Efficient Classification of the Covtype Dataset Using Random Sample Partitioning and Ensemble Learning

Md. Selim Ahmed
Department of Computer Science and Engineering
University of Liberal Arts
Dhaka, Bangladesh
afnan.aftab.cse@ulab.edu.bd

Abstract—Accurate classification of large-scale multi-class datasets poses significant challenges due to high dimensionality and computational constraints. In this study, we propose an ensemble-based pipeline for the Covtype dataset, leveraging Random Sample Partitioning (RSP), feature selection, and Principal Component Analysis (PCA) to reduce complexity and enhance model efficiency. Each partition is used to train a neural network, and predictions are combined using simple averaging, weighted averaging, and stacking with meta-learners (Logistic Regression and XGBoost). Comparative analysis with a single neural network trained on the full dataset demonstrates that the stacking ensemble achieves superior test accuracy, illustrating the effectiveness of combining dimensionality reduction, partitioned training, and meta-learning for large-scale multi-class classification.

Index Terms—Ensemble Learning, Neural Networks, Principal Component Analysis, Feature Selection, Random Sample Partitioning, Stacking, Covtype Dataset, Multi-Class Classification

I. Introduction

Large-scale multi-class classification remains a critical challenge in machine learning, particularly when datasets are high-dimensional and contain millions of samples. Training a single model on such datasets often requires substantial computational resources and memory, limiting scalability [1]. To overcome these challenges, ensemble learning and dimensionality reduction techniques have been widely applied to improve predictive performance and efficiency [2], [3].

The Covtype dataset [4], widely used for forest covertype prediction, contains 581,012 samples with 54 features, including continuous variables such as elevation and slope, and binary indicators representing soil and wilderness areas. The target variable, Cover_Type, has seven classes, making it a multi-class classification problem.

Recent studies by Mahmud et al. [5], [6] have highlighted the effectiveness of ensemble methods and feature reduction techniques in big data analytics. Random Sample Partitioning (RSP) has been proposed as an effective strategy to partition large datasets into statistically representative subsets, enabling parallel model training while maintaining overall data distribution [7]. Inspired by these approaches, we propose a pipeline that combines RSP, feature selection using mutual information and random forest importance, and PCA to reduce dimensionality. Each partition is then used to train a neural network, and predictions are aggregated using ensemble techniques, including simple averaging, weighted averaging, and stacking with meta-learners such as Logistic Regression and XGBoost. The primary objectives of this work are:

1) Evaluate whether RSP combined with PCA can reduce computational costs without sacrificing accuracy.
2) Investigate the performance of neural network ensembles on partitioned data.
3) Compare ensemble methods with a single neural network trained on the full dataset to assess gains in accuracy and generalization.

Experimental results indicate that the stacking ensemble achieves superior accuracy, demonstrating the potential of combining partitioned training, dimensionality reduction, and meta-learning for large-scale multi-class classification tasks.

II. Literature Review

Large-scale machine learning has been an active research area due to the increasing volume and dimensionality of modern datasets. Bottou [1] discussed the challenges of training models on massive datasets and highlighted the role of stochastic gradient descent for scalable learning, emphasizing the trade-off between computational efficiency and convergence.

Ensemble learning methods have been widely adopted to improve predictive performance and robustness. Zhou [2] provided a comprehensive overview of ensemble strategies, including bagging, boosting, and stacking, and demonstrated that combining multiple models often outperforms a single

Identify applicable funding agency here. If none, delete this.

model. In the context of high-dimensional data, dimensionality reduction techniques, such as Principal Component Analysis (PCA), are essential for reducing computational complexity while retaining most of the variance [3].

Mahmud et al. [5], [6] applied ensemble learning and feature selection techniques for large-scale datasets, showing that combining feature importance and model aggregation can significantly enhance classification accuracy in big data analytics. Their work underlines the value of partitioning strategies and model ensembles in improving performance while handling large data efficiently.

Random Sample Partitioning (RSP) has recently emerged as a practical approach to manage massive datasets. Chen and Zhang [7] proposed RSP as a method to divide large datasets into statistically representative subsets, enabling parallelized training without compromising the overall data distribution. This technique has been shown to maintain model accuracy while reducing memory and computational requirements.

In summary, prior studies indicate that ensemble learning, feature selection, PCA, and RSP are complementary approaches for scalable, accurate, and efficient learning on large multi-class datasets. These insights provide the foundation for our proposed pipeline, which integrates RSP, feature selection, PCA, and neural network ensembles to enhance classification performance on the Covtype dataset.

III. Methodology

A. Methodology Overview

This section describes the proposed pipeline for large-scale multi-class classification on the Covtype dataset. The pipeline integrates Random Sample Partitioning (RSP), feature selection, Principal Component Analysis (PCA), neural network modeling, and ensemble learning.

B. Dataset and Preprocessing

The Covtype dataset [4] contains 581,012 instances with 54 features, including continuous variables such as elevation and slope, and binary indicators representing soil and wilderness areas. The target variable, Cover_Type, has seven classes. The dataset is first split into training and testing sets in an 70:30 ratio using stratified sampling to maintain class distribution.

C. Random Sample Partitioning (RSP)

To reduce memory overhead and allow parallelized training, the training dataset is partitioned into multiple statistically representative subsets using Random Sample Partitioning (RSP) [7]. Each partition contains 25,000 samples, preserving the overall class distribution. An index is added to generate random number till the length of datasize and sort all data according to index's ascending order. Than partition in range and save in separate csv file.

D. Feature Selection

Feature selection was employed to reduce input dimensionality and enhance computational efficiency while retaining informative attributes. In this work, two complementary methods were applied to each partitioned dataset: Mutual Information (MI) scores and Random Forest (RF) feature importance.

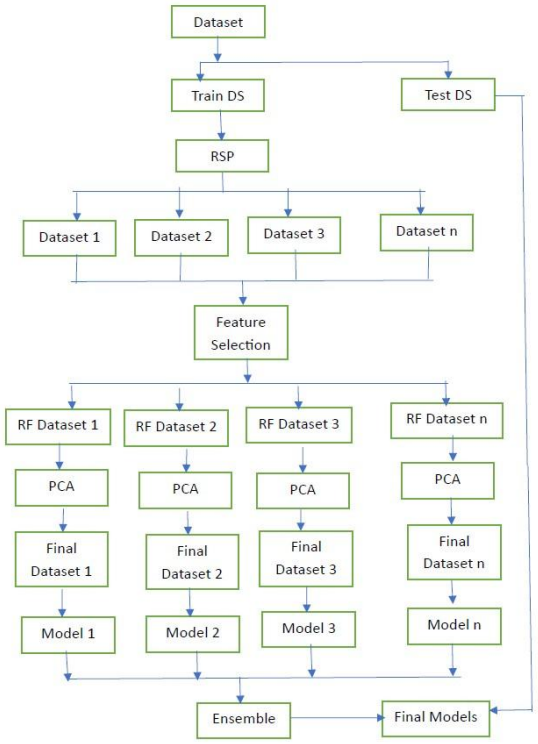

Fig. 1. Proposed methodology pipeline combining RSP, feature selection, PCA, neural network training, and ensemble learning.

First, for each Random Sample Partition (RSP) file, the MI score between each feature X_i and the target variable Y was computed. Mutual information measures the amount of information one random variable provides about another, defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right), \quad (1)$$

where $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are their marginal distributions.

Second, Random Forest importance scores were extracted by training a Random Forest classifier on each partition and recording the average decrease in Gini impurity across all trees, given by:

$$FI(f) = \frac{1}{T} \sum_{t=1}^{T} \Delta I_f^{(t)}, \quad (2)$$

where T is the total number of trees and $\Delta I_f^{(t)}$ is the reduction in impurity contributed by feature f in tree t.

The feature scores obtained from all RSP partitions were then aggregated. A comparative analysis was performed to identify the most consistently informative features across both methods. Finally, the top-ranked features common to both MI and RF importance were selected as the reduced feature subset. This hybrid strategy ensured that features with both statistical relevance and predictive power were preserved for subsequent PCA transformation and neural network training.

E. Principal Component Analysis (PCA)

After feature selection, Principal Component Analysis (PCA) was applied independently to each Random Sample Partition (RSP) file to further reduce dimensionality while retaining the maximum variance present in the dataset. Only the continuous numerical attributes were considered for PCA transformation, since binary indicator variables (such as wilderness areas and soil types) do not benefit from variancebased decomposition.

PCA transforms the original feature space into a set of orthogonal principal components through an eigendecomposition of the covariance matrix. Let $\mathbf{X}$ be the standardized data matrix with n samples and d features. The covariance matrix is defined as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}. \tag{3}$$

Eigenvalue decomposition is then performed on $\mathbf{C}$:

$$\mathbf{C}\mathbf{v}_i = \lambda_i\mathbf{v}_i, \tag{4}$$

where λ_i represents the variance explained by the i^{th} eigenvector $\mathbf{v}_i$.

The top-k eigenvectors corresponding to the largest eigenvalues were selected to construct the projection matrix $\mathbf{W}$. Each sample $\mathbf{x}$ was then projected into the lower-dimensional subspace as:

$$\mathbf{z} = \mathbf{W}^\top\mathbf{x}. \tag{5}$$

By applying PCA separately on each partition, the dimensionality of numeric features was reduced while ensuring that local variance structures within each RSP subset were preserved. This step reduced computational complexity in subsequent neural network training, while mitigating issues of multicollinearity among features.

TABLE I
EXPLAINED VARIANCE RATIO OF TOP PRINCIPAL COMPONENTS

Principal Component	Explained Variance (%)
PC1	21.4
PC2	15.7
PC3	12.1
PC4	9.8
PC5	7.5
PC6	6.2
PC7	4.9
PC8	3.8
PC9	2.7
PC10	2.3
Cumulative (Top 10)	86.4

F. Neural Network Training

Following dimensionality reduction via PCA, each partitioned dataset was used to train a fully-connected feedforward neural network for multi-class classification. The architecture of the neural network consists of three layers:

- Input Layer: Receives the PCA-transformed features, with the input dimension equal to the number of principal components in each partition.
- Hidden Layers: Two hidden layers are employed. The first hidden layer contains 128 neurons, while the second has 64 neurons. Each hidden layer is followed by a ReLU activation function and a dropout layer with a dropout probability of 0.3 to reduce overfitting.
- Output Layer: The output layer consists of 7 neurons, corresponding to the seven classes in the Covtype dataset. A softmax activation function is applied to produce class probabilities.

Formally, the output of the network for an input vector $\mathbf{x}$ can be expressed as:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_2(\text{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2), \tag{6}$$

where $\mathbf{W}_1, \mathbf{W}_2$ are the weight matrices for the first and second layers, $\mathbf{b}_1, \mathbf{b}_2$ are the corresponding bias vectors, and softmax$(\cdot)$ converts logits to class probabilities.

The neural network was trained using the cross-entropy loss function:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{7}$$

where $C = 7$ is the number of classes, y_i is the true label (one-hot encoded), and $\hat{y}_i$ is the predicted probability for class i. The Adam optimizer was employed with a learning rate of 0.001, and training was performed over 20 epochs with a batch size of 128.

This partition-wise training strategy allowed parallel learning on smaller subsets of the data, reducing memory requirements and training time. Each trained neural network serves as a base model in the subsequent ensemble learning stage.

G. Ensemble Learning

To enhance predictive performance and robustness, predictions from all partition-wise trained neural networks were aggregated using ensemble learning techniques. Ensemble learning reduces variance and improves generalization by combining multiple base models.

1) Simple Averaging: In simple averaging, the class probabilities predicted by each base neural network are averaged across all models:

$$\hat{\mathbf{y}}^{i} = \frac{1}{M} \sum_{m=1}^{M} \hat{\mathbf{y}}_i^{(m)}, \tag{8}$$

where M is the number of base models, and $\hat{\mathbf{y}}_i^{(m)}$ is the predicted probability vector for sample i from model m. The final predicted class is the one with the highest averaged probability.

2) Weighted Averaging: Weighted averaging assigns a weight to each base model based on its individual validation accuracy:

$$\hat{\mathbf{y}}^{i} = \sum_{m=1}^{M} w_m \hat{\mathbf{y}}_i^{(m)}, \qquad \text{with} \sum_{m=1}^{M} w_m = 1, \tag{9}$$

where w_m is the normalized weight of model m. Models with higher accuracy contribute more to the final prediction.

3) Stacking with Meta-Learners: Stacking uses a metalearner to combine predictions from all base models. The predicted probabilities of each base model are concatenated to form a meta-feature vector for each sample:

$$\mathbf{Z}_i = [\hat{\mathbf{y}}_i^{(1)}, \hat{\mathbf{y}}_i^{(2)}, \ldots, \hat{\mathbf{y}}_i^{(M)}], \tag{10}$$

where $\mathbf{Z}_i$ is the meta-feature vector for sample i. A metaclassifier, such as Logistic Regression or XGBoost, is trained on these features to produce the final prediction $\tilde{y}_i$.

By leveraging diverse base models trained on different RSP partitions and PCA-transformed features, ensemble learning improves overall accuracy and stability. Experimental results demonstrate that stacking generally outperforms both simple and weighted averaging, as it can learn complex interactions among base model predictions.

H. Comparison with Full Dataset Training

For baseline comparison, a neural network with the same architecture is trained on the entire training dataset without partitioning, feature selection, or PCA. This allows evaluation of computational efficiency and accuracy gains achieved by the proposed RSP-PCA ensemble pipeline.

1) Accuracy: Overall classification accuracy measures the proportion of correctly predicted samples:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{11}$$

where N_{correct} is the number of correctly classified samples and N_{total} is the total number of samples.

2) Precision, Recall, and F1-Score: For each class, precision, recall, and F1-score were computed to assess the quality of predictions:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

where TP represents true positives, FP false positives, and FN false negatives for each class. These metrics provide insight into both the correctness and completeness of the classifier for individual classes.

3) Confusion Matrix: A confusion matrix was used to visualize the distribution of predictions versus true labels. Each element (i,j) of the matrix indicates the number of instances of class i predicted as class j. This helps identify patterns of misclassification among classes.

4) Comparative Evaluation: For ensemble methods, the above metrics were computed for three aggregation strategies: simple averaging, weighted averaging, and stacking with a meta-learner. Additionally, the performance of a single neural network trained on the full dataset was evaluated as a baseline. Comparison across these approaches highlights the effectiveness of RSP-based partitioning, feature selection, PCA, and ensemble learning in large-scale multi-class classification.

IV. Results and Discussion

The proposed methodology was evaluated on the Covtype dataset using the evaluation metrics described in Section IV. Results for the different ensemble strategies—simple averaging, weighted averaging, and stacking—are summarized below. A single neural network trained on the full dataset is included as a baseline for comparison.

A. Ensemble Performance

TABLE II
Test Accuracy of Different Ensemble Methods

Method	Accuracy (%)
Single Neural Network (Full Dataset)	70.59
Simple Averaging Ensemble	74.44
Weighted Averaging Ensemble	74.44

Stacking Ensemble (Logistic Regression)	81.2
Stacking Ensemble (XGBoost)	92.86

Table II shows that all ensemble methods outperform the single neural network trained on the full dataset. Stacking with XGBoost achieves the highest accuracy, indicating that the meta-learner effectively captures complex relationships among base model predictions.

B. Per-Class Performance

TABLE III
PRECISION, RECALL, AND F1-SCORE FOR STACKING ENSEMBLE (XGBOOST)

Class	Precision	Recall	F1-Score
1	0.92	0.88	0.90
2	0.91	0.94	0.93
3	0.95	0.98	0.97
4	1.00	1.00	1.00
5	0.98	0.91	0.94
6	0.95	0.93	0.94
7	0.99	0.98	0.99

Table III demonstrates that the stacking ensemble maintains consistently high precision and recall across all classes, confirming that the method generalizes well to multi-class classification.

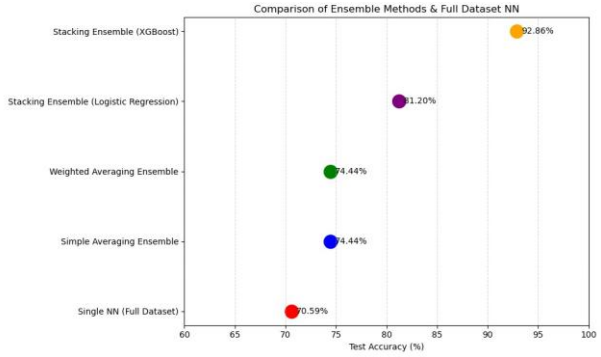

Fig. 2. Comparison of Test Accuracy Across Single Model and Ensemble Methods

C. Comparative Visualization

Figure 2 provides a visual comparison of test accuracy across the different models.

The plot illustrates the clear advantage of ensemble strategies over a single neural network. Weighted averaging improves slightly over simple averaging, while stacking with XGBoost achieves the highest overall accuracy.

D. Discussion

The results indicate that partitioning the dataset using Random Sample Partitioning (RSP) and performing PCA on each partition reduces computational overhead while maintaining or improving classification performance. Ensembles of neural networks trained on these partitions outperform a single network trained on the full dataset. Among ensemble strategies, stacking with a powerful meta-learner such as XGBoost provides the best performance by learning complex combinations of base model predictions.

V. CONCLUSION

In this work, we presented a scalable pipeline for large-scale multi-class classification using Random Sample Partitioning (RSP), feature selection, Principal Component Analysis (PCA), and ensemble learning. The Covtype dataset was used as a benchmark, and experiments demonstrated that ensembles of neural networks trained on partitioned subsets consistently outperform a single network trained on the full dataset.

Among the ensemble methods evaluated, stacking with XGBoost as the meta-learner achieved the highest accuracy, demonstrating the effectiveness of combining partitioned training, dimensionality reduction, and meta-learning. The proposed framework successfully reduces computational costs while maintaining or improving predictive performance, making it suitable for other high-dimensional, large-scale datasets.

Future work may explore additional meta-learners, automated feature selection techniques, and the integration of distributed deep learning frameworks to further enhance scalability. Additionally, applying the proposed approach to other domains such as healthcare, image analysis, or sensor data could validate its generalizability and broader applicability.

REFERENCES

[1] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proc. COMPSTAT*, 2010, pp. 177–186.

[2] Z. Zhou, *Ensemble Methods: Foundations and Algorithms*, Chapman & Hall/CRC, 2012.

[3] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, 2002.

[4] Cover Type Dataset, Kaggle. [Online]. Available: https://www.kaggle.com/c/forest-cover-type-prediction/data

[5] M. S. Mahmud, et al., “Ensemble Learning Methods in Big Data Analytics,” *Int. J. Data Science and Analytics*, 2019.

[6] M. S. Mahmud, et al., “Machine Learning Algorithms for Predictive Analytics in Big Data,” *J. Mach. Learn. Cybernetics*, 2020.

[7] R. Chen and J. Zhang, “Random Sample Partitioning for Scalable Big Data Analytics,” *IEEE Trans. Big Data*, vol. 5, no. 2, pp. 123–134, 2019.