Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 3, June 2024, pp. 1897~1912
ISSN: 2302-9285, DOI: 10.11591/eei.v13i3.6003

Ensemble learning based on relative accuracy approach and diversity teams

Mahmoud B. Rokaya[1,3], Kholod D. Alsufiani[2]
[1]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
[2]Computer Sciences Program, Turabah University College, Taif University, Taif, Saudi Arabia
[3]Department of Computer Science, Faculty of Science, University of Tanta, Tanta, Gharbia, Egypt

Article Info

Article history:

Received Feb 12, 2023
Revised Oct 14, 2023
Accepted Nov 14, 2023

Keywords:

Decision tree
Ensemble learning
Recurrent neural network
Support vector machine
Weighted voting

ABSTRACT

Ensemble learning, which involves combining the opinions of multiple experts to arrive at a better result, has been used for centuries. In this work, a review of the major voting methods in ensemble learning is explored. This work will focus on a new method for combining the results of individual learners. The method depends on the relative accuracy and diversity of teams. Instead of trying to assign weight to each different trainer, the concept of diversity teams is presented. Each team will vote as one player; however, the individual accuracies of each learner still be implemented. The concept of relaxing parameters that deal with each team as one player is presented. Our experiments demonstrate that traditional ensemble voting methods outperform individual learners. There is a limit to the superiority of the ensemble learner that any ensemble learner cannot go beyond. The proposed voting method gives the same results as the traditional ensemble voting methods, however, a different diversity of the proposed method from the traditional voting method or for different values of the relaxing parameter can be achieved.

Corresponding Author:

Mahmoud B. Rokaya
Department of Information Technology, College of Computers and Information Technology
Taif University
Taif 21944, Saudi Arabia
Email: mahmoudrokaya@tu.edu.sa

1. INTRODUCTION

Ensemble learning is defined as "ensemble learning depends on training a set of trainers and then using these trainers for implementing new data through taking a weighted vote of the trainer's results [1]. There are many combining methods, however, the most known methods are bagging and boosting [2]. Ensemble learning has many approaches and many works looked at considering many aspects. Some works focused on the types of trainers, either homogenous [3], [4] or heterogeneous [5], [6]. Some works looked at the purpose of using ensemble learning either for classifying [7]–[26], clustering [27]–[34], regression [35]–[37], or streaming [38]–[42].

Ensemble learning has many applications in almost all fields. In medicine, some works applied ensemble methods to predict the disease [43]–[50] or to classify the patients in each disease [50]–[56]. Also, ensemble learning is used in medicine for DNA prediction [57] or for DNA imbalanced splice site datasets [58]. Ensemble learning has applications in security [59]–[63]. Social media is not an exception, and many works depend on ensemble learning to achieve better performance or cover multi-data types for one learner [64]–[74]. Ensemble learning is used in commerce [75]–[78] and credit cards [79]–[81]. Image processing is a traditional field for machine learning and ensemble learning, for example [82], [83]. Industry [84]–[94], land science and geology [95]–[98], agriculture [99]–[102], weather [103]–[105], transportation [106]–[108], and education [109] have a huge amount of research to handle ensemble learning. However, there are a tremendous number of works in applying ensemble learning in all fields, but these works depend on developing some homogenous or heterogenous methods as individual learners, and when it comes to the merging step classical and simple merging methods are used.

The basic elements in ensemble learning are the diversity of individual learning and the method of merging trainers' results. Combining the individual learners' results is the core of ensemble learning. This work proposes a combining method depending on the dynamic voting method. Many works studied the voting methods in ensemble learning. Liu and Truszczynski [110] proposed a method for voting-based ensemble learning for partial lexicography. Araz and Spannowsky [111] proposed a combine-and-conquer that is based on Bayesian ensemble neural networks. Suchithra and Pai [112] evaluated the performance of bagging-based k-nearest. neighbor and proposed a voting rule method. Kim *et al* [113] proposed a two-stage weighting method for voting in ensemble learning. Some works proposed different methods for voting in classification problems [114]–[119]. In the related works section details of voting works will be discussed. Some applications concentrated on voting to present a solution to some problems [120]–[128].

In this work, a proposed voting method considered the methods of combining the results of the individual learning methods to get the results for new data examples. Most of the work concentrated on applications. In those studies, some learners were presented as learners who could solve a part of the problem or can achieve limited accuracy. The ensemble method is applied and the used methods in the process of combining either are the known methods or the rough voting. These works avoided the newer proposed methods because of the complexity of these methods. The current work presents a light method that can be applied for any type of learner and any type of application. The current work belongs to the work that searches for how to combine the results of the individual trains to get a global better performance. There are a lot of works to study the voting process.

The main idea of this work is to look at the trainers as a team. In fact, the practical work leads us to consider two teams. We will propose a method to define the extreme conflict trainer for each trainer based on the concept of maximum conflict, then each member of the resulting pairs will be assigned to one of the sets each of them is called a conflict team based on the concept of minimum conflict. We will construct based on the prediction vector for each trainer the diversity matrix which expresses the total number of different predictions for each pair of trainers. Based on this matrix the concept of conflict pairs and conflict teams are defined.

2. RELATED WORK

Smith and Martinez [116] tested the strength of ensemble learning to get rid of the outlier examples without the need for a filtering process prior to training the individual learners. They trained 9 weak trainers on unfiltered data and then developed an ensemble learner based on rough majority voting. The ensemble learner outperformed any individual learner regardless of the used filtering method to clean the data before the training process. This work proved the importance of ensemble learning and its power to deal with data with a lot of noisy or present outlier samples.

The goal of using ensemble learning is to reduce the variance for the sake of improving the accuracy of the whole system. Ensemble learning deals with all addressed problems in machine learning including feature selection, error correction, imbalanced classes, losing features, incremental learning, the concept of drift from nonstationary distributions, and others. Ensemble learning, despite getting attention in machine learning for a few years, is old and it might be parallel to the history of humanity [129]. In our daily lives, we apply the basic concept in ensemble learning when we ask some experts to gain a wide insight to be able to solve some problems that have multiple aspects. The early ensemble learning methods are bagging, boosting and ada-boost, stacked generalization, and mixture of experts [129]. The current work concentrates on developing a method to test the diversity of a given learner and then proposes a dynamic method to merge the learners' individual results for predicting new instances of results.

In the following part, some recent works that discussed the voting process and its applications to solve some problems will be explored. Araz and Spannowsky [111] developed an ensemble learning method where the ensemble learner gives feedback to each neural network to improve the representation. of the network hypothesis. To use the ensemble learner to modify the parameters of each individual learner is very promising and might bring the field to a new era in machine learning where the machine with the help of ensemble observations can change the method of thinking (modify internal hyperparameters) of each individual learner without the need of manual changing (human supervision).

Kim *et. al.* [113] introduced the WAVE method for voting in ensemble learning. The method depends on iterative procedures that assign two different weighting vectors, one weight vector for classifiers and another one for instances. The two vectors are connected in such a way that the vector of instances determines the weights of the classifier's weight vector. This method tried to pick the classifiers that have a bigger chance of picking the correct class of a given data instance. In fact, it tries to catch the winner classifiers.

Kuncheva and Rodríguez [114] proposed a type of ensemble of ensembles. In their work, they used four different methods for combining the results of the trainers. These methods are naïve Bayes, recall combiner, majority voting, and weighted majority voting. First, they choose one combining method and then generate the next combining method from the last one in a subsequent manner. This method is very heavy since it first needs to develop the trainers then add one ensemble method then go back to change the set of parameters in each trainer and hence generate the subsequent combining method.

Zhang *et al.* [115] developed a voting method in ensemble learning to deal with the imbalanced data classification problem. The method depends on weighted majority voting. However, the weight of each classifier is generated as the solution of an optimization problem based on a differential evolution algorithm. This means that it is required to solve one evolutionary optimization problem for each weight for each classifier after setting the original training phase for each classifier.

Liu and Truszczynski [110] presented a method for ensemble learning that depends on merging a set of small trees (partial lexicographic preference (PLP) trees). Instead of using a large tree. The tree is divided into a forest composed of small PLP- trees. According to their results, they proved that any combining method for the small trees in one ensemble learner will give a result that competes with the individual learners regardless of the combining method used to merge the merge the small trees. In fact, in the current work, we will show a similar result. However, the details show that despite different combining methods giving very near accuracies but depending on the combining method, one can develop a combining method that competes with the current voting methods and can generate several ensemble learners with near accuracies but have a significant diversity that might enable us to choose the most suitable combining method for a given application.

Cornelio *et al.* [117] adopted the approach of weak learners without tunning or pruning to the hyperparameters then margining them in one ensemble learner can compete with any sophisticated tunned learner. However, the margining method can affect the accuracy of the ensemble learner not only for its accuracy but also in the diversity that can be gained for each different ensemble learner that one can get through different combining methods. achieved through different methods. Rojarath and Songpan [119] addressed the issue of multi-class data. They proposed a cost-sensitivity matrix of the true positive (TP). This matrix, in conjunction with a probability measure, was used to assign a weight for a set of heterogenous trainers in an ensemble learning model. It is very important that the ensemble learner model be independent of the type of individual learners and the type of problem at hand. In the current work, the proposed combining method is independent of the type of each weak learner of the problem at hand domain.

Delgado [118] proposed, based on the confidence level CL that assigns a degree of support of each weak learner and bagging approach, a voting scheme. The degree of support measure depends on the probabilistic of the error of each individual classifier. When the number of weak leaners is odd the proposed ensemble voting approach can compete with the simple voting majority approach for the binary classification problems distribution. For multiclass problems, the degree of support depends on the error distribution of the classifiers and additional knowledge of the probability distribution over the classes. Restricting the number of classifiers to be odd and defining a different weighting method based on the number of classes present in the problems make this method very limited in its applicability.

Xu *et al.* [130] proposed a decision model based on an ensemble learning approach through the following two-stage scheme. This approach adopts the dynamic weighting of the base classifiers which can be learned from successful decisions history. The model depends on classic weighted majority voting. However, the approach depends on a continuous refreshing of the weights based on historical decisions.

Suchithra and Pai [112] used the nearest neighbour estimation and bagging ensemble method to propose an ensemble learner. Through implementing k-nearest label ranker, an ensemble model was proposed. The voting method was the voting rule selector (VRS) which was integrated with another traditional voting method to develop an ensemble learning model.

The above-mentioned works either adopt a very simple voting approach (simple voting majority) or propose a very complicated approach that cannot be applied in many situations. Based on a set of considerations the proposed voting was developed. These considerations are: voting scheme is essential in building a proper ensemble learning model. The voting method should be simple and independent from the problem class and the nature of the individual learners as well. The diversity of the weak learners should be checked before deciding the voting method. In this work, instead of considering the weak learners as individual players who work individually to achieve the team goal, the voting method based on the diversity matrix will distribute the learners into two cooperative teams. The weights will be designed to reflect the cooperation between the two teams to achieve better results. Without losing the generality of the approach, the method is applied in the case of binary classification. Each individual learner will be trained without any optimization approach to improve its hyper parameters. The proposed scheme can be applied to any set of homogenous or heterogenous set of learners.

3. METHOD

Suppose that we have a set of data D and we need to choose the best class for a given element $x \in D$ through choosing the best trainer from all trainers set that can be found to classify the elements of the set D. Bayes theorem can be used to find the best trainer t_{opt} among the set of all trainers T. t_{opt} must be the most probable trainer that can be used to classify elements of D. Hence according to Bayes theorem c_{opt} is given by (1):

$$t_{opt} = \arg\max_{t \in T} p\left({}^{t}/_{D}\right) = \arg\max_{t \in T} \frac{p\left({}^{D}/_{t}\right) \times p(t)}{p(D)} \equiv \arg\max_{t \in T} p\left({}^{D}/_{t}\right) \times p(t) \tag{1}$$

It is clear that we have an exhaustive search problem that might not be solved in a real-time since the size of the set T might be very large. The alternative solution is to try to relax the assumption from trying to find the best trainer among all set of trainers to find the best class of a given data element using a given set of trainers. Instead of designing a competition between the trainers to find the winning trainer, let them cooperate to find the best class for a given element from the set of tata [104]. As many trainers can be added as much as a better performance [111]. Weight voting was used in many works [113]–[115], [119]–[124]. The relative accuracy $acc_c(t)$ of a trainer $t \in T$ with respect to a class $c \in C$ can be stated as the times that t classified an element $x \in D$ correctly as an element from the class c divided by the total number of elements in the class c $acc_c(t) = \frac{|pre_t(x \in D) = c|}{|x_c \in D|}$ where $pre_t(x)$ is the predicted class of x by trainer t based on bayes theorem and relative accuracy, the best class c_{opt} can be found from the set of classes C based on the fact c_{opt} must be the most probable class given the set of all possible trainers T through (2):

$$c_{opt} = \underset{c \in C}{\operatorname{argmax}} \sum_{t \in T} p\left({}^{C}/_{t}\right) \times p_c\left({}^{t}/_{D}\right) \tag{2}$$

Figure 1 shows the structure of the proposed system that simulates how (2) can be implemented. We have a given number of trained weak learners on a given data to classify the data into a given number of classes. There are many methods to calculate the weights in (2). one method is the dynamic weighting approach to combine between the results of the weak learners to get an optimal classification. In such dynamic weighting, after getting the predicted class for all instances of data, the ensemble trainer can be built.

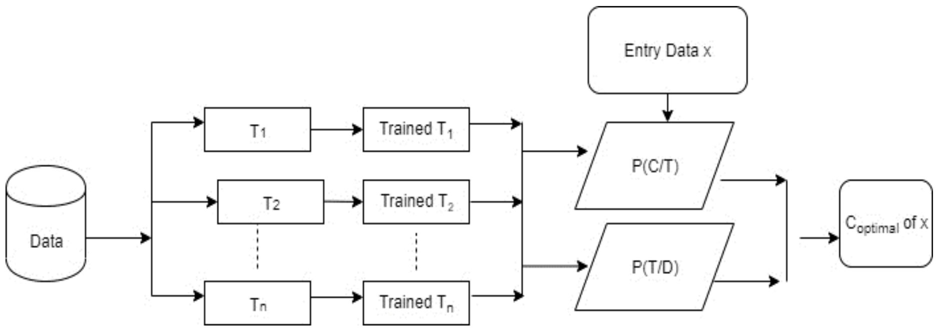

Figure 1. General view of ensemble learning model based on relative voting

Suppose we have n independent weak learner. $t_1, t_2, \ldots, t_n$ and m classes $c_1, c_2, \ldots, c_m$. One can define $P_{c_j}\left({}^{t_i}/_{D}\right)$ to represent the relative accuracy of t_i with respect to class c_j divided by the sum of relative accuracies of all weak learners with respect to class c_j.

$$P_{c_j}\left({}^{t_i}/_{D}\right) = \frac{acc_{c_j}(t_i)}{\sum_{k=1}^{n} acc_{c_j}(t_k)}, i = 1, \ldots, n \tag{3}$$

class of a given data x using t_i is $C_j, i = 1, \ldots, n$ and $j = 1, \ldots, m$. The optimal class $C_{optimal}$ of x can be determined by (4):

$$C_{optimal}(x_k) = \underset{C_j \in C}{\arg max} \sum_{t_i \in T} p_x(C_j/t_i) * P_{c_j}(t_i/D) \quad (4)$$

For entry,

$$x_k \in D, p_x\left({C_j}/{t_i}\right) = \begin{cases} 1 \; if \; pre(t_i, x_k) = C_j \\ 0 \; if \; pre(t_i, x_k) \neq C_j \end{cases} \quad (5)$$

where $pre(t_i, x_k)$ is the predicted class of x_k made by t_i. $p(t_i/D)$ give us an idea about which weak learner is correct most of the time. And $p({C_j}/{t_i})$ give us a tool to sum the accuracies of all weak learner gives their votes dynamically to the class C_j, So it is a tool of dynamic voting. In (5) represents the rough dynamic weighting approach. The trainer only votes for the class that this trainer assigned to the entry.

Note that there is no direct method to calculate $p_x\left({C_j}/{t_i}\right)$, (5) is one of many possible candidates that can be used to calculate $p_x\left({C_j}/{t_i}\right)$. The optimal values can be gotten through considering $p_x\left({C_j}/{t_i}\right)$ as weights and the problem can be stated as finding the optimal values of $p_x\left({C_j}/{t_i}\right)$ that reduce the error in results of (4). This will formulate the ensemble problem as a neural network with one input layer and one output layer and with no hidden layers. Figure 2 shows the structure of such a network.

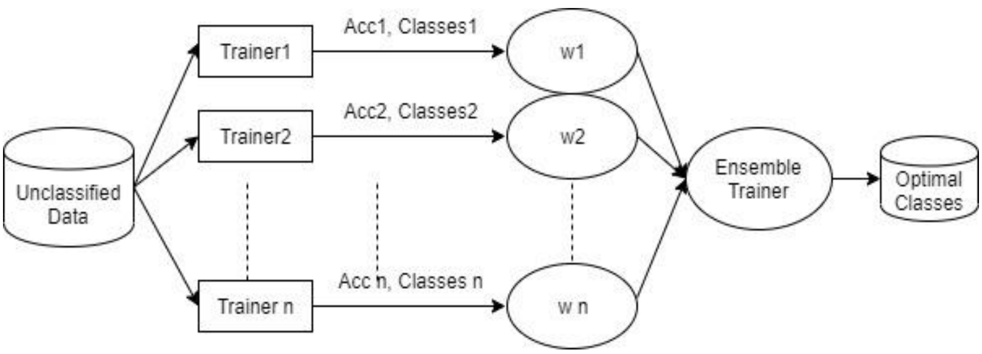

Figure 2. Ensemble learning as simple neural network containing one node

Also, it can be stated as an optimization problem. Find the weights $p_x\left({C_j}/{t_i}\right)$ that minimize the error or maximize the gain (accuracy) of the learner expressed in (4) and (5).

In the following, two examples will be provided how the proposed values for $p_x\left({C_j}/{t_i}\right)$ in (5) might solve problem of classic majority voting or its variations that cannot solved based on the classical majority voting method. However, there is no theoretical guarantee that enables us to claim that the proposed ensemble method based on (4) and (5) can give better performance than the classic majority voting method. As we will see in the results. The proposed method can give at least the same results as the classic majority methods. In the results, we will compare the results of the majority voting and variations of the proposed method.

Example 1: suppose we have 6 weak learners $t_1, t_2, \ldots, t_6$. Table 1 shows the values of using these 6 weak learners to classify a give data into two classes $\oplus$ and $\ominus$. Table 1 summarizes the calculation of $P_{c_j}\left({t_i}/{D}\right), i = 1,2, \ldots, 7$ and $j = 1,2$. And $p\left({C_j}/{t_i}\right), i = 1,2, \ldots, 7$ for a given item x. Since 3 trainers classified x as $\oplus$ and another 3 trainers classified x as $\ominus$, hence the classical voting will fail to assign the correct class of x. The optimal class of x can be calculated based on Bays naïve as follows:

Table 1. Dummy example 1 to show the validity of dynamic relative voting model

Learner	$Acc_{\oplus}$	$Acc_{\ominus}$	$P_{\oplus}(^{t_i}/_D)$	$P_{\ominus}(^{t_i}/_D)$	$p_x\left(^{\oplus}/_{wl_i}\right)$	$p_x\left(^{\ominus}/_{wl_i}\right)$	$\oplus$ Voting	$\ominus$ Voting
wl_1	0.90	0.70	0.18	0.15	1	0	0.18	0.00
wl_2	0.70	0.85	0.14	0.18	1	0	0.14	0.00
wl_3	0.90	0.70	0.18	0.15	1	0	0.18	0.00
wl_4	0.80	0.80	0.16	0.17	0	1	0.00	0.17
wl_5	0.70	0.90	0.14	0.19	0	1	0.00	0.19
wl_6	0.90	0.80	0.18	0.17	0	1	0.00	0.17
Ensemble decision							0.51	0.53

$\sum_{t\in T} p(^{\oplus}/_t) \times p_{\oplus}\left(^{\oplus}/_D\right) = 0.51 < \sum_{t\in T} p(^{\ominus}/_t) \times p_{\ominus}\left(^{\ominus}/_D\right)$=0.53, then, the ensemble prediction $x\epsilon \ominus$.

Example 2: this example shows that, the minority can gain a greater value than the majority and can decide the correct class. Table 2 shows the values of using these 7 weak learners to classify a give data into two classes $\oplus$ and $\ominus$. Table 2 summarizes the calculation of $P_{c_j}\left(^{t_i}/_D\right), i = 1,2, \dots ,7$ and $j = 1,2$. And $p\left(^{C_j}/_{t_i}\right), i = 1,2, \dots ,7$ for a given item x. Since 3 trainers classified x as $\oplus$ and another 4 trainers classified x as $\ominus$, hence the classical voting will assign the class of x as $\ominus$. While the optimal class of x based on the proposed method can be calculated as $\sum_{t\in T} p(^{\oplus}/_t) \times p_{\oplus}\left(^{\oplus}/_D\right) = 0.53 < \sum_{t\in T} p(^{\ominus}/_t) \times p_{\ominus}\left(^{\ominus}/_D\right)$=0.51. Then, the ensemble prediction $x\epsilon \oplus$. Dynamic voting based on the prediction of each trainer is assumed to give better results than the results of each individual trainer as well as better than other ensemble learning methods. In the following section we will explore the results of real experiments. Assigning $p_x\left(^{\oplus}/_{t_i}\right)$ to be zero or one is called rough relative majority. If we relaxed this value to be a number between zero and one, we might get a better result. Using a unified value for all trainers is called fixed relaxed relative majority voting.

Table 2. Dummy example 2 to show the validity of dynamic relative voting model

Learner	$Acc_{\oplus}$	$Acc_{\in}$	$P_{\oplus}(^{t_i}/_D)$	$P_{\ominus}(^{t_i}/_D)$	$p_x\left(^{\oplus}/_{t_i}\right)$	$p_x\left(^{\ominus}/_{t_i}\right)$	$\oplus$ Voting	$\ominus$ Voting
t_1	0.90	0.60	0.18	0.13	1	0	0.18	0.00
t_2	0.90	0.60	0.18	0.13	1	0	0.18	0.00
t	0.80	0.60	0.16	0.13	1	0	0.16	0.00
t_4	0.80	0.80	0.16	0.17	0	1	0.00	0.17
tl_5	0.70	0.80	0.14	0.17	0	1	0.00	0.17
t_6	0.90	0.80	0.18	0.17	0	1	0.00	0.17
t_7	0.60	0.80	0.12	0.17	0	1	0.00	0.17
Ensemble decision							0.53	0.51

In the experiments, many variations of (4) and (5) will be tested. The concept of diversity will be used to propose a different method to look at the ensemble learning and the voting process. Based on the diversity concept and the conflict trainer the trainers will be divided into 2 teams. Instead of assigning weights for each trainer two weights can be assigned for each team. The goal of the two teams s to assign the best class for a given input. To define the diversity matrix, assume that for a given data x a trainer t_1 assigned the class C_1 for x which we will assign it the value θ_1 and a trainer t_2 assigned the class C_2 for x which we will assign it the value θ_2 then then the sum of the absolute difference $|\theta_1(x) - \theta_2(x)|$ for each element x in the data set. In (6) define the entry in the diversity matrix for each two trainers.

$$div(t_j, t_k) = \sum_{x\in D} \left|\theta_j(x) - \theta_k(x)\right| \quad (6)$$

For each trainer, the conflict trainer is defined as the trainer that gives the maximum difference in the column of the trainer in the diversity matrix. Successive deletion and iteration will be used to define the pairs of conflicting trainers. First, locate the maximum value in the diversity matrix, this assigns the first two conflict trainers, The column and rows of those two trainers are omitted from the diversity matrix. The process is repeated for the resulting diversity matrix till we get an empty diversity matrix or a diversity matrix with one trainer. This unique trainer is called (if any) neutral trainer and it will be omitted from the right to vote. Figure 3 explains the repeated process of reducing the diversity matrix for 5 trainers.

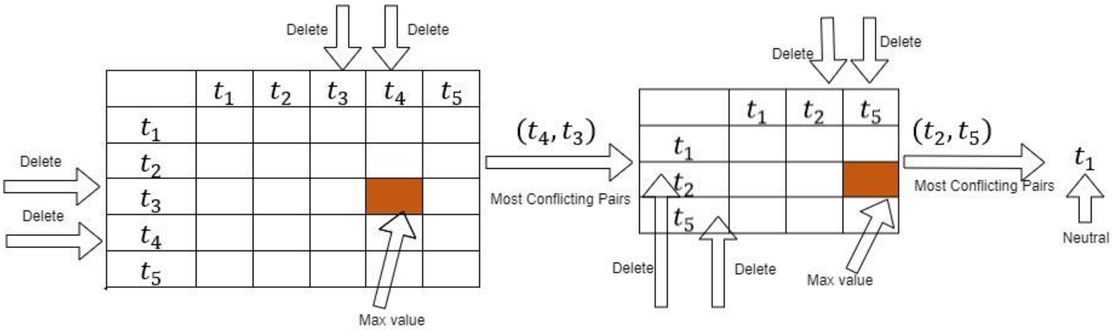

Figure 3. Using diversity matrix phase 1 in generating diversity teams (conflicting pairs)

Now, we have determined the pairs of most conflicting trainers. Suppose (t_i, t_k) and (t_j, t_l) are conflicting pairs. This means that t_i is conflicting to t_k and t_j is conflicting to t_l. Then what is the relation between t_k and t_j or t_i and t_l. From the original diversity matrix, the two trainers with minimal diversity will be added in one set. This process can be repeated to get two sets of diversity teams. A dynamic relaxed voting method can be proposed based on the concept of diversity teams. The dynamic relaxed relative majority voting depends heavily on the sets of conflictions. Figure 4 explains how to get such conflictions sets.

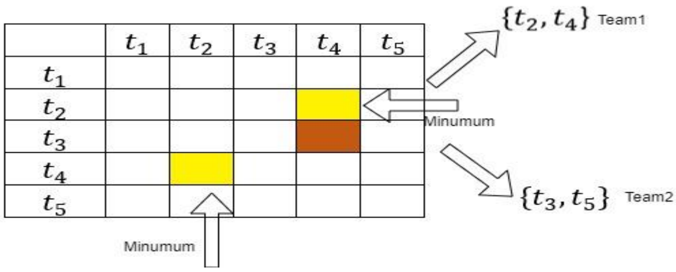

Figure 4. Using diversity matrix phase 2 in generating diversity teams (cooperative members)

In this work we can define 4 different types of voting strategies. In (4) and (5) represent the relative voting strategy. If the relative accuracy is replaced by the absolute accuracy of each trainer, one gets the absolute relative voting strategy. Depending on the concept of diversity matrix and the diversity teams, one can get many of voting strategies where we propose a value $\propto$ between [0,1] to represent $p_x\left({}^{C_j}/_{t_i}\right)$ for the first team and use $1-\propto$ to represent $p_x\left({}^{C_j}/_{t_i}\right)$ for the other team. A number of trials can be done to find the best value of $\propto$. In the experiments, 3 values were determined to give the highest accuracies. The corresponding ensemble trainers were called R1, R2, and R3.

In (7) to (9) correspond to relative absolute dynamic voting, where the relative accuracy for each trainer is replaced by the absolute accuracy $Acc_{t_i}(t_i/D)$.

$$C_{optimal}(x_k) = \arg\max_{C_j \in C} \sum_{t_i \in T} p_x(C_j/t_i) * P_{t_i}(t_i/D) \tag{7}$$

For entry,

$$x_k \in D, p_x\left({}^{C_j}/_{t_i}\right) = \begin{cases} 1 \; if \; pre(t_i, x_k) = C_j \\ 0 \; if \; pre(t_i, x_k) \neq C_j \end{cases} \tag{8}$$

$$P_{t_i}\left({}^{t_i}/_{D}\right) = \frac{acc(t_i)}{\sum_{k=1}^{n} acc(t_k)}, i = 1, \dots, n \tag{9}$$

The relaxed voting strategy rules can be given through replacing $P_x({}^{C_j}/_{t_i})$ by the relaxing parameter $\propto$ for one team and $1-\propto$ for the other team in (5). The majority voting rule is given by (11) and (12):

$$C_{optimal}(x_k) = \arg\max_{C_j \in C} \sum\nolimits_{t_i \in T} \Lambda_{C_j}(t_i(x)) \tag{11}$$

$$\Lambda_{C_j}(t_i(x)) = \begin{cases} 1 \; if \; t_i(x) = C_j \\ \\ 0 \; otherwise \end{cases} \tag{12}$$

4. RESULTS AND DISCUSSION

All related experiments and data can be accessed through the following link. The system consists of three types of weak trainers. The categories of weak trainers are as follows. Ordinary support vector machine (SVMT) trainer, decision trees trainer (DTT) and a set of deep learning trainers. The deep learning trainers are two dense connected traiers, one with one hot embedding (DSH) and one with embedding layer (DSE). Based on variations Recurrent net, two different trainers are built, GRNT and LSTMT. Finally, based on conventional nets a one CONVD1 was built (CONVD1T). So, the total number of weak trainers used in this work is seven. Figure 5 shows the architecture of the deep learning trainers. Figure 5(a) shows the architecture of DSH trainer structure. Figure 5(b) shows the architecture of DSE trainer. Figure 5(c) shows the architecture of GRN trainer. Figure 5(d) shows the architecture of LSTM trainer. Figure 5(e) shows the architecture of CONVID trainer.

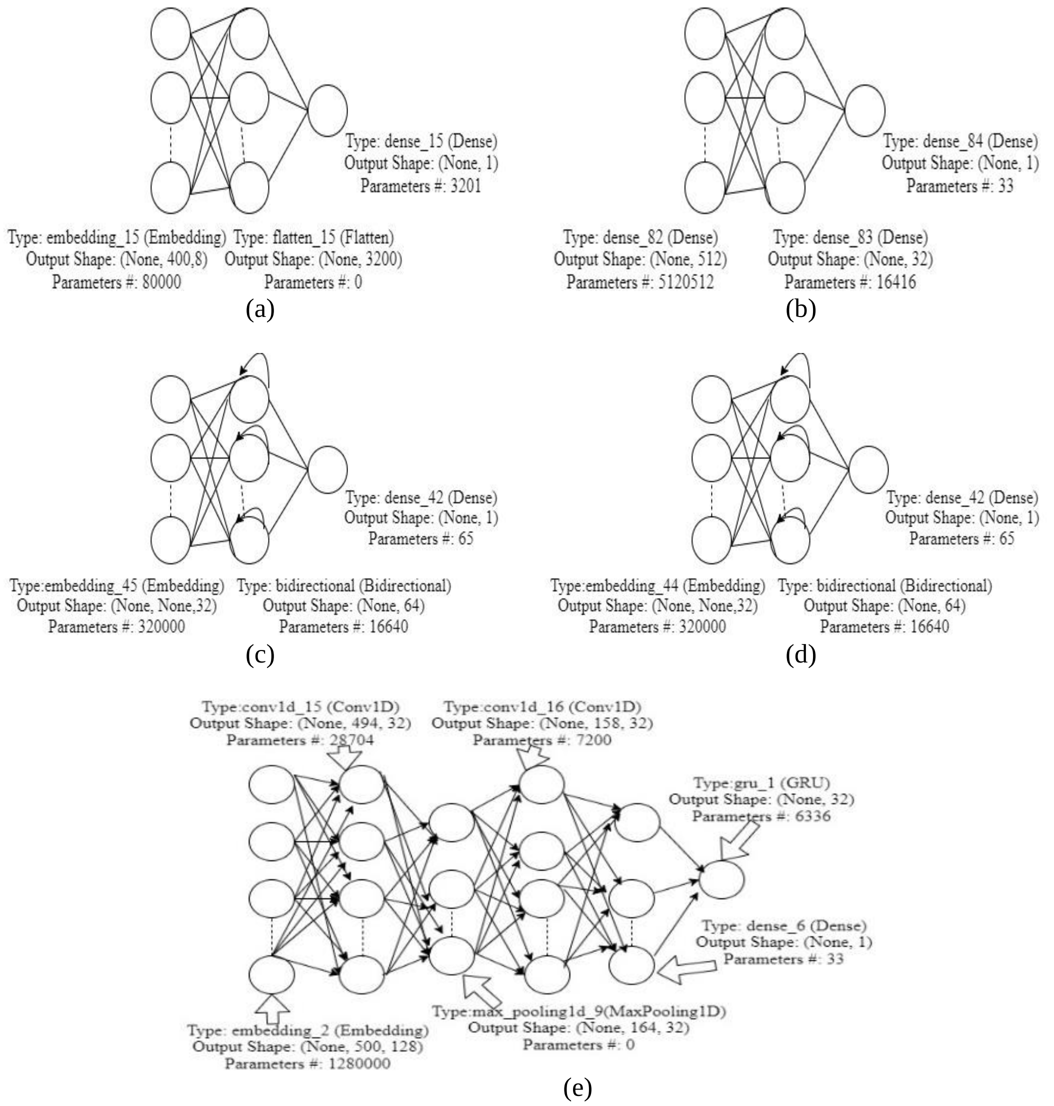

Figure 5. Neural networks trainers structure; (a) DSH trainer structure, (b) DSE trainer structure, (c) GRN trainer structure, (d) LSTM trainer structure, and (e) CONVID trainer structure

All trainers are trained using internet movie database (IMDB) dataset. IMDB consists of 50,000 reviews from the IMDB. IMDB has 50% negative reviews and 50% positive reviews. The set of 50000 reviews was divided into 25000 reviews for training and 25000 reviews for testing. Figure 6 shows the training and validation results for loss and accuracy of some trainers. Most of the trainers tend to get higher overfitting after few epochs. The results of testing accuracy for all trainers are listed shown in Table 3. The highest testing accuracy was achieved by DSE trainer, and the lowest testing accuracy was achieved by DTT trainer.

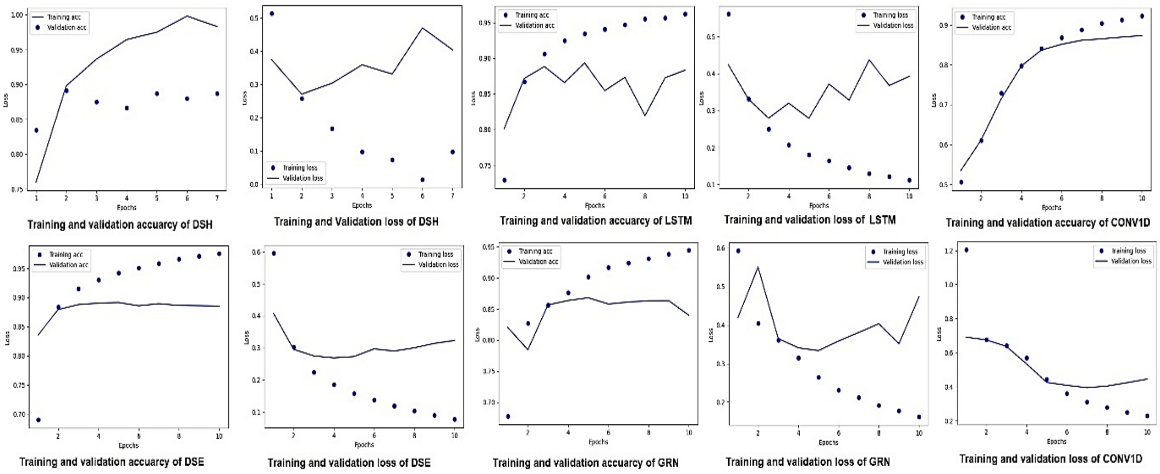

Figure 6. Results of absolute accuracy and loss functions during the training of the individual trainers

Table 3. Relative accuracies of the individual trainers with respect to the negative and positive classes

Trainer	Accuracy	Negative_acc (⊖)	Positive_acc(⊕)
DSH	0.87348	0.87872	0.86824
DSE	0.87576	0.87496	0.87656
GRNT	0.85604	0.8196	0.89248
LSTMT	0.85116	0.91736	0.78496
CONVD1T	0.77824	0.77152	0.78496
DTT	0.70384	0.69928	0.7084
SVMT	0.83408	0.84336	0.8248
Min	0.70384	0.69928	0.7084
Max	0.87576	0.91736	0.89248

Table 3 shows also the relative testing accuracies as well as the absolute accuracy of each trainer with respect to each type of review. Since there are two classes, one of them will be called the negative class and the other will be called the positive class. ⊖ stands for negative reviews (class) and ⊕ stands for positive reviews (class). Table 3 shows that the lowest accuracies came from the decision tree DTT trainer while the best positive accuracy achieved by GRNT trainer, and the best negative accuracy achieved by LSTMT trainer. This was expected since it is known that the best deep trainer that can handle time series or text streams is the recurrent nets.

To test the proposed approach, we will compare the accuracy of the proposed approach in contrast of the accuracy of the traditional majority voting and a relaxed version of the proposed approach where we will replace the relative accuracy with the absolute accuracy of each trainer. Figure 4 shows the details of the algorithm to apply the majority voting, relative accuracy approach and the absolute accuracy approach. Table 4 represents the div function (diversity matrix) for all trainers. The min difference between DSH and SVMT, and the max difference between DTT and CONVD1T.

The experiments were designed to the approach using variety methods to calculate $p\left({C_j}/{t_i}\right)$. The minimum difference between trainers was between DSH trainer and SVMT trainer and the maximum difference between trainers was between DTT trainer and CONVD1T trainer. This note will be used as an indicator to design values of $p\left({C_j}/{t_i}\right)$.

Table 4. Initial diversity matrix of the individual trainers

Trainer	DSH	DSE	GRNT	LSTMT	CONVD1T	DTT	SVMT
DSH	0	2718.429	3161.526	3518.903	9044.522	7432.308	2622.949
DSE	2718.429	0	2702.554	2668.054	8495.66	7663.706	4125.81
GRNT	3161.526	2702.554	0	3271.968	8924.985	7612.585	4230.079
LSTMT	3518.903	2668.054	3271.968	0	8908.16	7972.727	4703.086
CONVD1T	9044.522	8495.66	8924.985	8908.16	0	12584.69	10260.91
DTT	7432.308	7663.706	7612.585	7972.727	12584.69	0	7546
SVMT	2622.949	4125.81	4230.079	4703.086	10260.91	7546	0

The minimum value of $p\left({}^{C_j}/_{t_i}\right)$ will be 1-12584/25000=0.4967 and the maximum value will be 1-2622/25000=0.896. In the experiments, these values will be relaxed to be values between 0.5 and 1.0. Also, for each trainer the most conflict trainer is be defined as follows: the most conflict for a trainer t_i is the trainer t_j that give the maximum difference in the column of t_i. Table 5 shows the conflict trainers pairs.

Table 5. The diversity teams resulting from applying the diversity conflicting algorithm on the diversity matrix

Trainer	Conflict trainer
CONVD1T	DTT
LSTMT	SVMT
GRNT	DSH

In the experiments, there is one trainer (DSE trainer). This trainer is called the neutral trainer and it was assigned the weight zero for $p\left({}^{C_j}/_{t_i}\right)$ in the experiments. This means that it will be omitted from the voting process. Table 6 summarizes the results of all experiments. The first experiment tests the traditional majority voting method. The second tests the rough relative majority method and the last one tests a relaxed version of the rough relative majority with unified values for all trainers 0.9. Figure 7 represents the results of testing the relaxed relative majority based on the conflict concept.

Table 6. Results of relative accuracies of the optimal values of the relaxing parameter $\propto$

$\propto$	ne_acc_ens	po_acc_ens	acc_ens
0.528	0.88048	0.88496	0.88272
0.504	0.88056	0.88456	0.88256
0.564	0.87912	0.88512	0.88212

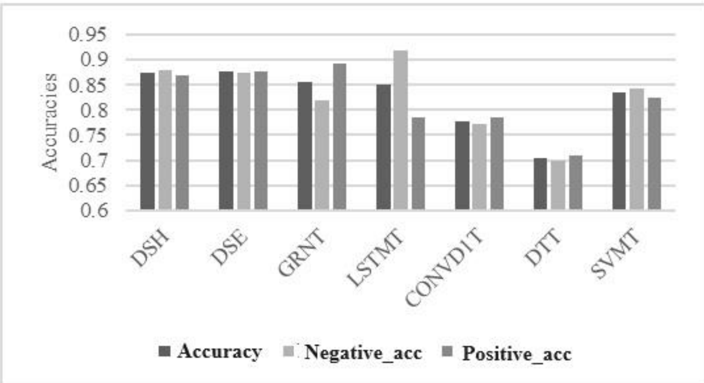

Figure 7. Relative accuracies of the individual trainers with respect to the negative and positive classes

Figure 8 shows the results of relaxed approach using a range of values between [0,1] for the relaxation parameter $\propto$. The worst performance was resulting from values of the relaxed parameter $\propto$ below than 0.2, however, it is still presenting a competing performance to the individual traines especially for values near 0.2. The values between 0.3 and 0.4 of the relaxation parameters ∝ gives identical performance of the three different types of accuracies. Also, the performance is still better than the individual trainers and compete with other ensemble methods. So, to get a balanced performance high performance, keep the value of the relaxing parameter ∝ between 0.3 and 0.4. It is clear that the best value of the relaxation parameter is near 0.5. Based on the experimental results, there is 3 values of ∝ that give the best performance. So, these 3 values are considered to give 3 different relaxed ensemble voting methods. For values greater than 0.5, the performance is still fine and gives acceptable results, however these results are not stable and also are not optimal. This analysis leads us to consider only the values of relaxed parameters that are at the top of each accuracy.

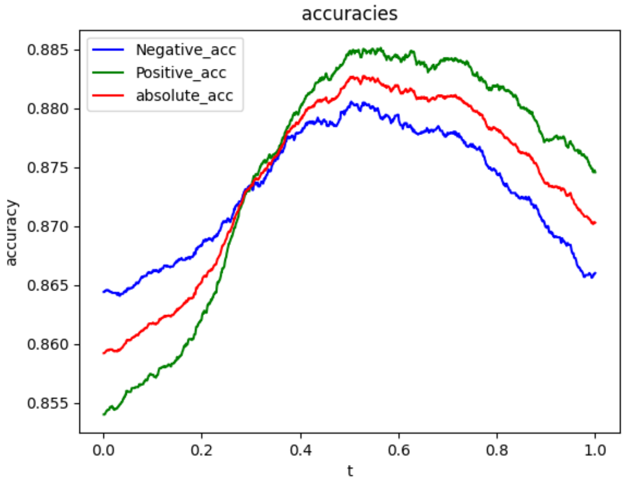

Figure 8. Relaxed ensemble results for different values of the relaxing ∝ in the interval [0,1]

The results confirm the known facts about ensemble learning. Any ensemble learning, regardless of the voting method, is superior for any induvial learner [129]. Also, the experiments proved that there is a limit of the superiority of the ensemble learner that can reach. In the above experiments, it was 90%. The accuracy limit depends on the accuracies of the individual learners. Table 7 shows the diversity matrix of the used voting strategies. It is noted that the rough relative voting strategy, absolute rough voting strategy and the majority voting strategy give identical results since the diversity difference between all of them is zero. However, considering the relaxing parameter ∝ with optimal values not only producing a competing result but also gives different diversity. This diversity is required, especially when different results from ordinary voting methods are required.

Table 7. Diversity matrix for ensemble models

	Relative accuracy	Majority accuracy of ensemble model	Relative absolute accuracy	R1	R2	R3
Relative accuracy	0	0	0	0	0	0
Majority accuracy of ensemble model	0	0	0	0	0	0
Relative absolute accuracy	0	0	0	0	0	0
R1	634	634	634	0	0	0
R2	690	690	690	110	0	0
R3	587	587	587	163	273	0

5. CONCLUSION

In this work, a revision of the ensemble learning methods and its applications in different fields was presented. A deep look at the voting strategies was explored. This work focused on a new method for combining the results of individual learners. The diversity concept was practically defined and based on this definition a proposed voting method was presented. The results of the experiments show that any voting strategy will lead to an ensemble learner that is superior to any individual learner. The diversity matrix of different ensemble learners shows that all ordinary voting strategies will lead to identical ensemble learners. However, the proposed relaxed voting method leads to real different ensemble learners that give different diversity from other ensemble learners based on the different values of the relaxing parameter or different voting strategy.

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 1–15, 2000, doi: 10.1007/3-540-45014-9_1.

[2] E. Yaman and A. Subasi, "Comparison of bagging and boosting ensemble machine learning methods for automated EMG signal classification," *BioMed Research International*, pp. 1–13, Oct. 2019, doi: 10.1155/2019/9152506.

[3] M. Soheili, A. M. E. Moghadam, and M. Dehghan, "Statistical analysis of the performance of rank fusion methods applied to a homogeneous ensemble feature ranking," *Scientific Programming*, pp. 1–14, Sep. 2020, doi: 10.1155/2020/8860044.

[4] C. Valle, F. Saravia, H. Allende, R. Monge, and C. Fernández, "Parallel approach for ensemble learning with locally coupled neural networks," *Neural Processing Letters*, vol. 32, no. 3, pp. 277–291, Dec. 2010, doi: 10.1007/s11063-010-9157-6.

[5] Z. Li, X. Y. Jing, X. Zhu, H. Zhang, B. Xu, and S. Ying, "Heterogeneous defect prediction with two-stage ensemble learning," *Automated Software Engineering*, vol. 26, no. 3, pp. 599–651, Sep. 2019, doi: 10.1007/s10515-019-00259-1.

[6] N. Ostvar and A. M. E. Moghadam, "HDEC: A heterogeneous dynamic ensemble classifier for binary datasets," *Computational Intelligence and Neuroscience*, pp. 1–11, Dec. 2020, doi: 10.1155/2020/8826914.

[7] T. Museba, F. Nelwamondo, K. Ouahada, and A. Akinola, "Recurrent adaptive classifier ensemble for handling recurring concept drifts," *Applied Computational Intelligence and Soft Computing*, pp. 1–13, Jun. 2021, doi: 10.1155/2021/5533777.

[8] D. Zheng, C. Qin, and P. Liu, "Adaptive particle swarm optimization algorithm ensemble model applied to classification of unbalanced data," *Scientific Programming*, pp. 1–13, Oct. 2021, doi: 10.1155/2021/7589756.

[9] J. Zhang, G. Lu, J. Li, and C. Li, "An ensemble classification method for high-dimensional data using neighborhood rough set," *Complexity*, pp. 1–12, Nov. 2021, doi: 10.1155/2021/8358921.

[10] H. Zhang, Y. Song, B. Jiang, B. Chen, and G. Shan, "Two-stage bagging pruning for reducing the ensemble size and improving the classification performance," *Mathematical Problems in Engineering*, pp. 1–17, Jan. 2019, doi: 10.1155/2019/8906034.

[11] J. Zou, X. Fu, L. Guo, C. Ju, and J. Chen, "Creating ensemble classifiers with information entropy diversity measure," *Security and Communication Networks*, pp. 1–11, May 2021, doi: 10.1155/2021/9953509.

[12] H. Chongomweru and A. Kasem, "A novel ensemble method for classification in imbalanced datasets using split balancing technique based on instance hardness (sBal_IH)," *Neural Computing and Applications*, vol. 33, no. 17, pp. 11233–11254, Sep. 2021, doi: 10.1007/s00521-020-05570-7.

[13] Y. Guo, X. Wang, P. Xiao, and X. Xu, "An ensemble learning framework for convolutional neural network based on multiple classifiers," *Soft Computing*, vol. 24, no. 5, pp. 3727–3735, Mar. 2020, doi: 10.1007/s00500-019-04141-w.

[14] M. Jang and S. Cho, "Observational learning algorithm for an ensemble of neural networks," *Pattern Analysis and Applications*, vol. 5, no. 2, pp. 154–167, Jun. 2002, doi: 10.1007/s100440200014.

[15] Y. Kim and J. Kim, "Convex hull ensemble machine for regression and classification," *Knowledge and Information Systems*, vol. 6, no. 6, pp. 645–663, Nov. 2004, doi: 10.1007/s10115-003-0116-7.

[16] S. B. Kotsiantis, "An incremental ensemble of classifiers," *Artificial Intelligence Review*, vol. 36, no. 4, pp. 249–266, Dec. 2011, doi: 10.1007/s10462-011-9211-4.

[17] V. Kumar and S. Minz, "Multi-view ensemble learning: an optimal feature set partitioning for high-dimensional data classification," *Knowledge and Information Systems*, vol. 49, no. 1, pp. 1–59, Oct. 2016, doi: 10.1007/s10115-015-0875-y.

[18] H. Liu and M. Cocea, "Nature-inspired framework of ensemble learning for collaborative classification in granular computing context," *Granular Computing*, vol. 4, no. 4, pp. 715–724, Oct. 2019, doi: 10.1007/s41066-018-0122-5.

[19] A. Narassiguin, M. Bibimoune, H. Elghazel, and A. Aussem, "An extensive empirical comparison of ensemble learning methods for binary classification," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 1093–1128, Nov. 2016, doi: 10.1007/s10044-016-0553-z.

[20] S. Priya and R. A. Uthra, "Retraction Note to: Comprehensive analysis for class imbalance data with concept drift using ensemble based classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 283–283, Apr. 2023, doi: 10.1007/s12652-022-04066-7.

[21] A. L. Prodromidis and S. J. Stolfo, "Cost complexity-based pruning of ensemble classifiers," *Knowledge and Information Systems*, vol. 3, no. 4, pp. 449–469, Nov. 2001, doi: 10.1007/pl00011678.

[22] S. Qiao *et al*., "LMNNB: Two-in-One imbalanced classification approach by combining metric learning and ensemble learning," *Applied Intelligence*, vol. 52, no. 7, pp. 7870–7889, May 2022, doi: 10.1007/s10489-021-02901-6.

[23] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.

[24] B. Verma and S. Z. Hassan, "Hybrid ensemble approach for classification," *Applied Intelligence*, vol. 34, no. 2, pp. 258–278, Apr. 2011, doi: 10.1007/s10489-009-0194-7.

[25] X. Zeng and T. R. Martinez, "Using a neural network to approximate an ensemble of classifiers," *Neural Processing Letters*, vol. 12, no. 3, pp. 225–237, 2000, doi: 10.1023/A:1026530200837.

[26] Z. H. Zhou, "When semi-supervised learning meets ensemble learning," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 1, pp. 6–16, Mar. 2011, doi: 10.1007/s11460-011-0126-2.

[27] G. Teng, C. He, J. Xiao, Y. He, B. Zhu, and X. Jiang, "Cluster ensemble framework based on the group method of data handling," *Applied Soft Computing*, vol. 43, pp. 35–46, Jun. 2016, doi: 10.1016/j.asoc.2016.01.043.

[28] W. Hua and L. Mo, "Clustering ensemble model based on self-organizing map network," *Computational Intelligence and Neuroscience*, pp. 1–11, Aug. 2020, doi: 10.1155/2020/2971565.

[29] J. Yan and W. Liu, "An ensemble clustering approach (consensus clustering) for high-dimensional data," *Security and Communication Networks*, pp. 1–9, May 2022, doi: 10.1155/2022/5629710.

[30] X. Huang, F. Qin, and L. Lin, "The Core cluster-based subspace weighted clustering ensemble," *Wireless Communications and Mobile Computing*, pp. 1–17, May 2022, doi: 10.1155/2022/7990969.

[31] Z. Zhang *et al*., "Research on PMF model based on BP neural network ensemble learning bagging and fuzzy clustering," *Complexity*, pp. 1–9, Jul. 2021, doi: 10.1155/2021/9985894.

[32] M. Koohzadi, N. M. Charkari, and F. Ghaderi, "Unsupervised representation learning based on the deep multi-view ensemble learning," *Applied Intelligence*, vol. 50, no. 2, pp. 562–581, Feb. 2020, doi: 10.1007/s10489-019-01526-0.

[33] M. Uchida, Y. Maehara, and H. Shioya, "Unsupervised weight parameter estimation method for ensemble learning," *Journal of*

Mathematical Modelling and Algorithms, vol. 10, no. 4, pp. 307–322, Dec. 2011, doi: 10.1007/s10852-011-9157-1.
[34] M.-L. Zhang and Z.-H. Zhou, "Exploiting unlabeled data to enhance ensemble diversity," *Data Mining and Knowledge Discovery*, vol. 26, no. 1, pp. 98–129, Jan. 2013, doi: 10.1007/s10618-011-0243-9.
[35] B. Mevik, V. H. Segtnan, and T. Næs, "Ensemble methods and partial least squares regression," *Journal of Chemometrics*, vol. 18, no. 11, pp. 498–507, Nov. 2004, doi: 10.1002/cem.895.
[36] M. Ruiz-Abellón, A. Gabaldón, and A. Guillamón, "Load forecasting for a campus university using ensemble methods based on regression trees," *Energies*, vol. 11, no. 8, pp. 2038–2050, Aug. 2018, doi: 10.3390/en11082038.
[37] N. Moniz, P. Branco, L. Torgo, and B. Krawczyk, "Evaluation of ensemble methods in imbalanced regression tasks," *Proceedings of Machine Learning Research*, vol. 74, pp. 129–141, 2017.
[38] S. Liu *et al.*, "Microcluster-based incremental ensemble learning for noisy, nonstationary data streams," *Complexity*, pp. 1–12, May 2020, doi: 10.1155/2020/6147378.
[39] A. Cano and B. Krawczyk, "ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams," *Machine Learning*, vol. 111, no. 7, pp. 2561–2599, Jul. 2022, doi: 10.1007/s10994-022-06168-x.
[40] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The online performance estimation framework: heterogeneous ensemble learning for data streams," *Machine Learning*, vol. 107, no. 1, pp. 149–176, Jan. 2018, doi: 10.1007/s10994-017-5686-9.
[41] Y. Sun and H. Dai, "Constructing accuracy and diversity ensemble using Pareto-based multi-objective learning for evolving data streams," *Neural Computing and Applications*, vol. 33, no. 11, pp. 6119–6132, Jun. 2021, doi: 10.1007/s00521-020-05386-5.
[42] G. Zhu and Q. Dai, "EnsPKDE&IncLKDE: a hybrid time series prediction algorithm integrating dynamic ensemble pruning, incremental learning, and kernel density estimation," *Applied Intelligence*, vol. 51, no. 2, pp. 617–645, Feb. 2021, doi: 10.1007/s10489-020-01802-4.
[43] S. A. Althubiti, S. Paul, R. Mohanty, S. N. Mohanty, F. Alenezi, and K. Polat, "Ensemble learning framework with GLCM texture extraction for early detection of lung cancer on CT images," *Computational and Mathematical Methods in Medicine*, pp. 1–14, Jun. 2022, doi: 10.1155/2022/2733965.
[44] M. Nilashi *et al.*, "Predicting parkinson's disease progression: evaluation of ensemble methods in machine learning," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–17, Feb. 2022, doi: 10.1155/2022/2793361.
[45] S. Sheikhi and M. T. Kheirabadi, "An efficient rotation forest-based ensemble approach for predicting severity of Parkinson's disease," *Journal of Healthcare Engineering*, pp. 1–9, Jun. 2022, doi: 10.1155/2022/5524852.
[46] J. Zhang, K. Xia, Z. He, Z. Yin, and S. Wang, "Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection," *Mathematical Problems in Engineering*, pp. 1–18, Feb. 2021, doi: 10.1155/2021/6622935.
[47] X. Y. Gao, A. Amin Ali, H. Shaban Hassan, and E. M. Anwar, "Improving the accuracy for analyzing heart diseases prediction based on the ensemble method," *Complexity*, pp. 1–10, Feb. 2021, doi: 10.1155/2021/6663455.
[48] G. Jinfeng, S. Qummar, Z. Junming, Y. Ruxian, and F. G. Khan, "Ensemble framework of deep CNNs for diabetic retinopathy detection," *Computational Intelligence and Neuroscience*, pp. 1–11, Dec. 2020, doi: 10.1155/2020/8864698.
[49] T. R. Mahesh *et al.*, "AdaBoost ensemble methods using k-fold cross validation for survivability with the early detection of heart disease," *Computational Intelligence and Neuroscience*, pp. 1–11, Apr. 2022, doi: 10.1155/2022/9005278.
[50] S. N. H. Bukhari *et al.*, "Machine learning-based ensemble model for zika virus t-cell epitope prediction," *Journal of Healthcare Engineering*, pp. 1–10, Oct. 2021, doi: 10.1155/2021/9591670.
[51] B. Sabeena, S. Sivakumari, and D. M. Teressa, "Optimization-based ensemble feature selection algorithm and deep learning classifier for Parkinson's disease," *Journal of Healthcare Engineering*, pp. 1–12, Apr. 2022, doi: 10.1155/2022/1487212.
[52] M. Saberi Anari, "A hybrid model for leaf diseases classification based on the modified deep transfer learning and ensemble approach for agricultural AIoT-based monitoring," *Computational Intelligence and Neuroscience*, pp. 1–15, Apr. 2022, doi: 10.1155/2022/6504616.
[53] M. Masud *et al.*, "A Pneumonia diagnosis scheme based on hybrid features extracted from chest radiographs using an ensemble learning algorithm," *Journal of Healthcare Engineering*, pp. 1–11, Feb. 2021, doi: 10.1155/2021/8862089.
[54] Z. Tao, H. Bing-Qiang, L. Huiling, S. Hongbin, Y. Pengfei, and D. Hongsheng, "18F-FDG-PET/CT whole-body imaging lung tumor diagnostic model: An ensemble E-ResNet-NRC with divided sample space," *BioMed Research International*, pp. 1–13, Apr. 2021, doi: 10.1155/2021/8865237.
[55] F. Ahmad, A. Farooq, and M. U. Ghani, "Deep ensemble model for classification of novel coronavirus in chest X-Ray images," *Computational Intelligence and Neuroscience*, pp. 1–17, Jan. 2021, doi: 10.1155/2021/8890226.
[56] M. Oloko-Oba and S. Viriri, "Ensemble of efficientnets for the diagnosis of tuberculosis," *Computational Intelligence and Neuroscience*, pp. 1–12, Dec. 2021, doi: 10.1155/2021/9790894.
[57] J. Wang *et al.*, "EDLMFC: an ensemble deep learning framework with multi-scale features combination for ncRNA–protein interaction prediction," *BMC Bioinformatics*, vol. 22, no. 1, pp. 1–19, Mar. 2021, doi: 10.1186/s12859-021-04069-9.
[58] A. Stanescu and D. Caragea, "An empirical study of ensemble-based semi-supervised learning approaches for imbalanced splice site datasets," *BMC Systems Biology*, vol. 9, no. 5, pp. 1–12, 2015, doi: 10.1186/1752-0509-9-S5-S1.
[59] D. Rani, N. S. Gill, P. Gulia, and J. M. Chatterjee, "An ensemble-based multiclass classifier for intrusion detection using internet of things," *Computational Intelligence and Neuroscience*, pp. 1–12, May 2022, doi: 10.1155/2022/1668676.
[60] Z. Wang, J. Liu, and L. Sun, "EFS-DNN: An ensemble feature selection-based deep learning approach to network intrusion detection system," *Security and Communication Networks*, pp. 1–14, Apr. 2022, doi: 10.1155/2022/2693948.
[61] W. Lian, G. Nie, B. Jia, D. Shi, Q. Fan, and Y. Liang, "An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning," *Mathematical Problems in Engineering*, pp. 1–15, Nov. 2020, doi: 10.1155/2020/2835023.
[62] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets," *Security and Communication Networks*, pp. 1–9, Jan. 2020, doi: 10.1155/2020/4586875.
[63] X. Yang, Y. Chen, X. Qian, T. Li, and X. Lv, "BCEAD: a blockchain-empowered ensemble anomaly detection for wireless sensor network via isolation forest," *Security and Communication Networks*, pp. 1–10, Nov. 2021, doi: 10.1155/2021/9430132.
[64] K. Sundararajan and A. Palanisamy, "Multi-rule based ensemble feature selection model for sarcasm type detection in Twitter," *Computational Intelligence and Neuroscience*, pp. 1–17, Jan. 2020, doi: 10.1155/2020/2860479.
[65] A. Hansrajh, T. T. Adeliyi, and J. Wing, "Detection of online fake news using blending ensemble learning," *Scientific Programming*, pp. 1–10, Jul. 2021, doi: 10.1155/2021/3434458.
[66] L. Yu, Y. Wu, J. Yang, and Y. Zhang, "Bullet subtitle sentiment classification based on affective computing and ensemble learning," *Wireless Communications and Mobile Computing*, pp. 1–9, Jun. 2021, doi: 10.1155/2021/5563104.
[67] A. Al-Hashedi *et al.*, "Ensemble classifiers for arabic sentiment analysis of social network (twitter data) towards COVID-19-related conspiracy theories," *Applied Computational Intelligence and Soft Computing*, pp. 1–10, Jan. 2022, doi: 10.1155/2022/6614730.

[68] M. M. V. Chalapathi, M. R. Kumar, N. Sharma, and S. Shitharth, "Ensemble learning by high-dimensional acoustic features for emotion recognition from speech audio signal," *Security and Communication Networks*, pp. 1–10, Feb. 2022, doi: 10.1155/2022/8777026.

[69] M. Fayaz, A. Khan, J. U. Rahman, A. Alharbi, M. I. Uddin, and B. Alouffi, "Ensemble machine learning model for classification of spam product reviews," *Complexity*, pp. 1–10, Dec. 2020, doi: 10.1155/2020/8857570.

[70] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8885861.

[71] Y. Chen, R. Chang, and J. Guo, "Emotion recognition of EEG signals based on the ensemble learning method: AdaBoost," *Mathematical Problems in Engineering*, pp. 1–12, Jan. 2021, doi: 10.1155/2021/8896062.

[72] X. Yu *et al.*, "Deep ensemble learning for human action recognition in still images," *Complexity*, pp. 1–23, Jan. 2020, doi: 10.1155/2020/9428612.

[73] M. H. Javed, Z. Yu, T. Li, T. M. Rajeh, F. Rafique, and S. Waqar, "Hybrid two-stream dynamic CNN for view adaptive human action recognition using ensemble learning," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 4, pp. 1157–1166, Apr. 2022, doi: 10.1007/s13042-021-01441-2.

[74] P. Wang and Z. Xu, "A novel consumer purchase behavior recognition method using ensemble learning algorithm," *Mathematical Problems in Engineering*, pp. 1–10, Dec. 2020, doi: 10.1155/2020/6673535.

[75] P. Wang, X. Li, Z. Qin, Y. Qu, and Z. Zhang, "Stock price forecasting based on wavelet filtering and ensembled machine learning model," *Mathematical Problems in Engineering*, pp. 1–12, Jun. 2022, doi: 10.1155/2022/4024953.

[76] G. Xing, S. Sun, and J. Guo, "A new decomposition ensemble learning approach with intelligent optimization for PM2.5 concentration forecasting," *Discrete Dynamics in Nature and Society*, pp. 1–11, Mar. 2020, doi: 10.1155/2020/6019826.

[77] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, Sep. 2019, doi: 10.1002/for.2585.

[78] P. Yazdanian and S. Sharifian, "E2LG: a multiscale ensemble of LSTM/GAN deep learning architecture for multistep-ahead cloud workload prediction," *Journal of Supercomputing*, vol. 77, no. 10, pp. 11052–11082, Oct. 2021, doi: 10.1007/s11227-021-03723-6.

[79] Y. Xie, A. Li, L. Gao, and Z. Liu, "A heterogeneous ensemble learning model based on data distribution for credit card fraud detection," *Wireless Communications and Mobile Computing*, pp. 1–13, Jul. 2021, doi: 10.1155/2021/2531210.

[80] S. R. Lenka, S. K. Bisoy, R. Priyadarshini, and M. Sain, "Empirical analysis of ensemble learning for imbalanced credit scoring datasets: a systematic review," *Wireless Communications and Mobile Computing*, pp. 1–18, Jun. 2022, doi: 10.1155/2022/6584352.

[81] D. Xu, X. Zhang, J. Hu, and J. Chen, "A novel ensemble credit scoring model based on extreme learning machine and generalized fuzzy soft sets," *Mathematical Problems in Engineering*, pp. 1–12, Jun. 2020, doi: 10.1155/2020/7504764.

[82] R. Han, P. Liu, G. Wang, H. Zhang, and X. Wu, "Advantage of combining ObiA and classifier ensemble method for very high-resolution satellite imagery classification," *Journal of Sensors*, pp. 1–15, Nov. 2020, doi: 10.1155/2020/8855509.

[83] S. Bakkali, Z. Ming, M. Coustaty, and M. Rusiñol, "EAML: ensemble self-attention-based mutual learning network for document image classification," *International Journal on Document Analysis and Recognition*, vol. 24, no. 3, pp. 251–268, 2021, doi: 10.1007/s10032-021-00378-0.

[84] G. Farias, E. Fabregas, I. Martínez, J. Vega, S. Dormido-Canto, and H. Vargas, "Nuclear fusion pattern recognition by ensemble learning," *Complexity*, pp. 1–9, Jun. 2021, doi: 10.1155/2021/1207167.

[85] C. Zhao, Z. Lin, J. Tan, H. Hu, and Q. Li, "A new transfer learning ensemble model with new training methods for gear wear particle recognition," *Shock and Vibration*, pp. 1–10, Jan. 2022, doi: 10.1155/2022/3696091.

[86] S. Zhao, Y. Zhang, H. Xu, and T. Han, "Ensemble classification based on feature selection for environmental sound recognition," *Mathematical Problems in Engineering*, pp. 1–7, Feb. 2019, doi: 10.1155/2019/4318463.

[87] H. Tian, M. Shuai, K. Li, and X. Peng, "An Incremental Learning Ensemble Strategy for Industrial Process Soft Sensors," *Complexity*, pp. 1–12, May 2019, doi: 10.1155/2019/5353296.

[88] Y. Zhou, J. Wang, and Z. Wang, "Multisensor-based heavy machine faulty identification using sparse autoencoder-based feature fusion and deep belief network-based ensemble learning," *Journal of Sensors*, pp. 1–26, Jun. 2022, doi: 10.1155/2022/5796505.

[89] H. Babajanian Bisheh, G. Ghodrati Amiri, and E. Darvishan, "Ensemble classifiers and feature-based methods for structural damage assessment," *Shock and Vibration*, pp. 1–14, Dec. 2020, doi: 10.1155/2020/8899487.

[90] S. M. T. U. Raju *et al.*, "An approach for demand forecasting in steel industries using ensemble learning," *Complexity*, pp. 1–19, Feb. 2022, doi: 10.1155/2022/9928836.

[91] D. R. Wijaya, F. Afianti, A. Arifianto, D. Rahmawati, and V. S. Kodogiannis, "Ensemble machine learning approach for electronic nose signal processing," *Sensing and Bio-Sensing Research*, vol. 36, pp. 1–11, Jun. 2022, doi: 10.1016/j.sbsr.2022.100495.

[92] A. Hamdi and H. Frigui, "Ensemble hidden Markov models with application to landmine detection," *Eurasip Journal on Advances in Signal Processing*, no. 1, pp. 1–15, Dec. 2015, doi: 10.1186/s13634-015-0260-8.

[93] S. Kumar, J. Singh, and O. Singh, "Ensemble-based extreme learning machine model for occupancy detection with ambient attributes," *International Journal of System Assurance Engineering and Management*, vol. 11, pp. 173–183, Jul. 2020, doi: 10.1007/s13198-019-00935-1.

[94] Q. Ruan, Q. Wu, Y. Wang, X. Liu, and F. Miao, "Effective learning model of user classification based on ensemble learning algorithms," *Computing*, vol. 101, no. 6, pp. 531–545, Jun. 2019, doi: 10.1007/s00607-018-0688-4.

[95] C. Guo and Z. Li, "Automatic rock classification algorithm based on ensemble residual network and merged region extraction," *Advances in Multimedia*, pp. 1–11, Mar. 2022, doi: 10.1155/2022/3982892.

[96] W. Li, Z. Fang, and Y. Wang, "Stacking ensemble of deep learning methods for landslide susceptibility mapping in the Three Gorges Reservoir area, China," *Stochastic Environmental Research and Risk Assessment*, vol. 36, no. 8, pp. 2207–2228, Aug. 2022, doi: 10.1007/s00477-021-02032-x.

[97] Z. Liang, C. Wang, and K. U. J. Khan, "Application and comparison of different ensemble learning machines combining with a novel sampling strategy for shallow landslide susceptibility mapping," *Stochastic Environmental Research and Risk Assessment*, vol. 35, no. 6, pp. 1243–1256, Jun. 2021, doi: 10.1007/s00477-020-01893-y.

[98] X. Yin, Q. Liu, Y. Pan, X. Huang, J. Wu, and X. Wang, "Strength of stacking technique of ensemble learning in rockburst prediction with imbalanced data: comparison of eight single and ensemble models," *Natural Resources Research*, vol. 30, no. 2, pp. 1795–1815, Apr. 2021, doi: 10.1007/s11053-020-09787-0.

[99] A. Khatri, S. Agrawal, and J. M. Chatterjee, "Wheat seed classification: utilizing ensemble machine learning approach," *Scientific Programming*, pp. 1–9, Feb. 2022, doi: 10.1155/2022/2626868.

[100] H. Li, Y. Jin, J. Zhong, and R. Zhao, "A fruit tree disease diagnosis model based on stacking ensemble learning," *Complexity*, pp. 1–12, Sep. 2021, doi: 10.1155/2021/6868592.

[101] P. Zhang, J. Meng, Y. Luan, and C. Liu, "Plant miRNA–lncRNA interaction prediction with the ensemble of CNN and IndRNN," *Interdisciplinary Sciences – Computational Life Sciences*, vol. 12, no. 1, pp. 82–89, Mar. 2020, doi: 10.1007/s12539-019-00351-w.

[102] L. Zhang, G. Li, X. Li, H. Wang, S. Chen, and H. Liu, "EDLm6APred: ensemble deep learning approach for mRNA m6A site prediction," *BMC Bioinformatics*, vol. 22, no. 1, pp. 1–15, Dec. 2021, doi: 10.1186/s12859-021-04206-4.

[103] V. M. Krasnopolsky, M. S. Fox-Rabinovitz, and A. A. Belochitski, "Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model," *Advances in Artificial Neural Systems*, pp. 1–13, May 2013, doi: 10.1155/2013/485913.

[104] H. Singh, M. R. Najafi, and A. Cannon, "Evaluation and joint projection of temperature and precipitation extremes across Canada based on hierarchical Bayesian modelling and large ensembles of regional climate simulations," *Weather and Climate Extremes*, vol. 36, p. 100443, Jun. 2022, doi: 10.1016/j.wace.2022.100443.

[105] K. Dhibi, M. Mansouri, K. Bouzrara, H. Nounou, and M. Nounou, "Reduced neural network based ensemble approach for fault detection and diagnosis of wind energy converter systems," *Renewable Energy*, vol. 194, pp. 778–787, Jul. 2022, doi: 10.1016/j.renene.2022.05.082.

[106] A. Rasaizadi and S. Seyedabrishami, "Stacking ensemble learning process to predict rural road traffic flow," *Journal of Advanced Transportation*, pp. 1–12, Jun. 2022, doi: 10.1155/2022/3198636.

[107] Y. Rui, W. Lu, Z. Yi, R. Wu, and B. Ran, "A novel hybrid model for predicting traffic flow via improved ensemble learning combined with deep belief networks," *Mathematical Problems in Engineering*, pp. 1–16, Oct. 2021, doi: 10.1155/2021/7328056.

[108] P. Yildirim, U. K. Birant, D. Birant, and M. H. Y. Moghaddam, "EBOC: ensemble-based ordinal classification in transportation," *Journal of Advanced Transportation*, pp. 1–17, Mar. 2019, doi: 10.1155/2019/7482138.

[109] L. K. Smirani, H. A. Yamani, L. J. Menzli, and J. A. Boulahia, "Using ensemble learning algorithms to predict student failure and enabling customized educational paths," *Scientific Programming*, pp. 1–15, Apr. 2022, doi: 10.1155/2022/3805235.

[110] X. Liu and M. Truszczynski, "Voting-based ensemble learning for partial lexicographic preference forests over combinatorial domains," *Annals of Mathematics and Artificial Intelligence*, vol. 87, no. 1–2, pp. 137–155, Oct. 2019, doi: 10.1007/s10472-019-09645-7.

[111] J. Y. Araz and M. Spannowsky, "Combine and conquer: event reconstruction with bayesian ensemble neural networks," *Journal of High Energy Physics*, no. 4, pp. 1–23, Apr. 2021, doi: 10.1007/JHEP04(2021)296.

[112] M. S. Suchithra and M. L. Pai, "Evaluating the performance of bagging-based k-nearest neighbor ensemble with the voting rule selection method," *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 20741–20762, Jun. 2022, doi: 10.1007/s11042-022-12716-3.

[113] H. Kim, H. Kim, H. Moon, and H. Ahn, "A weight-adjusted voting algorithm for ensembles of classifiers," *Journal of the Korean Statistical Society*, vol. 40, no. 4, pp. 437–449, Dec. 2011, doi: 10.1016/j.jkss.2011.03.002.

[114] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles," *Knowledge and Information Systems*, vol. 38, no. 2, pp. 259–275, Feb. 2014, doi: 10.1007/s10115-012-0586-6.

[115] Y. Zhang, B. Liu, J. Cai, and S. Zhang, "Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution," *Neural Computing and Applications*, vol. 28, no. S1, pp. 259–267, Dec. 2017, doi: 10.1007/s00521-016-2342-4.

[116] M. R. Smith and T. Martinez, "The robustness of majority voting compared to filtering misclassified instances in supervised classification tasks," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 105–130, Jan. 2018, doi: 10.1007/s10462-016-9518-2.

[117] C. Cornelio, M. Donini, A. Loreggia, M. S. Pini, and F. Rossi, "Voting with random classifiers (VORACE): theoretical and experimental analysis," *Autonomous Agents and Multi-Agent Systems*, vol. 35, no. 2, pp. 1–31, Oct. 2021, doi: 10.1007/s10458-021-09504-y.

[118] R. Delgado, "A semi-hard voting combiner scheme to ensemble multi-class probabilistic classifiers," *Applied Intelligence*, vol. 52, no. 4, pp. 3653–3677, Mar. 2022, doi: 10.1007/s10489-021-02447-7.

[119] A. Rojarath and W. Songpan, "Cost-sensitive probability for weighted voting in an ensemble model for multi-class classification problems," *Applied Intelligence*, vol. 51, no. 7, pp. 4908–4932, Jul. 2021, doi: 10.1007/s10489-020-02106-3.

[120] M. Sultan Zia, M. Hussain, and M. Arfan Jaffar, "A novel spontaneous facial expression recognition using dynamically weighted majority voting based ensemble classifier," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25537–25567, Oct. 2018, doi: 10.1007/s11042-018-5806-y.

[121] Y. Li and Y. Luo, "Performance-weighted-voting model: an ensemble machine learning method for cancer type classification using whole-exome sequencing mutation," *Quantitative Biology*, vol. 8, no. 4, pp. 347–358, Dec. 2020, doi: 10.1007/s40484-020-0226-1.

[122] E. Tasci, "Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition," *Multimedia Tools and Applications*, vol. 79, no. 41–42, pp. 30397–30418, Nov. 2020, doi: 10.1007/s11042-020-09486-1.

[123] S. Madichetty and M. Sridevi, "Identification of medical resource tweets using Majority Voting-based Ensemble during disaster," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–18, Dec. 2020, doi: 10.1007/s13278-020-00679-y.

[124] T. A. Pham and H.-L. T. Vu, "Application of ensemble learning using weight voting protocol in the prediction of pile bearing capacity," *Mathematical Problems in Engineering*, pp. 1–14, Jul. 2021, doi: 10.1155/2021/5558449.

[125] R. Ahuja and S. C. Sharma, "Stacking and voting ensemble methods fusion to evaluate instructor performance in higher education," *International Journal of Information Technology (Singapore)*, vol. 13, no. 5, pp. 1721–1731, Oct. 2021, doi: 10.1007/s41870-021-00729-4.

[126] D. Balamurugan, S. S. Aravinth, P. C. S. Reddy, A. Rupani, and A. Manikandan, "Multiview objects recognition using deep learning-based wrap-CNN with voting scheme," *Neural Processing Letters*, vol. 54, no. 3, pp. 1495–1521, Jun. 2022, doi: 10.1007/s11063-021-10679-4.

[127] M. U. Salur and İ. Aydın, "A soft voting ensemble learning-based approach for multimodal sentiment analysis," *Neural Computing and Applications*, vol. 34, no. 21, pp. 18391–18406, Nov. 2022, doi: 10.1007/s00521-022-07451-7.

[128] U. Zahoora, M. Rajarajan, Z. Pan, and A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier," *Applied Intelligence*, vol. 52, no. 12, pp. 13941–13960, Sep. 2022, doi: 10.1007/s10489-022-03244-6.

[129] R. Polikar, "Ensemble Learning," in *Ensemble Machine Learning*, New York, NY: Springer New York, 2012, pp. 1–34. doi: 10.1007/978-1-4419-9326-7_1.

[130] C. Xu, W. Chang, and W. Liu, "Data-driven decision model based on local two-stage weighted ensemble learning," *Annals of Operations Research*, vol. 325, no. 2, pp. 995–1028, Jun. 2023, doi: 10.1007/s10479-022-04599-2.

BIOGRAPHIES OF AUTHORS

Mahmoud B. Rokaya received the Bachelor of Science degree in Mathematics from Tanta University, Egypt, 1997, and the Master of Science degree in Operations Research, Tanta University, Egypt. He received the Doctor of Engineering degree in Information Science and System Engineering Design from High School of Engineering, Tokushima University, Japan. Currently He works as Associate Professor Information Technology Department, Taif University, Saudi Arabia. His research interests include information management, artificial intelligence, information retrieval, and natural language processing. He can be contacted at email: mahmoudrokaya@tu.edu.sa.

Kholod D. Alsufiani holds a BSc degree with honors in Computer Science from Taif University, followed by an MSc degree in Computer Science from Middlesex University in London, which she obtained in 2013. In 2020, she successfully completed her PhD in Computer Science from Middlesex University. Currently serving as an Assistant Professor and Department Head of the Mathematics Department at Turaba University College-Taif university. Kholod's research interests are centered around the fields of information visualization, human-centered artificial intelligence, human-centered design, and artificial intelligence. She can be contacted at email: K.kholod@tu.edu.sa.