



# Cost-sensitive probability for weighted voting in an ensemble model for multi-class classification problems

Artittayapron Rojarath<sup>1</sup> · Wararat Songpan<sup>1</sup>

Accepted: 30 November 2020 / Published online: 6 January 2021  
© The Author(s) 2021

## Abstract

Ensemble learning is an algorithm that utilizes various types of classification models. This algorithm can enhance the prediction efficiency of component models. However, the efficiency of combining models typically depends on the diversity and accuracy of the predicted results of ensemble models. However, the problem of multi-class data is still encountered. In the proposed approach, cost-sensitive learning was implemented to evaluate the prediction accuracy for each class, which was used to construct a cost-sensitivity matrix of the true positive (TP) rate. This TP rate can be used as a weight value and combined with a probability value to drive ensemble learning for a specified class. We proposed an ensemble model, which was a type of heterogenous model, namely, a combination of various individual classification models (support vector machine, Bayes, K-nearest neighbour, naïve Bayes, decision tree, and multi-layer perceptron) in experiments on 3-, 4-, 5- and 6-classifier models. The efficiencies of the propose models were compared to those of the individual classifier model and homogenous models (Adaboost, bagging, stacking, voting, random forest, and random subspaces) with various multi-class data sets. The experimental results demonstrate that the cost-sensitive probability for the weighted voting ensemble model that was derived from 3 models provided the most accurate results for the dataset in multi-class prediction. The objective of this study was to increase the efficiency of predicting classification results in multi-class classification tasks and to improve the classification results.

**Keywords** Ensemble learning · Multi-class data · Cost-sensitive learning · True positive

## 1 Introduction

Currently, the problem of Multiclass classification with decision making processes is a fundamental problem in supervised learning which is an important problem in classify results by multiple label-classes [1–3]. For example, the works [4–6] applied to real-life situations such as text categorization, entity recognition, and disease diagnose respectively. Therefore, finding an appropriate method or strategy to solve the multi-class classification problem is important. However, it is difficult to solve multi-class classification problems in order to the scope of a decision for a problem is the decision-making process probably be more complex than the problems of binary classifications [7]. The decision-making process [4, 8–10] is another problem that has been studied in this research. Part of the decision to classify the result class of the data set. There is

often a problem with multi-class dataset data due to improper classification of results due to the large collection and distribution of class results. From the increasing number of multi-class data sets Therefore, new machine learning algorithms [11–13] need to be developed to improve the efficiency of the results class prediction and find that there are multiple learning models that can be used to solve the same problem [14–16]. Ensemble learning is a machine learning process to improve the efficiency of predictions [17–19], [20] using a strategy to combine multiple learning predictions and helps to reduce the problem of inappropriate model selection by combining all models. This method is popular and widely used to improve performance than individual models [13, 21, 22]. According to the effectiveness of the ensemble learning method, it is necessary to create Creating a new ensemble model to improve the model's accuracy and stability [16, 20, 23]. The main challenges of creating a new ensemble model are how to combine strategy and methods. Determine the assigning weight probability, which is the method preceding the final class result classification process for multi-class data. In aspect of designing an efficient new ensemble model method [24–26], there are 4 key points to be considered: dataset, based

✉ Wararat Songpan  
wararat@kku.ac.th

<sup>1</sup> Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand

model, combination strategy and method of assigning probability weight. According to previous studies [16, 27, 28] the researcher tends to not pay much attention to multi-class data sets due to the multi-class classification. There is a complex decision-making class classification results that are difficult to manage [29]. To deal with this important problem, there are 2 main methods that are used to create classification methods for multi-class data: The traditional base model method and the ensemble model method. At the traditional base model method [30–40], the model used to classify the resultant class, such as classifying the resultant class. The nature of the decision tree, which is the decision tree method [41], is used to predict the pattern recognition class of the individual base model. [5, 42–44] Accuracy depends on the factors of the prediction of the result of class [45]. Although the traditional base model method can classify the result class in the case of multi-class data sets, it has difficulty in the decision of classifying the correct results class without information bias is greater than the ensemble model. Therefore, the ensemble learning model [46], is considered to be more appropriate for managing Multi-class classification, complex results in which they can improve the classification performance by using the combination method (multiple classifier) in machine learning [47–49]. Decision-making, the method of assigning weight (assigning weight probability to class result) is one of the challenges for classifying new result classes in multi-class data. [50–52] Therefore, ensemble learning methodology has been considered for reducing class bias occurrence from classification results [4, 53]. Maintaining good classification performance comes from combining result classes. By determining the appropriate probability weight and obtaining a higher classification accuracy for multi-class data [54, 55]. The latest techniques for the combination method are selection base model and assigning method. Many studies [40, 56–58] attempt to optimize the probability weight obtained from the prediction of the class results from the ensemble model. It focuses on the combination strategy process, which is a combination of weighted voting methods, which is a method for assigning appropriate weight values to classifications that do not affect the effectiveness of the model and the accuracy of the classification is derived from the improvement of assigning weight. In this step, the true positive value of the prediction of the resulting class increases and decreases. The misclassified data sets of multi-class data are compared to the base classification model and traditional ensemble model [42, 50, 59]. Another important problem is the lack of complete training samples in the data set, which causes training data insufficient data and the difficulty of using a combination strategy to combine models to create good classification methods for multi-class data. Therefore, recent work focuses on supervised machine learning methods, including cost-sensitive methods. Approach [57] in order to overcome the decision problem of classifying the appropriate outcomes. Most jobs that provide

high accuracy (Measure the efficiency of the model by confusion matrix methods including Precision, Recall, F-measure, G-mean, and Accuracy). Only for multi-class data classification that has more than 2 result classes, the accuracy resulting from each the range of the number of classes. The result shows that even if the dataset has more classes, which should not decrease the accuracy. While evaluating performance using the confusion matrix method of the base classification model and traditional ensemble model, the performance evaluation is lower than the new ensemble model. This is possible because of the information bias. The selection of the result class from a single pattern without randomization appropriate randomization results in incorrect results class predictions due to incorrect probability weight assignments. [5, 60, 61] Current work is trying to improve the efficiency of the method combination by setting cost new cost-sensitive weight (new assigning weight) for more accurate probability weight of class result.

In our research, we present a novel supervised cost-sensitive weighted based on Ensemble learning classification. For process resolution decision-making of multi-class classification. The framework that we have proposed is cost-sensitive probability weighted based on ensemble learning, which differs from the initial classification work [42, 62] that focuses on the pattern recognition of The classification of an individual base model that does not take into account the predisposition of the class result data obtained from predictions from the model may lead to information bias. Complex multi-class data without parameter adjustment prior to testing data, making cost-sensitive probability weighted methods stable in the process [63] while achieving accuracy It must be assessed by the efficiency of good classifying multi-class data. In addition, the cost-sensitive methods weighted based on ensemble learning that are presented are different from the traditional ones that are usually simple votes. [5, 20, 42, 64]. Who use the weight voting method in the assigning probability weight method because cost-sensitive weighted based on ensemble learning. The key principle is to determine the new weight of the final class result. It is used before the result class classification procedure to help reduce the risk of bias data management at the test samples from the base model integration, the proposed cost-sensitive weighted methods have been tested on multi-class benchmark datasets and have been implemented. The comparison of other state-of-the-art methods demonstrates the effectiveness of the framework presented in the areas of Precision, Recall, F-measure, G-mean and Accuracy.

To conclude, our main contributions are four folds:

1. Our proposed method offers cost-sensitive weighted-based on ensemble learning, which is a novel supervised combination model based on Ensemble learning classification Based on machine learning methods to improve the classification of data from individual models to ensemble

models for problem solving processes. The costed-weighted approach is to learn multi-class samples without adjusting the parameters of the base model, providing a good evaluation of the model. Than the base model method and the traditional ensemble model.

2. Our proposed method presents the novel supervised learning of the cost-sensitive probability weighting ensemble learning method, where the combination strategy eliminates information bias from the decision to classify wrong outcomes, which is a good way of consolidating when that strategy gives value misclassified by decreasing.
3. The methods is used TPrate-weight, which is a novel combination strategy from the improvement of the probability weight to combine the base model to provides a better classification performance compared to the many existing methods that are tested using benchmarks dataset.
4. In this research, our proposed method present efforts to improve the process decision-making for multi-class data sets by suggesting the combination methods in the constructing ensemble model of the framework

## 2 Literature review

### 2.1 Ensemble model

The ensemble learning approach employs diverse classification models to enhance the prediction efficiency on various datasets. Although the individual models provide satisfactory prediction accuracy and precision on the dataset, they are sometimes confronted with the problem of bias when specifying the dataset that is obtained from the prediction or specifying the parameter. One of the approaches to resolving the bias is the joint decision approach, which is called ensemble learning with one additional base classification model. The decision is later combined via voting [64]. In this study, weighted voting was deployed to enhance the efficiency in predicting the classification model. Ensemble learning was utilized in various research studies to enhance the prediction efficiency. Reference [42] applied ensemble learning to multi-class ensemble classification and proposed the Kalman-filter-based heuristic ensemble (KFHE). Based on this approach, ensemble learning was conducted during the final procedure. The data were deployed with Kalman filters to combine various individual models with multi-class classification. The experiment compared the KFHE and the original ensemble model and calculated the efficiency of KFHE for testing data without noise and data with noisy class labels. The KFHE realized a significantly better value.

Ensemble learning has also been used to enhance the efficiency of classifiers. In reference [5], natural language

processing (NLP) was conducted via ensemble learning, which employed the voting technique to combine classifiers for named-entity recognition. This approach was crucial for NLP. This study recommended two solutions for generating the ensemble model, which differ in terms of their effects in enhancing the efficiency. It was hypothesized that the reliability of predicting each classifier differed among the output classes. Therefore, the ensemble system must search for each class that was most suitable for the classifier through voting, such as binary voting. Additionally, it must determine the number of votes for each class for the single classifiers, such as the real votes. The applied model classified and selected without domain knowledge or language-specific resources. The results for each language demonstrated that multi-objective optimization (MOO) with real voting could yield higher efficiency than the individual classifiers in the ensemble. For selecting a suitable weight, all the parameters should be set to the most suitable values at the same time. To realize this objective, multi-objective optimization was implemented to enhance the efficiency in evaluating the quality of one additional classification.

In addition to improving the efficiency of machine learning via ensemble learning, reference [20] deployed extreme learning machines to improve the efficiency of the classification model via an advanced ensemble with various models. Extreme learning was efficient in its general operation. However, overfitting of the training data could be likely. To resolve the problem of low efficiency, a heterogeneous ensemble was implemented. This study proposed the advanced ELM ensemble (AELME) for classification. This approach was comprised of the regularized ELM, the 2-norm-optimized ELM (ELML2), and the kernel ELM. The ensemble was generated by training the ELM classifier, which was randomly selected on a subset of the training data with resampling. The resampling subset was selected for the classification of the training data. Each classifier was learned for selecting the data subset randomly via the ELM algorithm. The AELM-Ensemble was developed using the objective function to increase the diversity and accuracy in the group of the final ensemble. The class labels of the unseen data were predicted via majority voting, which were combined with the predictions from ensembles in AELME. The results of the study demonstrate that AELME yielded higher accuracy than the other models on benchmark datasets. Classifying the training data of the subset and combining heterogeneous ELM classifiers yielded high accuracy in the overall operation.

Moreover, in multi-objective optimization, ensemble learning is typically used to increase the efficiency of the classifier. Reference [4] conducted a sentiment analysis. An ensemble was generated via the weighted voting of multi-objective methods that are based on the differential evolution algorithm for text sentiment classification with supervised machine learning. Most approaches that employ ensemble learning

for sentiment analysis manage the features to enhance the prediction efficiency. The study attempted to develop a multi-objective method by increase the efficiency via a weighted voting scheme to fix the suitable weights for each classifier and output class to increase the prediction efficiency for all the algorithms and the sentiment classification. Thus, a multi-objective differential evolution algorithm (MODE) was implemented in this study. Several mutation operators were applied to enhance the efficiency of the differential evolution algorithm. The weight of each classifier for each output class was prescribed by the MODE algorithm, whereas the weight of each classifier for each output class was combined by the ensemble to yield the final prediction for each output class.

Ensemble learning is one of the learning algorithms that has the highest efficiency in supervised learning with groups of models. The prediction strategy is to combine the predictions that are derived from multiple learning algorithms to generate the final result [65] and to increase the prediction efficiency [17]. The ensemble learning model utilized the advantages of each base model when combining the models. Then, the resulting model was utilized to generate the final result by combining the subdivisions from the prediction [66]. The model that combines the learning models was regarded as an efficient approach for improving the classification performance, and it realized outclassed management with a small sample size, high dimensionality, and data with complicated structures, such as the individual models [67]. Furthermore, ensemble learning was one of the machine learning approaches that was deployed to combine models to improve the result of each individual model. This approach relies on the combination of the output of sets of learning models according to specified rules to obtain a better model than an individual learner [68]. Ensemble learning is regarded as an efficient machine learning technique, which is comprised of diverse components for a single task instead of various sub-tasks. Later, the base learning machines were combined to form ensemble learning machines. Compared to the individual algorithms, it was found that the ensemble techniques could reduce the error in determining the mean and combine the multiple classification models to reduce overfitting of the training data. Many research studies have demonstrated the efficiency of ensemble learning, which is an easy technique. In addition, they indicated that ensemble learning realizes higher efficiency than the individual algorithms with the same complexity [27]. Ensemble learning is the combination of the outputs of sets of learning models according to specified rules to obtain a better model than single learners. Ensemble learning has been widely implemented in applications such as image recognition, speech recognition, and industrial process monitoring. Ensemble learning consists of two main procedures: The first procedure involves training. The ensemble model is derived from the base learning algorithms. The second procedure involves prediction. The output of the model is

combined to generate the decision. The selection of the ensemble model can be divided into two stages: The first stage fixes the functions or criteria for evaluating and ranking the model. The second stage deploys a search algorithm to search the group for the best model. The performances of ensemble methods depend on the training data. A severe problem of the ensemble learning algorithm during the stage of learning is maintaining the performance without duplicating the base model. In generating a different model from the feature space, the base models did not reduce the accuracy compared to the individual model, whereas the efficiency of all the ensemble models was enhanced [69]. Regarding ensemble learning, six vital techniques have been employed for analysis. However, the homogenous model is used as a technique for ensemble learning by voting from the separation data. The traditional homogenous model is described as follows.

### 2.1.1 AdaBoost (adaptive boosting)

AdaBoost [31, 70, 71] is a machine learning algorithm and one of the approaches that has been derived from ensemble learning. The approach combines weak models from various classification models. The main strategy of the AdaBoost algorithm concerns the generation of the model weights and sample weights. Once in the process of repeated training, the model weights and sample weights were trained by the weak model. The weights of the samples, which were inaccurately predicted, would increase to focus on the next training step. The model weights were calculated based on the error rate of each weak model.

### 2.1.2 Bagging (bootstrap aggregating)

Bootstrap aggregating (bagging) [72–75] is one of the primary ensemble methods that uses bootstrap sampling. With bagging, the ensemble, as a homogeneous ensemble, was combined to generate the prediction model with data resampling. Bagging was used to generate a subset for resampling, and aggregating was used to generate a subset for bootstrapping. The main objective of bagging was to reduce the error from the variance of the unstable base classifier.

### 2.1.3 Stacking

Stacking [16, 76, 77] is an approach that combines models. Stacking was deployed to reduce the error rate by reducing the bias in the data. The main strategy of stacking is to combine the outputs of various prediction models.

### 2.1.4 Random forest

A random forest [29, 78–80] or a random decision forest is an ensemble learning method for classification and regression. Moreover, random forest was one of the most powerful and



successful machine learning algorithms and combined the diversity of randomized decision trees and prediction via averaging.

### 2.1.5 Random subspace

Random subspace [28] was proposed in a decision forest. Random subspace consists of multiple decision trees that are generated in multiple random subspaces, which are combined to form a classifier. Random subspace was the approach of the classifier ensemble, which was deployed to resolve the problems of noise and more redundant data than the single classifier. The random subspace of the training dataset was improved similarly to bagging. This improvement was conducted in the feature space, especially in the instance space. This approach was derived from the random subspace for generating the base classifier when the dataset had complicated or irrelevant properties.

### 2.1.6 Voting

Voting [62] is regarded as the easiest approach for combining individual classification algorithms. To select the rules for combining the classifier ensemble, voting is the decision rule for selecting a single class out of several alternatives. Voting predicts the class via majority voting. Voting is the approach that is most frequently implemented in ensemble learning. The pattern of voting includes unweighted and weighted voting. Unweighted voting involves simple voting or majority voting, while weighted voting involves simple weighted voting.

A literature review summary is presented in Table 1. The base learner model depends on the experimental data. For model combination, a strategy of combining the predictions of the classifications that are made is applied to the ensemble. There are two main types of ensembles: homogeneous ensembles and heterogeneous ensembles (Fig. 1).

Homogeneous ensembles used the same learning algorithm, whereas, heterogeneous ensembles use various learning algorithms, such as the classifier combination model. As an example of the research of Zhibin Wu et al. [73] in which the heterogeneous ensembles are constructed from two basic methods in comparing the model's performance. Thus, creating models using the principles of a traditional ensemble model is the bagging method. By creating the ANN model as the 1st ensemble model and the SVM model as the 2nd ensemble model, these methods will be combined. A sample voting technique to create a new ensemble model, then compare to find the best method.

## 3 Proposed framework

Our proposed framework is a type of heterogeneous ensemble. This study focused on enhancing the efficiency of the data

classification. Several classification models are typically employed for the test, such as the naive Bayes, multilayer perceptron, and decision tree models. These approaches were employed to evaluate the efficiency to identify the best approach. The model that was tested by those methods was called the base classifier model. The efficiency of the base classifier model was evaluated to identify the best approach. It was posited that best approach for the dataset depended on the attributes of the input dataset. Therefore, the strategy of testing the base models resulted in testing of the ensemble model, which is a combination of various models, to obtain a model with higher efficiency in testing the datasets and with higher accuracy. This study used a dataset with multiple data types to evaluate the efficiency and probability of the most suitable class and could encourage the dataset to realize the best accuracy. To combine the models, the base model was used to generate an ensemble model for obtaining the predicted class and the weights for calculating and combining the models. The weights from the probability of the class occurrence were calculated with the true positive (TP) rate, which measures the reliability of the class that was predicted by the model in comparison with the actual class. In addition, our purposed model uses the reliability rate as a parameter of the probability weight model to select a class. Before generating the model, the dataset was pre-processed. This procedure prepared the dataset before model testing. In this research study, the test for model efficiency was divided into three parts, namely, testing of the base model, the original ensemble model, and the proposed new ensemble model or the TPweight-voting ensemble model that consists of 3TP-Ensemble, 4TP-Ensemble, 5TP-Ensemble and 6TP-Ensemble. All the models, starting with the base model, consist of six models: the decision tree, k-nearest neighbours, support vector machine, multilayer perceptron, naive Bayes, and Bayesian network approaches. The base model was regarded as the beginning model for generating each new ensemble model. Next, the original ensemble model, which was comprised of another six approaches, namely, AdaBoost, bagging, random forest, random subspace, stacking, and voting, is presented. The models were combined via the traditional approach. Finally, the new ensemble model, namely, the TPweight-voting ensemble model, is presented. This model consists of four approaches: the 3TP-Ensemble model, the 4TP-Ensemble model, the 5TP-Ensemble model and the 6TP-Ensemble model. The TPweight-voting ensemble model consists of the base model and realizes improved prediction performance by combining various datasets into a new model to increase the prediction efficiency. This study aims at developing a more efficient model for multi-class classification datasets than the original individual models. In this study, the original ensemble models were compared.

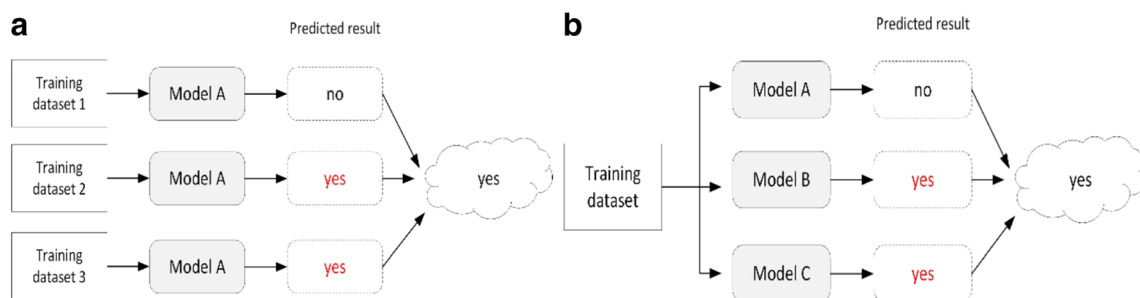
Before the data are input for the test, they are pre-processed. The procedure for preparing the data before testing is as follows: First, various datasets that were derived from the

**Table 1** Literature classification according to the base learner and the use of homogeneous and heterogeneous ensembles

Year	Authors	Datasets	Base learner	Homogeneous	Heterogeneous
2014	Monther Alhamdoosh, Dianhui Wang [69]	9 datasets on regression tasks from UCI	ANN	x	
2015	Jingjing Cao, Sam Kwong, Ran Wang, Xiaodong Li, KeLi, Xiangfei Kong [64]	20 UCI data sets and Finance event series data	CSSV-based ELM algorithms (class-specific soft voting)	x	
2016	Aytunç Onan et al. [4]	9 public sentiment analysis datasets	NB, SVM, LR, BLR, LDA		x
2016	Hamit Erdala et al. [74]	Publicly available dataset that was supplied by the Turkish Supervisory Body (BDDK)	Dstump, Rtree, REPTree	x	
2017	Adnan O. M. et al. [20]	10 diverse datasets from a machine learning repository (UCI)	Svm (LibSVM)		x
2018	Zhen-Ya Wang, Chen Lu, Bo Zhou [67]	datasets that were collected from a rotary actuator system	PNNs: probabilistic neural networks (feedforward neural networks)	x	
2018	Diego P.P. et al. [27]	2 sets real-world datasets from the UCI Machine Learning repository	RNN	x	
2018	Mücahid Barstug˘an et al. [71]	ECG signals from the MIT Arrhythmia Database	SVM	x	
2018	Haoyuan Hong et al. [72]	a landslide inventory map with a total of 237 landslide locations	DT	x	
2018	Zhibin Wu et al. [73]	five climate zones in China and the location of Changsha city	ANN, SVM (Bagging)	x	
2018	Jing Gu et al. [28]	four UCI benchmark clustering datasets	KNN	x	
2019	Arjun Pakrashi, Brian Mac Namee. [42]	30 UCI datasets.	Kalman-filter-based heuristic ensemble (KFHE)	x	
2019	Pablo Pérez-Gállego et al. [65]	32 datasets	AG, PAG, HDy (selective method)		x
2019	Zhixiong Li, Dazhong Wu, Chao Hu, Janis Terpenney [17]	218 testing data sets of the 2008 IEEE PHM Data Challenge	particle filters (PFs), Kalman filter and hidden Markov model	x	
2019	A. Galicia et al. [66]	Australian solar photovoltaic data	DT, GBT, RF	x	
2019	Junhua Zheng, Hongjian Wang, Zhihuan Song, Zhiqiang Ge [68]	Industrial benchmark	K-NN	x	
2019	Yuyan Wang et al. [16]	Prostate cancer dataset	DT, SVM, RBF		x

UCI dataset (Center for Machine Learning and Intelligent Systems) were input for testing. Before testing the model with the datasets, the datasets were handled in two steps. The first step involved preparing the data. This step started with data cleaning, which involved cleaning the data and deleting the instances with large amounts of missing data. The second step replaced the missing values via imputation to make the data smoother. The final data preparation stage was data transformation. This stage transformed the data type according to the

testing input pattern of the classification model. Part of the input data described the input attributes. Each dataset contains two or more classes to facilitate learning in multi-class classification problems. After adapting the datasets, the classification models were tested, and the obtained accuracy values were compared to identify the best approach for testing on the multiple datasets after the input data were suitably managed for testing with the classification model as the basis for generating the new ensemble model.



**Fig. 1** Categories of Ensemble classifier models. (a) Homogeneous ensemble. (b) Heterogeneous ensemble

Figure 2 illustrates that the overall generation of the new ensemble model, which is comprised of two main parts: base model classification and ensemble model classification. The first part involves selecting a suitable model for combining and generating the new ensemble model. The second part generates the new ensemble model and is divided into two procedures. The notations that are used in this paper are presented in Table 2. The framework imported of multi-class dataset which has the data set divided into 2 sets, Training dataset and Testing dataset, then import and test with the based model. The reason why the proposed model used train and independent splitting in order to the training set is a set in which the model uses the imported dataset train to verify and assess the accuracy of the model. However, to measure error using data set is independent of all the data for choosing the best model which is simulation to real case study. The split independent train-test data has advantage with case of many attributes to be consider many classes in the experiment. The independent data splitting method will be not bias and overfitting when is considered and applied to real situation that shown the robustness of this methodology. In addition, it will be in the selection process based model. When preparation of testing data into the model, at this stage, the accuracy and predicted result will be obtained in Fig. 3.

After obtaining the correctness, the next step will select the based model to create a new ensemble model and also will combine strategy with the base model that work together. Each model from the testing dataset have predicted class result of each model. The  $Class_n$  value from this process is in the process of generating the heterogenous ensemble model, in which each classes are assigned a new Weight value for each class result. The equation is  $ProbC_n * Combine\ strategyC_n$ .

This results in the Probability weight of the class result multiplied by the method or strategy used to combine the models. Combine strategy  $C_n$  is divided into four methods: Combine strategy  $C_n$  with TP rate, Recall, Precision, and Fmeasure. For example, the method presented by the research is Combine strategy  $C_n$  with TP rate-weight, which details the creation of the heterogenous ensemble model and the calculation example as shown in Fig. 4.

Figure 4 shows the TP weight class calculation process, which contains a combination strategy by using TP rate, calculated with the probability of class result (probability weight). This method is caused by a combination of a variety of base models to achieve a new weight Class N. For example, the probability value of class N generated by a model M and multiplies with the TP rate of class N from model M.

The principle of TP-rate calculation is to calculate the value of the model that the class is actually accurately predicted value from 0 to 1. The true positive calculation of the class is divided by the total number of that class, where True positive is the relative value of the predicted class result and actual class result when the predicted class only meets the actual value. The result of class from the calculation was example that shown how to calculate the new weight of class as Fig. 5: Christian was calculated, the TP rate of 0.667 is derived from the predicted value of model1. In addition, model1 as bayse model was given the probability value that were from 0 to 1 which was 0.211. The new weight of model1 of class 1 as probability weigh of m1 mulplies with TP rate fo class1 as 0.141 approximatly.

After that the processes find the new weight class of all classes in that datasets when calculating to find all new weight classes that have occurred, therefore, The further calculated

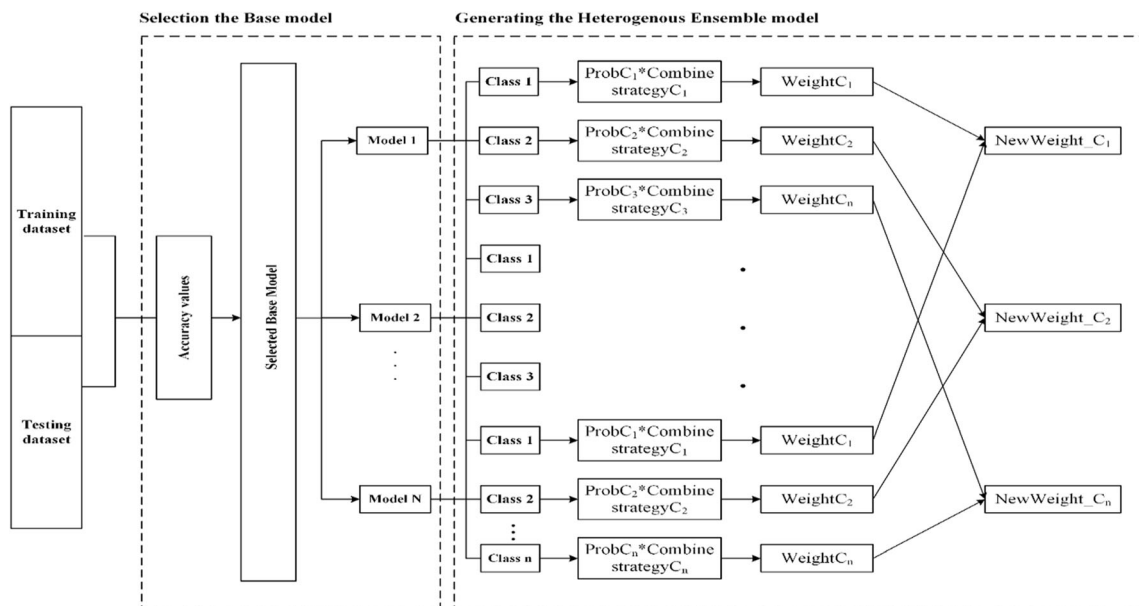


Fig. 2 Overview of the Newweight-voting ensemble model

**Table 2** Description of notations for generating the heterogenous ensemble model

Notation	Description
$c_m$	class set, where $m$ in the number of classes in each dataset that is generated in the testing samples
$d_w$	predicted result set, where $w$ refers to the total number of instances in the testing dataset
$e_t$	set for the TPweight model, where $t$ refers to the number of generated TPweight ensemble models
$f_k$	set for the Fmeasure-weight model, where $k$ refers to the number of generated Fm-weight ensemble models
$F_r$	Fmeasure-weight ensemble
$g_w$	Set of Newweight class, where $w$ is the number of instances in the testing dataset ( $T_s$ )
$M_n$	base models
$m_b$	number of base models for testing
$n_d$	set for the New-weight model, where $d$ refers to the number of generated New ensemble models
$n_u$	training sample set, where $u$ is the total number of instances in the training dataset
$p_e$	set for the Precision-weight model, where $e$ refers to the number of generated Prec-weight ensemble models
$Pr_r$	Precision-weight ensemble
$P_w$	probability value set, where $w$ refers to the total number of instances in the testing dataset
$r_c$	set for the Recall-weight model, where $r$ refers to the number of generated Rec-weight ensemble models
$r_w$	true-positive rate
$R_r$	Recall-weight ensemble
$s_w$	sample set, where $w$ is the total number of instances in the testing dataset
$t_j$	Combination strategy, where $j$ refers to the total number of strategies
$T_n$	training dataset
$T_s$	testing dataset
$T_w$	set of weights that were derived from the TPweight set, where $w$ refers to the total number of instances in the testing dataset
$T_r$	TP rate
$T_{rw}$	TPweight values
$w_d$	set for the Newweight model, where $d$ refers to the number of generated New ensemble models

for the TP weight ensemble model method will be created a ensemble model, if the number of model N is 3 the new weight class of m1 to m3 shown 3 TP-weight Ensemble model in Fig. 6. The TP weight ensemble model is an example of a combination of 3 based models, which called the 3TP-weight ensemble model method. The 3TP-weight were composed of

mlp, svm and bayes methods. The Christian class was calculated as new weight class of Christian from mlp model are combined with new weight class from Christian of svm model and new weight class of Christian from bayes model. For example, the 3TP-weight ensemble model is 0.287 by the 9th instances of classes: Christian with the 3TP-weight ensemble model was

**Fig. 3** Instance of corectness TP rate

$$\begin{aligned}
 TruePositiveRate &= \frac{TruePositiveClassN}{ClassN} \\
 &= \frac{Predict\ class}{Actual\ class}
 \end{aligned}$$



	predictedreligion	religion
1		
2	Muslim	Muslim
3	Marxist	Christian
4	Muslim	Muslim
5	Muslim	Muslim
6	Catholic	Catholic
7	Hindu	Ethnic
8	Catholic	Catholic
9	Catholic	Catholic
10	Marxist	Catholic
11	Marxist	Marxist
12	Catholic	Catholic
13	Catholic	Catholic
14	Others	Muslim
15	Marxist	Marxist
16	Ethnic	Ethnic
17	Christian	Catholic
18	Muslim	Catholic
19	Muslim	Muslim
20	Muslim	Muslim



The true positive-weighted class can be calculated with the equation below.

$$TPweightclass = weight \max \frac{\sum_{j=3}^{class} (P * (T_r))_w}{M_b}$$

$$Class N = \frac{[(Probweight)_{m1} * (TPrateC1)_{m1}] + [(Probweight)_{m2} * (TPrateC1)_{m2}] + [(Probweight)_{m3} * (TPrateC1)_{m3}]}{M_b}$$

Fig. 4 Process of TP-rate weight ensemble model

maximum value. Therefore, the TP weight is calculated of each class, which will be calculated as the WeightCn value of each class result from each model, the WeightCn value is the new probability weight for each class result and then use the weight averaging method to combine the WeightCn of each class from each model to get a NewWeight\_Cn and select the class that

The New class with TPrate can be calculated with Equation.

$$New \text{ weight Class } N = (Probweight)_{m1} * (TPrateC1)_{m1}$$

Where

$$TruePositiveRate = \frac{TruePositiveClassN}{ClassN} = \frac{14}{21}$$

Then *TPrate of class: Christian* = 0.667

Fig. 5 Example of prediction with TP-rate weight ensemble model

The New class with TPrate can be calculated with Equation.

$$New \text{ Weight Class } N = (Probweight)_{m1} * (TPrateC1)_{m1}$$

instance	actual	predicted	Muslim	Marxist	Christian	Catholic	Ethnic	Buddhist	Hindu	Others
1	Muslim	1: Muslim	0.929	0.048	0	0	0.001	0.018	0.003	0.001
2	Christia	2: Marxist	0.004	0.654	0.092	0.211	0.002	0.035	0.001	0
3	Muslim	1: Muslim	0.953	0.035	0	0	0.002	0.004	0.005	0
4	Muslim	1: Muslim	0.835	0.005	0.001	0.001	0.002	0.003	0.136	0.019
5	Catholic	4: Catholic	0.012	0.044	0.012	0.884	0.012	0.008	0.014	0.013
6	Ethnic	7: Hindu	0.134	0.012	0.17	0.003	0.307	0.024	0.329	0.022
7	Catholic	4: Catholic	0	0.011	0	0.983	0.001	0.002	0.003	0
8	Catholic	4: Catholic	0	0.018	0	0.977	0.001	0.003	0.001	0
9	Catholic	2: Marxist	0.032	0.336	0.211	0.119	0.064	0.229	0.002	0.007
10	Marxist	2: Marxist	0	0.967	0.009	0.01	0	0.007	0	0.007
11	Catholic	4: Catholic	0.023	0.08	0.041	0.828	0.026	0.001	0.001	0
12	Catholic	4: Catholic	0.001	0.091	0.003	0.898	0.004	0.001	0	0
13	Muslim	8: Others	0.377	0.003	0	0	0	0.036	0.093	0.491
14	Marxist	2: Marxist	0.003	0.556	0.015	0.414	0.004	0.007	0.001	0
15	Ethnic	5: Ethnic	0.029	0	0	0	0.969	0	0.002	0
16	Catholic	3: Christia	0	0	0.98	0.015	0	0	0	0.005
17	Catholic	1: Muslim	0.961	0	0.018	0.005	0.012	0	0.004	0
18	Muslim	1: Muslim	0.994	0	0	0	0	0	0.005	0.001
19	Muslim	1: Muslim	0.5	0	0.001	0.015	0.484	0	0	0
20	Christia	5: Ethnic	0.086	0	0.41	0.001	0.457	0.001	0.044	0

$$New \text{ Weight Class } N = (Probweight)_{m1} * (TPrateC1)_{m1}$$

$$Class : Christian = 0.211 \times 0.667 = 0.1407 \cong 0.141$$

### 3TP-weight Ensemble model

$$TPweightclass = weight \max \frac{\sum_{r=3}^{class} (P^*(T_r))_w}{M_b}$$

$$TPweight \text{ Class N} = \frac{[New \text{ Weight class N}]_{m1} + [New \text{ Weight class N}]_{m2} + [New \text{ Weight class N}]_{m3}}{M_b}$$

instance	mlp_Christian	svm_Christian	bayes_Christian	weight_Christian
1	0.000	0.071	0.000	0.024
2	0.520	0.167	0.061	0.249
3	0.000	0.047	0.000	0.016
4	0.000	0.119	0.001	0.040
5	0.000	0.119	0.008	0.042
6	0.640	0.167	0.113	0.307
7	0.000	0.095	0.000	0.032
8	0.000	0.119	0.000	0.040
9	0.579	0.143	0.141	0.287
10	0.062	0.143	0.006	0.070
11	0.054	0.143	0.027	0.075
12	0.000	0.095	0.002	0.032
13	0.000	0.119	0.000	0.040
14	0.001	0.143	0.010	0.051
15	0.000	0.095	0.000	0.032
16	0.657	0.167	0.654	0.492
17	0.001	0.119	0.012	0.044
18	0.000	0.119	0.000	0.040
19	0.000	0.095	0.001	0.032

**Class Christian** =  $\frac{mlp + svm + bayes}{M}$

**Class Christian** =  $\frac{0.579 + 0.143 + 0.141}{3}$   
= **0.287**

instance	weight:Muslim	weight:Marxist	weight:Christian	weight:Catholic	weight:Ethnic	weight:Buddhist	weight:Hindu	ensemble_result
1	0.614	0.059	0.024	0.040	0.018	0.002	0.000	Muslim
2	0.041	0.254	0.249	0.149	0.018	0.003	0.000	Marxist
3	0.620	0.055	0.016	0.024	0.010	0.000	0.000	Muslim
4	0.560	0.023	0.040	0.016	0.001	0.000	0.000	Muslim
5	0.072	0.036	0.042	0.481	0.031	0.005	0.000	Catholic
6	0.085	0.033	0.307	0.024	0.107	0.002	0.000	Christian
7	0.020	0.051	0.032	0.515	0.037	0.000	0.000	Catholic
8	0.050	0.042	0.040	0.523	0.027	0.000	0.000	Catholic
9	0.059	0.179	<b>0.287</b>	0.058	0.036	0.019	0.000	<b>Christian</b>
10	0.040	0.422	0.070	0.172	0.009	0.001	0.000	Marxist
11	0.026	0.062	0.075	0.469	0.043	0.000	0.000	Catholic
12	0.041	0.132	0.032	0.399	0.028	0.000	0.000	Catholic
13	0.456	0.022	0.040	0.008	0.000	0.003	0.000	Muslim
14	0.021	0.221	0.051	0.397	0.019	0.001	0.000	Catholic
15	0.070	0.014	0.032	0.040	0.593	0.000	0.000	Ethnic
16	0.020	0.029	0.492	0.053	0.018	0.000	0.000	Christian
17	0.381	0.022	0.044	0.025	0.254	0.000	0.000	Muslim
18	0.630	0.021	0.040	0.024	0.009	0.000	0.000	Muslim
19	0.440	0.014	0.032	0.072	0.219	0.000	0.000	Muslim
20	0.062	0.014	0.132	0.024	0.332	0.000	0.000	Ethnic

Fig. 6 Process of the TP weight voting ensemble model

identified in the voting stage. The overall procedure is described in Algorithm 1.

**Algorithm 1** Newweight-voting ensemble learning approach

**Require**

**Input:** Training dataset  $T_n$  and Testing dataset  $T_s$

**Procedure**

- 1: Base model generation;
- 2: Generate models  $\{m_1, m_2, \dots, m_b\}$ ;
- 3: Call select model from the accuracy values in Algorithm 2;
- 4: Base model for the combination process;
- 5: Newweight generation;
- 6: Generate Combination strategy set  $T = (t_1, t_2, \dots, t_j)$ ;
- 7: Generate probability value set  $P = (P_1, P_2, \dots, P_w)$ ;
- 8: Generate the Newweight ensemble model  $\{w_1, w_2, \dots, w_d\}$ ;
- 9: Call calculation with the Newweight ensemble process in Algorithm 3;
- 10: Call voting Newweight class in Algorithm 4;
- 11: Newweight-voting for the Newweight ensemble result in all the samples;
- 12: Class Newweight-voting ensemble result for the samples in  $T_s$

**Output:** Newweight-voting ensemble learning

Algorithm 1 provides an overview of all the processes in the construction of the Newweight-voting ensemble learning model. First, the dataset was input. The dataset was divided into two parts: the training dataset, which is denoted by  $T_n$ , and the testing dataset, which is denoted by  $T_s$ . The process of generating the Newweight-voting ensemble learning model was divided into two main procedures: generating the base model and generating the Newweight model. Building the new ensemble model begins with the generation of the base model. The model set  $M = (m_1, m_2, \dots, m_b)$ , where  $b$  is the number of base models for testing, is generated. Another process involves selecting the model based on the accuracy on the testing dataset ( $T_s$ ). Algorithm 2 presents the procedure for selecting the model for generating the new ensemble model, which results in the base model that is used in the process of combining the models. In the second procedure, the Newweight model was generated by generating a set of weights from the Newweight set  $W$ . In addition, a set of probabilities was generated, where the probability value set is  $P = (P_1, P_2, \dots, P_w)$  and  $w$  refers to the total number of instances in the testing dataset ( $T_s$ ). Moreover, the set for the Newweight model ( $w_1, w_2, \dots, w_d$ ) was generated, and  $d$  refers to the number of generated New ensemble models. In Algorithm 3, the calculation function for the Newweight ensemble mode is implemented. The class with the largest Newweight was selected as the classification result of the Newweight-voting ensemble model, as presented in Algorithm 4. Then, the model obtains Newweight, which is the probability weight that is selected as the suitable weight of the new class for the New ensemble result in every sample set ( $T_s$ ).

The final result that is obtained from all the processes is the Newweight-voting ensemble learning model.

**Algorithm 2** Selecting the base models

**Require**

**Input:** Training sample set  $N = (n_1, n_2, \dots, n_u)$ ;

Testing sample set  $S = (s_1, s_2, \dots, s_w)$ ;

Class set  $C = (c_1, c_2, \dots, c_m)$ ;

Accuracy values;

**Procedure**

- 1: Base model generation;
- 2: Generate the probability values  $\{P_1, P_2, \dots, P_w\}$ ;
- 3: Generate the prediction results  $\{d_1, d_2, \dots, d_w\}$ ;
- 4: Calculate the accuracy values.
- 5: Actual classes of the samples in  $T_s$ ;
- 6: Predicted classification results for the samples in  $T_s$ ;
- 7: Sorted in descending order based on the accuracy values;

**Output:** Model for generating the new ensemble model

Algorithm 2 describes the selection of the base model for the process of combining the models. In this step, the training dataset was input, where the training sample set is  $N = (n_1, n_2, \dots, n_u)$  and  $u$  is the total number of instances in the training dataset. The testing dataset contains the sample set  $S = (s_1, s_2, \dots, s_w)$ , where  $w$  is the total number of instances in the testing dataset from the class set  $C = (c_1, c_2, \dots, c_m)$ , in which  $m$  is the number of classes in each dataset that is generated from the testing samples ( $S$ ) and the accepted accuracy value. In the step of selecting the model, the generated base model

accepted the probability set  $P = (P_1, P_2, \dots, P_w)$  and the predicted result set  $D = (d_1, d_2, \dots, d_w)$  that was derived from the probability of classifying the testing sample set ( $T_s$ ). For generating the base model, the actual class and the predicted class from the samples in  $T_s$  were considered. The accuracy value is equal to the number true positives in the dataset multiplied by one hundred. Then, the obtained result is divided by the number of observations in the testing set ( $S$ ). Having calculated the accuracy value of the base model, the model use to generate a new ensemble model, which was selected by sorting all the

models in descending order of their accuracy values. The step of selecting the model for the combination involves deleting the base model that has the lowest accuracy. The base models were repeatedly deleted until three models remained; these models were combined. Thus, the model combination was the 3new-Ensemble, 4new-Ensemble, 5new-Ensemble and 6new-Ensemble model. After generating the base model and obtaining the accuracy value for selecting the base models, the final result was the various models for generating the Newweight-voting ensemble model.

### **Algorithm 3** Calculation process of the New ensemble model

#### **Require**

**Input:** Testing sample set  $S = (s_1, s_2, \dots, s_w)$ ;

Class set  $C = (c_1, c_2, \dots, c_m)$ ;

Probability set  $P = (P_1, P_2, \dots, P_w)$ ;

Based model set  $M = (m_1, m_2, \dots, m_b)$ ;

Combination strategy set  $T = (t_1, t_2, \dots, t_j)$ ;

#### **Procedure**

1. Calculate the probability values ( $P$ ) for the sample set ( $S$ ) via Eq. 1, which are in  $[0, 1]$ ;
2. Call predicting the result from the base model ( $M$ );
3. Generate the prediction result  $f(x) = \{d_1, d_2, \dots, d_w\}$ ;
4. Call calculation the combination strategy via Eq. 2, 4, 6, 8 from the confusion matrix in the table;
5. Generate the combination strategy;  $\{t_1, t_2, \dots, t_j\}$ ;
6. Consider each class ( $C$ );
7. Call calculating the combination strategy process with probability values via Eq. 3, 5, 7, 9
8. Generate the new ensemble model  $\{w_1, w_2, \dots, w_d\}$ ;
9. Consider each sample set ( $S$ );
10. Calculate the combination strategy of the sample set in each model;
11. Call combining model with the combine strategy ( $T$ );
12. Generate the combining model  $\{w_1, w_2, \dots, w_d\}$ ;
13. Call combining model with Prec-weight ensemble ( $Pr_r$ );
14. Generate the Prec-weight ensemble  $\{p_1, p_2, \dots, p_c\}$ ;
15. Call combining model with Rec-weight ensemble ( $R_r$ );
16. Generate the Rec-weight ensemble  $\{r_1, r_2, \dots, r_c\}$ ;
17. Call combining model with FM-weight ensemble ( $F_r$ );
18. Generate the Fm-weight ensemble  $\{f_1, f_2, \dots, f_s\}$ ;
19. The new ensemble learning model from weight combined the combination strategy values;

**Output:** The combination strategy of each class

Algorithm 3 presents a process for calculating weight in the new ensemble model. The traditional weight is replaced by the new weight from the combination strategy that consists of 4 methods for determining the new weight. Traditional base model combined with TP-weight method, Precision-weight,

Recall-weight and Fmeasure-weight method. The input data in this process were the testing sample set  $S = (s_1, s_2, \dots, s_w)$ , which was calculated for generating the ensemble model with the class set ( $C$ ) and probability set ( $P$ ) that were derived from the testing sample set ( $T_s$ ). Next, the base model set  $M = (m_1,$

**Table 3** Confusion matrix table for each model

#Dataset = N	Predicted class <sub>1</sub>	Predicted class <sub>2</sub>	...	Predicted class <sub>N</sub>
Actual class <sub>1</sub>	TPclass <sub>1</sub> (#True Positive with class1)	FNclass2(#False Negative with class1 as class2)	...	FNclass <sub>N</sub> (#False Negative with class1 as class N)
Actual class <sub>2</sub>	FNclass2 (#False Negative with class2 as class1)	TPclass2 (#True Positive with class 2)	...	FNclass <sub>N</sub> (#False Negative with class2 as classN)
Actual class <sub>N</sub>	FNclass <sub>N</sub> (#False Negative with class N as class1)	FNclass <sub>N</sub> (#False Negative with class N as class2)		TPclass <sub>N</sub> (#True Positive with class N)

$m_2, \dots, m_b$ ) is accepted, where  $b$  is the number of tested base models. Algorithm 3 begins by calculating the probability values ( $P$ ) of the sample set ( $S$ ) via the measure for each model, and the values were between 0 and 1. The collection of probabilities is expressed in Eq. 1. The class result is predicted with the base model ( $M$ ), and the prediction result set  $D = (d_1, d_2, \dots, d_w)$  of the testing sample ( $T_s$ ) is generated. This process results in the actual class and the predicted class.

$$[(Prob C_1^{m_1}), (Prob C_2^{m_2}), \dots, (Prob C_N^{m_N})] \tag{1}$$

True Positive rate-weight ensemble learning is a method in which various factors Comes from the rate that the class predicted correctly. The TP rate was calculated and used to determine the new weight of the new ensemble model, namely, TP-weight. The procedure for calculating the TPweight values by considering the probability of the class result from the prediction model multiplied by the weight, which was derived from the calculated TP rate ( $T_{rw}$ ) of each testing sample ( $T_s$ ). The TP rate was calculated from the number of predicted classes that corresponded to the actual class in each sample set divided by the total number for all classes, as expressed in Eq. 2, from the confusion matrix in Table 3.

$$Tr_{C_i}^{m_i} = \left[ \left( \frac{TPClass C_1^{m_1}}{N_{m_1}} \right), \left( \frac{TPClass C_2^{m_2}}{N_{m_2}} \right), \dots, \left( \frac{TPClass C_N^{m_N}}{N_{m_N}} \right) \right] \tag{2}$$

To combine the models, the TP-weight values of each class were combined through the average weight, which combined the TP-weight from each model to determine the suitable class with the largest TPweight. The class became the TP-weight class ensemble result. The TP-weight class can be calculated via Eq. 3.

$$TPweight C_i^{m_i} : \begin{cases} \max_{1 \leq n < C_i} TPweight C_i^{m_i} > \alpha, & f(x) = f(\hat{x}) \\ \frac{1}{m_i} * \sum_{i=1}^{m_i} (Prob_{C_i}^{m_i} * Tr_{C_i}^m) \end{cases} \tag{3}$$

For every class ( $C$ ) in the TP-weight class ensemble result, an equation for the probability set  $P = (P_1, P_2, \dots, P_w)$  was obtained, where  $P$  was a set of probabilities of the class that were derived from the base model. The probability was multiplied by the weight from calculating the TP-weight values ( $T_r$ ). The weight from the result that the model could correctly predict ( $D$ ) was compared to the actual results of the testing samples ( $T_s$ ), where  $w$  is the number of instances in the testing dataset ( $T_s$ ) divided by the number in the base model set  $M = (m_1, m_2, \dots, m_b)$ , in which  $b$  is the number of base models that are used in the test to generate the ensemble model.

Precision-weight ensemble learning is a method in which various factors Comes from measuring the accuracy of the model by considering class by class. The Precision was calculated and used to determine the new weight of the new ensemble model, namely, Precision-weight. The procedure for calculating the Precision-weight values by considering the probability of the class result from the prediction model multiplied by the weight, which was derived from the calculated Precision (Pr) of each testing sample (Ts). Precision can be calculated from the ratio of the rows that the class predicted correctly based on the actual class. As for the total number of rows from the class result that is predicted correctly and incorrect prediction of the class being considered. Which will look for every class, the results in the import data set as in Eq. 4.

$$Prec_{C_i}^{m_i} = \left[ \left( \frac{TPClass C_1^{m_1}}{TP_{m_1} + FP_{m_1}} \right), \left( \frac{TPClass C_2^{m_2}}{TP_{m_2} + FP_{m_2}} \right), \dots, \left( \frac{TPClass C_N^{m_N}}{TP_{m_N} + FP_{m_N}} \right) \right] \tag{4}$$

To combine the models, the Precision-weight values of each class were combined through the average weight, which combined the Prec-weight from each model to determine the suitable class with the largest Prec-weight. The class became the Prec-weight class ensemble result. The Prec-weight class can be calculated via Eq. 5.

$$Precweight C_i^{m_i} : \begin{cases} \max_{1 \leq n < C_i} Precweight C_i^{m_i} > \alpha, & f(x) = f(\hat{x}) \\ \frac{1}{m_i} * \sum_{i=1}^{m_i} (Prob_{C_i}^{m_i} * Prec_{C_i}^m) \end{cases} \tag{5}$$



**Table 4** Description of the datasets

UCI data set	# of Attributes	# of Instances	# of Classes	Data Types
Balance Scale	5	625	3	Integer
Eucalyptus Soil	20	736	5	Integer, Text
Grass Grub	9	155	4	Integer, Text
Lymphography	16	148	4	Text
Mushroom	23	8124	7	Text
Car Evaluation	7	1728	4	Text
Flags	30	129	8	Integer, Text
Urban Land Cover	148	675	6	Integer
Vehicle	19	846	4	Integer
User Knowledge	6	403	4	Integer

For every class ( $C$ ) in the Prec-weight class ensemble result, an equation for the probability set  $P = (P_1, P_2, \dots, P_w)$  was obtained, where  $P$  was a set of probabilities of the class that were derived from the base model. The probability was multiplied by the weight from calculating the Prec-weight values ( $Pr_p$ ). The weight from the result that the model could correctly predict ( $D$ ) was compared to the actual results of the testing samples ( $T_s$ ), where  $w$  is the number of instances in the testing dataset ( $T_s$ ) divided by the number in the base model set  $M = (m_1, m_2, \dots, m_b)$ , in which  $b$  is the number of base models that are used in the test to generate the ensemble model.

Recall-weight ensemble learning is a method in which various factors Comes from measuring the accuracy of the model by considering class by class. The Recall was calculated and used to determine the new weight of the new ensemble model, namely, Recall-weight. The procedure for calculating the Recall-weight values by considering the probability of the class result from the prediction model multiplied by the weight, which was derived from the calculated Recall ( $R_r$ ) of each testing sample ( $T_s$ ). The Recall value can be calculated from the ratio that the class predicts correctly to the actual class values. The total number of rows that the class predicts is incorrect and is not considered combined with all the correctly predicted rows, which will be finding the resulting class in the imported data set, as in Eq. 6.

$$Rec_{C_i}^{m_i} = \left[ \left( \frac{TP_{Class} C_1^{m_1}}{TP_{m_1} + FN_{m_1}} \right), \left( \frac{TP_{Class} C_2^{m_2}}{TP_{m_2} + FN_{m_2}} \right), \dots, \left( \frac{TP_{Class} C_N^{m_N}}{TP_{m_N} + FN_{m_N}} \right) \right] \quad (6)$$

To combine the models, the Recall-weight values of each class were combined through the average weight, which combined the Rec-weight from each model to determine the suitable class with the largest Rec-weight. The class became the Rec-weight class ensemble result. The Rec-weight class can be calculated via Eq. 7.

$$Recweight C_i^{m_i} : \begin{cases} \max_{1 \leq n < C_i} Recweight C_i^{m_i} > \alpha, & f(x) = f(\hat{x}) \\ \frac{1}{m_i} * \sum_{i=1}^{m_i} (Prob_{C_i}^{m_i} * Rec_{C_i}^{m_i}) \end{cases} \quad (7)$$

For every class ( $C$ ) in the Rec-weight class ensemble result, an equation for the probability set  $P = (P_1, P_2, \dots, P_w)$  was obtained, where  $P$  was a set of probabilities of the class that were derived from the base model. The probability was multiplied by the weight from calculating the Rec-weight values ( $R_r$ ). The weight from the result that the model could correctly predict ( $D$ ) was compared to the actual results of the testing samples ( $T_s$ ), where  $w$  is the number of instances in the testing dataset ( $T_s$ ) divided by the number in the base model set  $M = (m_1, m_2, \dots, m_b)$ , in which  $b$  is the number of base models that are used in the test to generate the ensemble model.

Fmeasure-weight ensemble learning is a method in which various factors is derived from the weights of the probability of occurrence of the resulting class. The F-measure was calculated and used to determine the new weight of the new ensemble model, namely, Fm-weight. The procedure for calculating the Fm-weight values by considering the probability of the class result from the prediction model multiplied by the weight, which was derived from the calculated F-measure (Fr) of each testing sample (Ts). The F-measure can be calculated from the combination of the performance indicators of the two class result classification: precision and recall. The F-measure shows the average measurement of the accuracy and accuracy of that class can be calculated. Can be obtained from the following equation, where the F-measure is equal to the precision multiplied by the recall value multiplied by 2 times the precision combined with the recall value as in the 8th equation

$$F-Measure = \frac{2 \times precision \times recall}{precision + recall} \quad (8)$$

To combine the models, the Fm-weight values of each class were combined through the average weight, which combined the Fm-weight from each model to determine the suitable class with the largest Fm-weight. The class became the Fm-weight class ensemble result. The Fm-weight class can be calculated via Eq. 9.

Fmweight  $C_i^{m_i}$

$$: \begin{cases} \max_{1 \leq n < C_i} Fmweight C_i^{m_i} > \alpha, & f(x) = f(\hat{x}) \\ \frac{1}{m_i} * \sum_{i=1}^{m_i} (Prob_{C_i}^{m_i} * Fm_{C_i}^m) \end{cases} \quad (9)$$

For every class ( $C$ ) in the Fm-weight class ensemble result, an equation for the probability set  $P = (P_1, P_2, \dots, P_w)$  was obtained, where  $P$  was a set of probabilities of the class that were derived from the base model. The probability was multiplied by the weight from calculating the Fm-weight values ( $F_i$ ). The weight from the result that the model could correctly predict ( $D$ ) was compared to the actual results of the testing

samples ( $T_s$ ), where  $w$  is the number of instances in the testing dataset ( $T_s$ ) divided by the number in the base model set  $M = (m_1, m_2, \dots, m_b)$ , in which  $b$  is the number of base models that are used in the test to generate the ensemble model. According to the equation, the number of classes with *class*  $n$  starts with  $n = 3$ , where  $n$  is the number of the class that was tested in the dataset, with at least three classes for the test with the objective of determining the multi-class classification results. Then, the class for which the combination strategy weight including the TP-weight, Prec-weight, the Rec-weight and the Fm-weight that exceeds  $\alpha$ , which a threshold parameter that is set to 0.8 to yield the maximum accuracy rate in this study, was voted as the Newweight class of the Newweight -voting ensemble learning model.

**Table 5** Parameter settings for each algorithm

Algorithm	Abbreviation	Parameter settings
Naive Bayes	NB	useKernelEstimator = False
Multilayer Perceptrons	MLP	hiddenLayers = a learningRate = 0.3 trainingTime = 500
Bayesian	Bayes	estimator = SimpleEstimator -A 0.5 searchAlgorithm = K2 -P 1 -S BAYES
Support Vector Machine	SVM	c = 1 filter type = Normalize training data kernel = PolyKernel
Decision Trees	DT	confidenceFactor = 0.25 numFolds = 3 unpruned = False
K-Nearest Neighbours	k-nn	KNN = 1, 3, 5 crossValidate = False distanceWeighting = No distance weighting meanSquared = False nearestNeighbourSearchingAlgorithm = LinearNNSearch -A distanceFunction = EuclideanDistance -R first-last
K-Nearest Neighbours	k-nn	KNN = 1, 3, 5 crossValidate = False distanceWeighting = No distance weighting meanSquared = False nearestNeighbourSearchingAlgorithm = LinearNNSearch -A distanceFunction = EuclideanDistance -R first-last
Adaptive Boosting	Adaboost	classifier = DecisionStump numIterations = 10 weightThreshold = 100
Bootstrap Aggregating	Bagging	classifier = REPTree, minNum = 2.0, minVarianceProp = 0.001, numFolds = 3 bagSizePercent = 100 numIterations = 10
Random Forest	RF	maxDepth = 0 numTrees = 10
Random Subspace	RS	classifier = REPTree, minNum = 2.0, minVarianceProp = 0.001, numFolds = 3 numIterations = 10 subSpaceSize = 0.5
Stacking	-	metaClassifier = Zero numFolds = 10
Vote	-	combinationRule = Average of Probabilities
Naive Bayes	NB	useKernelEstimator = False

After the calculation in Algorithm 3, the new weight was obtained. The combination strategy set  $T$ , was generated, and every sample set ( $T_s$ ) in each model was calculated. Then, the models were combined by the new-weight ensemble model set  $T = (t_1, t_2, \dots, t_j)$ , where  $j$  refers to the total number of strategies. In the final process, The new weight ensemble was calculated continuously until the number of new weight ensemble values in the

testing sample set  $S = (s_1, s_2, \dots, s_w)$  was equal to the number in the testing dataset ( $T_s$ ). The obtained Newweight ensemble was deployed to determine the Newweight ensemble class by the Newweight ensemble-voting maximum class. The results of this process were the Newweight ensemble values of each class, namely, each class of every model had a new weight that was derived from the new ensemble model.

#### **Algorithm 4** Voting maximum class Newweight ensemble model

##### **Require**

**Input:** Testing sample set  $S = (s_1, s_2, \dots, s_w)$ ;

Class set  $C = (c_1, c_2, \dots, c_m)$ ;

Base model set  $M = (m_1, m_2, \dots, m_b)$ ;

Newweight set  $W = (w_1, w_2, \dots, w_d)$ ;

##### **Procedure**

- 1: Calculate the Newweight values ( $W$ ) for all the classes ( $C$ );
- 2: Call voting maximum Newweight value of each class;
- 3: Consider each sample set ( $S$ );
- 4: Generate the Newweight class set  $G = (g_1, g_2, \dots, g_w)$ ;
- 5: New classes of the ensemble model;
- 6: Generate a new ensemble model;
- 7: Calculate the Newweight class for the accuracy values;

**Output:** The classes of samples in  $T_s$

Algorithm 4 describes the process of selecting the Newweight class that has the largest weight as the classification result of the Newweight -voting ensemble learning model. The input data consist of the testing samples ( $S$ ) and the class ( $C$ ) of the testing dataset ( $T_s$ ). Moreover, the sets of the base model ( $M$ ) and Newweight ( $W$ ) were input. The procedure of selecting the weight started with the selection of the largest Newweight values that were calculated from every class compared to all the classes in the dataset and from every sample set ( $S$ ). Next, the Newweight class set  $G = (g_1, g_2, \dots, g_w)$  was generated. This procedure generates a new class ensemble result set ( $G$ ), where  $g$  is the new class that is obtained by calculating the Newweight values and  $w$  is the number of instances in the testing dataset ( $T_s$ ). This procedure results in new classes of ensemble models. Afterwards, a new ensemble model, namely, the Newweight-voting ensemble model, was generated, and the accuracy value was calculated from the testing samples ( $S$ ) through the obtained Newweight class. Eventually, the results that were obtained by selecting the largest weight for the Newweight class were the sample classes in  $T_s$  or all classes in the testing dataset ( $T_s$ ) from the new ensemble model.

## 4 Experimental results

This section describes the results of testing the model classifications. Ten datasets were considered in the test, which were

derived from the UCI dataset (Center for Machine Learning and Intelligent Systems). Each dataset is detailed in Table 4. The table describes the attributes of the datasets. Moreover, for each of the ten datasets, it specifies the numbers of attributes, instances, and classes. The data were collected as integers and text files for the tests of the model. The largest dataset contained 8124 instances, and the smallest dataset contained 129 instances. The largest number of attributes was 148, and the smallest number was 5 attributes; hence, the test considers the model and the dataset that collected most and fewest attributes. Finally, the attributes of the classes were collected. It was a test of the multi-class classification performance, where the testing dataset with the most classes had 8 classes and that with the fewest classes had 3 classes. For testing of the model and the data, various datasets were considered. The datasets with the largest amount and the smallest amount of data generated the classifiers or predictors efficiently if there were many data instances to be considered in the test. Then, three types of classification models (the base model, the original ensemble model, and the TPweight-voting ensemble model) were tested. In this study, a test was administered to compare two ensemble models, namely, the original ensemble model and the new ensemble model. The obtained accuracy indicates that the Newweight ensemble model realized higher efficiency of the prediction class than the evaluation of the weight of the classification result using the combination strategy, which

**Table 6** TP-weight cost-sensitive probability of ensemble model

Dataset	Evaluation measure	Homogeneous Ensembles											Heterogeneous Ensembles (Propose Model)					
		Individuals Classification models											Model					
		NB	MLP	Bayes	SVM	DT	k-nn	Adaboost	Bagging	RF	RS	Stacking	Vote	3TP	4TP	5TP	6TP	
Balance Scale	precision	0.740	0.938	0.764	0.904	0.726	0.760	0.563	0.745	0.784	0.639	0.216	0.216	0.904	0.904	0.861	0.788	
	recall	0.588	0.882	0.619	0.825	0.570	0.612	0.391	0.594	0.644	0.470	0.121	0.121	0.825	0.825	0.755	0.651	
	F-Measure	0.655	0.909	0.684	0.862	0.638	0.678	0.462	0.661	0.707	0.542	0.155	0.155	0.862	0.862	0.804	0.713	
	G-mean	0.660	0.910	0.688	0.863	0.643	0.682	0.469	0.665	0.711	0.548	0.162	0.162	0.863	0.863	0.806	0.716	
Eucalyptus Soil	Accuracy	74.038	93.750	76.442	90.385	72.596	75.962	56.250	74.519	78.365	63.942	21.635	21.635	<b>90.385</b>	90.385	86.058	78.846	
	precision	0.308	0.388	0.376	0.428	0.316	0.248	0.272	0.280	0.356	0.340	0.236	0.236	0.432	0.368	0.364	0.364	
	recall	0.084	0.115	0.108	0.133	0.086	0.063	0.071	0.073	0.102	0.096	0.060	0.060	0.160	0.127	0.125	0.125	
	F-Measure	0.131	0.177	0.168	0.203	0.135	0.101	0.113	0.116	0.159	0.149	0.095	0.095	0.233	0.189	0.186	0.186	
Grass Grub	G-mean	0.160	0.211	0.202	0.239	0.165	0.125	0.139	0.143	0.191	0.180	0.119	0.119	0.263	0.216	0.213	0.213	
	Accuracy	30.800	38.800	37.600	42.800	31.600	24.800	27.200	28.000	35.600	34.000	23.600	23.600	<b>43.200</b>	36.800	36.400	36.400	
	precision	0.462	0.462	0.365	0.385	0.308	0.346	0.308	0.288	0.308	0.308	0.288	0.288	0.481	0.423	0.462	0.404	
	recall	0.222	0.222	0.161	0.172	0.129	0.150	0.129	0.119	0.129	0.129	0.119	0.119	0.236	0.196	0.222	0.184	
Lymphography	F-Measure	0.300	0.300	0.224	0.238	0.182	0.209	0.182	0.169	0.182	0.182	0.169	0.169	0.316	0.268	0.300	0.253	
	G-mean	0.320	0.320	0.243	0.258	0.199	0.228	0.199	0.185	0.199	0.199	0.185	0.185	0.337	0.288	0.320	0.273	
	Accuracy	46.154	46.154	36.538	38.462	30.769	34.615	30.769	28.846	30.769	30.769	28.846	28.846	<b>48.077</b>	42.308	46.154	40.385	
	precision	0.857	0.776	0.857	0.837	0.755	0.796	0.714	0.776	0.776	0.776	0.592	0.592	0.857	0.857	0.857	0.857	
Flags	recall	0.667	0.535	0.667	0.631	0.507	0.565	0.455	0.535	0.535	0.535	0.326	0.326	0.667	0.667	0.667	0.667	
	F-Measure	0.750	0.633	0.750	0.719	0.607	0.661	0.556	0.633	0.633	0.633	0.420	0.420	0.750	0.750	0.750	0.750	
	G-mean	0.756	0.644	0.756	0.726	0.619	0.671	0.570	0.644	0.644	0.644	0.439	0.439	0.756	0.756	0.756	0.756	
	Accuracy	<b>85.714</b>	77.551	<b>85.714</b>	83.673	75.510	79.592	71.429	77.551	77.551	77.551	59.184	59.184	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	
Mushroom	precision	0.437	0.310	0.447	0.360	0.305	0.327	0.233	0.290	0.351	0.371	0.246	0.246	0.438	0.443	0.419	0.418	
	recall	0.101	0.062	0.105	0.076	0.061	0.066	0.043	0.057	0.073	0.079	0.046	0.046	0.115	0.117	0.107	0.107	
	F-Measure	0.164	0.103	0.170	0.125	0.101	0.110	0.072	0.095	0.121	0.131	0.077	0.077	0.182	0.185	0.171	0.170	
	G-mean	0.210	0.139	0.216	0.165	0.136	0.147	0.100	0.128	0.160	0.172	0.106	0.106	0.224	0.228	0.212	0.212	
Car Evaluation	Accuracy	43.722	30.982	44.682	35.968	30.465	32.681	23.301	28.951	35.081	37.149	24.631	24.631	<b>43.796</b>	44.276	41.876	41.839	
	precision	0.740	0.665	0.747	0.674	0.644	0.642	0.604	0.637	0.618	0.587	0.587	0.587	0.708	0.663	0.644	0.642	
	recall	0.486	0.398	0.495	0.408	0.376	0.374	0.337	0.369	0.350	0.321	0.321	0.321	0.519	0.396	0.376	0.374	
	F-Measure	0.587	0.498	0.596	0.508	0.475	0.473	0.433	0.468	0.447	0.415	0.415	0.415	0.599	0.496	0.475	0.473	
Flags	G-mean	0.600	0.515	0.608	0.524	0.492	0.490	0.451	0.485	0.465	0.434	0.434	0.434	0.606	0.513	0.492	0.490	
	Accuracy	<b>73.958</b>	66.493	74.653	67.361	64.410	64.236	60.417	63.715	61.806	58.681	58.681	58.681	70.833	66.319	64.410	64.236	
	precision	0.492	0.692	0.708	0.662	0.523	0.554	0.369	0.323	0.723	0.646	0.323	0.323	0.769	0.723	0.723	0.692	

Table 6 (continued)

Dataset	Evaluation measure	Individuals Classification models										Homogeneous Ensembles					Heterogenous Ensembles (Propose Model)					
		NB	MLP	Bayes	SVM	DT	k-nn	Adaboost	Bagging	RF	RS	Stacking	Vote	3TP	4TP	5TP	6TP					
Urban Land Cover	recall	0.110	0.223	0.232	0.198	0.121	0.136	0.069	0.057	0.247	0.188	0.057	0.057	0.331	0.272	0.278	0.246					
	F-Measure	0.179	0.337	0.350	0.305	0.197	0.219	0.117	0.097	0.369	0.292	0.097	0.097	0.463	0.395	0.402	0.363					
	G-mean	0.232	0.393	0.405	0.362	0.252	0.275	0.160	0.136	0.423	0.349	0.136	0.136	0.505	0.443	0.448	0.413					
	Accuracy	49.231	69.231	70.769	66.154	52.308	55.385	36.923	32.308	72.308	64.615	32.308	32.308	<b>76.923</b>	72.308	72.308	69.231					
	precision	0.804	0.845	0.815	0.851	0.792	0.804	0.232	0.804	0.774	0.815	0.149	0.149	0.839	0.841	0.857	0.863					
	recall	0.362	0.407	0.382	0.439	0.349	0.354	0.042	0.363	0.332	0.374	0.024	0.024	0.395	0.417	0.429	0.441					
F-Measure	0.499	0.549	0.520	0.579	0.485	0.492	0.071	0.500	0.465	0.513	0.042	0.042	0.537	0.558	0.571	0.584						
	G-mean	0.539	0.586	0.558	0.611	0.526	0.534	0.098	0.540	0.507	0.552	0.060	0.060	0.576	0.592	0.606	0.617					
Accuracy	80.357	84.524	81.548	85.119	79.167	80.357	23.214	80.357	77.381	81.548	14.881	14.881	83.929	84.118	85.714	<b>86.310</b>						
Dataset	Evaluation measure	Individuals Classification models										Homogeneous Ensembles					Heterogenous Ensembles					
Vehicle	precision	0.411	0.766	0.606	0.734	0.688	0.663	0.401	0.732	0.730	0.723	0.238	0.238	0.709	0.674	0.681	0.702					
	recall	0.189	0.522	0.339	0.479	0.424	0.396	0.182	0.477	0.475	0.466	0.094	0.094	0.448	0.408	0.415	0.440					
	F-Measure	0.259	0.621	0.435	0.580	0.524	0.496	0.251	0.578	0.575	0.567	0.135	0.135	0.549	0.508	0.515	0.541					
	G-mean	0.279	0.632	0.454	0.593	0.540	0.513	0.270	0.591	0.589	0.580	0.150	0.150	0.564	0.524	0.531	0.556					
User Knowledge Modelling	Accuracy	41.135	76.596	60.638	73.404	68.794	66.312	40.071	73.239	73.050	72.340	23.759	23.759	<b>70.922</b>	67.376	68.085	70.213					
	precision	0.848	0.972	0.869	0.793	0.910	0.814	0.476	0.924	0.959	0.924	0.234	0.234	0.938	0.903	0.903	0.931					
F-Measure	recall	0.651	0.922	0.689	0.561	0.772	0.593	0.232	0.802	0.885	0.802	0.093	0.093	0.834	0.757	0.757	0.818					
	G-mean	0.737	0.946	0.768	0.657	0.835	0.686	0.312	0.859	0.921	0.859	0.133	0.133	0.883	0.824	0.824	0.871					
Accuracy	84.828	97.241	86.897	79.310	91.034	81.379	47.586	92.414	95.862	92.414	23.448	23.448	<b>93.793</b>	90.345	90.345	93.103						



**Table 7** comparison of all measure of cost-sensitive probability for ensemble model

Dataset	Evaluation measure	TP-rate	3TP	4TP	5TP	6TP	3Pre	4Pre	5Pre	6Pre	3Rec	4Rec	5Rec	6Rec	3F-m	4F-m	5F-m	6F-m
Balance Scale	precision	0.904	0.904	0.904	0.861	0.788	0.904	0.904	0.861	0.788	0.918	0.918	0.918	0.913	0.918	0.913	0.913	0.879
	recall	0.825	0.825	0.825	0.755	0.651	0.825	0.825	0.755	0.651	0.848	0.848	0.848	0.840	0.848	0.840	0.840	0.785
	F-Measure	0.862	0.862	0.862	0.804	0.713	0.862	0.862	0.804	0.713	0.882	0.882	0.882	0.875	0.882	0.875	0.875	0.829
	G-mean	0.863	0.863	0.863	0.806	0.716	0.863	0.863	0.806	0.716	0.882	0.882	0.882	0.876	0.882	0.876	0.876	0.831
Eucalyptus Soil	Accuracy	90.385	90.385	90.385	86.058	78.846	90.385	90.385	86.058	78.846	91.826	91.826	91.826	91.346	91.826	91.346	91.346	87.98
	precision	0.432	0.368	0.364	0.364	0.364	0.432	0.368	0.364	0.364	0.212	0.212	0.2	0.2	0.212	0.204	0.191	0.187
	recall	0.160	0.127	0.125	0.125	0.125	0.160	0.127	0.125	0.125	0.063	0.063	0.058	0.058	0.063	0.060	0.056	0.054
	F-Measure	0.233	0.189	0.186	0.186	0.186	0.233	0.189	0.186	0.186	0.097	0.097	0.090	0.090	0.097	0.093	0.086	0.084
G-mean	G-mean	0.263	0.216	0.213	0.213	0.213	0.263	0.216	0.213	0.213	0.115	0.115	0.108	0.108	0.115	0.110	0.103	0.101
	Accuracy	<b>43.200</b>	36.800	36.400	36.400	36.400	43.200	36.800	36.400	36.400	21.224	21.224	20	20	21.224	20.408	19.183	18.775
	precision	0.481	0.423	0.462	0.404	0.404	0.481	0.423	0.462	0.404	0.442	0.442	0.442	0.480	0.442	0.384	0.461	0.461
	recall	0.236	0.196	0.222	0.184	0.236	0.236	0.196	0.222	0.184	0.209	0.15	0.209	0.235	0.209	0.172	0.222	0.222
Grass Grub	F-Measure	0.316	0.268	0.300	0.253	0.316	0.316	0.268	0.300	0.253	0.283	0.209	0.283	0.316	0.283	0.238	0.3	0.3
	G-mean	0.337	0.288	0.320	0.273	0.337	0.337	0.288	0.320	0.273	0.304	0.227	0.304	0.336	0.304	0.257	0.320	0.320
	Accuracy	<b>48.077</b>	42.308	46.154	40.385	48.077	48.077	42.308	46.154	40.385	44.23	34.615	44.23	48.076	44.23	38.461	46.153	46.153
	precision	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816
Lymphography	recall	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
	F-Measure	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.689	0.689	0.689	0.689	0.689	0.689	0.689	0.689
	G-mean	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.698	0.698	0.698	0.698	0.698	0.698	0.698	0.698
	Accuracy	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	<b>85.714</b>	81.632	81.632	81.632	81.632	81.632	81.632	81.632	81.632
Mushroom	precision	0.438	0.443	0.419	0.418	0.438	0.438	0.443	0.419	0.418	0.451	0.459	0.444	0.429	0.451	0.460	0.443	0.432
	recall	0.115	0.117	0.107	0.107	0.115	0.115	0.117	0.107	0.107	0.121	0.124	0.117	0.111	0.120	0.124	0.117	0.112
	F-Measure	0.182	0.185	0.171	0.170	0.182	0.182	0.185	0.171	0.170	0.190	0.195	0.186	0.177	0.190	0.196	0.185	0.178
	G-mean	0.224	0.228	0.212	0.212	0.224	0.224	0.228	0.212	0.212	0.233	0.238	0.229	0.219	0.233	0.239	0.228	0.220
Car Evaluation	Accuracy	43.796	44.276	41.876	41.839	43.796	43.796	44.276	41.876	41.839	45.125	45.937	44.497	42.983	45.162	<b>46.048</b>	44.387	43.205
	precision	0.708	0.663	0.644	0.642	0.708	0.663	0.663	0.644	0.642	0.649	0.649	0.640	0.630	0.654	0.654	0.642	0.618
	recall	0.519	0.396	0.376	0.374	0.519	0.396	0.396	0.376	0.374	0.381	0.381	0.372	0.362	0.387	0.387	0.374	0.350
	F-Measure	0.599	0.496	0.475	0.473	0.599	0.496	0.496	0.475	0.473	0.48	0.480	0.471	0.460	0.486	0.486	0.473	0.447
G-mean	G-mean	0.606	0.513	0.492	0.490	0.606	0.513	0.513	0.492	0.490	0.497	0.497	0.488	0.477	0.503	0.503	0.490	0.465
	Accuracy	<b>70.833</b>	66.319	64.410	64.236	70.223	66.319	66.319	64.410	64.236	64.93	64.93	64.062	63.020	65.451	65.451	64.236	61.805
	precision	0.769	0.723	0.723	0.692	0.769	0.723	0.723	0.723	0.692	0.707	0.707	0.723	0.707	0.723	0.723	0.723	0.707
	recall	0.331	0.272	0.278	0.246	0.331	0.272	0.272	0.278	0.246	0.256	0.256	0.271	0.256	0.271	0.271	0.271	0.256
Flags	F-Measure	0.463	0.395	0.402	0.363	0.463	0.395	0.395	0.402	0.363	0.377	0.377	0.394	0.377	0.394	0.394	0.394	0.377

Table 7 (continued)

Dataset	Evaluation measure	TP-rate	Cost-sensitive probability	Precision	Cost-sensitive probability	Recall	Cost-sensitive probability	F-measure	Cost-sensitive probability
Urban Land Cover	G-mean	0.505	0.443	0.448	0.413	0.426	0.443	0.443	0.426
	Accuracy	<b>76.923</b>	72.308	69.231	69.231	70.769	72.307	72.307	70.769
	precision	0.839	0.841	0.863	0.863	0.773	0.857	0.851	0.851
	recall	0.395	0.417	0.429	0.441	0.299	0.428	0.416	0.428
	F-Measure	0.537	0.558	0.571	0.584	0.431	0.571	0.559	0.571
	G-mean	0.576	0.592	0.606	0.617	0.481	0.606	0.595	0.595
Vehicle	Accuracy	83.929	84.118	<b>86.310</b>	86.310	77.380	85.714	85.119	85.714
	precision	0.709	0.674	0.681	0.702	0.751	0.748	0.762	0.695
	recall	0.448	0.408	0.415	0.440	0.502	0.502	0.497	0.431
	F-Measure	0.549	0.508	0.515	0.541	0.602	0.602	0.579	0.532
	G-mean	0.564	0.524	0.531	0.556	0.614	0.614	0.627	0.547
	Accuracy	70.922	67.376	68.085	70.213	<b>75.177</b>	74.822	76.241	73.404
User Knowledge Modelling	precision	0.938	0.903	0.931	0.931	0.979	0.931	0.979	0.917
	recall	0.834	0.757	0.757	0.818	0.940	0.818	0.834	0.786
	F-Measure	0.883	0.824	0.871	0.883	0.824	0.824	0.847	0.847
	G-mean	0.885	0.827	0.827	0.873	0.959	0.872	0.884	0.849
	Accuracy	<b>93.793</b>	90.345	90.345	93.103	97.931	93.103	91.724	91.724

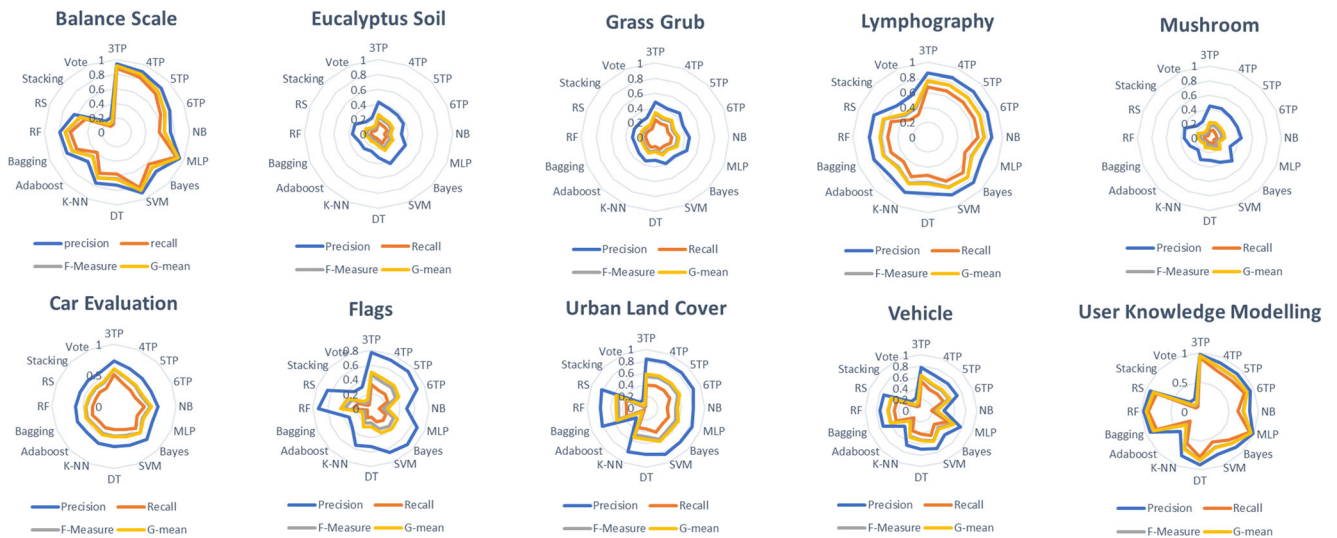


Fig. 7 Evaluation measure (Precision, Recall, F-Measure, and G-Mean) performances of each data set and model

encouraged the probability of the class to have higher efficiency (Table 5).

### 5 Discussion

The weight of cost-sensitive have divided into 4 weight measure as true-positive rate. True Positive rate-weight ensemble learning, Precision-weight ensemble learning, Recall-weight ensemble learning and Fmeasure-weight ensemble learning as Table 7. Each combination strategy is assessed by the overall performance measurements based on cost-sensitive learnings. The TP-rate cost-sensitive probability method was given the best performance measurement methods compared to other methods. The advantage of True positive rate-weight ensemble learning concept as expertise of models is used to able to predict only correct a class that is also high accuracy only each classes. Therefore, the experiment of TP weight will be better than other weighting measure. Our propose model focused on different weight measures is used for value combination from N ensemble methods. The overall performance as Precision, Recall, F-Measure, G-mean and Accuracy are measure repeatedly that shown robustness of models definitely.

The overall performances of the classification models with between 3 and 7 classes are compared in Tables 6 and 7. This table compares the precision, recall, F-measure, G-measure and accuracy values of the base model classifications of all six models and six homogenous models. The proposed model with the best performance value was 3TP-ensemble model, as the values on nine datasets indicate that the 3TP-ensemble model yields the best results. On another dataset, the best results are obtained by the 6TP-ensemble, which is of the same type as the proposed model. The homogenous ensemble model that performed the best was the random forest model, whereas the approaches of stacking and voting yielded lower accuracy rates. The Bayes classification models realized satisfactory performance. Therefore, our proposed model was recently improved to enhance the efficiency in predicting the base classification results and to yield high precision, recall, F-measure and G-mean values in classifying the datasets, as shown in Fig. 7.

The multi-classes datasets in Fig. 8 were divided into three groups: First, the datasets with between 3 and 4 classes were grouped, which are balance scale, lymphographic, vehicle, grass grab, car evaluation, and user knowledge modelling. The best model is 3TP-ensemble, whereas the percentage accuracy rate is the same as those of classification models MLP

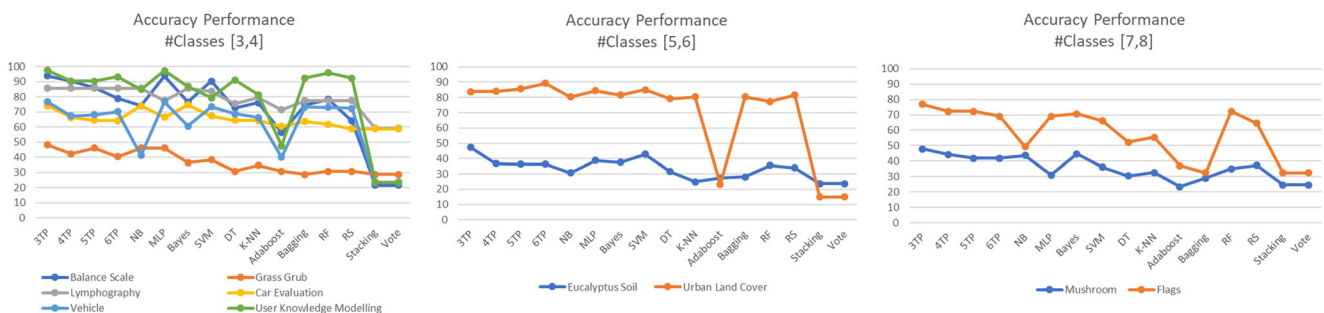
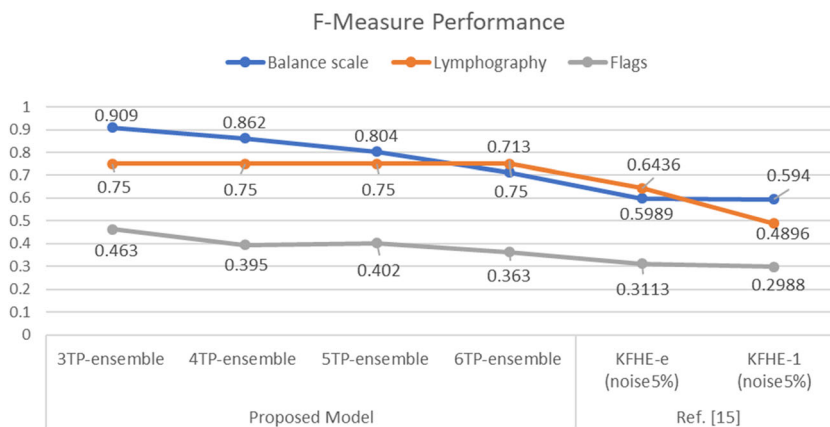


Fig. 8 Accuracy performance versus the number of classes for multi-class classification

**Fig. 9** F-measure performance comparison with other works



and Bayes. Second, the datasets with 5 and 6 classes were grouped, which are eucalyptus soil and urban land cover. The highest accuracy rates are realized by 3TP and 6TP, respectively. This type of proposed model yields the best results. Finally, the datasets with 7 and 8 classes were grouped.

In addition, the F-measure performance of 3TP-ensemble is compared with those of other homogenous models in Reference [14], which proposed KFHE-e (noise 5%) and KFHE-1 (noise 5%), which were also based on the limitations of current multi-class classification ensemble algorithms. Our proposed model, namely, 3TP-ensemble, yields F-measure values of 0.909, 0.75, and 0.463 on the balance scale (3 classes), lymphography (4 classes) and flags (8 classes) datasets, as shown in Fig. 9. The percentage accuracy of 3TP-ensemble is the highest, namely, 85.74%, on the Lymphography dataset (4 classes) compared with mlp:NS ECOC V1, mlp:NS ECOC V2, svm: NS ECOC V1 and svm: NS ECOC V2 in reference [4] and NP-AVG and NP-MAX in reference [37], as shown in Fig. 10. The experimental results demonstrate the robustness and high performance measure values of the 3TP-Ensemble model.

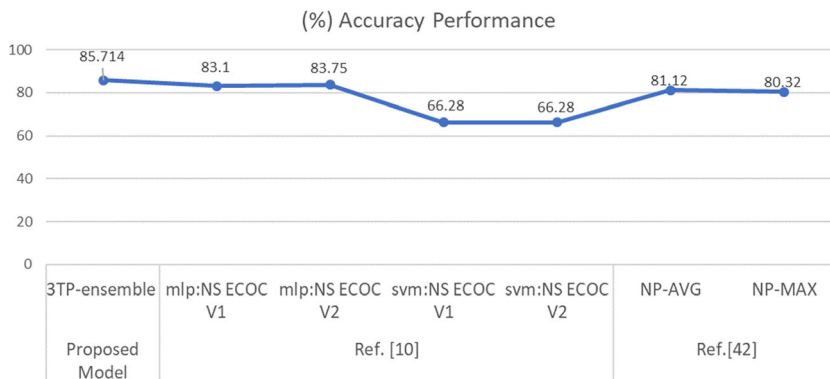
The TPweight ensemble could yield the most suitable predicted class among the new multi-classes of the classification result with increased accuracy. According to our approach, the model that provided the highest average accuracy value was

the TPweight-voting ensemble model. Comparing the best accuracy values on each dataset between the homogenous ensemble model and the newly proposed ensemble model, the TPweight ensemble realized higher efficiency for predicting the result than the ten datasets and the original ensemble model. The TPweight-voting ensemble model was proposed for increasing the efficiency of predicting the classes in various datasets with multi-class labels in practice. The model encouraged the prediction in each sample set  $T_s$  to obtain the most suitable class, which was derived by parameter  $\alpha$ , which reflected the reliability of probability weight for each model, in combination with the TP rate. According to the experimental results for this approach, it could increase the accuracy value of the prediction to a higher value than those that were realized by other models of the original ensemble model.

### 6 Conclusions

In this research, we present a novel supervised cost-sensitive weighted based on Ensemble learning classification. By the framework of the method cost-sensitive probability weighted based on ensemble learning is introduced with the machine learning concept of cost-sensitive weighted ensemble learning

**Fig. 10** Accuracy performance comparison with other works (Lymphography dataset)



which takes advantage of combining results to improve predictive performance -sensitive weighted is designed to reduce class bias occurring as a result of classification. The focus is on the combination strategy that combines weighting with the method of weight voting, which is a method for determining the appropriate weight value. The method presented is based on instructional learning. That is, to learn the model first by dividing test data sets into training data and testing data in order to design a class result. For a multi-class dataset test data set, we demonstrate comprehensive results when compared to methods and apply it to 10 multi-class data sets. We clearly show that the method cost-sensitive probability. Our weighted based probability on ensemble learning has superior methods other state-of-the-art features in Accuracy, Recall, F-measure, G-mean and Accuracy, and reduce misclassified performance. And achieve accuracy values based on good evaluation of classifying multi-class data. Our methods are efficient and stable for problem sets in decision making processes. The multi-class dataset for future work will explore the management of selection base model for creating a new ensemble model to enhance the classification of data sets resulting from the combined prediction.

**Acknowledgments** This study was financially supported in part of the researcher development project of Department of Computer Science, Faculty of Science, Khon Kaen University, THAILAND. And we would like to thank you Asst.Prof.Dr. Mahasak Ketcham, Secretary of Artificial Intelligence Association of Thailand as advisory research.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Agarwal N, Balasubramanian V, Jawahar C (2018) Improving multiclass classification by deep networks using DAGSVM and triplet loss. *Pattern Recogn Lett* 112:184–190
- Eghbali N, Montazer G (2017) Improving multiclass classification using neighborhood search in error correcting output codes. *Pattern Recogn Lett* 100:74–82
- Silva-Palacios D, Ferri C, Ramirez-Quintana M (2017) Improving performance of multiclass classification by inducing class hierarchies. *Procedia Comput Sci* 108:1692–1701
- Onan A, Korukoğlu S, Bulut H (2016) A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Syst Appl* 62:1–16
- Saha S, Ekbal A (2013) Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data Knowl Eng* 85:15–39
- Maron R, Weichenthal M, Utikal J, Hekler A, Berking C, Hauschild A, Enk A, Haferkamp S, Klode J, Schadendorf D, Jansen P, Holland-Letz T, Schilling B, Kalle C, Fröhling S, Gaiser M, Hartmann D, Gesierich A, Kähler K, Wehkamp U, Karoglan A, Bär C, Brinker T (2019) Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks. *Eur J Cancer* 119:57–65
- Kang S, Cho S, Kang P (2015) Multi-class classification via heterogeneous ensemble of one-class classifiers. *Eng Appl Artif Intell* 43:35–43. <https://doi.org/10.1016/j.engappai.2015.04.003>
- Webb C, Ferrari M, Lindström T, Carpenter T, Dürr S, Garner G et al (2017) Ensemble modelling and structured decision-making to support emergency disease management. *Prev Vet Med* 138:124–133
- Goodarzi L, Banihabib M, Roozbahani A (2019) A decision-making model for flood warning system based on ensemble forecasts. *J Hydrol* 573:207–219
- Wheaton M, Topilow K (2020) Maximizing decision-making style and hoarding disorder symptoms. *Compr Psychiatry* 101:152187
- Silva-Palacios D, Ferri C, Ramirez-Quintana M (2017) Improving performance of multiclass classification by inducing class hierarchies. *Procedia Comput Sci* 108C:1692–1701
- Vranjković V, Struharik R, Novak L (2015) Hardware acceleration of homogeneous and heterogeneous ensemble classifiers. *Microprocess Microsyst* 39(8):782–795
- Chaudhary A, Kolhe S, Kamal R (2016) A hybrid ensemble for classification in multiclass datasets: an application to oilseed disease dataset. *Comput Electron Agric* 124:65–72
- Xu J, Wang W, Wang H, Guo J (2020) Multi-model ensemble with rich spatial information for object detection. *Pattern Recogn* 99: 107098
- Yijing L, Haixiang G, Xiaoa L, Yanana L, Jinling L (2016) Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. *Knowl-Based Syst* 94:88–104
- Wang Y, Wang D, Geng N, Wang Y, Yin Y, Jin Y (2019) Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection. *Appl Soft Comput* 77:188–204
- Li Z, Wu D, Hu C, Terpeny J (2019) An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction. *Reliab Eng Syst Saf* 184:110–122
- Bertini Junior J, Nicoletti M (2019) An iterative boosting-based ensemble for streaming data classification. *Inf Fusion* 45:66–78
- Sabzevari M, Martínez-Muñoz G, Suárez A (2018) Vote-boosting ensembles. *Pattern Recogn* 83:119–133
- Abuassba A, Zhang D, Luo X, Shaheryar A, Ali H (2017) Improving classification performance through an advanced ensemble based heterogeneous extreme learning machines. *Comput Intell Neurosci* 2017:1–11
- Cai Y, Liu X, Zhang Y, Cai Z (2018) Hierarchical ensemble of extreme learning machine. *Pattern Recogn Lett* 116:101–106
- Drotár P, Gazda M, Vokorokos L (2019) Ensemble feature selection using election methods and ranker clustering. *Inf Sci* 480:365–380
- Moustafa S, ElNainay M, Makky N, Abougabal M (2018) Software bug prediction using weighted majority voting techniques. *Alex Eng J* 57(4):2763–2774
- Samma H, Lahasan B (2020) Optimized two-stage ensemble model for mammography mass recognition. *IRBM* 41:195–204
- La Cava W, Silva S, Danai K, Spector L, Vanneschi L, Moore J (2019) Multidimensional genetic programming for multiclass classification. *Swarm Evol Comput* 44:260–272



26. Brucker F, Benites F, Sapozhnikova E (2011) Multi-label classification and extracting predicted class hierarchies. *Pattern Recogn* 44:724–738
27. Mesquita D, Gomes JP, Rodrigues L, Oliveira S, Galvão R (2018) Building selective ensembles of randomization based neural networks with the successive projections algorithm. *Appl Soft Comput* 70:1135–1145
28. Gu J, Jiao L, Liu F, Yang S, Wang R, Chen P, Cui Y, Xie J, Zhang Y (2018) Random subspace based ensemble sparse representation. *Pattern Recogn* 74:544–555
29. Zhou Y, Qiu G (2018) Random forest for label ranking. *Expert Syst Appl* 112:99–109
30. Hamze-Ziabari S, Bakhshpoori T (2018) Improving the prediction of ground motion parameters based on an efficient bagging ensemble model of M5' and CART algorithms. *Appl Soft Comput* 68:147–161
31. Hui Y, Shuli L, Rongxiu L, Jianyong Z (2018) Prediction of component content in rare earth extraction process based on ESNs-Adaboost. *IFAC-Papersonline* 51(21):42–47
32. Tang L, Tian Y, Pardalos P (2019) A novel perspective on multiclass classification: regular simplex support vector machine. *Inf Sci* 480:324–338
33. Benjumedá M, Bielza C, Larrañaga P (2019) Learning tractable Bayesian networks in the space of elimination orders. *Artif Intell* 274:66–90
34. Trabelsi A, Elouedi Z, Lefevre E (2019) Decision tree classifiers for evidential attribute values and class labels. *Fuzzy Sets Syst* 366:46–62
35. Zhang Y, Cao G, Wang B, Li X (2019) A novel ensemble method for k-nearest neighbor. *Pattern Recogn* 85:13–25
36. Heidari M, Shamsi H (2019) Analog programmable neuron and case study on VLSI implementation of multi-layer perceptron (MLP). *Microelectron J* 84:36–47
37. Jiang L, Zhang L, Yu L, Wang D (2019) Class-specific attribute weighted naive Bayes. *Pattern Recogn* 88:321–330
38. Guggari S, Kadappa V, Umadevi V (2018) Non-sequential partitioning approaches to decision tree classifier. *Future Comput Inform J* 3(2):275–285
39. Zhou X, Wang X, Hu C, Wang R (2020) An analysis on the relationship between uncertainty and misclassification rate of classifiers. *Inf Sci* 535:16–27
40. Kuncheva L, Rodríguez J (2012) A weighted voting framework for classifiers ensembles. *Knowl Inf Syst* 38(2):259–275
41. Rooney N, Patterson D (2007) A weighted combination of stacking and dynamic integration. *Pattern Recogn* 40:1385–1388
42. Pakrashi A, Mac Namee B (2019) Kalman filter-based heuristic ensemble (KFHE): a new perspective on multi-class ensemble classification using Kalman filters. *Inf Sci* 485:456–485
43. Wang Z, Srinivasan R (2017) A review of artificial intelligence based building energy use prediction: contrasting the capabilities of single and ensemble prediction models. *Renew Sust Energ Rev* 75:796–808
44. Brembo E, Eide H, Lauritzen M, van Dulmen S, Kasper J (2020) Building ground for didactics in a patient decision aid for hip osteoarthritis. Exploring patient-related barriers and facilitators towards shared decision-making. *Patient Educ Couns* 103(7):1343–1350
45. Ding R, Palomares I, Wang X, Yang G, Liu B, Dong Y et al (2020) Large-scale decision-making: characterization, taxonomy, challenges and future directions from an artificial intelligence and applications perspective. *Inf Fusion* 59:84–102
46. Shortland N, Alison L, Thompson L (2020) Military maximizers: examining the effect of individual differences in maximization on military decision-making. *Personal Individ Differ* 163:110051
47. Yang X, Lo D, Xia X, Sun J (2017) TLEL: a two-layer ensemble learning approach for just-in-time defect prediction. *Inf Softw Technol* 87:206–220
48. Mesgarpour M, Chaussalet T, Chahed S (2017) Corrigendum to “ensemble risk model of emergency admissions (ERMER)”. *Int J Med Inform* 108:65–67
49. Lin L, Wang F, Xie X, Zhong S (2017) Random forests-based extreme learning machine ensemble for multi-regime time series prediction. *Expert Syst Appl* 83:164–176
50. Tan Y, Shenoy P (2020) A bias-variance based heuristic for constructing a hybrid logistic regression-naïve Bayes model for classification. *Int J Approx Reason* 117:15–28
51. Ceschi A, Costantini A, Sartori R, Weller J, Di Fabio A (2019) Dimensions of decision-making: an evidence-based classification of heuristics and biases. *Personal Individ Differ* 146:188–200
52. Trajdos P, Kurzynski M (2018) Weighting scheme for a pairwise multi-label classifier based on the fuzzy confusion matrix. *Pattern Recogn Lett* 103:60–67
53. Zhang L, Shah S, Kakadiaris I (2017) Hierarchical multi-label classification using fully associative ensemble learning. *Pattern Recogn* 70:89–103
54. Mao S, Jiao L, Xiong L, Gou S, Chen B, Yeung S-K (2015) Weighted classifier ensemble based on quadratic form. *Pattern Recognit* 48(5):1688–1706
55. Kim H, Kim H, Moon H, Ahn H (2011) A weight-adjusted voting algorithm for ensembles of classifiers. *J Korean Stat Soc* 40(4):437–449
56. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. *Pattern Recogn* 48(5):1623–1637
57. García V, Mollineda R, Sánchez J (2014) A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowl-Based Syst* 59:66–74
58. Tao X, Li Q, Guo W, Ren C, Li C, Liu R, Zou J (2019) Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification. *Inf Sci* 487:31–56
59. Rosdini D, Sari P, Amrania G, Yulianingsih P (2020) Decision making biased: how visual illusion, mood, and information presentation plays a role. *J Behav Exp Financ* 27:100347
60. Liu Y, Gunawan R (2017) Bioprocess optimization under uncertainty using ensemble modeling. *J Biotechnol* 244:34–44
61. Galicia A, Talavera-Llames R, Troncoso A, Koprinska I, Martínez-Álvarez F (2019) Multi-step forecasting for big data time series based on ensemble learning. *Knowl-Based Syst* 163:830–841
62. More S, Gaikwad P (2016) Trust-based voting method for efficient malware detection. *Procedia Comput Sci* 79:657–667
63. Guan D, Yuan W, Ma T, Lee S (2014) Detecting potential labeling errors for bioinformatics by multiple voting. *Knowl-Based Syst* 66:28–35
64. Cao J, Kwong S, Wang R, Li X, Li K, Kong X (2015) Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing* 149:275–284
65. Pérez-Gállego P, Castaño A, Ramón Quevedo J, José del Coz J (2019) Dynamic ensemble selection for quantification tasks. *Inf Fusion* 45:1–15
66. Wei Y, Sun S, Ma J, Wang S, Lai K (2019) A decomposition clustering ensemble learning approach for forecasting foreign exchange rates. *J Manuf Sci Eng* 4(1):45–54
67. Wang Z, Lu C, Zhou B (2018) Fault diagnosis for rotary machinery with selective ensemble neural networks. *Mech Syst Signal Process* 113:112–130
68. Zheng J, Wang H, Song Z, Ge Z (2019) Ensemble semi-supervised fisher discriminant analysis model for fault classification in industrial processes. *ISA Trans* 92:109–117
69. Alhamdoosh M, Wang D (2014) Fast decorrelated neural network ensembles with random weights. *Inf Sci* 264:104–117
70. Chen J, Yang C, Zhu H, Li Y, Gong J (2019) Simultaneous determination of trace amounts of copper and cobalt in high

- concentration zinc solution using UV–vis spectrometry and Adaboost. *Optik* 181:703–713
71. Barstuğan M, Ceylan R (2018) The effect of dictionary learning on weight update of AdaBoost and ECG classification. *J King Saud Univ Comp Inf Sci*. <https://doi.org/10.1016/j.jksuci.2018.11.007>
  72. Hong H, Liu J, Bui D, Pradhan B, Acharya T, Pham B et al (2018) Landslide susceptibility mapping using J48 decision tree with AdaBoost, bagging and rotation Forest ensembles in the Guangchang area (China). *CATENA* 163:399–413
  73. Wu Z, Li N, Peng J, Cui H, Liu P, Li H, Li X (2018) Using an ensemble machine learning methodology-bagging to predict occupants' thermal comfort in buildings. *Energ Build* 173:117–127
  74. Erdal H, Karahanoğlu İ (2016) Bagging ensemble models for bank profitability: an empirical research on Turkish development and investment banks. *Appl Soft Comput* 49:861–867
  75. Sun J, Lang J, Fujita H, Li H (2018) Imbalanced enterprise credit evaluation with DTE-SBD: decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. *Inf Sci* 425:76–91
  76. Healey S, Cohen W, Yang Z, Kenneth Brewer C, Brooks E, Gorelick N et al (2018) Mapping forest change using stacked generalization: an ensemble approach. *Remote Sens Environ* 204:717–728
  77. Sun W, Trevor B (2018) A stacking ensemble learning framework for annual river ice breakup dates. *J Hydrol* 561:636–650
  78. Gong H, Sun Y, Shu X, Huang B (2018) Use of random forests regression for predicting IRI of asphalt pavements. *Constr Build Mater* 189:890–897
  79. Shipway N, Barden T, Huthwaite P, Lowe M (2019) Automated defect detection for fluorescent penetrant inspection using random Forest. *NDT&E Int* 101:113–123
  80. Partopour B, Paffenroth R, Dixon A (2018) Random forests for mapping and analysis of microkinetics models. *Comput Chem Eng* 115:286–294

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.