ISSN 2063-5346

A ROBUST HETEROGENEOUS ENSEMBLE LEARNING FRAMEWORK FOR DISTRIBUTED DATA MINING ON BIOMEDICAL DATA SOURCES

S.Eswar Reddy[1], Biswajit Brahma[2], Dr. Subhendu Kumar Pani[3]

Article History: Received: 10.05.2023 Revised: 29.05.2023 Accepted: 09.06.2023

Abstract

Data mining methods have been proposed for finding hidden information in databases. When data is massive, dispersed, and heterogeneous, data mining and knowledge extraction become difficult. Classification is a common prediction task in data mining. Countless AI calculations have been proposed for the reason. Group learning consolidates numerous base classifiers to work on the exhibition of individual order calculations. In particular, distributed data mining relies heavily on ensemble learning. In this way, investigation of gathering learning is vital to apply it in true information mining issues. Ensemble learning is a well-established technique in machine learning that involves combining the predictions of multiple models to improve overall accuracy. In the context of distributed data mining, where data is spread across multiple locations or nodes, ensemble learning becomes more challenging. However, a novel approach to ensemble learning in distributed data mining has been proposed that addresses this challenge. In this paper, we propose a way to deal with build group of classifiers and review its presentation utilizing famous learning calculations on an assortment of freely accessible datasets from biomedical space.

Keywords: meta-learning, classifier ensemble, ensemble learning ensemble method, classification performance, meta-classifier.

[1]Asst.Professor,Comp.Sc ,United School of Business Management (USBM),Odisha, India
sereddy@gmail.Com

[2]Senior Data & Visualization Engineer, McKesson Corporation, USA
Biswajit.Brahma@gmail.com

[3]Professor, Krupajal Engineering College, Odisha, India
pani.subhendu@gmail.com

DOI:10.48047/ecb/2023.12.9.128

1. Introduction

Ensemble learning is a popular technique in machine learning that involves combining the predictions of multiple models to improve overall accuracy. In the context of distributed data mining[1,2], where data is spread across multiple locations or nodes, ensemble learning becomes more challenging. However, a novel approach to ensemble learning in distributed data mining has been proposed that addresses this challenge[3].

The approach involves using a hierarchical clustering algorithm to group similar data points together into clusters. Each cluster is then assigned to a different node, where a base model is trained on the data within that cluster. The base models are then combined using a consensus-based method to create an ensemble model.

The consensus-based method involves aggregating the predictions of the base models using a weighted voting scheme, where the weight assigned to each base model is proportional to its accuracy on the validation set. This ensures that the base models with higher accuracy have more influence on the final prediction.

To further improve the accuracy of the ensemble model, a meta-model is trained on the predictions of the base models[4,5]. The meta-model learns the relationship between the predictions of the base models and the true labels, and uses this relationship to make more accurate predictions on the test set.

Numerous pervasive distributed computing environments have emerged as a result of technological advancements in computing and communication over wired and wireless networks. In many of these environments, the amount of data available has increased dramatically. It prompts a huge scope information examination issue and offers a chance to foster mechanized information digging procedures for finding designs in the enormous information and removing helpful information from it. The issue of information mining is additionally exasperated because of the way that generally speaking, the information is disseminated over many registering hubs and stays heterogeneous. There are a number of reasons why data is distributed, like ownership and privacy. For instance, a few datasets concerning pivotal business data (for example Master card extortion, illegal tax avoidance) may be claimed by various associations found geologically in a few areas and they have certified motivations to keep the information hidden. They may, however, be interested in sharing these data for useful information and the company's benefit. As a result, the issues that modern data mining techniques face include not only the size of the data that needs to be mined, but also the distribution of that data and the heterogeneity of it.

A broadly taken on way to deal with the conveyed information mining issue is to apply different AI calculations utilizing equal and gradual learning strategies. In particular, numerous studies have utilized the meta-learning approach extensively[6]. Meta-learning is a form of ensemble learning in which a number of learning algorithms serve as base learners and construct local models from distributed data sources. A higher-level learning algorithm then combines these local models to create a final model of the distributed data.A general framework of ensemble learning in distributed data mining is shown in Figure-1.

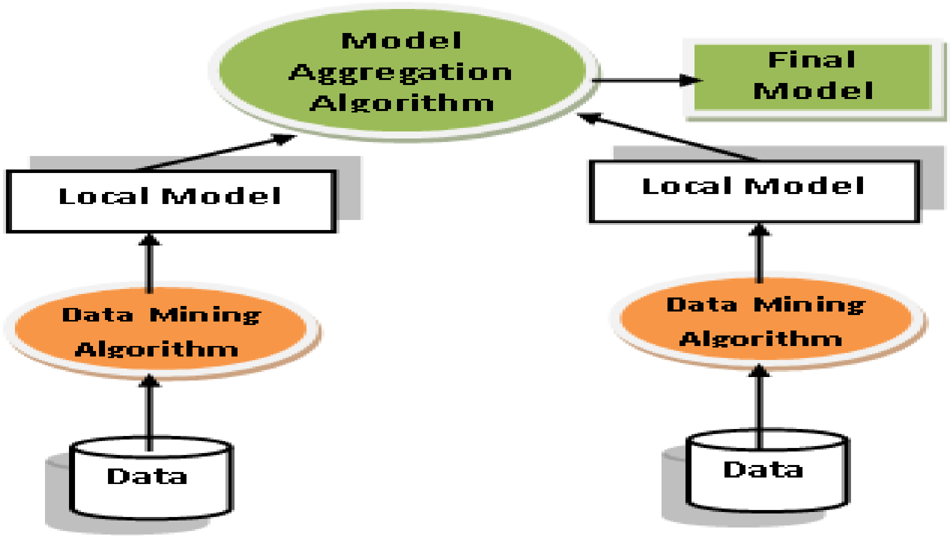

Figure-1: Ensemble Learning Framework

Numerous pervasive distributed computing environments have emerged as a result of technological advancements in computing and communication over wired and wireless networks. In many of these environments, the amount of data available has increased dramatically. It prompts a huge scope information examination issue and offers a chance to foster mechanized information digging procedures for finding designs in the enormous information and removing helpful information from it. The issue of information mining is additionally exasperated because of the way that generally speaking, the information is disseminated over many registering hubs and stay heterogeneous. There are a number of reasons why data is distributed, like ownership and privacy. For instance, a few datasets concerning pivotal business data (for example Master card extortion, illegal tax avoidance) may be claimed by various associations found geologically in a few areas and they have certified motivations to keep the information hidden. They may, however, be interested in sharing these data for useful information and the company's benefit. As a result, the issues that modern data mining techniques face include not only the size of the data that needs to be mined, but also the distribution of that data and the heterogeneity of it.

2. Fundamental Algorithms for Ensemble Learning:

Here are some examples of popular ensemble learning:

Bagging	Bagging, or Bootstrap Aggregating, is a popular ensemble learning technique that involves training multiple models on bootstrap samples of the training data and then aggregating their predictions to make a final prediction.
Boosting	Boosting is another popular ensemble learning technique that involves sequentially training multiple models, where each subsequent model is trained to correct the errors of the previous model.
Stacking	Stacking is a meta-ensemble learning technique that involves training multiple base models on the training data

	and then using their predictions as input to a higher-level model, which makes the final prediction
Diversity-based Ensemble Learning	Diversity-based ensemble learning techniques aim to increase the diversity of the base models by using different algorithms, subsets of features, or different parameter settings
Deep Ensemble Learning	Deep Ensemble Learning involves training multiple neural networks with different architectures or initialization weights, and then aggregating their predictions.
Bayesian Ensemble Learning	Bayesian Ensemble Learning involves using Bayesian methods to estimate the probability distribution over the model parameters, and then combining the predictions of multiple models sampled from this distribution.
Ensemble Learning for Imbalanced Data	Ensemble learning techniques have been adapted to handle imbalanced datasets, where the number of instances of one class is much larger than the other. Techniques such as boosting, bagging, and cost-sensitive learning have been used to address this problem

3. Our Approach

The study is carried out in a methodical manner, which is outlined in the following sections.

3.1 Datasets Used

The study's datasets come from the UCI Machine Learning Repository, which can be accessed by the general public. The sample is carefully chosen to represent diversity and a variety of statistical characteristics relevant to the biomedical application domain[7,8,9,10,11,12]. Table-1 sums up the qualities of the datasets. Detailed description of the used Dataset is shown in Fig 2.1,Fig 2.2 Fig 2.3.

Table-1: Dataset Characteristics

Dataset	#Instances	#Classes	#Attributes	#Missing Values (%)
Hypothyroid	3772	2	30	0
Prostate Cancer	21	2	12601	0
Ovarian Cancer	253	2	15155	0

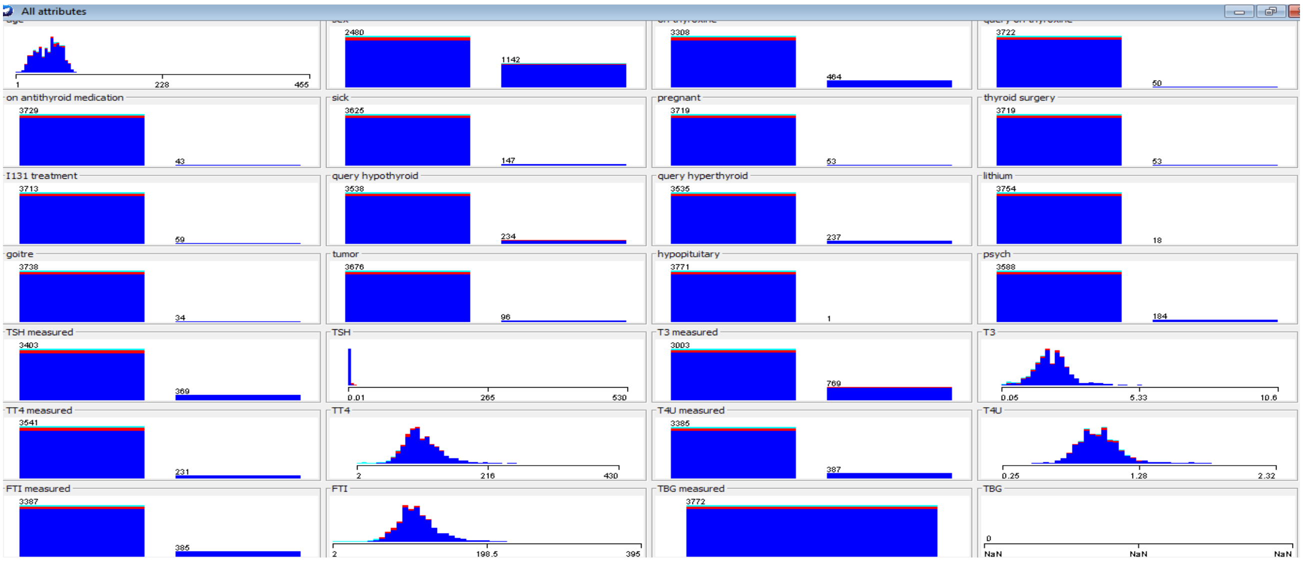

Fig 2.1: Hypothyroid Dataset

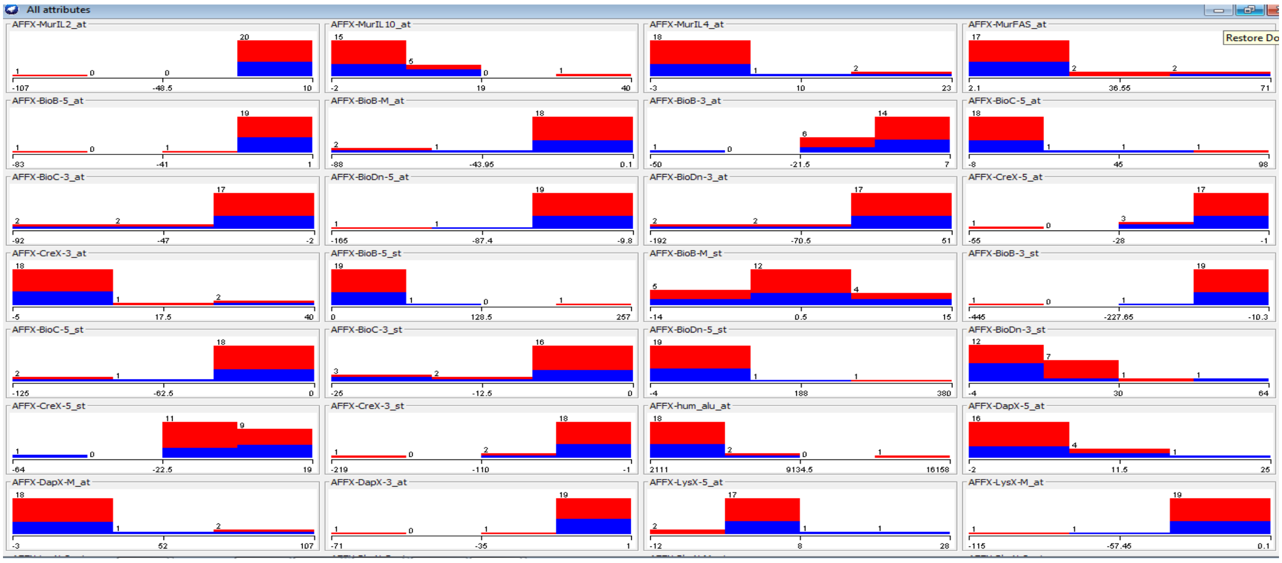

Fig 2.2: Prostate Cancer Dataset

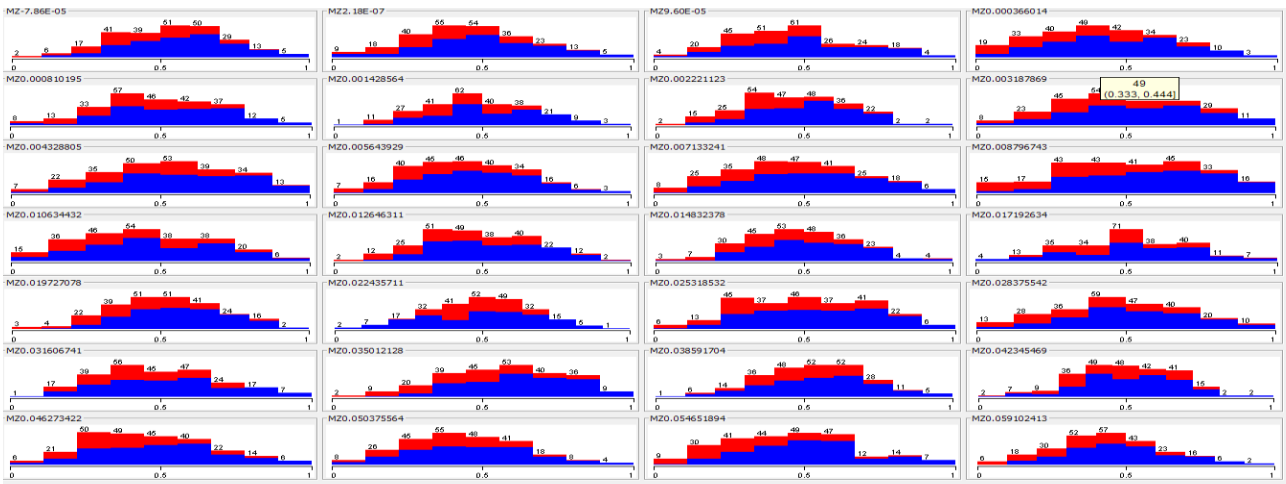

Fig 2.3: Ovarian Cancer Dataset

3.3 Ensemble Selection Techniques

Ensemble selection[13,14,15] is a technique in machine learning where multiple models, known as base learners or weak learners, are combined to create a stronger and more robust predictive model. The idea behind ensemble selection is that by combining the predictions of multiple models, the resulting ensemble can often outperform any individual model in terms of accuracy and generalization.

There are different approaches to ensemble selection, but one common technique is known as model averaging. In model averaging, each base learner is trained on a different subset of the training data or with different configurations, and their predictions are combined by averaging or voting. This approach helps reduce the impact of individual model biases and errors.

Another approach is called stacking or meta-learning, where the predictions of the base learners are used as features to train a meta-model, also known as a meta-classifier or combiner. The meta-model learns how to best combine the predictions of the base learners to make the final prediction. This technique can be particularly effective when the base learners have complementary strengths or expertise in different areas of the input space.

Ensemble selection can also involve dynamically selecting subsets of the base learners to form ensembles. This can be done based on their performance on a validation set or through more advanced techniques like genetic algorithms or optimization algorithms.

The benefits of ensemble selection include improved generalization, reduced over fitting, increased stability, and better handling of complex or noisy datasets. By combining the predictions of multiple models, ensemble selection can capture different aspects of the data and provide a more comprehensive and accurate prediction.

It's important to note that ensemble selection requires training and maintaining multiple models, which can increase computational complexity and memory requirements. However, the benefits of improved performance often outweigh these drawbacks, especially in scenarios where accuracy is crucial.

A set of three popular meta-classifiers is selected for ensemble learning and comparative evaluation. They are Decorate, AdaBoost and Bagging. A brief description of each of these algorithms is provided here.

3.3.1 Meta Classifier

DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) is an algorithm that is used to create diverse ensembles of classifiers. It was introduced by Melville and Mooney in 2005.

DECORATE is based on the idea of generating artificial training examples that are specifically designed to challenge the base classifier. These artificial examples are created by modifying the existing training examples using a process called "oppositional relabeling." The goal is to create diverse hypotheses by providing the base classifier with challenging and contrasting examples.

The algorithm follows these steps:

1. Generate an initial ensemble of classifiers using the base classifier.
2. Generate artificial training examples by relabeling the original training examples based on their misclassification patterns.

3. Add the artificial examples to the training set and retrain the ensemble of classifiers.
4. Evaluate the performance of the ensemble on a validation set and select the best classifiers.
5. Repeat steps 2-4 until a stopping criterion is met (e.g., a predefined number of iterations or convergence).
6. Combine the selected classifiers to form the final ensemble.

AdaBoost (Adaptive Boosting) is a popular boosting algorithm that trains a sequence of classifiers in succession. It was introduced by Freund and Schapire in 1996.The basic idea behind AdaBoost is to iteratively train a series of weak classifiers, where each weak classifier focuses on the instances that were difficult to classify by the preceding classifiers. In AdaBoost, "weak classifiers" typically refer to classifiers that perform slightly better than random guessing, such as decision stumps (simple decision trees with a single split).

The AdaBoost algorithm follows these steps:

1. Initialize the weights for each training instance to be equal.
2. For each iteration:
a. Train a weak classifier on the training data, giving more importance to instances that were misclassified by the previous classifiers. The weight of each instance is adjusted based on its misclassification rate.
b. Calculate the error rate of the weak classifier, which represents how well it performed on the training data.
c. Compute the weight of the weak classifier based on its error rate. The weight indicates the importance of the weak classifier in the ensemble.
d. Update the weights of the training instances, increasing the weights of misclassified instances to focus on the difficult examples.
3. Repeat steps 2 until a predefined number of iterations is reached or a termination condition is satisfied.
4. Combine the weak classifiers by assigning weights to each classifier based on their performance, creating the final boosted ensemble.

By iteratively adjusting the weights of the training instances and giving more importance to difficult examples, AdaBoost focuses on learning from the mistakes made by the previous classifiers. The final ensemble is a weighted combination of the weak classifiers, where the weight of each classifier depends on its accuracy in the training process.

Bagging (Bootstrap Aggregating) is a popular ensemble learning technique that involves generating multiple subsets of the original training set through random sampling with replacement. These subsets, called bootstrap samples, are used to train individual classifiers. The final prediction of the ensemble is typically determined by aggregating the predictions of each classifier through a majority vote.

The steps involved in bagging are as follows:

1. Generate multiple bootstrap samples: Randomly select subsets of the original training set by sampling with replacement. Each bootstrap sample has the same size as the original training set, but some instances may be repeated, while others may be omitted.
2. Train individual classifiers: For each bootstrap sample, train a separate classifier using a chosen learning

algorithm. These classifiers are typically referred to as base classifiers or weak learners. Each classifier is trained independently on its respective bootstrap sample.
3. Make predictions: Use the trained classifiers to make predictions on new unseen instances or the test set.
4. Aggregate predictions: Combine the predictions of the individual classifiers using a majority vote (for classification problems).

The class label that receives the most votes across all classifiers is selected as the final prediction of the bagging ensemble. For regression problems, the predictions of the individual classifiers can be averaged to obtain the final prediction.

The key idea behind bagging is that by generating multiple bootstrap samples and training separate classifiers on them, it introduces diversity in the training process. Each classifier focuses on different subsets of the data, potentially capturing different patterns and reducing the variance of the ensemble's predictions. The majority vote or averaging of predictions helps to consolidate the diverse opinions of the individual classifiers and produce a more robust and accurate final prediction.

Bagging can be applied to various learning algorithms, such as decision trees, random forests, and support vector machines. It is particularly effective when the base classifiers are unstable or have high variance. By reducing variance and improving generalization, bagging can help improve the overall performance and robustness of the ensemble.

It's worth noting that while bagging can improve performance, it does not address the issue of bias in the base classifiers. Boosting algorithms, such as AdaBoost, are designed to handle bias and focus on difficult examples by adjusting instance weights. Bagging and boosting are two popular ensemble learning techniques, each with its own strengths and characteristics.

The process of ensemble selection typically involves the following steps:

Model Generation: Generate a set of base models by applying different learning algorithms or by using the same algorithm with different parameter settings or subsets of the training data.

Model Evaluation: Evaluate the performance of each base model using an appropriate evaluation metric (e.g., accuracy, precision, recall, etc.) on a validation set or through cross-validation.

Ensemble Construction: Select a subset of the base models based on their individual performance. This selection can be done using various approaches such as greedy algorithms, meta heuristic optimization techniques, or statistical methods.

Ensemble Combination: Combine the selected base models to form the final ensemble model. The combination can be performed using different strategies, such as averaging the predictions of the base models, weighted averaging, or using more advanced techniques like stacking or boosting.

3.4 Weka Spark

Weka is a popular open-source machine learning and data mining tool that provides a comprehensive set of algorithms and tools for data preprocessing, classification, regression, clustering, association rules mining, and feature selection. It is widely used for data analysis and predictive modeling tasks. On the other hand, Spark is an open-source distributed computing system designed for big data processing and analytics. It provides a programming model that allows for distributed data processing and in-memory computing, making it wellsuited for large-scale data processing tasks. While Weka and Spark are separate tools, they can be used together to leverage the capabilities of both. In fact, there is a project called "Weka-Spark" that aims to integrate Weka with Apache Spark, allowing users to perform distributed machine learning tasks using the algorithms and tools provided by Weka on large-scale datasets.

3.5 Experiment Design

Our analysis assesses three meta-classifiers for staggered outfit discovering that thinks about four base classifiers. However, in order to reduce computational complexity, we limit ensemble construction to two levels. A level-1 meta-classifier and a base classifier are used in the single ensemble learning, such as Decorate + J48, while a level-2 meta-classifier, a level-1 meta-classifier, and a base classifier are used in the layered ensemble learning, such as Decorate + Bagging + J48.

To measure classification accuracy, we carry out the experiment using a test mode of 10-fold cross validation. The 10-overlap cross approval evades one-sided results and gives vigor to the arrangement. In addition, the default values of a classification algorithm's parameters are chosen.

The classifier ensemble is constructed and the performance is studied using the following steps.

Step-1: Using Weka and Weka Spark, run each of the candidate classification algorithms one by one on each of the datasets to record their classification accuracy.

Step-2: Select the classifiers for the Weka Spark environment that consistently outperform Weka in terms of accuracy across the datasets. For ensemble learning, it is assumed that these classifiers serve as base classifiers.

Step-3: Form dataset situations by utilizing blends of picked base classifiers with all the datasets.

Step-4: Run all the outfit calculations on the planned dataset situations to record arrangement precision of the single gathering approach in Weka Spark climate

Step-5: Utilizing various ensemble algorithm combinations, create a two-layered ensemble.

Step-6: In the Weka Spark environment, run each of the two-layered ensembles on the dataset scenarios to record the classification performance.

The setup of Weka Information Stream Climate utilizing Weka Spark to run classifiers is displayed in Figure-2.

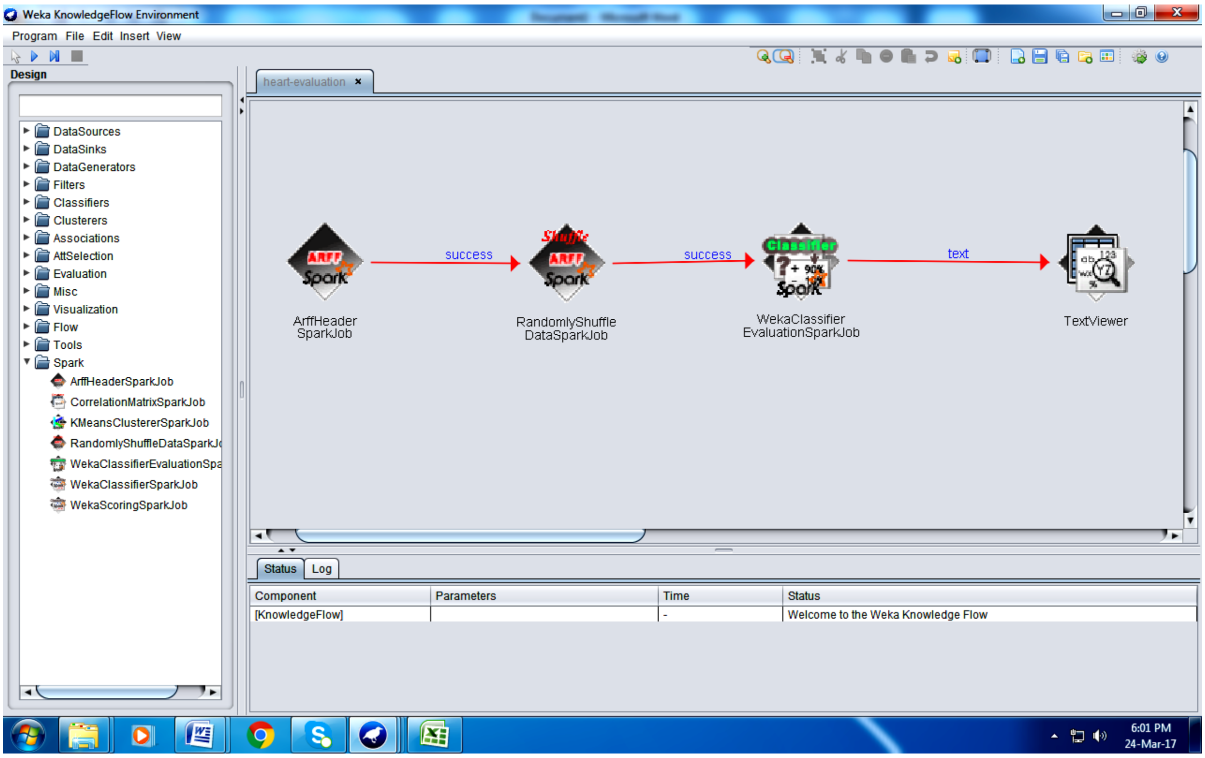

Figure-2: Weka Spark Configuration

4. Result Analysis

The classification accuracy of the candidate classifiers on the selected datasets after applying step-1 of the experiment are shown in Figure-3. Two classifiers i.e. NB and SMO are knocked out in the step-2 of our experiment as they fail to perform consistently across the datasets. Only IBk and J48 are considered for subsequent stages of the experimental study.

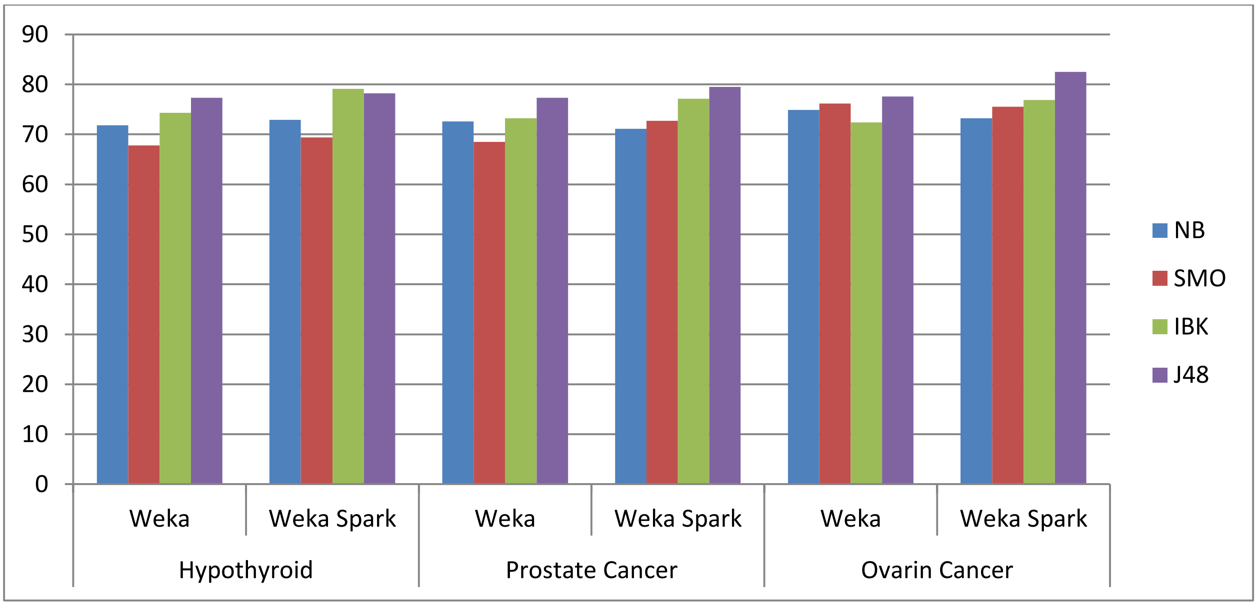

Figure-3: Performance of candidate classifiers based on Acuracy

Tool	Dataset	NB	SMO	IBK	J48
WEKA	Hypothyroid	71.8	67.8	74.3	77.3
WEKA	Prostate Cancer	72.6	68.5	73.2	73.4
WEKA	Ovarian Cancer	74.9	76.2	72.4	77.6
WEKA-SPARK	Hypothyroid	72.9	69.4	79.1	78.2
WEKA-SPARK	Prostate Cancer	71.1	72.7	77.1	79.5
WEKA-SPARK	Ovarian Cancer	73.2	75.5	76.9	82.5

The dataset scenarios formulated using these two classifiers for ensemble learning are shown in Table-2.

Table-2: Dataset Scenarios

Dataset	Classifier	Scenario
Hypothyroid	J48	HYJ48
	IBk	HYIBk
Prostate Cancer	J48	PJ48
	IBk	PIBk
Ovarian Cancer	J48	OVJ48
	IBk	OVIBk

The accuracy percentage of single ensemble learning against selected base classifiers in different dataset scenarios is shown in Table-3 and performance of single ensemble learning in terms of percentage of improvement in accuracy is depicted in Figure-4. It reveals that classification accuracy of Decorate ensemble is consistently significant across the scenarios. Also AdaBoost provides better accuracy than that of the base classifier except one scenario (i.e. BCJ48) wherein there is no change in accuracy. However, Bagging algorithm has inconsistent performance.

Table-3: Accuracy of Single Ensemble vs. Base Classifier

Dataset Scenario	Decorate	AdaBoost	Bagging	Base Classifier
HYJ48	82.7403	79.5934	78.5445	77.5734
HYIBk	86.3168	83.1699	83.5196	80.7224
PJ48	80.9779	80.0665	82.28	79.2552
PIBk	87.6052	81.6573	81.7458	81.3552
OVJ48	89.5385	88.7978	86.8448	87.5856
OVIBk	88	81.8037	83.6656	81.4433

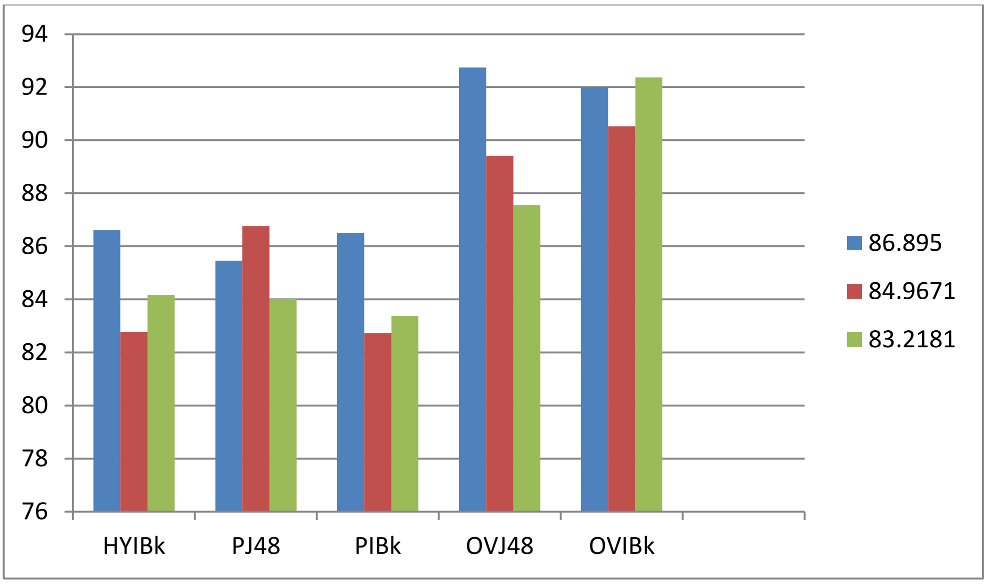

Figure-4: Accuracy Improvement of Single Ensemble

The accuracy percentage of layered ensemble learning against base classifiers on different dataset scenarios is shown in Table-4 and performance of single ensemble learning in terms of percentage of improvement in accuracy is depicted in Figure-5. It shows that ensemble learning with Decorate + AdaBoost combination provides significantly better performance and consistent across the scenarios. It also shows that ensemble learning with Decorate + Bagging combination is somewhat better and consistent across the scenarios. However, other combinations such as AdaBoost + Decorate, Bagging + Decorate, AdaBoost + Bagging and Bagging + AdaBoost show negative performance. It can be established from this experimental data that Decorate when placed as level-2 meta-classifier performs well consistently across the datasets considered in the study and improves the classification accuracy between two to eleven percent (i.e. 2% - 11%) as compared to the accuracy achieved by the base level classifier.

Table-4: Accuracy of Layered Ensemble vs. Base Classifier

Dataset Scenario	Decorate + AdaBoost	Decorate + Bagging	Bagging + Decorate	Bagging + AdaBoost	AdaBoost + Decorate	AdaBoost + Bagging	Base Classifier
HYJ48	86.895	84.9671	83.2181	84.5196	82.7713	81.7224	78.6734
HYIBk	86.6154	82.7692	84.1678	76.8252	79.2727	80.3217	79.6224
PJ48	85.4635	86.7656	84.0313	83.5104	83.1198	82.9896	80.0365
PIBk	86.5052	82.7292	83.3802	80.6458	78.1719	80.125	80.2552
OVJ48	92.7407	89.4079	87.5556	89.4074	86.4444	88.2963	87.7778
OVIBk	92	90.5185	92.3704	87.5556	85.3333	87.1852	85.3333

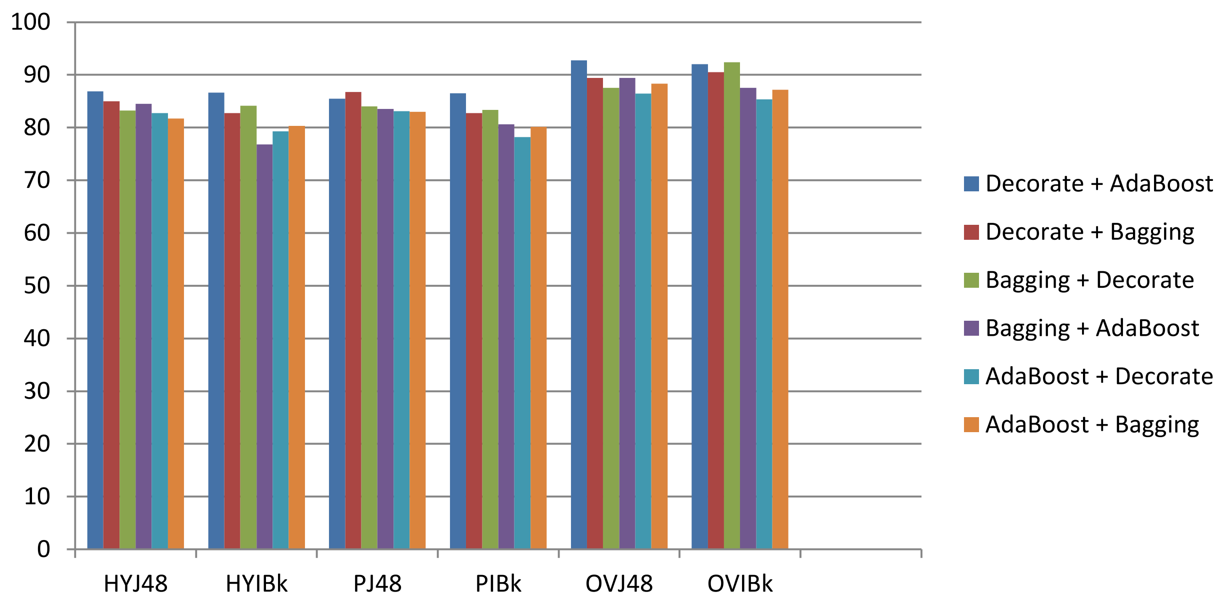

Figure-5 Accuracy Improvement of Layered Ensemble

5. Conclusion

In this research, we chose two base classifier out of four thought about at first. On biomedical datasets, the chosen base classifiers were combined with well-known meta-classifiers in multi-level ensemble learning. The distributed framework Weka Spark was used for the simulations. The generated data lead us to the conclusion that the Decorate algorithm performs exceptionally well and offers significant classification accuracy in comparison to that of the individual base classifiers. When used as a level-2 classifier in the multi-level ensemble construction, it exhibits further improvements in its classification performance across the datasets. However, when it is used as the lavel-1 meta-classifier in the ensemble, its classification performance becomes inconsistent. In any case Enhance performs well and is the most ideal decision for all chose datasets. It very well may be additionally said that meta-learning further develops order precision over base-learning and the exact information got in this study offers areas of strength for a to it.

The proposed method has been tested on a number of benchmark datasets, and the results show that it is more accurate than existing ensemble learning methods. In addition, the strategy is scalable and capable of handling massive datasets spread across multiple nodes. In conclusion, the proposed approach to ensemble learning in distributed data mining offers a promising solution to the challenges of combining models trained on data distributed across multiple nodes.

References:

1. Fu, Y. Distributed data mining: An overview. Newsl. IEEE Tech. Comm. Distrib. Process. 2001, 4, 5–9.
2. Moshkov, M. Decision trees and reducts for distributed decision tables. In Proceedings of the Monitoring, Security, and Rescue Techniques in Multi agent Systems, MSRAS 2004, Plock, Poland, 7–9 June 2004; Advances in Soft Computing; Dunin-Keplicz, B., Jankowski, A., Skowron, A., Szczuka, M.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 28, pp. 239–248.
3. S.; Nandhini, M. A framework for distributed data mining heterogeneous classifier. Comput.
Commun. 2019, 147, 58–75.

4. Meta-Health: Learning-to-Learn (Meta-learning) as a Next Generation of Deep Learning Exploring Healthcare Challenges and Solutions for Rare Disorders: A Systematic Analysis 2023, Archives of Computational Methods in Engineering
5. A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning, Neurocomputing, Volume 349, 2019, pp. 202-211.
6. Meta-learning in natural and artificial intelligence.Current Opinion in Behavioral Sciences, Volume 38, 2021, pp. 90-95
7. Aderghal, K., Boissenin, M., Benois-Pineau, J., Catheline, G., & Karim, A., 2017, 01, Classification of sMRI for AD Diagnosis with Convolutional Neuronal Networks: A Pilot 2-D+e Study on ADNI. In (p. 690–701).
8. Böhle M., Eitel F., Weygandt M., Ritter K. Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based Alzheimer's Disease classification. Front. Aging Neurosci. 2019:194.
9. Cheng, D., & Liu, M., 2017, 09, Classification of Alzheimer's Disease by cascaded convolutional neural networks using pet images. In (p. 106–113).
10. Duc N.T., Ryu S., Qureshi M.N.I., Choi M., Lee K.H., Lee B. 3d-deep learning based automatic diagnosis of Alzheimer's Disease with joint MMSE prediction using resting-state fMRI. Neuroinformatics. 2020;18(1):7 1–86.]
11. Etminani K., Soliman A., Davidsson A., Chang J.R., Martínez-Sanchis B., Byttner S., Ochoa-Figueroa M. A 3D deep learning model to predict the diagnosis of dementia with Lewy bodies, Alzheimer's Disease, and mild cognitive impairment using brain 18F-FDG PET. Eur. J. Nucl. Med. Mol. Imaging. 2022;49(2):563–584.
12. Farooq A., Anwar S., Awais M., Rehman S. A deep cnn based multi-class classification of Alzheimer's Disease using mri. 2017 IEEE Int. Conf. Imaging Syst. Tech. (Ist.) 2017:1–6.
13. Shirai, S., Kudo, M., Nakamura, A.: Bagging, Random Subspace Method and Biding. LNCS, vol. 5342, pp. 801–810. Springer, Heidelberg (2008).
14. Skurichina, L., Kuncheva, L.: An Experimental Study on Diversity for Bagging and Boosting with Linear Classifiers. Information Fusion 3, 245–258 (2002)
15. Redpadth, D.B., Levart, K.: Boosting Feature Selection. LNCS, vol. 3086, pp. 305–314. Springer, Heidelberg (2005)