applied sciences

MDPI

Article

A New Ensemble Strategy Based on Surprisingly Popular Algorithm and Classifier Prediction Confidence

Haochen Shi, Zirui Yuan, Yankai Zhang, Haoran Zhang and Xiujuan Wang *

School of Computer Science, Beijing University of Technology, Beijing 100124, China; shihaochen0418@163.com (H.S.); ziruiyuan7@163.com (Z.Y.); z123zyk456@126.com (Y.Z.); zhrgz040809@126.com (H.Z.)
* Correspondence: xjwang@bjut.edu.cn

Abstract: Traditional ensemble methods rely on majority voting, which may fail to recognize correct answers held by a minority in scenarios requiring specialized knowledge. Therefore, this paper proposes two novel ensemble methods for supervised classification, named Confidence Truth Serum (CTS) and Confidence Truth Serum with Single Regression (CTS-SR). The former is based on the principles of Bayesian Truth Serum (BTS) and introduces classification confidence to calculate the prior and posterior probabilities of events, enabling the recovery of correct judgments provided by a confident minority beyond majority voting. CTS-SR further simplifies the algorithm by constructing a single regression model to reduce computational overhead, making it suitable for large-scale applications. Experiments are conducted on multiple binary classification datasets to evaluate CTS and CTS-SR. Experimental results demonstrate that, compared with existing ensemble methods, both of the proposed methods significantly outperform baseline algorithms in terms of accuracy and F1 scores. Specifically, there is an average improvement of 2–6% in accuracy and an average increase of 2–4% in F1 score. Notably, on the Musk and Hilly datasets, our method achieves a 5% improvement compared to the traditional majority voting approach. Particularly on the Hilly dataset, which generally exhibits the poorest classification performance and poses the greatest prediction challenges, our method demonstrates the best discriminative performance. validating the importance of confidence as a feature in ensemble learning.

Keywords: ensemble learning; supervised classification; Bayesian Truth Serum; confidence

check for updates

Academic Editor: Pedro Couto

Received: 19 January 2025
Revised: 7 March 2025
Accepted: 7 March 2025
Published: 10 March 2025

Citation: Shi, H.; Yuan, Z.; Zhang, Y.; Zhang, H.; Wang, X. A New Ensemble Strategy Based on Surprisingly Popular Algorithm and Classifier Prediction Confidence. *Appl. Sci.* **2025**, *15*, 3003. https://doi.org/10.3390/app15063003

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Ensemble learning has gained increasing prominence in machine learning in recent years, emerging as a pivotal tool for enhancing model performance. It refers to methodologies that combine predictions from multiple independent models to improve overall performance. While single models may underperform when addressing complex problems due to overfitting or inherent biases, the core philosophy of ensemble learning lies in leveraging model diversity to overcome performance bottlenecks caused by overfitting or bias in individual models, thereby achieving superior accuracy, generalization capability, and robustness compared to single models [1,2].

With the advancement of deep learning, ensemble learning techniques and their application scenarios have continued to expand and evolve. Traditional majority voting, despite its simplicity, often delivers suboptimal results when handling complex or specialized data and tasks, necessitating more flexible and effective strategies to integrate the strengths of diverse models [3].

Ensemble learning is often regarded as the machine learning counterpart of the "wisdom of crowds" [4]. Conventional methods, such as majority voting and weighted averaging, operate under the assumption that the majority of predictions are correct. Random forest, for instance, exemplifies the effectiveness of majority voting [5]. While these methods perform well in many scenarios, their efficacy may diminish when individual classifiers exhibit significant diversity and minority classifiers demonstrate higher confidence in specific samples. This is particularly evident in cases requiring specialized knowledge to determine ground-truth answers, where the correct judgments of minority classifiers risk being overshadowed by erroneous predictions from the majority. Such limitations can lead to degraded overall performance.

Recent advances in crowd wisdom theory have introduced novel perspectives for ensemble learning. To address the aforementioned constraints, we propose an ensemble strategy that integrates Bayesian Truth Serum (BTS) with the concept of surprise popularity [6–8]. BTS leverages two critical pieces of information: participants' direct answers to questions and their predictions about the group's responses. By computing the prior probability of events and deriving posterior probabilities for participants, BTS evaluates their credibility based on posterior probabilities and confidence levels. This approach diverges from traditional majority-centric methods by prioritizing participants who provide correct answers while accurately predicting others' responses, as such individuals are more likely to possess genuine expertise. Recent work by McCoy et al. [9] further extends this theory through a hierarchical Bayesian framework for collective decision-making.

The surprise popularity algorithm shares similarities with BTS, as it also analyzes participants' answers and their inferences about group responses. For challenging, less-understood problems, this method identifies experts as those who exhibit high confidence in their answers and recognize that most others may fail to provide correct solutions. Under high-difficulty scenarios, the algorithm prioritizes opinions from such experts. Recent studies demonstrate its unique advantages in complex tasks like ranking problems [10].

Our strategy further incorporates classification confidence, which evaluates the reliability of each classifier's prediction for specific samples. By quantifying confidence levels, we determine which classifiers merit greater trust in their predictions [11]. When classifiers disagree on a sample, confidence metrics enable weighted aggregation of their judgments. In the context of surprise popularity, confidence also adjusts posterior probabilities to identify correct minority predictions, thereby enhancing the ensemble's ability to extract accurate results from minority classifiers. Experimental results demonstrate that our method outperforms traditional ensemble strategies (e.g., random forest, AdaBoost) and individual supervised learning models across diverse binary datasets, achieving higher accuracy and F1 scores.

This study refines the BTS framework by introducing classification confidence and integrating surprise popularity principles. First, we construct diverse base classifiers and compute their confidence levels for each sample. Using these confidences, we derive prior and posterior probabilities, comparing their discrepancies to determine whether to prioritize correct minority predictions over majority errors. Additionally, we propose a single regression model strategy with reduced spatial overhead. The key contributions of this work include the following:

1. Designing an ensemble strategy that integrates classification confidence with surprise popularity;
2. Proposing a prior regression method based on classification confidence and defining posteriors using confidence metrics;
3. Validating the algorithm on public datasets, with experimental results demonstrating superior performance over conventional ensemble methods. Our research highlights

the potential of this approach and provides new directions for future studies and applications in ensemble learning.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 details the proposed ensemble strategy combining classification confidence and surprise popularity. Section 4 presents experimental evaluations. Finally, we conclude the paper.

2. Related Work

The ensemble learning is inspired by people's decision-making processes in real life, and uses the wisdom of crowds: by integrating the opinions of a large number of individuals, it is often possible to obtain better decision-making solutions than individuals. The goal is to build a collection of individual classifiers. By voting on the decisions of individual classifiers in the ensemble, highly accurate classification decisions can be obtained. Ensemble learning has excellent results in solving classification, regression, prediction, decision-making, and other problems [12]. The following summarizes the relevant research works on ensemble learning from the perspectives of algorithm development and application fields.

2.1. Development of Ensemble Learning

In recent years, with the continuous emergence of various new technologies, decision making has become increasingly difficult. The concept of ensemble learning has evolved continually in research; more and more new ideas have been integrated with it, driving the continuous optimization of model performance based on ensemble learning. The theories and methods of ensemble learning have gradually become more complex and diversified. With integrating different theories and methods, it not only expands the depth of application of ensemble learning but also plays a significant role in improving model generalization ability, enhancing robustness, and addressing complex problems.

Continuing the evolution of boosting-family algorithms, Tianqi Chen [13] proposed the XGBoost algorithm in 2016, which is an improvement on the GBDT (gradient boosting decision trees) algorithm. This algorithm fits the second-order derivative of the loss function from the previous round and adds a regularization term beyond the objective function, achieving significant improvements in accuracy. Upon its release, the model received widespread recognition and application in the academic community. In 2017, Microsoft [14] introduced a new boosting framework model called LightGBM. This algorithm builds on GBDT by incorporating gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) techniques. It eliminates most of the data with smaller gradients, reducing the impact of low-gradient parts on information gain. Compared to the XGBoost algorithm, this approach significantly enhances computational speed while maintaining accuracy.

In 2004, Geoffrey et al. [15], based on the hypothesis that the improvement in accuracy of multistrategy ensemble methods is due to the increased diversity among the ensemble members formed, investigated techniques that utilize simple ensemble learning methods to explore the relationship between diversity among members and ensemble error. They developed three multistrategy ensemble learning techniques. The results demonstrated positive effects, reducing the error rate and validating their hypothesis.

Zhang et al. [16] established a semi-supervised support vector machine (S3VM) model composed of base learners, which are based on different interference factors. They also proposed an ensemble method, named EnsembleS3VM, which leverages clustering evaluation techniques. Meanwhile, they combined two multiclass classification strategies to reduce runtime and improve classification accuracy. This method can be used for classification decision making on imbalanced data and data with unknown distributions.

Yu et al. [17] proposed a progressive subspace ensemble learning (PSEL) method that simultaneously considers data sample space and feature space. PSEL first employs random subspace technology to generate a set of subspaces. Then, it introduces a progressive selection process based on a new cost function that incorporates both current and long-term information to sequentially select classifiers. Finally, a weighted voting scheme is used to summarize the predicted labels and obtain the final results. The results of nonparametric tests indicate that PSEL performs well on most real datasets and outperforms many state-of-the-art classifier ensemble methods.

Svargiv et al. [18] studied a novel ensemble learning algorithm based on learning automata called LABEL, which adjusts the contribution of each base learner by dynamically allocating influence coefficients. This adjustment can be made according to different conditions of the problem space, thereby enhancing the performance of the overall ensemble model and improving detection accuracy. This method avoids treating base learners as equal, ignoring the differences and advantages among them. In subsequent research, this method can be applied to problems with nonlinear and unpredictable data. Table 1 demonstrates a comparison of the differences between our study and existing research.

Table 1. Differences between existing studies and this study.

Method	Year	Model	Dataset	Learning Paradigms
MTS, HMTS, DMTS [8]	2023	Perceptron, LR, RF, SVM, MLP.	UCI Repository.	Supervised, semi-supervised.
Decision-tree algorithm C4.5 [12]	2000	Randomization, bagging, and boosting.	UCI Repository.	Supervised.
XGBoost [13]	2016	Tree boosting, C4.5, randomization, sparsity-aware algorithm, weighted quantile sketch.	Allstate, Higgs Boson, Yahoo! LTRC, Criteo.	Supervised.
LightGBM [14]	2017	GOSS, EFB.	Allstate Insurance Claim, Flight Delay, LETOR, KDD10, KDD12.	Supervised.
Multistrategy Ensemble Learning [15]	2004	Bagging, wagging, boosting, SASC.	UCI Repository.	Supervised.
EnsembleS3VM [16]	2018	SVM, LapSVM, S3VMs, S4VMs.	UCI Repository, PolSAR.	Semi-supervised.
PSEL [17]	2016	Decision tree.	UCI Repository, alizadeh—2000—v1, armstrong—2002—v1 and other 16 cancer datasets.	Supervised.
LAbEL [18]	2020	SVM, random forest, naïve Bayes, logistic regression.	Heart disease dataset, Breast cancer Wisconsin (Diagnostic) dataset, Yelp, Abalone dataset, Gender voice.	Supervised.
Our Study	2025	Gbt, LR, RF, SVM, MLP.	UCI Repository, musk, magazine.	Supervised.

2.2. Applications of Ensemble Learning

As ensemble learning techniques continue to advance and mature, their application scenarios have gradually expanded to various fields, such as medical testing, security monitoring, and more. An increasing number of industries and enterprises are leveraging

predictive methods derived from ensemble learning principles to detect and analyze target objects. By integrating the advantages of multiple models, these methods have significantly improved detection accuracy and robustness, especially in target recognition and prediction tasks in complex environments. The widespread application of this technology has not only enhanced system reliability but also provided a more precise basis for decision support and risk management.

Khanh Pham et al. [19] applied ensemble learning to develop a classification model that accurately estimates slope stability. The model constructs an ensemble classifier through two primary techniques: parallel learning and sequential learning. By identifying and learning failure modes of slopes under different conditions, the model is able to assess slope safety, thereby improving the accuracy of disaster warnings.

Luis Gutierrez-Espinoza et al. [20] used ensemble learning to detect fake reviews, integrating random forest, bagging, and AdaBoost, with support vector machines (SVMs) and multilayer perceptrons (MLP) serving as weak classifiers with optimized hyperparameters. The research results indicate that after optimizing the hyperparameters, the accuracy of using MLP as a standalone classifier is 68.2%, while the accuracy of the classifier integrating MLP with AdaBoost reaches 77.3%. This demonstrates that ensemble learning methods can effectively classify fake reviews. The study used a self-collected dataset of fake restaurant reviews to train the classifiers.

Yang et al. [21] proposed an image denoising method based on ensemble learning. This method utilizes different types of image prior knowledge to construct an ensemble denoising model composed of two denoising bases with shrinkage transformations. Through a resampling scheme, multiple denoising bases are sequentially combined multiple times to construct an integrated image denoiser. After performance validation, it can be verified that the ensemble image denoiser operates effectively.

Wang et al. [22] investigated a multikernel ensemble learning method for software defect classification and prediction. By leveraging ensemble learning techniques, they developed a multikernel classifier that combines the advantages of multikernel learning and ensemble learning. Using the widely-used NASA MDP dataset as test data, the results demonstrated that MKEL significantly outperforms several representative and state-of-the-art defect prediction methods.

An et al. [23] proposed a three-layer deep ensemble learning framework named DELearning for the classification and diagnosis of Alzheimer's disease. This framework aims to integrate multisource data and tap into "expert wisdom" using deep learning algorithms. Neural networks serve as the base classifiers. In the voting layer, two sparse autoencoders are trained for feature learning. In the stacking layer, a nonlinear feature weighting method based on deep belief networks is proposed to rank the base classifiers. In the optimization layer, oversampling and threshold shifting are used to address cost-sensitive issues. An optimized prediction is obtained through an ensemble of probability predictions based on similarity calculations. Based on clinical datasets, the detection accuracy is 4% higher than that of six well-known ensemble classifiers.

An overview of current research both domestically and internationally reveals that ensemble learning and ensemble methods are continuously evolving and improving, tending towards higher accuracy, efficiency, and robustness. They play a crucial role in an increasing number of interdisciplinary fields. However, regarding the core mechanism behind many current ensemble learning methods—the majority voting mechanism—Luo and Liu [8] expressed dissent. They argue that in situations where special knowledge is required to obtain the correct answer, the result selected by the majority voting mechanism may not fully represent the correct answer. In cases where the majority is incorrect, they propose deciding whether to adopt the minority answer by comparing the prior and poste-

rior probabilities of the data points. Table 2 presents the various parameters of different ensemble learning methods.

Table 2. Different studies in specific fields.

Methods	Field	Year	Model	Dataset	Learning Paradigms
Parallel learning, Sequential learning [19]	Slope stability	2021	Homogeneous ensemble: DT, SVM, ANN, KNN. Heterogeneous ensemble: KNN, SVM, SGD, GP, QDA, GNB, DT, ANN.	153 slope cases.	Supervised.
Bagging, AdaBoost [20]	Fake Reviews	2020	DT, RF, SVM, XGBT, MLP.	Restaurant Dataset.	Supervised.
Sequential learning [21]	Image Denoising	2020	Local base denoiser, Nonlocal base denoiser.	400 cropped images of size 180 × 180 from the Berkeley segmentation dataset, BSD68.	Supervised.
MKEL [22]	Software defect prediction	2016	SVM.	Datasets from NASA MDP.	Supervised.
DELearning [23]	Alzheimer's disease classification	2020	Bayes Network, naïve Bayes, J48 and so on, DBN, RBM.	UDS.	Supervised.

3. Methodology

3.1. Problem Definition

Let $\boldsymbol{D} = \{(\boldsymbol{x_i}, l_i)\}_{i=1}^{\boldsymbol{M}}$ be the sample set, $\boldsymbol{D_{tr}} \subset \boldsymbol{D}$ is training set, where $(\boldsymbol{x_i}, l_i)$ is the i-th training sample, $\boldsymbol{x_i} = (X_{i1}, X_{i2}, \cdots X_{iT}) \in R^T$, $l_i \in \Phi = \{\omega_1, \omega_2, \cdots \omega_L\}$, M is the number of training sample, T is the number of features for sample, and L is the number of classes; $\mathbb{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_N\}$ represents the set of N base classifiers trained on $\boldsymbol{D_{tr}}$. For each sample $\boldsymbol{x_i}$, $\boldsymbol{f_j}$ independently performs the classification, that is,

$$\boldsymbol{x_i} \xrightarrow{f_j} O_i^j = \boldsymbol{f_j}(\boldsymbol{x_i}) \tag{1}$$

The ensemble learning algorithm, by applying a given strategy $\boldsymbol{\mathcal{T}}$, provides the final classification result O_i for the sample.

$$O_i = \boldsymbol{\mathcal{T}}\left(O_i^1, O_i^2, \cdots O_i^N\right) \tag{2}$$

3.2. Bayesian Truth Serum

The ensemble learning approach in this paper is primarily based on Bayesian Truth Serum (BTS), a mechanism design method grounded in information theory, which incentivizes participants to report truthful information without external validation. Originally proposed by Drazen Prelec in 2004 [6], BTS aims to address certain information asymmetry issues, such as in surveys or prediction markets. The core of this method involves asking participants two questions:

- **The participant's own answer to the question.**
- **The participant's prediction of how others will answer the question.**

Then, Bayesian statistical methods are used to calculate the consistency between each participant's answer and their prediction of others' responses. This approach can identify

participants who not only provide the correct answer but also accurately predict others' responses, thereby rewarding them. This process encourages participants to provide more truthful and accurate information.

The basic steps of BTS are as follows:

I. **Collect answer**: Participants answer a question and provide a probability distribution of how likely they think others will choose each option.

II. **Calculate prior probability**: Calculate the prior probability for each answer option, typically the average distribution of responses provided by all participants.

$$P_k = \frac{\sum_{i=1}^{N} I_{i,k}}{N} \tag{3}$$

Here, $I_{i,k}$ is an indicator function where the i-th participant chooses the k-th option, and N represents the number of participants.

III. **Calculate posterior probability**: Calculate each participant's posterior probability based on their report and the prior probability.

$$Q_{i,k} = \frac{P_k y_{i,k}}{\sum_{j=1}^{N} P_j y_{i,j}} \tag{4}$$

Here, $y_{i,k}$ is the probability that the i-th participant believes others will choose the k-th option.

IV. **Scoring Mechanism**: Calculate a score based on each participant's posterior probability and their provided confidence; a higher score indicates that their answer is more likely to be close to the truth.

3.3. Confidence Truth Serum

3.3.1. Algorithm Overview

Confidence Truth Serum (CTS) introduces the concept of Bayesian Truth Serum into the ensemble strategy. It uses base classifiers to represent individual decisions within a group and establishes a predictive mechanism that allows the truth to reside in the hands of a knowledgeable minority. This mechanism prevents the wisdom of a single individual within the group from being disregarded due to majority voting, thus recovering the correct answers held by the minority and providing a comprehensive understanding of the uncertainty surrounding the event. Unlike the approach in reference [6], this ensemble strategy considers both the classification results and the confidence level of the base classifiers.

Specifically, for each sample $\boldsymbol{x_i}$, the basic classifier $\boldsymbol{f_j}$ classifies independently, yielding the class output result O_i^j and the prediction probability C_i^j, that is,

$$\boldsymbol{x_i} \xrightarrow{f_j} \left(O_i^j = \boldsymbol{f_j}(\boldsymbol{x_i}), C_i^j = P_j(O_i^j|\boldsymbol{x_i}) \right) \tag{5}$$

Based on comprehensive strategy $\boldsymbol{\mathcal{T}}$, the final classification result is provided for the sample: O_i.

$$O_i = \boldsymbol{\mathcal{T}}\left((O_i^1, C_i^1), (O_i^2, C_i^2), \cdots, (O_i^N, C_i^N) \right) \tag{6}$$

C_i^j, often referred to as classification confidence in many machine learning models, primarily serves to quantify each classifier's certainty in a given prediction, thereby optimizing the decision-making process in ensemble learning. The calculation method for confidence is effectively and reasonably defined by us based on the fundamental principles of the classifiers used and the relevant information provided during their prediction processes. By clarifying the calculation methods for the confidence levels of each base learner on the

data, we ensure that each classifier has a confidence value for each data point, which is then applied to subsequent computational processes.

Based on the above definitions, we can obtain the output results of all base classifiers on the training set.

$$\boldsymbol{C} = \begin{pmatrix} P_1(O_1^1|\boldsymbol{x}_1) & P_2(O_1^2|\boldsymbol{x}_1) & \cdots & P_N(O_1^N|\boldsymbol{x}_1) \\ P_1(O_2^1|\boldsymbol{x}_2) & P_2(O_2^2|\boldsymbol{x}_2) & \cdots & P_N(O_2^N|\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_1(O_M^1|\boldsymbol{x}_M) & P_2(O_M^2|\boldsymbol{x}_M) & \cdots & P_N(O_M^N|\boldsymbol{x}_M) \end{pmatrix} \tag{7}$$

$$\boldsymbol{O} = \begin{pmatrix} f_1(\boldsymbol{x}_1) & f_2(\boldsymbol{x}_1) & \cdots & f_N(\boldsymbol{x}_1) \\ f_1(\boldsymbol{x}_2) & f_2(\boldsymbol{x}_2) & \cdots & f_N(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(\boldsymbol{x}_M) & f_2(\boldsymbol{x}_M) & \cdots & f_N(\boldsymbol{x}_M) \end{pmatrix} \tag{8}$$

Similarly, for a test sample $\boldsymbol{x_p}$, the classification results of all base classifiers can be obtained:

$$\boldsymbol{O}(\boldsymbol{x_p}) = (f_1(\boldsymbol{x_p}), f_2(\boldsymbol{x_p}), \ldots f_N(\boldsymbol{x_p})) = \left(O_p^1, O_p^2, \ldots O_p^N\right) \tag{9}$$

$$\boldsymbol{C}(\boldsymbol{x_p}) = \left(P_1\left(O_1^1|\boldsymbol{x_p}\right), P_1\left(O_1^1|\boldsymbol{x_p}\right), \ldots P_1\left(O_1^1|\boldsymbol{x_p}\right)\right) = \left(C_p^1, C_p^2, \ldots C_p^N\right) \tag{10}$$

3.3.2. Ensemble Strategy $\mathcal{T}$

(1) According to the explanation of the Bayesian Truth Serum algorithm in Section 3.2, we need to calculate the prior probability P and posterior probability Q for each answer option (i.e., the classification result for each sample).

Prior probability and posterior probability are concepts rooted in statistics. The prior refers to the estimated likelihood of an event occurring based on prior knowledge or historical information, before any data are observed. It reflects an initial belief about the occurrence of an event in the absence of new evidence. The posterior probability, on the other hand, is the updated probability of the event after observing the data, representing the belief after taking new evidence into account and making a prediction. Therefore, in this paper, the prior probability P is defined as the result of each base classifier in the ensemble $\mathbb{F}$ predicting the classification outcomes of other base classifiers, also referred to as peer prediction information. The posterior probability Q is calculated by aggregating the category prediction probabilities (classification confidence) of all base classifiers for an unknown event.

Taking a supervised binary classification task as an example, the calculation process is as follows:

Based on the matrix $\boldsymbol{O}$, for the k-th base classifier, calculate the consistency ratio of the other $N-1$ classifiers agreeing with its classification result according to Equation (11), denoted as y_i^k (representing the belief score of the k-th classifier for the i-th sample):

$$y_i^k = \frac{\sum_{j \neq k,\, O_i^j = O_i^k} C_i^j}{\sum_{j \neq k} C_i^j} \tag{11}$$

The result y_i^k is used as the new label for each training sample, and C_i^k is added as an additional one-dimensional feature to the sample, forming the feature set $\boldsymbol{x'_i} = \left\{X_1, X_2, X_3 \ldots\ldots X_n, C_i^k\right\}$. This forms a new training dataset $\boldsymbol{D_k} = \left\{\boldsymbol{x'}_i, y_i^k\right\}$, which is then divided into $\boldsymbol{D_k^+}$ and $\boldsymbol{D_k^-}$ according to the original label of the samples, "1" or "0".

Here, confidence is added as a new feature because traditional regression model training usually utilizes feature vectors by optimizing an objective function, such as using the least squares error to fit the model $\hat{y} = w^T + b$. However, in ensemble learning, the classification confidence C_i^k of each base learner for a sample is also an important source of information. Adding classification confidence as a new one-dimensional feature increases the dimensionality of the training samples, allowing the regression model to utilize sample information more comprehensively, leading to a more optimal parameter model and, thus, improving prediction accuracy.

To avoid the phenomenon where directly applying Equation (11) results in the prior and posterior probabilities always being identical, it is necessary to approximate the prior by training a regression model (see reference [6]); therefore, we, respectively, train two regression models for each individual classifier based on the two training sets $\boldsymbol{D_k^+}$ and $\boldsymbol{D_k^-}$.

We repeat the above process for each base classifier.

For each test sample $(\boldsymbol{x_i}, l_i)$, we calculate the prior probability of label 1 using the corresponding regression model based on the actual prediction results of each classifier.

$$\overline{p}_k(x_i) = \begin{cases} p_k^+(x_i) & , if\ O_i^k = 1 \\ 1 - p_k^-(x_i) & , if\ O_i^k = 0 \end{cases} \tag{12}$$

After obtaining the predicted regression results from all individual classifiers, we calculate the prior and posterior probabilities for each sample:

$$P(\boldsymbol{x_i}, 1) := \frac{\sum_k \overline{p}_k(x_i)}{K} \tag{13}$$

$$Q(\boldsymbol{x_i}, 1) := \frac{\sum_{k, O_i^k = 1} C_i^k}{\sum_{k=1}^N C_i^k} \tag{14}$$

If $P(\boldsymbol{x_i}, 1) < Q(\boldsymbol{x_i}, 1)$, the surprising answer 1 will be considered as the true answer. Otherwise, the final classification result is considered to be 0.

Unlike Bayes' theorem, the posterior probability in this paper is not an update or replacement of the prior probability but, rather, a comparison with it. Moreover, the two probability values are derived based on classification confidence, providing a more accurate reflection of the reliability of event estimation. Applying the "surprisingly popular" concept, we determine whether to adopt the latest decision result. In this process, estimated probabilities are replaced by the confidence level of individual decisions, placing greater emphasis on those individuals within the minority who possess the correct answer, thereby ensuring more accurate decision making. In fact, the calculation formula in [8] represents the extreme case where confidence equals 1.

(2) It can be seen that the process of calculating the "surprising" degree of label 1 in the above binary classification task essentially assumes the answer to be 1, which is conceptually similar to hypothesis testing. Similarly, we can assume the answer to be 0 during the algorithm's calculation process to obtain the corresponding ensemble result. In fact, the above process of calculating the prior and posterior probabilities for label 1 or label 0 does not affect the final classification result. Taking the construction of a multiple linear regression model as an example, we provide a brief proof under the extreme case where confidence is equal to 1:

Proof:

Set

$$k_1 = \sum_k 1(f_k(x_t) = 1)$$

$$k_0 = \sum_k 1(f_k(x_t) = 0)$$

$$K = k_1 + k_0$$

For the regression model of each basic classifier, we set

$$p_k^+ = \beta_k + \theta_k X + \varepsilon_k$$

$$p_k^- = \beta_k' + \theta_k' X + \varepsilon_k'$$

For **Label = 1**:

$$\text{if } f(x_t) = 1, \overline{p}_k(x_t) = p_k^+$$

$$\text{else if } f(x_t) = 0, \overline{p}_k(x_t) = 1 - p_k^-$$

$$\therefore P(X_t, 1) = \frac{\sum_k \overline{p}_k(x_t)}{K}$$

$$= \frac{\sum_{k1} \beta_k + \left(\sum_{k1} \theta_k\right) X_t + \sum_{k1} \varepsilon_k - \sum_{k0} \beta_k' - \left(\sum_{k0} \theta_k'\right) X_t - \sum_{k0} \varepsilon_k' + k_0}{K}$$

$$= \frac{k_0 + \sum_{k1} \beta_k + \sum_{k1} \varepsilon_k - \sum_{k0} \beta_k' - \sum_{k0} \varepsilon_k' + \left(\sum_{k1} \theta_k - \sum_{k0} \theta_k'\right) X_t}{K}$$

As the parameters β, θ, and ε are fixed after the regression model is trained, we can set

$$n = \sum_{k1} \beta_k + \sum_{k1} \varepsilon_k - \sum_{k0} \beta_k' - \sum_{k0} \varepsilon_k'$$

$$m = \left(\sum_{k1} \theta_k - \sum_{k0} \theta_k'\right) X_t$$

$$\therefore P(X_t, 1) = \frac{k_0 + n + m}{K}$$

$$Q(X_t, 1) = \frac{k_1}{K} = 1 - \frac{k_0}{k}$$

For **Label = 0**:

$$P(X_t, 0) = \frac{\sum_{k0} \beta_k' + \sum_{k0} \varepsilon_k' - \sum_k \beta_k - \sum_{k1} \theta_k + \left(\sum_{k0} \theta_k' - \sum_{k1} \theta_k\right) X_t + k_1}{K}$$

$$= \frac{k_1 - n - m}{K}$$

$$Q(X_t, 0) = \frac{k_0}{K} = 1 - \frac{k_1}{K}$$

According to the role for **Label = 1**:

$$\text{when } ans = 1, \text{ that is } P < Q, \ \frac{k_0 + n + m}{K} < 1 - \frac{k_0}{K},$$

$$\text{equals } 2k_0 + n + m < K \ (a)$$

According to the role for **Label = 0**:

$$\text{when } ans = 1, \text{ that is } P > Q, \ \frac{k_1 - n + m}{K} > 1 - \frac{k_1}{K},$$

$$\text{equals } 2k_1 - n - m > K;$$

$$\text{that is, } 2k_0 + n + m < K \ (b) \square$$

This is equivalent to Equation (a), indicating that the classification rules are the same.

3.4. Confidence Truth Serum with Single Regression (CTS-SR)

As previously mentioned, in a binary classification problem, the CTS algorithm requires the construction of a total of $2N$ regression models for N base classifiers. This incurs significant memory overhead, especially in the context of ensemble learning where the number of classifiers is large. Additionally, each model is trained only on samples with specific labels, so when the training sample set is small, the regression model has limited data for training. To reduce memory overhead and make full use of the training samples to fit the regression model, this paper proposes a variant of the CTS algorithm, called CTS-SR, which only requires the construction of a single regression model based on all data.

Specifically, in this algorithm, we add the classifier's classification result, named $\boldsymbol{f_k(x_i)}$, for each sample as a one-dimensional feature, in addition to the confidence named $\boldsymbol{C_i^k}$ as a new feature, that is, $\boldsymbol{x_i} = \left\{\boldsymbol{x_1, x_2, x_{3.....}x_n, C_i^k, f_k(x_i)}\right\}$. Still, we let y_i^k, whose meaning is as above, be the new label for each training sample (the calculation process for y_i^k is the same as above). This forms a new training dataset $\boldsymbol{D_k} = \left\{\boldsymbol{x_i, y_i^k}\right\}$, and each individual classifier trains a regression model p_k on $\boldsymbol{D_k}$.

For each testing sample $(\boldsymbol{x_i, y_i}) \in \boldsymbol{T}$, we calculate the prior and posterior probabilities for the sample using each classifier's regression model.

$$P(X_i) := \frac{\sum_k p_k(x_i)}{K} \tag{15}$$

$$Q(X_i) := \frac{\sum_k C_i^k\left(O_i^k = 1\right)}{\sum_{k=1}^{N} C_i^k} \tag{16}$$

The decision rule of this new method differs from CTS because this approach does not incorporate the concept of a predetermined answer. If we still apply $P(X_i) < Q(X_i)$, then whether the answer is 1 or 0 holds no practical significance.

In fact, the prior probability P in this paper describes the consistency of the classifier ensemble's decision for a new sample based on past learning experience. If $P > Q$, this indicates a high level of agreement among the classifiers, in which case we follow the majority voting result. However, when $P < Q$, it suggests that the decision on this event differs from previous instances, indicating a higher level of surprise. To obtain the result from knowledgeable individuals who hold the correct answer, we still apply the "surprisingly popular" concept. This concept implies that in a survey on an obscure question, those who know the true answer often predict a lower consensus rate. We denote this value as R_i^k, In other words, these individuals truly know the answer and believe that most of the group does not. We therefore select these individuals' judgments as the final answer.

As the machine cannot determine whether its answer is correct, we use confidence to express this. We assume that classifiers with high confidence in their classification result and low predicted consensus are more likely to hold the correct answer. Therefore, the decision rule is expressed as follows:

$$O_i = \begin{cases} \text{MajorityVote}\left(O_i^1, O_i^2, \ldots, O_i^N\right) & , if\ P > Q \\ O_i^k \mid \text{argmax}_k\left(C_i^k - R_i^k\right) & , if\ P < Q \end{cases} \tag{17}$$

3.5. Analysis

The CTS and CTS-SR algorithms proposed in Sections 3.3 and 3.4 introduce the concept of unexpected popularity based on the Bayesian Truth Serum theory. The former integrates the confidence levels of the classifiers and decides whether to follow the majority or minority results by comparing the prior and posterior probabilities of the event itself.

Similarly, CTS-SR also emphasizes confidence as a critical metric, fully considering the answers of expert classifiers with specialized knowledge in scenarios with high decision difficulty, making it more aligned with real-world prediction scenarios. Both methods break through the traditional strategies that focus on majority results, thoroughly considering the expertise and confidence levels of decision makers, and constructing a more comprehensive decision-making process and scenario, thereby enabling a more integrated and detailed judgment of unknown events. However, the proposed algorithms still have certain limitations. As the calculation of relevant parameters during training and prediction mostly relies on the confidence levels of the classifiers, defining and computing confidence poses a challenge and may result in increased time overhead. Additionally, when the classification results of all classifiers are highly consistent, CTS may struggle to recover unexpectedly correct answers, which requires further extensive research.

4. Experiment

This section shows the experimental content and results of the application of the CTS and CTS-SR algorithms to binary classification tasks. By using common binary classifiers to build an ensemble learning set, the two methods are applied to a total of eight processed binary classification task datasets. The HMTS in [6] is compared with the common integrated learning method and the self-recurrence HMTS in important indexes. Experimental results show that our algorithm is significantly better than HMTS and other comparison algorithms on most datasets, and we also carry out corresponding theoretical analysis of the results.

4.1. Experimental Setup

In terms of datasets, we consider six binary data commonly found in UCI and two datasets, magazine and o-beauty, used for the detection of naval reviews. For the two text datasets, magazine and All-Beauty, we used the Bert model to extract feature words and convert them into numerical data. Then principal component analysis was adopted to reduce dimension to a certain extent, and all datasets were normalized. We chose 50% as the training set and 50% as the test set. Detailed information is shown in Table 3.

Table 3. The information of datasets.

Datasets	# of Instances	# of Features	Numeric Type	Description of Content
Australian	690	14	Integer, Float	Credit card applicant information.
German	1000	24	Integer	Credit history characteristics.
spam	4601	57	Integer, Float	Content characteristics of the emails.
biodeg	1055	41	Integer, Float	Chemical characteristics of the substances.
magazine	2374	10	Double	Comments of magazine in Amazon.
all-beauty	508	768	Double	Characters of Beauty products.
musk	476	166	Integer	Structural characteristics of the molecules.
Hilly	1212	100	Double	Characteristics of the terrain.

# indicates that the value represents a quantity.

In order to analyze the effectiveness of the Confidence Truth Serum method in the binary classification task, we selected three common ensemble learning methods (majority voting, random forest, and AdaBoost), and HMTS as the comparison group. These algorithms have greatly improved the prediction effect of the classifier.

Majority vote is a simple ensemble learning method that combines the prediction results of multiple classifiers and determines the final classification results through voting. It is suitable for multiple classifiers with similar performance. By synthesizing the advantages

of multiple models, better results can be obtained than a single model. Its classifier settings are consistent with CTS.

Random forest is an ensemble learning method based on decision trees, which makes classification or regression by constructing multiple decision trees and combining their prediction results. It has the characteristics of sample randomization and feature randomization, and has strong anti-overfitting ability; it is especially suitable for processing high-dimensional data. In order to ensure the same number of classifiers, a random forest composed of 15 decision trees is selected.

AdaBoost (adaptive boosting) is an ensemble learning method based on weighted combined weak classifiers. It improves the accuracy of the model by gradually adjusting the weights in the dataset, focusing on correcting misclassified data. In the experiment, 15 weak decision trees are used as the base classifier.

HMTS (Heuristic Machine Truth Serum): The idea is to train a regression model for each classifier to predict the percentage of agreement on predictions from other classifiers for each specific data point. After obtaining the classifier's prediction label and the predicted peer prediction information, the prior is again approximated using the predicted peer prediction information for each classifier, and the average is calculated and compared to the posterior. Its base classifier setup is consistent with CTS.

In terms of evaluation indicators, we mainly choose the traditional accuracy and $F1$ score. TP means that the actual label is positive and the classification is positive; TN means that the actual is negative and the classification is negative, FP means that the actual is negative and the classification is positive; FN means that the actual is positive and the classification is negative. The calculation formula of the evaluation index is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$F1 = 2 \times \frac{PrecisIon \times Recall}{PrecisIon + Recall} \tag{21}$$

In the experiment, we consider five common binary classifiers that can obtain their classification confidence, namely, gradient lift tree (gbt), logistic regression, random forest (RF), support vector machine (SVM), and multilayer perceptron (MLP). In order to enhance the robustness of the model, we selected three different noise rates to flip data labels to improve the prediction ability of the learner in the face of unknown data and enrich the types of learners. Based on this, we obtained a total of 15 different classifiers using our algorithm experiments and other integration methods.

As the idea of this paper is mainly based on confidence, we perform the following calculation for the classification confidence of the above classifier:

For random forests, weighted voting is used to calculate the confidence:

$$C = \frac{\sum(\boldsymbol{w_i} \times \boldsymbol{v_i})}{\sum \boldsymbol{w_i}}$$

where w_i represents the weight of the i-th decision tree and the predicted votes of the tree for this category, and v_i represents the number of votes that the tree assigns to that category.

For logistic regression, the output probability represents the probability that each sample belongs to a certain class. We output the maximum result in each class of probabilities, indicating confidence:

$$C = \max(p_1, p_2, \cdots\cdots, p_i)$$

where p_i represents the prediction probability for the i-th class, and the function max() represents the maximum value for all terms in parentheses.

For support vector machine, distance measurement algorithm is combined with sigmoid function. First, decision_function is used to calculate the distance, dis, between the test sample and the support vector, and then we use the function sigmoid to map the distance to a value in the interval [0,1], representing the probability that the sample belongs to a positive class, to represent the confidence degree:

$$C = \frac{1}{1 + e^{-dis}}$$

where dis is the distance from the test sample to the support vector.

For multilayer perceptrons, we output the maximum result output in each class of probabilities to represent the confidence:

$$C = \max(p_1, p_2, \cdots\cdots, p_i)$$

where p_i, the prediction probability for the i-th class, is expressed, and the $\max()$ function maximizes all terms in parentheses.

For the gradient lifting tree, we sample $\boldsymbol{x_i}$, calculate its distance to the k-th classifier, where dis_i represents the distance from the sample point to the classification hyperplane, $i = 1, 2, ..., k$, and bring it into the classification function $sigmoid(x)$ to obtain the confidence before normalization $prob_{+i}$:

$$prob_{+i} = sigmoid(dis_i) = \frac{1}{1 + e^{-dis_i}}$$

Then the $prob_{+i}$ is normalized, that is, the probability of the sample being Class i is obtained, in another word, it is the confidence degree:

$$C = \frac{prob_{+i}}{\sum_{j=1}^{k} prob_{+j}}$$

4.2. Experimental Results

4.2.1. Experiment 1: Comparison with Baseline Algorithms

To verify the effectiveness and superiority of the two CTS algorithms proposed in this paper across various binary classification tasks, we tested all eight datasets with the CTS algorithms and other baseline algorithms. We recorded the accuracy and F1 scores for each dataset, with Table 4 showing the accuracy of each ensemble algorithm on different datasets and Table 5 displaying their F1 scores. In both tables, bolded values represent the best result on each dataset. It can be seen that our proposed CTS algorithm achieved the highest accuracy on five datasets, while the CTS-SR algorithm achieved the highest accuracy on two datasets. Across both algorithms, they achieved the highest F1 score on six datasets. The results significantly outperform traditional ensemble learning strategies and HMTS, clearly demonstrating the applicability and superiority of our proposed ensemble strategy in various binary classification tasks.

Table 4. Comparison of accuracy among different ensemble algorithms.

Accuracy (%)	CTS	CTS-SR	Major	RF	AdaBoost	HMTS
Australian	86.95	87.24	86.95	84.92	**88.69**	83.18
German	**75.6**	74.2	74.8	71.4	72.6	72.2
spam	**94.08**	89.56	93.61	92.95	90.83	92.78
biodeg	85.98	**87.12**	86.36	83.52	81.25	84.84
magazine	**81.04**	80.53	80.28	78.60	75.82	77.84
all-beauty	85.82	**86.61**	85.43	82.67	81.10	85.03
musk	**84.13**	81.43	80.83	80.23	75.44	83.53
Hilly	**55.94**	50.82	51.65	54.62	48.18	55.11

Table 5. Comparison of F1 score among different ensemble algorithms.

F1 Score (%)	CTS	CTS-SR	Major	RF	AdaBoost	HMTS
Australian	84.94	85.23	84.84	82.31	**86.59**	79.16
German	**49.58**	37.07	46.15	45.69	42.67	48.52
spam	**92.08**	84.99	91.71	90.96	88.24	92.07
biodeg	79.89	**80.34**	79.77	75.63	74.01	78.60
magazine	**87.32**	82.65	87.28	85.63	84.39	84.64
all-beauty	**82.52**	**82.52**	82.46	78.43	75.75	80.80
musk	**81.27**	76.69	76.74	75.73	73.02	80.96
Hilly	59.48	57.53	**65.03**	56.69	64.23	57.63

The traditional majority voting and random forest ensemble algorithms heavily rely on the majority results within the group, meaning that the answers affirmed by most base learners significantly influence the final decision. Our proposed CTS algorithm, to some extent, breaks through the limitations of traditional classification, especially by considering and correctly recovering the correct results provided by minority classifiers even when the majority of classifiers agree. As shown in Table 4, CTS improves accuracy by more than 4% compared to majority voting on the musk and Hilly datasets, and by 6% and 9% compared to AdaBoost on the magazine and musk datasets, respectively. This indicates that our method can correctly classify more instances. The data in Table 5 further demonstrate that CTS improves the F1 score by 2–7% compared to traditional random forest and AdaBoost ensemble algorithms. This suggests that our proposed method not only accurately classifies more instances but also maintains a good balance between precision and recall, reflecting the completeness and balance of the approach. This performance improvement is attributed to our introduction of classifier confidence as a crucial factor in the decision-making process, allowing the correction of answers by a few highly confident expert classifiers when the majority provides incorrect results.

4.2.2. Experiment 2: Comparison with Single Classifiers

To further validate the versatility of the two CTS algorithms, we continued to compare them with several common individual classifiers in supervised learning on the same datasets and recorded their performance. Tables 6 and 7, respectively, present the accuracy and F1 scores of each method across the datasets, with the table settings consistent with Table 4. It is evident that the CTS algorithm achieved the highest accuracy on five datasets, while the CTS-SR algorithm also attained the highest accuracy on two datasets. Together, the two algorithms achieved the highest F1 scores on three datasets. Their performance is significantly superior to that of individual machine learning classifiers. This demonstrates the broad superiority of the strategies we have proposed.

Table 6. Accuracy result compared to single classifier.

Accuracy (%)	CTS	CTS-SR	DT	LightGBM	MLP	SVM
Australian	86.95	**87.24**	82.60	85.79	81.73	86.66
German	**75.6**	74.2	67.2	73.6	68.2	74.0
spam	**94.08**	89.56	89.48	90.65	93.69	92.13
biodeg	85.98	**87.12**	79.16	84.84	85.60	86.93
magazine	**81.04**	80.53	73.79	76.49	80.87	80.62
all-beauty	85.82	86.61	72.83	84.25	83.46	86.61
musk	**84.13**	81.43	69.46	75.74	78.74	79.64
Hilly	**55.94**	50.82	54.78	53.46	50.19	49.17

Table 7. F1 Score compared to single classifier.

F1 Score (%)	CTS	CTS-SR	DT	LightGBM	MLP	SVM
Australian	**84.94**	85.23	80.00	82.63	78.70	84.86
German	49.58	37.07	37.95	**49.78**	46.37	44.91
spam	92.08	84.99	86.76	**93.51**	91.95	89.69
biodeg	79.89	**80.34**	68.64	79.66	77.26	80.22
magazine	87.32	82.65	82.58	87.38	**87.75**	87.39
all-beauty	82.52	82.52	73.97	80.78	79.41	**83.16**
musk	**81.27**	76.69	62.63	72.40	75.17	73.64
Hilly	59.48	57.53	58.29	57.88	63.17	**63.24**

4.2.3. Experiment 3: Friedman–Nemenyi Test

To verify that our algorithm has a significant performance difference compared to the baseline on multiple datasets, we conduct the Friedman–Nemenyi test on various ensemble algorithms. By ranking the performance of each algorithm on each dataset and calculating the average rank of each algorithm across all datasets, we then compute the Friedman statistic χ_F^2 using the formula shown in Equation (22). Here, n represents the number of datasets, k represents the number of algorithms, and rm_j represents the average rank of the j-th algorithm. Based on the degrees of freedom and the significance level, we calculated the p-value. If the p-value is less than the significance level α, we reject the null hypothesis, indicating that there is a significant difference in algorithm performance. After the Friedman test rejects the null hypothesis, we further use the Nemenyi test to determine which algorithms have significant differences in performance. The critical difference (CD) value is used to judge whether the difference in average ranks between two algorithms is significant. The CD value is calculated using the formula shown in Equation (23), where $q_{\alpha,k}$ is the critical value obtained from the Nemenyi test table. If the difference in average ranks between two algorithms is greater than the CD value, their performance is considered significantly different.

In our experiments, we set $n = 8$, $k = 6$, degrees of freedom $df = k - 1 = 5$, and α = 0.05. From the table, we obtained $q_{\alpha,k}$ = 2.850. Based on these values, we conducted the Friedman–Nemenyi test using accuracy and F1 score, as described below.

$$\chi_F^2 = \frac{12n}{k(k+1)}\left(\sum_{j=1}^{k}(rm_j)^2 - \frac{k(k+1)^2}{4}\right) \tag{22}$$

$$CD = q_{\alpha,k}\sqrt{\frac{k(k+1)}{6n}} \tag{23}$$

(1) Friedman–Nemenyi Test Based on Accuracy

Based on the accuracy values in Table 4, the Friedman statistic was calculated to be 17.125, yielding a p-value of 0.004, which is less than α. This indicates that there are significant differences among the compared ensemble algorithms. Subsequently, the Nemenyi post hoc test was conducted, and the critical difference (CD) was calculated to be 2.666. Figure 1 displays the average ranks of the ensemble algorithms for accuracy. If the two horizontal line segments in the figure do not overlap, it signifies a significant performance difference between the two algorithms. It can be seen that the horizontal lines corresponding to CTS do not intersect with those of RF and AdaBoost. Additionally, the difference calculated from their labeled average rank values is greater than the CD value of 2.666. Moreover, CTS has the smallest average rank value, indicating that our algorithm significantly outperforms the other two, further underscoring the effectiveness of our method.

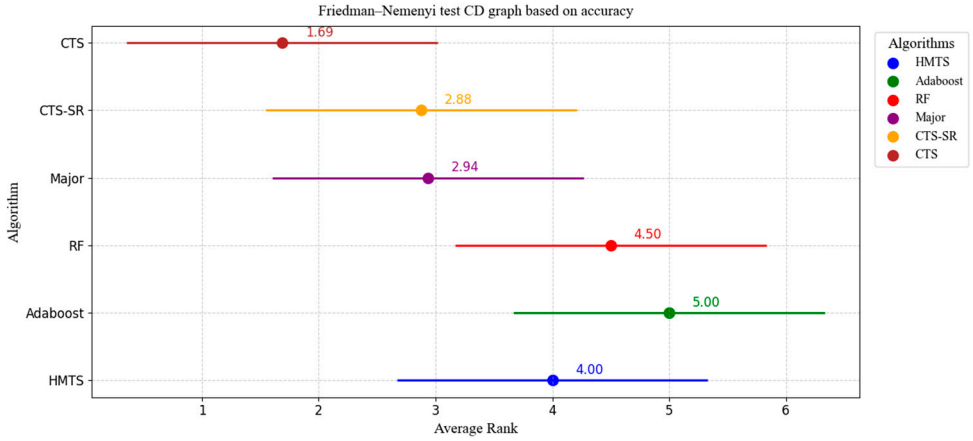


Figure 1. Friedman–Nemenyi test CD graph based on accuracy.

(2) Friedman–Nemenyi Test Based on F1 score

Based on the F1 score values in Table 5, the Friedman statistic was calculated to be 14.411, yielding a p-value of 0.0132, which is less than α. This indicates that there are significant differences among the compared ensemble algorithms. Subsequently, the Nemenyi post hoc test was conducted, and the critical difference (CD) was calculated to be 2.666. Similarly, it can be seen from Figure 2 that our algorithm significantly outperforms other ensemble methods in terms of F1 score, further illustrating the reliability of our proposed algorithm.

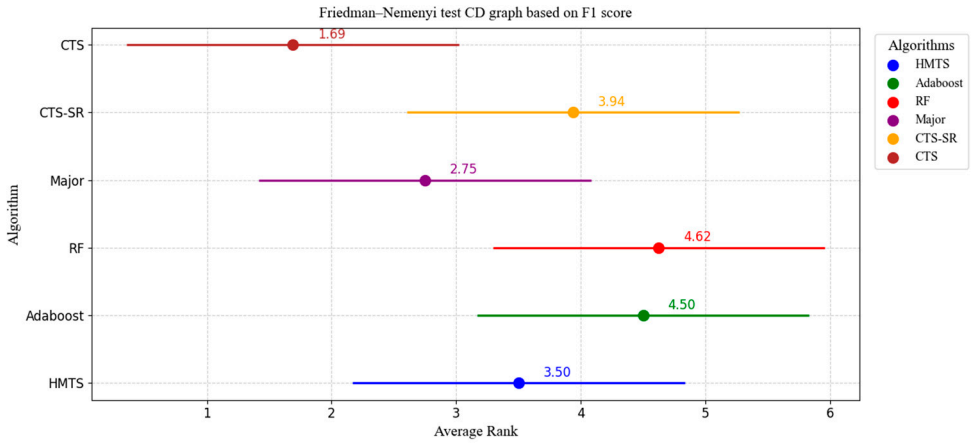


Figure 2. Friedman–Nemenyi test CD graph based on F1 score.

4.2.4. Experiment 4: Calculation and Analysis of Correlation Coefficient

To demonstrate that confidence, as an inherent attribute of the classifier, can significantly enhance the predictive capability of the regression model when used as a sample feature, we calculated and recorded the correlation coefficients between the confidence-based feature vectors of 15 base classifiers and the label vectors on each dataset. These were then compared with vectors composed of other sample features.

As shown in Figure 3, X_{ij} represents the j-th feature of the i-th sample in the dataset, C_i^k represents the classification confidence of the k-th classifier for the i-th sample, and y_i^k is the regression value of the k-th classifier for the i-th sample. $\boldsymbol{X_{ij}}(i = 1, 2, 3 \ldots\ldots n)$ is the vector composed of the first feature, $\boldsymbol{C_i^k}(i = 1, 2, 3 \ldots\ldots, n)$ is the vector composed of confidence values, and $\boldsymbol{y_i^k}(i = 1, 2, 3 \ldots\ldots, n)$ is the vector of regression values. The correlation coefficient is calculated as follows:

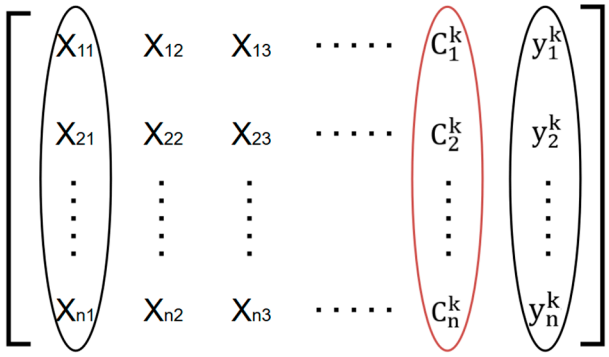


Figure 3. Illustration for correlation coefficient computing.

Let the two vectors be defined as follows:

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$$

$$\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$$

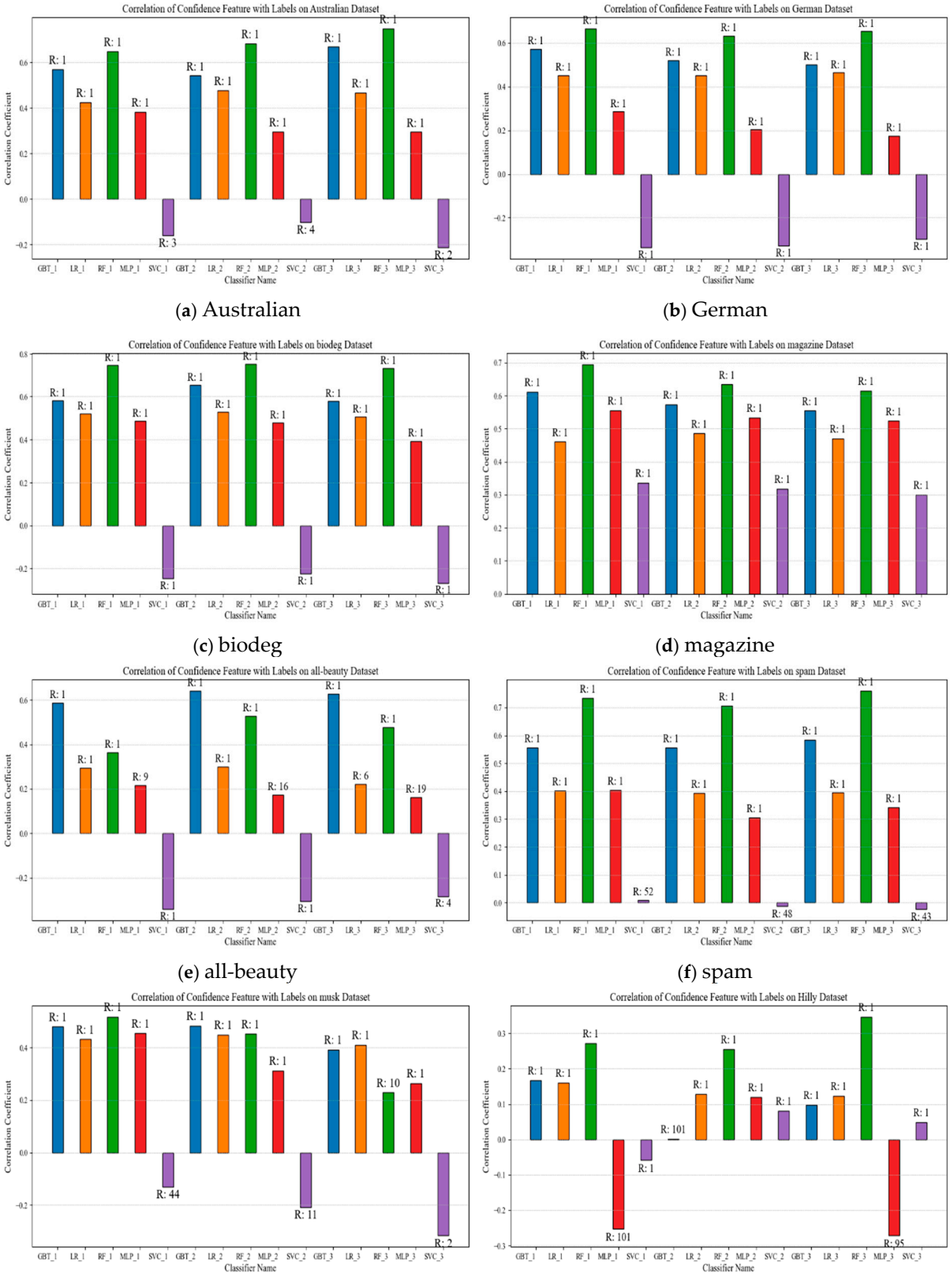
$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

The formula for the correlation coefficient ρ is as follows:

$$\rho = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} \times \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{24}$$

The specific experimental results are shown in Figure 4. As we are only concerned with the strength of the correlation between vectors, the ranking here considers the absolute values of the correlation coefficients between the two vectors. It can be seen that the confidence feature vector generally has the highest correlation coefficient with the regression values among all feature vectors, significantly enhancing the predictive capability of the regression model and providing support for prior and posterior probability calculations and classification accuracy.



(**a**) Australian (**b**) German

(**c**) biodeg (**d**) magazine

(**e**) all-beauty (**f**) spam

(**g**) musk (**h**) Hilly

Figure 4. The result of ρ on different datasets. Note: The mark named R above the bars means the rank of confidence vector between all the feature vectors.

Readers may notice that the confidence vector of the SVM is often negatively correlated with its regression value vector. This indicates that as the confidence of the SVM increases,

the consistency ratio of its peer predictions decreases. This situation aligns precisely with the decision rule in our proposed DTS algorithm when $P < Q$: namely, to heed the opinions of experts who are highly confident in their decision and believe that most others are unaware of the correct answer. This further reinforces the decision-making concept of CTS-SR when $P < Q$ in our proposed method.

4.2.5. Experiment 5: Data Analysis

To further analyze the effectiveness of the CTS algorithm, we recorded the recovery of minority correct answers by CTS compared to majority voting, shown in Table 8. As seen from the previous results, the accuracy of each ensemble algorithm on the Hilly dataset is lower compared to other datasets. However, Table 8 shows that the proportion of correct answers recoverable by CTS is the highest among all datasets, further demonstrating that our proposed strategy can accurately recover minority answers in challenging decision-making scenarios, reducing the drawbacks of majority voting.

Table 8. Number and proportion of recoveries on each dataset.

Datasets	# of Recovery/# of Test	Proportion
Australian	3/345	0.9%
German	12/500	2.4%
spam	36/2301	1.6%
biodeg	10/528	1.9%
magazine	33/1187	2.8%
all-beauty	5/254	2.0%
musk	19/334	5.7%
Hilly	84/606	13.9%

# indicates that the value represents a quantity.

In this section, we evaluated CTS and CTS-SR on common binary classification datasets by measuring accuracy and F1 scores, and compared them with popular ensemble learning algorithms as well as traditional single-classifier models in supervised learning. The experimental results demonstrate that our proposed algorithms achieved significant advantages on most datasets, improving accuracy by 4–6% across diverse test data. Furthermore, the Friedman–Nemenyi test was used to rank the ensemble algorithms and revealed significant performance differences, highlighting the superiority and generality of our proposed methods. Additionally, through Experiment 5, we further demonstrated the ability of CTS to recover correct answers compared to the traditional majority voting strategy, indicating that our proposed method effectively addresses the drawbacks of majority decision making and underscores the importance of confidence in ensemble learning.

5. Conclusions

This paper proposes two new ensemble methods for supervised classification, CTS and CTS-SR, which are based on confidence and incorporate the "surprisingly popular" concept. By using confidence levels instead of direct classification results, we generalize HMTS to a more general case. CTS obtains the prior and posterior probabilities of an event by statistically analyzing the prediction consistency and confidence levels of learners. This allows us to determine whether to recover the correct answers held by a minority group, rather than simply applying majority voting. CTS-SR uses confidence to represent the real-world scenario where the correct answer is known, thereby achieving adherence to the correct decisions of a few experts with lower computational overhead. Experiments demonstrate the applicability of two proposed ensemble methods in various situations, and we explain the reasons behind the results. A key aspect of the CTS algorithm in this

paper is treating the classification confidence of base learners—an inherent attribute—as an additional one-dimensional feature of the samples. This enables the regression model to utilize sample information more comprehensively, leading to significant improvements in the following aspects:

(1) Enhancing feature representation capability: It can capture the uncertainty information of sample classifications under different classifiers, enriching the representation of sample features.
(2) Improving the model's robustness: Introducing classification confidence increases model diversity, reduces the risk of overfitting, and enhances robustness.
(3) Boosting prediction performance: Classification confidence reflects the consistency of samples under different classifiers. By integrating this information, samples can be predicted more accurately.

6. Limitations and Future Work

Although the two CTS algorithms we proposed demonstrate significant superiority in various scenarios, they still have certain limitations. Our algorithm introduces classification confidence as a key variable in the computation process. However, some classifiers do not inherently provide a method for calculating confidence on data, which necessitates us to define the confidence calculation formula based on the principles of the classifiers. This poses certain challenges to the accuracy and rationality of our computations. Additionally, the calculation of confidence somewhat increases the time overhead. Moreover, the current CTS algorithm is only applicable to supervised learning classification tasks, and its universality still needs improvement compared to the HMTS algorithm proposed in [8]. Research on semi-supervised and unsupervised classification algorithms based on the CTS concept, as well as applying our method to more real-life decision-making scenarios, remains an important part of our future work.

Author Contributions: H.S. drafted the main manuscript text, prepared Figure 1, Figure 2, Figure 3, Figure 4, Tables 1–5, provided the detailed proof process, proposed and refined the CTS-SR algorithm, supplied most of the datasets used in the experiments, and designed, implemented, and adjusted the experiments. X.W. proposed the ensemble strategy that combines the "surprisingly popular" algorithm with classification confidence, designed the method details, conducted data analysis, designed and adjusted the experiments, and finalized revisions of the manuscript text. Z.Y. formatted the paper according to the requirements, provided part of the datasets used in the experiments, and wrote the first section of the text. Y.Z. researched the current state of ensemble learning, collected relevant references, and provided the baseline. H.Z. provided the calculation process for confidence, implemented parts of the confidence calculation in the experiments. Y.Z. and H.Z. wrote the second section of the text. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Natural Science Foundation (4202002).

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Polikar, R. Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45. [CrossRef]
2. Dietterich, T.G. Ensemble Methods in Machine Learning. In Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000. [CrossRef]
3. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
4. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]

5. Ho, T.K.; Hull, J.J.; Srihari, S.N. Decision Combination in Multiple Classifiers Systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 66–75.
6. Prelec, D. A Bayesian Truth Serum for Subjective Data. *Science* **2004**, *306*, 462–466. [CrossRef] [PubMed]
7. Prelec, D.; Seung, H.S.; Mccoy, J. A solution to the single-question crowd wisdom problem. *Nature* **2017**, *541*, 532–535. [CrossRef] [PubMed]
8. Luo, T.; Liu, Y. Machine truth serum: A surprisingly popular approach to improving ensemble methods. *Mach. Learn.* **2022**, *112*, 789–815. [CrossRef]
9. Mccoy, J.; Prelec, D. A Bayesian Hierarchical Model of Crowd Wisdom Based on Predicting Opinions of Others. *Manag. Sci.* **2023**, *70*, 5931–5948. [CrossRef]
10. Hosseini, H.; Mandal, D.; Shah, N.; Shi, K. Surprisingly Popular Voting Recovers Rankings, Surprisingly! In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–27 August 2021. [CrossRef]
11. Schapire, R.E.; Singer, Y. Improved Boosting Algorithms Using Confidence-rated Predictions. *Mach. Learn.* **1999**, *37*, 297–336. [CrossRef]
12. Dietterich, T.G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Mach. Learn.* **2000**, *40*, 139–157. [CrossRef]
13. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
14. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
15. Webb, G.I.; Zheng, Z. Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 981–991. [CrossRef]
16. Zhang, D.; Jiao, L.; Bai, X.; Wang, S.; Hou, B. A robust semi-supervised SVM via ensemble learning. *Appl. Soft Comput.* **2018**, *65*, 632–643. [CrossRef]
17. Yu, Z.; Wang, D.; You, J.; Wong, H.-S.; Wu, S.; Zhang, J.; Han, G. Progressive Subspace Ensemble Learning. *Pattern Recognit.* **2016**, *60*, 692–705. [CrossRef]
18. Svargiv, M.; Masoumi, B.; Keyvanpour, M.R. A new ensemble learning method based on learning automata. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *13*, 3467–3482. [CrossRef]
19. Phama, K.; Kim, D.; Park, S.; Choi, H. Ensemble learning-based classification models for slope stability analysis. *Catena* **2021**, *196*, 104886. [CrossRef]
20. Gutierrez-Espinoza, L.; Abri, F.; Namin, A.S.; Jones, K.S.; Sears, D.R.W. Ensemble Learning for Detecting Fake Reviews. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020. [CrossRef]
21. Yang, X.; Xu, Y.; Quan, Y.; Ji, H. Image Denoising via Sequential Ensemble Learning. *IEEE Trans. Image Process.* **2020**, *29*, 5038–5049. [CrossRef] [PubMed]
22. Wang, T.; Zhang, Z.; Jing, X.; Zhang, L. Multiple Kernel Ensemble Learning for Softwaredefect prediction. *Autom. Softw. Eng.* **2016**, *23*, 569–590. [CrossRef]
23. An, N.; Ding, H.; Yang, J.; Au, R.; Ang, T.F.A. Deep ensemble learning for Alzheimer's disease classification. *J. Biomed. Inform.* **2020**, *105*, 103411. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.