & w_i \\ w_2 & w_3 & w_3 & w_2 \\ . & . & . & . \\ w_i & w_i & w_i & w_i \end{bmatrix} * \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ . \\ c_j \end{bmatrix}, \tag{4}$$

3.1.1. The proposed multi precision search spaces formulation function

In general, the generation of a WVE of ML classifiers may consider mostly two phases that are i) Using various candidate ML algorithms to generate potential base members' classifiers that are to be used to form the WVE combinations, and ii) selection of base models optimal weights based on the WVE combination grounded by a accuracy performance criteria.

Proposition 3.1.1. If instead an ordered weights coefficients matrix *W*[*k*][*n*] can be automatically generated from the permutation of an explicit vector W[n] *that is* referred to as the search spaces *Sp* and Spz_+ *here* in, of weight values that satisfy the WVE weights coefficients domain constraints in equation (2), with a variable matrix *C*[*j*][*d*] of *j* base expert's class probability predictions on dataset *D containing d* total instances. Such that the resultant WVE combination k constant predictions matrix *Y[k][d] or Y_pred,* can be obtained from the product of the ordered weights coefficients matrix *W*[*k*][*n*] and variable matrix *C*[*j*][*d*] as shown in equation (5), which is augmented from equation (4) with the appending of the dimension of the dataset instances $I_{d,}$ for practical optimization purposes.

$$\mathrm{Y[k][d]} = \begin{bmatrix} k_1 w_1 & \cdots & k_1 w_n \\ \vdots & \ddots & \vdots \\ k_k w_n & \cdots & k_k w_1 \end{bmatrix} * \begin{bmatrix} c_1 I_1 & \cdots & c_1 I_d \\ \vdots & \ddots & \vdots \\ c_j I_1 & \cdots & c_j I_d \end{bmatrix}, \tag{5}$$

At that juncture, assuming that the *variable* matrix *C*[j][d] of m classifiers class probability predictions on instances *I* of dataset D with length d are provided, and an initialization function for explicitly formulation of values for generating the weight coefficients matrix *W*[k][n] which satisfy WVE weights constraints in equation (2) can be derived and developed as part of an automatic weighting values generation algorithm, then a Brute-exhaustive optimization procedure can be applied to search one optimal combination set from the automatically created WVE combinations predictions matrix in equation (5). This whose general form is that in equation (3). Whereas, equation (4) serves to compute the general form in equation (3) as a product of the of the weight coefficients *W*[k][n] and variable matrices *C*[j][d] of j individual classifiers probability predictions on supplied dataset d as represented in equation (5). But rather this time, the weight coefficients is automatically generated, hence the complete WVE general form in equation (3) will be automatically generated. It is to be proved that the general WVE combination matrix form representation in equation (3) can be automatically generated. In such, specifically for practical optimization purposes, brute exhaustive search can be automatically applied as long as dataset D with instances exist.

Proof: First, the variable K which represents the combinations counts is introduced into equation (3) to obtain a new representation form as in equation (6),

$$\mathrm{Y[k]} = \begin{bmatrix} k_1 w_1 c_1 & \cdots & k_1 w_n c_j \\ \vdots & \ddots & \vdots \\ k_k w_n c_1 & \cdots & k_k w_1 c_j \end{bmatrix} * \begin{bmatrix} c_1 I_1 & \cdots & c_1 I_d \\ \vdots & \ddots & \vdots \\ c_j I_1 & \cdots & c_j I_d \end{bmatrix}, \tag{6}$$

Then in subsequent sub sections 3.1.2, and 3.1.3, a function is derived to initialize the weights variable values, and incorporated as part of the proposed "1EXP(-)Z^+ initial term based arithmetic sequences multi precision search spaces algorithm function for systematic brute exhaustive optimization of intelligent small WVE (1EXP(-)Z^+-$_{\mathrm{IT}}$ASMPSS-BEO-$_{\mathrm{IS}}$WVE)".

3.1.2. 1EXP(-)Z^+-$_{\mathrm{IT}}$ASMPSS-BEO-$_{\mathrm{IS}}$WVE weight coefficients values formulation function closed loop equation

Whereas, as observed in [30], Taylors series can often be used for the derivation of algorithmic system's closed loop equation that expresses a particular problem domain. Herein, through lemma 3.1 and proposition 3.1.1, arithmetic sequence are used as a basis for the search space generation. These were applied as follows: A search space referenced by a positive integer denoted as Z^+, which is first set with $Z^{+=}1$ to represent the search space 1, whose precision is the first term a_o of an arithmetic sequence A in equation (7),

$$A = \sum_{n=1}^{(1/a_0)} (a_0 + a_0 * n), \tag{7}$$

3.1.3. 1EXP(-)Z+ based weights coefficients matrix and search spaces matrix computation

In order to automatically generate the search space values, represented as weights coefficients values matrix. First, the function $F(Z^+)$ in equation (8) whose computational representation is provided by the function in equation (9), is proposed herein to automatically initialize the first terms a_o's of the sequence, for all search spaces as sequence referenced by Z+, the positive integers greater than zero.

$$F(Z^+) = 1\exp\,(-)\,Z^+, \text{ for } Z+ >= 1, \tag{8}$$

$$F(Z^+) \text{ or } a_o = 1\exp\text{-}Z^+, \text{ or simply } 1/(1\exp Z^+), \tag{9}$$

Then $F(Z^+)$ which *is* a_o from equation (9), is used to serve as the basis for computationally generating the respective second to n^{th} terms a_1 to a_n of the arithmetic sequence A by using the arithmetic sequence's closed loop equation[1] (10).

$$a_n = a_o + d*n, \tag{10}$$

Consequently, by substituting a_o in the arithmetic sequence expression A from equation (7) by the proposed initialization function equation (9) values, to obtain arithmetic sequence A_{Z+} in equation (11).

$$A_{z+} = \sum_{z+=1}^{(1/1\exp\text{-}Z^+)} (1\exp\text{-}Z^+ + 1\exp\text{-}Z^+ * n), \tag{11}$$

Whereas, values a_0 to a_n, are used herein to represent weights values respectively w_1 to w_n specifying the WVE weights values domain. the permutation of the values a_0 to a_n as weights w_1 to w_n, to the list of WVE's constituting base models is later performed to complete the development of the proposed function algorithm for generating the search domain as WVE combinations proposed in proposition 3.1.1. This of which will be brute exhaustively searched. Finally, The weights coefficients values matrix as search space SP_{Z+} of K combinations is formulated as a permutation of sequences $Az+$ in equation (11) that represents the automatically computed weighting values by the WVE's list of base model C[*j*] in equation (12), to form the final automatically generated combinations matrix in equation (13).

$$SP_{Z+} = permutation\,(A_{Z+}, C[j]), \tag{12}$$

[1] O. Levin, "2.2: Arithmetic and Geometric Sequences," *Mathematics LibreTexts*, 2019.

$$SP_{Z+} = permutation((\sum_{n=1}^{(L)} (1\text{exp-}Z^{+} + 1\text{exp-}Z^{+} * \text{n})), C[j]) \tag{13}$$

where *L* is the reciprocal of the initialized fractional value based on $1eZ^{+}$, and Z+ is greater or equal to 0. A re-arrangement of the generated permutation spaces *Spz+*, from equation (13) in order of the dimensions of the variable matrix representing available classifiers class probability predictions would represent weighted output predictions for K combinations as expressed in equation (14).

$$\begin{bmatrix} y_1 k_1 \\ \vdots \\ y_k k_k \end{bmatrix} = \begin{bmatrix} k_1 1\text{exp-z}^{+} c_1 & \cdots & k_1 1\text{exp-z}^{+} * \text{ n } c_j \\ \vdots & \ddots & \vdots \\ k_k 1\text{exp-z}^{+} * \text{ n} c_1 & \cdots & k_k 1\text{exp-z}^{+} c_{(n^j)} \end{bmatrix}, \tag{14}$$

And by decomposing the matrix in equation (14) into its constant, coefficients, and variable matrices as defined in [31,32], equation (15) is obtained, which computes for the constant matrix as an output prediction as a product of the coefficients, variable matrices of the general WVE matrix form in equation (14).

$$\begin{bmatrix} y_1 k_1 \\ \vdots \\ y_k k_k \end{bmatrix} = \begin{bmatrix} k_1 1\text{exp-}z^{+} & \cdots & k_1 1\text{exp-}z^{+} * \text{ n} \\ \vdots & \ddots & \vdots \\ k_k 1\text{exp-}z^{+} * \text{ n} & \cdots & k_k 1\text{exp-}z^{+} \end{bmatrix} * \begin{bmatrix} c_1 \\ \vdots \\ c_2 \end{bmatrix}, \tag{15}$$

This of which when subjected to class probability predictions on dataset with d instances, the vector Y or Y[k][d] of equation (15) could finally be calculated as the argument max of the product of weights coefficient matrix and classifiers class probability predictions, which is then scored for accuracy against the true targets as observed in the data set D with I instances, for each k combination the accuracy is compared with the previous maximum score to pick it as a new maxim if the previous is small otherwise the algorithm proceed to the next combination iteration k. Until terminations conditions, the kth combinations with maximum accuracy is return as optimal WVE combination configuration set.

Whereas the final automatically generated search combinations in equation (14) is similar to the general WVE matrix equation (4) which was decomposed from matrix (3) in Lemma 3.1. In addition, as the WVE's full-form matrix in equation (13), and its corresponding variable and constant matrices in equation (14) are also similar to the WVE matrix forms in equations (5) and (6) in the initial proposition 3.1.1. Then, it entails that the automatically derived matrix form based on the proposed arithmetic sequences weights coefficients formulation function can well serve for representation of K possible WVE combinations in equation (1). Hence, the general WVE combination matrix form representation in equation (3) can be automatically generated. These could be implemented, which can then serve as automatic synthetic search space for brute exhaustive search.

3.1.4. The complete 1EXP(-)Z^{+}-$_{IT}$ASMPSS-BEO-$_{IS}$WVE algorithm

A pseudo-code of the straightforward implementation of the derived 1EXP(-)Z^{+} initial term-based sequences formulation and weights coefficients matrix generation algorithm is presented in Table 1. The algorithm execution starts at step 1, in step 2, the first sequence or search space reference is set to 1. Repeatedly from setp 3 until step 11, the search spaces *SPZ*$^+$ are generated. In step 12 the brute exhaustive procedure is called to search the formed search space SPZ$^+$ and return an optimal weights configuration set from the corresponding *SPZ*$^+$ based on class probability predictions C[*j*][*d*]. In step 13, the next sequence is initialized. In step 14, the algorithm checks if objective criteria and computational capacity are still not limited; the process repeats until either one or both of the termination conditions are satisfied. Finally, it provides the optimization results in step 15 before ending the execution in step 16. Whereas IEEE's mathematical co-processor FPA units permissible operation denoted by e substitutes 'exp' for practical implementation in the algorithm.

Table 1: The Complete 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE Algorithm.

Input: Base epxerts Probability predictions, true targets
Step 1: Start
Step 2: initialize search space precision (Z) = 1
Step 3: REPEAT
Step 4: Compute first term of sequence as ao = 1e – Z
Step 5: Initialize Search_space reference N = 1
Step 6: REPEAT
Step 7: Compute nth term an, an = (1e - Z) + ((1e - Z) * N)
Step 8: Sequence = ∪.Sequence+ an
Step 9: Increment N = N + 1
Step 10: UNTIL N < = 1eZ
Step 11: SPZ$^+$ = permutations(Sequence, E[j])
Step 12: Brute_Exhaustive_ optimization(SPZ$^+$, C[j][d])
Step 13: Increment Z = Z + 1
Step 14: UNTIL Z reach computational lim. or combination k
Step 15: Display optimization results
Step 16: End
Output: Optimal WVE subsets weights configuration

The 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE computational complexity was then asymptotically analyzed by calculating the proposed algorithms instructions lines asymptotic execution time as follows: substituting the complex for each execution line from the algorithm in Table 1, the total complexity could then represented as :

Total Complexity (TC) = F(Z^+) = {1}+{1}+{1}+{1e-Z^+}+{1}+{1}+{((1eZ^+) + ((1e-Z^+)*(N-1))} +{1}+{1} +{1eZ $^+$}+{1}+{1}+{1}+{1}+{1}+{1}

$$TC=\{11\}+\{1e^{-N}\}+\{((e^{-N})+\Big((e^{-N})*[(N-1)]\Big)\}+\{1e^{N}\} \tag{16}$$

From equation (16), it can be seen that {1e^N} is the highest order term, which is the worst-case scenario. Therefore, the algorithm has a worst-case scenario exponential complexity of {1e^N}. When this type of computational time complexity might be undesired in cases where search space precision grows so large, the upper bound running time could even fast be reached when the search spaces are integrated into the brute exhaustive-based search heuristics algorithm execution that would mainly arise from the size of ensemble base expert predictions to be weight estimated during optimization. Whereby, algorithm execution acceleration procedures namely, the constraining of search spaces with weights points, coupled with the vectorization of data structures thereof, and computation on reasonable computational hardware resources were used to facilitate for rapid execution of the algorithm computations in attempts to provide the algorithm execution run time minimization.

3.2. Experimentation

3.2.1. Datasets and base ML models

As part of this study, the dataset used to experiment the algorithms effectiveness in formulating search spaces on which the optimal weights of a WVE's could be estimated, was primarily obtained from the Tanzania Agriculture Research Institute (TARI), under the African Soil Information Services (AFSIS), and ministry of agriculture. The dataset contained sixteen (16) features, 15 of which are the key soil chemical properties necessary for the determination of fertility level as defined in [33], and the corresponding maize yields in harvested tons estimates mapping as index to fertility. With respect to ML classifiers, a total of seven algorithms classifiers were used, namely the support vector machine (SVM) [34], DecisionTreeClassifier (DT), GaussianNB (NB), KNeighborsClassifier (KNN), AdaBoostClassifier (AdaBoost), GradientBoostingClassifier(GB), and RandomForestClassifier(RF) [35].

3.2.2. Performance evaluation

In order to evaluate the performance of the proposed $1EXP(-)Z^{+}$-$_{IT}$ASMPSS-BEO-$_{IS}$WVE, an asymptotic analysis of its $1EXP(-)Z^{+}$ initial term based search space sequence formulation function algorithm procedures codes was performed to determine the algorithm computational complexity. Furthermore, the proposed algorithm's hardware clock cycles based execution times, and size complexity were obtained by executing it and profiling its search space function on the Intel(R) Core(TM) i7-855OU CPU @ 1.99 GHz with 16 GB RAM, as well as in the Core i8 hardware with 64 GB RAM, 64-bit operating system, which produced a result set constituting of similar results from Core i7, with more additional better results due the Core i8 hardware capacity which permitted for more computations. Accuracies and receiver operating characteristic's area under the curve (ROC AUC) of the WVE's found from search spaces generated by the proposed algorithm were used to evaluate its effectiveness. These are functions of the basic confusion matrix respective true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Accuracy as defined in equation (17), is the proportion of all predictions that are correctly identified as "Positive class" and "negative class".

$$\text{Accuracy} = (TP + TN) / (TP+TN+FP+FN), \tag{17}$$

Whereas accuracy is the most intuitive performance measure of model performance, the area under the curve (AUC) of the ROC AUCs, which for a multiclass problem can be computed by equation (18), was used as the main measure of effectiveness by observing the scores of the various WVE that are estimated from the formulated previously deemed effectual search spaces that resulted from the proposed $1EXP(-)Z^{+}$ initial-term based arithmetic sequences search spaces with accuracy maximization as an objective function.

$$\mathrm{AUC} = \frac{1}{c(c\text{-}1)} \sum_{j=1}^{c} \sum_{k>j}^{c} (\mathrm{AUC}(j|k) + \mathrm{AUC}(k|j)), \tag{18}$$

"where c denotes a total number of classes, AUC (j | k) represent AUC having positive class j, and negative class k" [36]. This of which can also be plotted and presented through ROC-AUC curves as false positive rate (1-specificity) against the true positive rate (sensitivity) [37,38,39].

4. Results and discussion

Following multiple execution of the proposed algorithm function and its overall search heuristic procedure runs, the results for its both its efficiency and effectiveness performance evaluation could be presented here in for the proposed 1EXP(-)Z^{+}-$_{IT}$ASMPSS-BEO-$_{IS}$WVE function efficiency and its brute exhaustive search based integration procedure effectiveness in optimizing for high performance WVE. Additionally, following an asymptotic analysis of the proposed algorithm's overall sequences generation function of equation (11) in sub section 3.1.2., it could be seen that, the 1EXP(-)Z^{+} based function would be mathematical validity for computing the WVE weighting values combinations as expressed by the WVE 1EXP(-)Z^{+} based matrix in equation (14), or its corresponding weight coefficients values matrix in equation (15).

As shown in Figure 1 of the derived 1EXP(-)Z^{+} initial term based arithmetic sequences formulation function expressions 3-D graphical display of its valid computational space, portrayed asymptotic optimality to the WVE constrained boundaries in equation (2). Whereas it can be observed the sequences initial term values represented by the y-axis in Figure 1 may get smaller as much as but never equal to 0 on the y-axis, hence the weights greater than 0 constraint is always maintains through that presented asymptotic characteristic, in turn the size of the sequence may grow larger to as much as the reciprocal of the 1EXP(-)Z^{+} as read form the x-axis based on the initialized sequence's first term on the y-axis. Based on those facts, the proposed function is considered mathematical valid for an optimal algorithmic system computational implementation. At that juncture a function for explicitly formulating values as weight coefficients W[k][n] which satisfy WVE weights constraints in equation (2) could be derived based on the proposed 1EXP(-)Z^{+} initial term arithmetic sequences , then a brute-exhaustive optimization could be applied to search one optimal combination set, hence that function provide for an algorithmic computational implementation. As shown in Figure 1 of the sequences formulation function, it is evident that the proposed weights coefficients values formulation algorithm's function is asymptotic optimal to the WVE weights domain constraints in equation (2), hence this study deduce that it can be computationally implemented to compute for the WVE combinations predictions Y[k][d], which is computed as the argument max function of the product of weights coefficient matrix and classifiers class probability predictions as presented in equation (15), which is then scored for accuracy against the true targets as observed in the data set D with I instances, for each k combination the accuracy is compared with the previous maximum score to pick it as a new maxim if the previous is small otherwise the algorithm proceed to the next combination iteration k. Until terminations conditions, the kth combinations with maximum accuracy is return as optimal WVE combination configuration set.

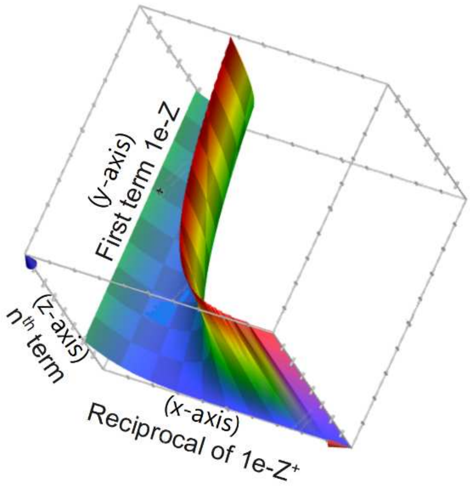


Figure 1: 1EXP(-)Z^+ based Sequence Initial term function asymptotic optimality to WVE weights constraints

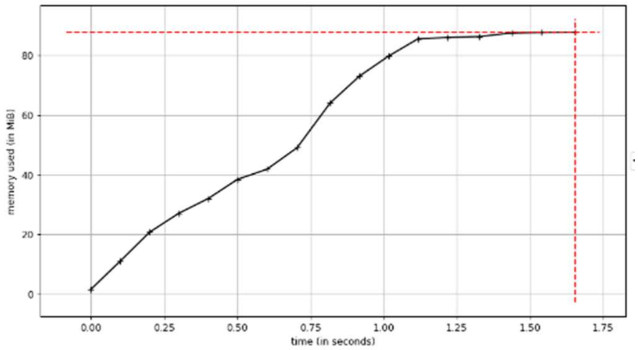


Figure 2: 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE sequences formulations total hardware clock cycles time in the most stable search space reference $Z^+ = 2$

4.1. 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE and optimization efficiency

Results of the algorithm efficiency are presented. Figure 2 displays results of the proposed 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE sequences formulations total hardware execution time profile in stable search space reference $Z^+ = 2$, the algorithm could be observed to consume approximately 90 MiBs, with total optimization execution time of approximately 1.7 seconds to formulate the sequences in for search space with reference Z^+=2 having precision factor 0.01, in Core i8 64 GB RAM, which maybe reasonable in WVE optimization procedure.

4.2. 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE effectiveness

The proposed 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE was highly effective in formulating multi-precision 1EXP(-)Z^+ based sequences that were processed to generate search spaces with varying combinations sizes in both search spaces 1 and 2, of which executions across referenced to these spaces were observably converging following execution of the proposed implementation a countless number of times. With 10 different sequence values in search space one (1), 100 in two(2), and 1000 in search space three(3) where the experimental core i8 64 GB hardware capacity limitation was reached to invoke the termination criteria, as a result forming an incomplete search space which was stored in log files. Among other reason, that could be explained by IEEE 754 standard for FPN system's FPA requirements specifying hardware's math co-processor world bit size memory limitations for FPA [40].

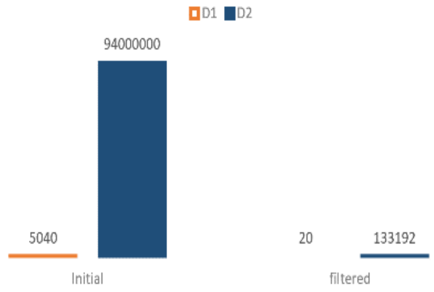


Figure 3: WVE initial and filtered potential search combinations in the stable domain search space 1 and 2

As annotated by the search space domain 2 filtered combinations plot in Figure 3. It can be seen that, unlike in search space 1, where five thousand and forty (5040) combinations were initially generated and filtered expressively by using the WVE weights boundary constraints in equation (2) as a reduction strategy that lead into only twenty candidate solutions, whereas these may be tractable by trial and error heuristic procedure, it would be a tedious task to do the same in search space 2, was the total number of generated combinations grew exponentially to one hundred and thirty-three thousand nine hundred and ninety-two (133,192) further filtered combinations of candidate solutions subsets which is a reduction from the initial formed ninety-four million (94,000,000) combinations due the maximum weight coefficients value being constrained to max of 1. Such combinations would be challenging to formulate without a computational, algorithmic implementation, such as the one proposed in this study, to effectively find optimal weights configuration sets based on prediction accuracy performance maximization as main objective criteria through brute exhaustive searching by considering the available hardware capacity.

In order to scrutinize the search space precision effect on the optimality based on accuracies of the various best WVE subsets, the proposed 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE was executed in search spaces 1 and 2, which have respective precision factors of 0.1 and 0.01. The partially logged combinations were processed independently of the package where it could not complete execution search space reference $Z^+ = 3$, with a precision factor of 0.001. As observed in Figure 4, search space with lower precision led to lower WVE accuracies, unlike those with higher precisions which showed to produce WVEs model with higher prediction accuracies. This fact explains not only that refined solution values could be achieved in search spaces with more higher precisions, but also it represent a good indication about the effectiveness of the proposed 1EXP(-)Z^{+}-$_{IT}$ASMPSS-BEO-$_{IS}$WVE algorithm function in generating search spaces as one of the key requirement for the successful execution of the consequent search procedure, as illuminated in [24], that how search spaces are a determinant factor as they have a significant effect in the overall optimization algorithm procedures implementation such as in finding WVE optimal subset, other than its diversity and constituting individual base model accuracies.

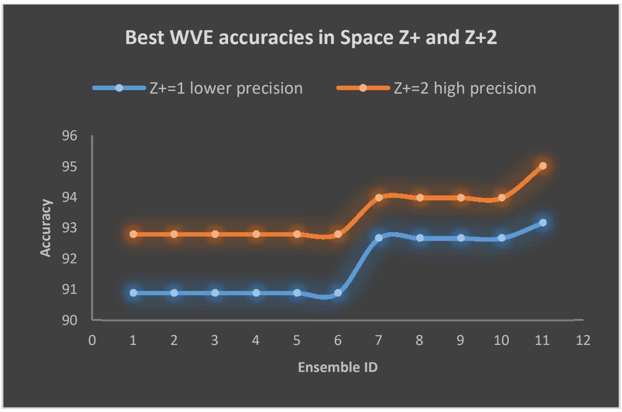
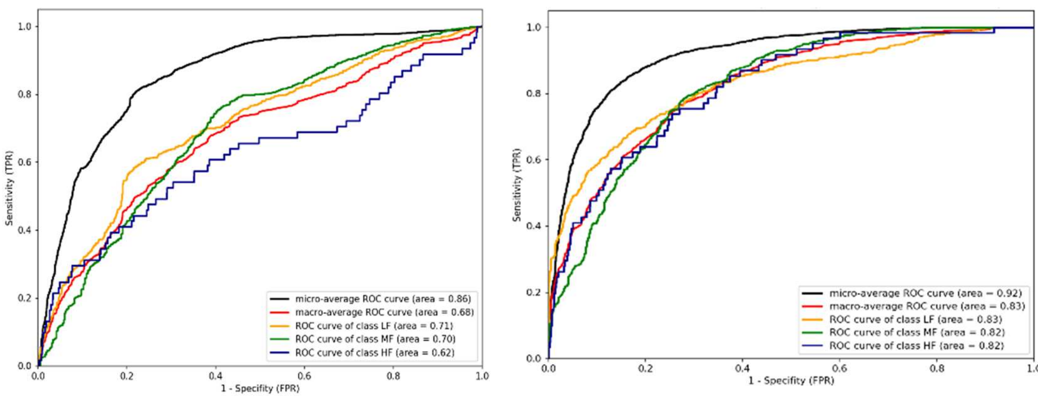


Figure 4: Best WVE accuracies in Space $Z^{+} = 1$ and $Z^{+} = 2$

Finally, as shown in Figure 5 of ROC AUC's plots for the three base models WVE that consisted of RF, SV, and KNN (See Figure 5 (a)), as well as another with four base models namely GB, RF, SVM and KNN (See Figure 5 (b)). These of which were brute optimized in search space having 0.01 precision and scale weights coefficient matrices, the effectiveness of the proposed 1EXP(-)Z^{+}-$_{IT}$ASMPSS-BEO-$_{IS}$WVE algorithm in generating effective search space with respect to the number of WVE's base models could be inferred.



(a) (b)

Figure 5: Optimal WVE results in respective 0.01, and 0.001precisons and scales search space for three and four WVE base models ROC plots and AUC results

It could be observed that, while the optimal WVE with three base model was good in correctly predicted target classes low and medium with respective ROC AUC scores of 71%, 70%, it almost guesses the high target class with 62% ROC AUC scores which is close to the 50% (0.5) cut point. In the contrary, the one with four base models exhibited a very good ability in increasingly discriminating and providing correct prediction for all low, medium, and high target classes with respective ROC AUC scores of 83%, 82%, and 82%. Whereas, as presented in the results, ROC AUC results of both these WVE's clearly outperforms results in another study by [41], other than the increased WVE diversity, such an achievement was due to the increase in the practical search space resulting from extra permuted base model.

5. Conclusion

In this paper, a 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE algorithm for optimizing weighted voting ensembles by using multi precisions search spaces was proposed to generate search spaces by using a 1EXP(-)Z^+ initial term based arithmetic sequences generation function algorithm which is mathematically valid to WVE weights domain constraints.

The proposed algorithm was observed to be effective in formulating multi precision search spaces and finding appropriate weights configurations sets across the 1EXP(-)Z^+ computationally generated multi precision WVE's base experts vs weights combinations search spaces. Whereby, ninety four million (94,000,000) possible values were formulated in the stable search space 2 whose sequence initial term value is 0.01, with 100 values as search space weights points, whereby by using four (4) base models, ninety four million (94,000,000) combinations could be generated, these which reduced through WVE weights constraints, into one hundred thirty three thousand nine hundred and ninety two (133,192) candidates. An optimal GB, RF, SVM, and KNN classifiers WVE could be obtained at a score of 94% prediction accuracy, with 83% AUC score for the macro average and 92% for the micro AUC score, which was 6% higher than a previously obtained RF, SV and KN combinations micro AUC score of 86%. Nevertheless, due to massive computational requirements that prematurely halts execution in search space 3 using Core i8 hardware with 64 GB RAM, with independent processing of the partially logged combinations to find a combination of GB, DT, RF, SVM, and KNN classifiers scoring an accuracy of 98.98%. Therefore, while the proposed algorithm has been effectively to optimize WVE combinations through its search space 2 and 3 generated weights coefficient matrices, it cannot be applied to very large WVEs.

Future work could be to investigate metaheuristics for improving the efficiency of the proposed 1EXP(-)Z^+-$_{IT}$ASMPSS-BEO-$_{IS}$WVE algorithm. Also, to experiment implementation of the proposed algorithm through quantum computation in capitalizing the rich qubit storage structures to deal with memory limitations, and inclusion of large WVEs optimization.

Acknowledgements. The work is supported by the African Development Bank (AfDB), United Republic of Tanzania, through project No.: P-Z1-IA0-016 and grant No.: 210015503281. And also, the Nelson Mandela African Institution of Science, Institute of Finance Management, and Technology (NM-AIST) and Tanzania Agricultural Research Institute (TARI). Finally, the authors are grateful for the reviewer's valuable comments.

Conflicts of Interest. The authors declare no conflicts of interest.

Authors' contributions. All authors contributed equally to this work.

REFERENCES

1. W.Nuankaew, P.Nuankaew, D.Doenribram, C.Jareanpon, Weighted voting ensemble for depressive disorder analysis with multi-objective optimization, *Current Applied Science and Technology*, 23(1) (2022) 10–55003.
2. A.Ekbal and S.Saha, Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach, *ACM Transactions on Asian Language Information Processing*, 10(2) (2011) 1–37.
3. D.Li, L.Luo, W.Zhang, F.Liu and F.Luo, A genetic algorithm-based weighted ensemble method for predicting transposon-derived piRNAs, *BMC bioinformatics*.17(1) (2016) 1–11.
4. Z.H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press; 2012. 238.
5. J.Liao, S.G.Teo, P.P.Kundu, T.Truong-Huu, An ensemble framework for unsupervised network anomaly detection, *Institute of Electrical and Electronics Engineers International Conference on Cyber Security and Resilience (CSR)* (2021).
6. J.Ast, R.Wasseghi, P.Nyhuis, A comparison of methods for determining performance based employee deployment in production systems, *Production Engineering*. 15(3) (2021) 335–342.
7. Simon. Answer to "How is Greedy Technique different from Exhaustive Search?". Stack Overflow. 2015.
8. L.Z.Chun-de Yang, K.Shu xian and L.Wang, Optimization of biclustering algorithm based on greedy randomized adaptive search procedure, *Journal of Mathematics and Informatics*, 12 (2018) 63–77.
9. Team TAE. Genetic Algorithm Introduction with Example Code. Medium. 2021.
10. Kurz CF, Maier W, Rink C. A greedy stacking algorithm for model assembling and domain weighting, *BMC Research Notes*, 13(1) (2020)1–6.
11. I.Ariyanti, M.A.Ganiardi and U.Oktari, Mobile application searching of the shortest route on delivery order of CV. Alfa Fresh With Brute Force Algorithm, *Jurnal Rancang Bangun dan Teknologi*, 19(3) (2019)120–130.
12. A.Angulo, D.Rodríguez, W.Garzón, D.F.Gómez, A.Sumaiti and S.Rivera, Algorithms for bidding strategies in local energy markets: Exhaustive search through parallel computing and metaheuristic optimization, *Algorithms*,14(9) (2021) 269.
13. A.He, J.He, R.Kim, D.Like and A.Yan, An ensemble-based approach for classification of high-resolution satellite imagery of the Amazon Basin, *Institute of Electrical and Electronics Engineers MIT Undergraduate Research Technology Conference (URTC), 2017*.
14. O.D.Okey, S.S.Maidin, P.Adasme, R.Lopes Rosa, M.Saadi, D.Carrillo Melgarejo, et al., BoostedEnML: efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning, *Sensors*,22(19) (2022) 7409.
15. Z.H.Zhou, Ensemble learning, *Encyclopedia of Biometrics*, doi. 2009;10:978–0.
16. H.M.Gomes, J.P.Barddal, F.Enembreck, A.Bifet, A survey on ensemble learning for data stream classification, *ACM Computing Surveys*, 50(2) (2017) 1–36.

17. T.Löfström, On effectively creating ensembles of classifiers: Studies on creation strategies, diversity and predicting with confidence, Department of Computer and Systems Sciences, Stockholm University; 2015.
18. P.E.Pintelas, I.E.Livieris, Ensemble algorithms and their applications. MDPI-Multidisciplinary Digital Publishing Institute; 2020.
19. J.Escorcia-Gutierrez, M.Gamarra, R.Soto-Diaz, M.Pérez, N.F.Madera and R.Mansour, Intelligent Agricultural Modelling of Soil Nutrients and pH Classification Using Ensemble Deep Learning Techniques. 2022;
20. M.Shahhosseini, G.Hu and H.Pham, Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. arXiv preprint arXiv:190805287. 2019.
21. S.T.Zouggar and A.Adla, A new function for ensemble pruning. In: Dargam F, Delias P, Linden I, Mareschal B, editors. Decision Support Systems VIII: Sustainable Data-Driven and Evidence-Based Decision Support. Cham: Springer International Publishing; 313 (2018) 181–190.
22. J.Brownlee, How to Develop a Weighted Average Ensemble With Python. Machine Learning Mastery. 2021.
23. I.Partalas, G.Tsoumakas and I.P.Vlahavas, Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection. In: ECAI. 2008. p. 117–21.
24. Mouret JB, Clune J. Illuminating search spaces by mapping elites. arXiv preprint arXiv:150404909. 2015;
25. T.Dauzhenka, P.J.Kundrotas and I.A.Vakser, Computational feasibility of an exhaustive search of side-chain conformations in protein-protein docking, *Journal of Computational Chemistry*, 39(24) (2018) 2012–2021.
26. C.Junk, L.Delle Monache, S.Alessandrini, G.Cervone and L.Von Bremen, Predictor-weighting strategies for probabilistic wind power forecasting with an analog ensemble, *Meteor Z*, 24(4) (2015) 361–379.
27. P.Pedamkar, Brute Force Algorithm: A Quick Glance of Brute Force Algorithm. EDUCBA, 2019.
28. M.Pospíšil, Representation of solutions of systems of linear differential equations with multiple delays and non-permutable variable coefficients, *Mathematical Modelling and Analysis*, 25(2) (2020) 303–322.
29. J.H.M.Wedderburn, On matrices whose coefficients are functions of a single variable, *Transactions of the American Mathematical Society*,16(3) (1915) 328–332.
30. M.H.Suhhiem and M.H.Lafta, Modified numerical method for solving Fredholm integral equations, *Journal of Mathematics and Informatics*, 16 (2019) 67–76.
31. R.Bellman, Introduction to matrix analysis, *Society of Industrial and Applied Mathematics*, 1997.
32. R.K.Kittappa, A representation of the solution of the nth order linear difference equation with variable coefficients, *Linear Algebra and its Applications*,193 (1993) 211–222.
33. E.K.Bünemann, G.Bongiorno, Z.Bai, R.E.Creamer, G.De Deyn, R.de Goede, et al. Soil quality–A critical review, *Soil Biology and Biochemistry*,120 (2018) 105–125.
34. J.X.Deng and G.Gan, A survey of stock forecasting model based on artificial intelligence algorithm, *Journal of Mathematics and Informatics*, 7 (2017) 73–78.

35. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Machine Learning in Python, (2011) 6.
36. Y.Yang, K.Zheng, C.Wu, X.Niu and Y.Yang, Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks, *Applied Sciences*, 9(2) (2019) 238.
37. N.A.Obuchowski and J.A.Bullen, Receiver operating characteristic (ROC) curves: review of methods with applications in diagnostic medicine, *Physics in Medicine & Biology*, 63(7) (2018) 07TR01.doi: 10.1088/1361-6560/aab4b1
38. D.K.McClish, Analyzing a portion of the ROC curve, *Medical decision making,* 9(3) (1989) 190–195.
39. J.V.Carter, J.Pan, S.N.Rai and S.Galandiuk, ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves, *Surgery*.=, 159(6) (2016) 1638–1645.
40. M.S.Committee, 754-2019-IEEE standard for floating-point arithmetic. 2019.
41. R.A.Viscarra Rossel, R.Rizzo, J.A.M.Demattê, T.Behrens, Spatial modeling of a soil fertility index using visible–near-infrared spectra and terrain attributes, *Soil Science Society of America Journal*, 74(4) (2010) 1293–1300.